Proceedings of the sixth ESSLLI Student Session

August 2001, Helsinki

Kristina Striegnitz (Ed.)

Preface

The student session of the European Summer School in Logic, Language and Information provides a forum where students at any level can present their own work to an international audience of researchers and other students. It aims at creating an environment where students can, on the one hand, discuss their ideas with experts in the area and get helpful and inspiring feedback on their work, and, on the other hand, meet and exchange ideas with other students. Submissions are invited in any of the six ESSLLI subject areas: logic, language, computation, logic & language, logic & computation, and language & computation.

It is now the sixth time that a student session is organized as part of the ESSLLI. As in the previous years, the number of submissions reflects the interest that exists in the ESSLLI student session: 63 submissions from all over the world were received this year. The program committee accepted 18 papers for a 30 minute presentation and 11 for a poster presentation, 28 of which appear in this volume.

I am very grateful to the program committee – to the co-chairs, who did an excellent job in finding reviewers and coordinating the reviewing in their section, and to the area experts who assisted us with their expertise and experience.

Of course, the student session wouldn't have been possible without the numerous reviewers. I want to thank them for taking the time to read and comment on the submitted papers, which not only helped us in making the final decision, but is also important as feedback for the student authors.

Finally, I would like to thank everybody who supported me with help and advice, especially Geert-Jan Kruijff, Ivana Kruijff-Korbayova, and Ahti Pietarinen of the local organizing committee.

I am looking forward to the student session in Helsinki, which promises many interesting presentations and stimulating discussions. I hope that it will be a success for all participants.

Kristina Striegnitz — Saarbrücken, June 2001
ESSLLI 2001 Student Session Chair

Programm Committee

LOGIC
Ani Nenkova, Columbia University, USA
Yde Venema, University of Amsterdam, Netherlands

LANGUAGE
Malvina Nissim, University of Pavia, Italy
Ruth Kempson, King's College London, UK

COMPUTATION
Jan Schwinghammer, Saarland University, Germany
Gilles Dowek, INRIA Rocquencourt, France

LOGIC & LANGUAGE
Raffaella Bernardi, University of Utrecht, Netherlands
Patrick Blackburn, INRIA Lorraine, France

LOGIC & COMPUTATION
Carsten Lutz, University of Aachen, Germany
Ilkka Niemelä, Helsinki University of Technology, Finland

LANGUAGE & COMPUTATION
Susanne Salmon-Alt, INRIA Lorraine, France
Shuly Wintner, University of Haifa, Israel

Reviewers

Marco Aiello, Jan Alexanderson, Thorsten Altenkirch, Elisabeth André, Hiroshi Aoyama, Carlos Areces, Mira Ariel, Alessandro Artale, Paolo Baldan, Jörg Bauer, Michael Beeson, Lev Beklemishev, Agnes Bende-Farkas, Brandon Bennett, Claire Beyssade, Jean-Yves Beziau, Patrick Blackburn, Ulrich Bodenhofer, Eerke Boiten, Johan Bos, Sebastian Brandt, Bryson Brown, Armelle Brun, Silvia Bruti, Mary Buchholtz, Isabella Burger, Miriam Butt, Jean Carletta, Stephen Clark, Francis Corblin, John Constable, Gilles Dowek, Antonin Dvorak Regine Eckhardt, Jan van Eijck, Katrin Erk, Tim Fernando, Olivier Ferret, Melvin Fitting, Christophe Fouquere, Nissim Francez, Enrico Franconi, Anette Frank, Bertrand Gaiffe, Kim Gerdes, Bart Geurts, Chiara Ghidini, Valentin Goranko, Bernd Grobauer, John Hannan, Roland Hausser, Dominik Heckmann, Klaus von Heusinger, Dirk Heylen, Colin Hirsch, Nancy Ide, Rosalie Iemhoff, Amar Isli, Gerhard Jäger, Tomi Janhunen, Maarten Janssen, Sylvain Kahane, Ruth Kempson, Steve King, Bartek Klin, Alexander Koller, Nobo Komagata, Geert-Jan Kruijff, Ivana Kruijff-Korbayova, Daniela Kurz, Celine Kuttler, François Lamarche, Frédéric Landragin, Maria Lapata, Alain Lecomte, Patrice Lopez, Carsten Lutz, Katja Markert, Maarten Marx, Tomoko Matsui, Erica Melis, Daniel Moldt, Paola Monachesi, Christof Monz, Richard Moot, Glyn Morrill, Stefan Müller, Daniele Nardi, Ilkka Niemela, Oystein Nilsen, Malvina Nissim, Rick Nouwen, Richard Oehrle, Martin Otto, Valeria de Paiva, Simon Parsons, Lawrence C. Paulson, Nikolay Pelov, Gerald Penn, Massimo Poesio, François Pottier, Stephanie Pouchot, Owen Rambow, Anne Reboul, Steve Reeves, Philip Resnik, Olivier Ridoux, Laurent Romary, Andreas Rossberg, Louisa Sadler, Christer Samuelsson, Anoop Sarkar, Ulrike Sattler, Marina Sbisa, Peter Schotch, Svetlana Sheremetyeva, Patrick St.-Dizier, Maarten Stol, Jiri Sustal, Mariet Theune, Yannick Toussaint, David Traum, Dimiter Vakarelov, Achille C. Varzi, Yde Venema, Andrei Voronkov, Peter Wiemer-Hastings, Frank Wolter, Paul Wong, Michael Zock

Contents

Computing adverbial quantifier domains

David Ahn
University of Rochester, Computer Science Department
davidahn@cs.rochester.edu[1]

Abstract. This paper describes a method for computing the domain of quantification of an adverbially quantified sentence. This method relies on the accommodation of presuppositions in the scope of a quantificational adverb and on the resolution of the domain in context. This paper also describes a computational system for processing such sentences based on this method.

1 Introduction

This paper is concerned with the computation of the logical form of adverbially quantified sentences, which are those sentences modified by an explicit quantificational adverb (qadverb), such as *always, usually, sometimes*, or *never*. The primary challenge in computing the logical form of an adverbially quantified sentence (which we will call qadverb sentences) is determining the domain of quantification (there are additional challenges associated with interpreting such a logical form, including the strange modal nature and quantificational force of generics, but we are leaving those aside for this paper). We propose that qadverbs quantify over situations which are restricted both by presuppositions of the scope of the qadverb and by context. Furthermore, we propose that such domain restrictions can be computed by a method based on the presupposition resolution algorithm of van der Sandt (1992), and we demonstrate this with a DRT-based grammar and parser based on the system of Blackburn and Bos (1999).

2 Quantificational adverbs

Syntactically, a qadverb can modify either a verb phrase or a sentence:

(1) John usually takes out the trash.

(2) Usually, John takes out the trash.

[1]This work was supported by NSF research grant IIS-0082928

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

Semantically, however, we take them to be sentential operators, because they give rise to scope ambiguities with quantified noun phrases (NPs):

(3) Someone usually takes out the trash.

In fact, we take the logical form of a qadverb sentence to be similar to that of a sentence containing a quantified NP. The qadverb, like a quantificational determiner, corresponds to an operator which takes three arguments. The first argument is a discourse marker, which serves as the variable of quantification. The second argument (the restrictor) indicates the domain of quantification; the third argument (the nuclear scope) indicates the predication which is asserted of the members of the domain. In the DRT-based logical form language, the second and third arguments are each a DRS. The variable of quantification is introduced in the restrictor DRS, which is accessible to the nuclear scope DRS, much like the arguments of a conditional operator. For a quantificational determiner, the restrictor is provided by the remainder of the NP, and the nuclear scope, by the remainder of the sentence. For a qadverb, the situation is not so straightforward.

Unlike a quantificational determiner, there is no syntactically determined relationship between a qadverb and constituents which might provide its restrictor. The earliest work on qadverb sentences (starting with Lewis, 1975) focuses on examples in which an *if*- or *when*-clause provides the restrictor:

(4) If a man owns a donkey, he always beats it.

Without such a clause (and sometimes, even with one), any part of a qadverb sentence may map to the restrictor argument. Milsark (1974) provides an oft-cited example:

(5) Typhoons arise in this part of the Pacific.

This sentence has two readings. One is an implausible reading in which the set of typhoons serves as the restrictor argument; on such a reading, it is taken to be a general property of typhoons that they arise in a particular part of the Pacific. In the other, more natural, reading, the restrictor argument is the set of situations involving the indicated part of the Pacific; on such a reading, the occasional arising of a typhoon is taken to be a property of situations in a particular part of the Pacific. De Swart (1991) and Rooth (1985) give examples of sentences in which even an explicit *when*-clause does not necessarily map to the restrictor:

(6) When John was young, he often took walks in the gardens.

(7) John usually shaves when he showers.

In (6), the *when*-clause only serves to set the situation in time. In (7), focal stress determines whether the *when*-clause maps to the restrictor.

Focal stress and explicit *if*- and *when*-clauses are just two of the many factors that have been implicated in the determination of the restrictor argument of a qadverb. Diesing (1992) proposes with her Mapping Hypothesis that it is syntactic structure which is primarily responsible for determining what overt material (in particular, which NPs) in a qadverb sentence is associated with the qadverb restrictor. Her theory requires the syntactic machinery of GB and fails to take into account much of the semantic complexity of the data. Opposed to this syntactic view are those who claim that it is the topic-focus or topic-comment structure of a sentence that determines its division into restrictor and matrix. According to this view, the topic of a qadverb sentence is its restrictor; its focus, the matrix (proponents of this approach include Chierchia, 1992 and Jäger, 1997; see Partee, 1995 for a list of others). This approach is closely related to Rooth's observations on the association of qadverb restrictors with focal stress (exemplified by (7), above). Cohen (1996) proposes that qadverbs quantify over appropriate sets of alternatives which may be generated in a variety of ways, including by focal stress or by presuppositions.

Several other authors have noticed that presuppositions of the scope of a qadverb are incorporated into its restrictor (Schubert and Pelletier 1987; Berman 1991):

(8) John usually beats Marvin at ping pong.

(9) John usually regrets missing a lecture by Chomsky.

Schubert and Pelletier note that in a qadverb sentence with a presuppositional verb, such as *beat*, the qadverb quantifies only over situations which satisfy the presupposition associated with the verb (in the case of (8), situations in which John plays Marvin at ping pong). Berman provides examples for other kinds of presuppositions, such as factives (*regrets* in (9)) and aspectual verbs. The range of presuppositional material considered by Schubert and Pelletier and by Berman is somewhat narrow, and both consider presupposition incorporation to be only a part of process of determining a qadverb restrictor. When we consider the full range of presuppositions which may be accommodated by a qadverb restrictor, however, we find that most of the overt material which has been claimed to be incorporated into the restrictor consists of presuppositions of the scope.

3 Presuppositions

A wide range of linguistic phenomena gives rise to presuppositions. Keenan (1971) includes as examples of presupposition triggers: definite descriptions, factive predicates, cleft constructions, selectional restrictions, temporal subordinate clauses, certain aspectual verbs, iteratives, and presuppositional adverbs; to this list, van der Sandt (1988) adds focal stress, lexical presupposition, and quantifiers. Additionally, Milsark (1977) and Fodor and Sag (1982) distinguish between two readings of indefinites, one of which corresponds to existential quantification and the other of which is essentially presuppositional. If we simply allow that sentential presuppositions are what determine a qadverb restrictor, we have an account which subsumes the above-mentioned accounts. Explicit *if*- and *when*-clauses fall together with other presupposed subordinate clauses. The NPs with which Diesing is concerned fall into one of the following classes: definite descriptions, which are the paradigm case of presuppositionality; quantified NPs, which are taken to presuppose their domains; and indefinites, which on certain interpretations are presuppositional. The most illustrative cases of topic-focus articulation (cleft constructions and focal stress) are also subsumed as presuppositions.

In order to make our presuppositional theory work, we need a separate account of presuppositions. The most widely accepted account of presuppositional phenomena is that of van der Sandt (1992), which is well-suited to our purposes. Van der Sandt proposes that presuppositions are simply informationally rich anaphors. Like an ordinary anaphor, a presupposition requires an antecedent, and if one is found in the appropriate context, then the presupposition is satisfied. Unlike an ordinary anaphor, though, the inability to find an antecedent does not necessarily lead to failure. Since a presupposition is informationally rich, i.e. it has its own descriptive content, an antecedent may be constructed for it, under the appropriate conditions.

Van der Sandt casts his account in a version of DRT, which allows him to handle projection in configurational terms, using the DRT notion of accessibility. On his analysis, presupposing elements introduce their elementary presuppositions in a local DRS just as anaphoric elements introduce anaphors in the local DRS. Each elementary presupposition must then be resolved, either by binding or accommodation. The resolution process proceeds from the local DRS for an elementary presupposition along its projection line looking for an appropriate marker which can be unified with the elementary presupposition. If one is found, then an anaphoric link is created between the presupposition and its antecedent, and the presupposition has been satisfied by binding. If one is not found, the presupposition must be accommodated at some accessible level in which constraints on binding, consistency, informativeness, and efficiency can be satisfied. Accommodation simply consists of copying the presupposition to the embedding DRS.

4 Qadverb analysis

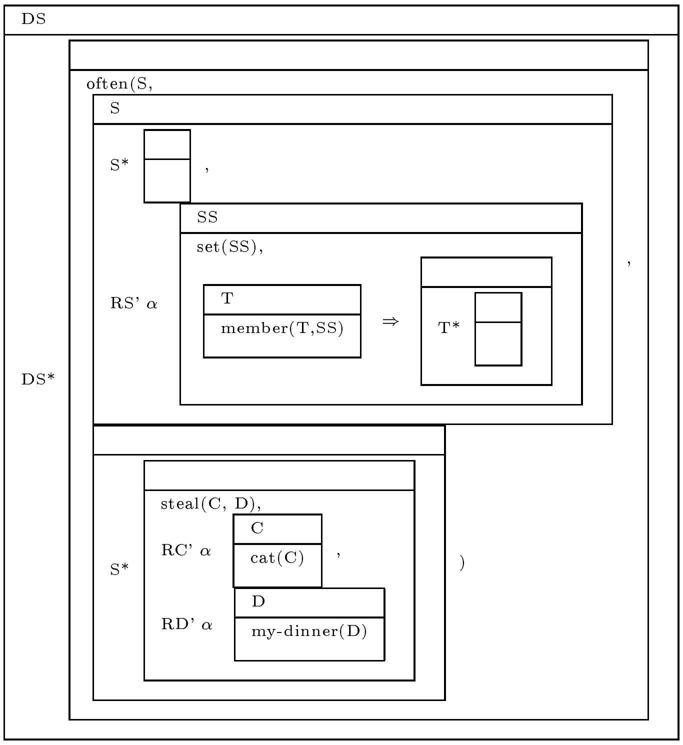
Following Berman (1987), Heim (1990), and von Fintel (1995), we take qadverbs to be quantifiers over situations (rather than unselective quantifiers or quantifiers over time as others have proposed), which allows us to restrict both temporally and informationally the entities being quantified over and also allows us to make the connection between qadverb restrictors and presuppositions clearer. Following Poesio (1994), we take presuppositions to be situation descriptions, and the resolution of a presupposition to consist of binding or accommodating a situation parameter associated with the description. Since there is no descriptive content initially associated with a qadverb restrictor, the restrictor in our initial logical form consists solely of a situational discourse marker with no conditions. The presupposition resolution process resolves presuppositions of the scope with the restrictor by binding the situation parameters associated with these presuppositions to the situational discourse marker introduced in the restrictor. This binding of situation parameters accounts for the presupposition accommodation observed by Schubert and Pelletier and Berman and also accounts for most of the other overt material which has been claimed to restrict the domain of quantification of qadverbs.

As we mentioned above, the domain of quantification of a quantificational determiner is presupposed. A qadverb also presupposes its domain of quantification. In both cases, what is presupposed is the existence of a set corresponding to the quantification domain. Thus, for a qadverb, the restrictor is taken to be a situation description which provides the membership conditions of the presupposed set of situations. This set of situations is associated with a situation parameter of its own. Note that this situation parameter is a separate entity from the situational discourse marker introduced in the restrictor. This discourse marker indicates the situations over which the qadverb quantifies and, as described above, binds presuppositions in the nuclear scope. The situation parameter associated with the domain presupposition provides the resource situation in which those entities are situated. It is the resolution of this parameter in the context of the rest of the discourse which results in the anaphoricity of qadverb domains that is observed by von Fintel (1995).

For example, consider the sentence (10). (We will illustrate one plausible resolution, though there are others.) The initial representation for this sentence (11) introduces a global discourse situation *DS*, which supports the dinner-stealing proposition. (The support relation * between a situational discourse marker and a DRS is transparent to accessibility.) The quantificational condition deriving from the qadverb *often* takes three arguments: *S*, which is the variable of quantification; the restrictor DRS, which contains a situation parameter *RS'* associated with the resource domain presupposition; and the nuclear scope DRS, which contains situation parameters associated with the presuppositions corresponding to *the cat* and *my dinner*. Situational parameters are distinguished by a prime and are associated to a presupposition via the α relation.

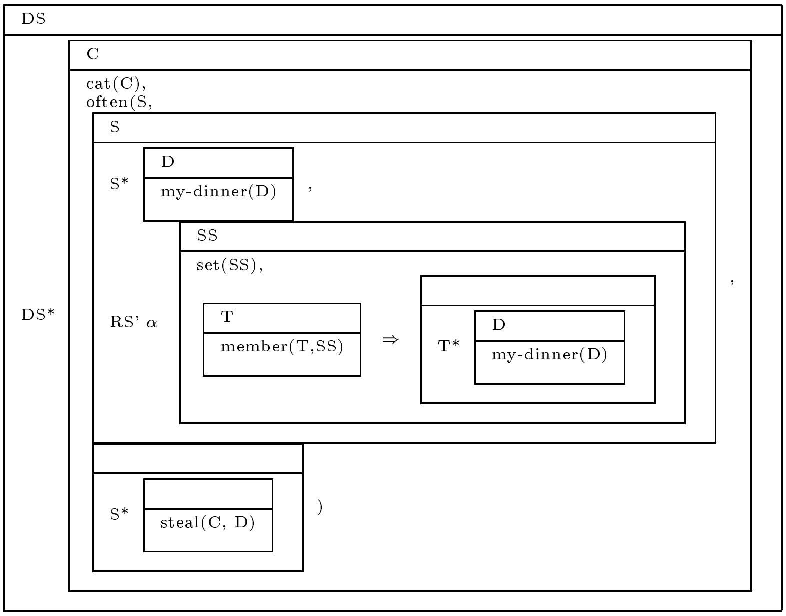
(10) The cat often steals my dinner

(11)



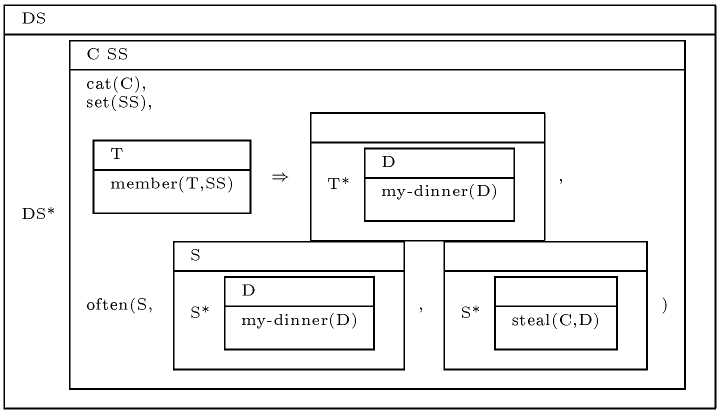
In the representation (12), the presupposition for *the cat* has been accommodated globally by binding *RC'* to *DS* and adding the conditions of *RC'* to the situation supported by *DS*. The presupposition corresponding to *my dinner* has been accommodated in the restrictor, by binding *RD'* to *S* and adding the conditions of *RD'* to the situation supported by *S*. These conditions are also added to resource domain presupposition.

(12)



The resolution process is completed by binding *RS'* to *DS* and adding its conditions, as well, resulting in representation (13).

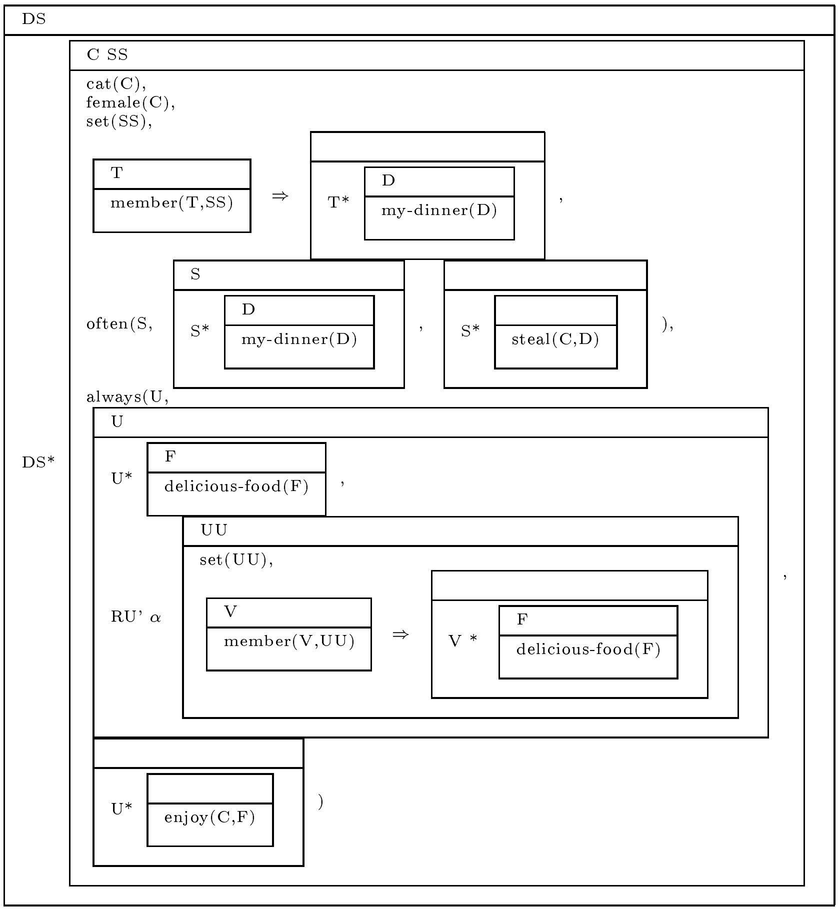
(13)



The discourse may be continued with sentence (14), which, after resolving the presuppositions associated with *she* and *the delicious food*, results in the representation (15).

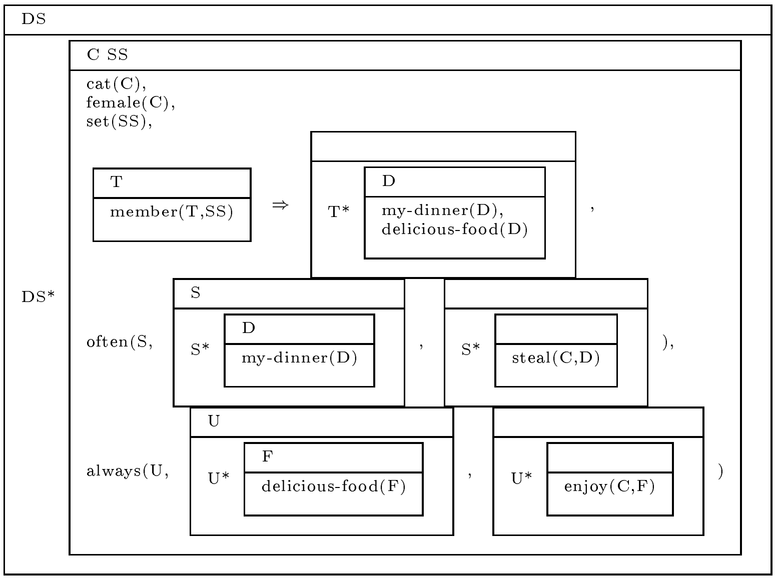
(14) She always enjoys the delicious food.

(15)



The final representation (16) is completed by binding *RU'* to *DS* and equating *UU* to *SS*. In collapsing the membership conditions for *UU* and *SS*, *F* and *D* are also equated.

(16)



In the above example, the resource domains associated with the qadverbs were resolved by accommodation or by binding. In binding the resource domain for *always* to the accommodated set for *often*, we equated the two sets. In general, however, the relation between an existing set and a resource domain which is bound to it is not equality. Consider the following example:

(17) John ate every piece of fruit in the bowl. Most (of the) oranges were tasty.

In this case, the set of fruit should bind the set of oranges, even though the two sets are not necessarily equal. Instead, the set of oranges should be taken to be a subset of the set of fruit. We see the same situation with qadverb domains:

(18) My friends always vacation in strange places. My best friend John usually goes to Antarctica.

The set of situations accommodated for the first sentence consists of situations in which the speaker's friends go on vacation. The resource domain for the second sentence is the set of situations in which John goes somewhere. Clearly, this set should be bound to a subset of the first set—those situations in which John goes on vacation. Thus, we expand the notion of binding to include binding a presupposed set to a subset of an existing set.

As the example (10)-(16) demonstrates, presuppositions arising in the scope of a qadverb are not necessarily resolved with the restrictor. Whether or not a presupposition must be bound to the situational discourse marker introduced in the restrictor depends on a number of factors. Most important is whether or not the sentence is coherent without such binding. Since we claim that a qadverb essentially quantifies over situations (which default to time intervals of contextually determined granularity in the absence of any bound presuppositions), any presuppositions that are resolved at a discourse level superordinate to the quantification must be constant for each of the situations over which the qadverb quantifies. For many presupposed elements, this is not possible. Consider, for example:

(19) The fog usually lifts before noon here.

There cannot be a single, unique instance of fog which repeatedly lifts; instead, the definite description must be bound to the qadverb restrictor, resulting in separate instances of fog for each lifting event. We are still developing a way to formalize this notion of persistence so that it can be used to evaluate possible resolutions.

One shortcoming of our analysis of qadverbs is that once the resource domain is resolved, the connection between the set of situations constituting the resource domain and the set of situations over which the qadverb quantifies is lost. One way to maintain this connection is to include an additional condition in the restrictor which indicates that the situation corresponding to the variable of quantification is a member of the resource domain. For example (11), the condition would be *member(S,SS)*. Once this is done, the additional conditions in the restrictor would seem to be redundant, except that they are necessary to maintain accessibility for the nuclear scope.

Geurts and van der Sandt (1999) present an account of quantifier domain restriction that is in spirit very similar to ours. They, however, introduce a new kind of entity—a propositional discourse marker—which allows them to handle acessibility non-structurally. On their analysis, a quantifier is not a relation between descriptions of sets but between propositional discourse markers, which are sets of embedding functions specified as separate conditions. Thus, there is a direct connection between the presupposed resource domain and the restrictor argument. They build in conservativity by stipulating that the propositional discourse marker associated with the nuclear scope is an extension of the marker associated with the restrictor and, as a result, they get accessibility of the restrictor from the nuclear scope for free.

5 Computational implementation

We are presently at work on a small computational system to process qadverb sentences. It is based on the DRT parser in the textbook by Blackburn and Bos (1999), which is focused on the problems of quantifier scope ambiguity and presupposition resolution. Thus far, we have added grammar rules and lexical entries to allow for plural nouns and quantificational adverbs and modified existing rules and entries to associate situational discourse markers with DRSs. We do use the existing notation for presuppositions, which differs somewhat from ours, and have not modified it to associate situation parameters with presuppositions. Thus, instead of introducing a presupposition inside the DRS in which its extent begins, a presupposition is introduced via an alpha expression which takes scope over the DRS in which its extent begins.

We have had to modify the presupposition resolution algorithm itself. The original algorithm resolves each presupposition before resolving any presuppositions triggered within its scope; in order for the resource domain of a qadverb to have any descriptive content to be resolved, however, the presuppositions of the nuclear scope (which falls within the scope of the resource domain presupposition) must be resolved first. Also, a resource domain presupposition (both for qadverbs and for quantified NPs, for which we have also added such a presupposition) may be bound either to an existing set or to a subset of an existing set. The semantic macro that produces the semantic portion of the lexical entry for a qadverb is as follows:

```
advSem(Sym, qadv, lambda(P, lambda(X,
       alfa(SS,qresource,
            drs([SS],[set(SS),
                      drs([S],[member(S,SS)])>drs([],[])]),
            drs([],[qcond(Sym,S,drs([S],[S * drs([],[])]),
                                drs([],[S * P@X])]))))))
```

Since a qadverb is syntactically a VP operator, its first argument is a predicate, and its second, an individual. The resulting expression is a resource domain presupposition (SS is the set of situations) whose scope is the quantificational condition (Sym is the translation of the qadverb). Presuppositions arising from $P@X$ (i.e. P applied to X) may be accommodated in the restrictor. Material from the restrictor is copied into the membership conditions of the resource domain, and the resource domain is then resolved.

This system is still a work-in-progress. We have not yet built a mechanism to check consistency when binding sets, and we still have to introduce situation parameters into presuppositional expressions to mirror our analysis more closely. We are considering trying to use propositional discourse markers, a la Geurts and van der Sandt (1999), since in addition to handling the connection between the restrictor and the resource domain more neatly, such an approach would obviate the need for a different order of resolution just for qadverbs. Changes would be necessary to handle accessibility through linked propositional discourse markers, as well as through structural embedding. Nonetheless, the current system computes qadverb domains in accordance with the theoretical analysis outlined above, using only presuppositional information.

References

Berman, S. (1987). Situation-based semantics for adverbs of quantification. In *University of Massachusetts Occasional Papers in Linguistics*, Volume 12, pp. 45–68. Graduate Linguistic Student Association, University of Massachusetts, Amherst. Also published in the Proceedings of the Sixth Annual West Coast Conference on Formal Linguistics, 17–31, Stanford Linguistics Association, Stanford University, 1987.

Berman, S. (1991). *On the semantics and logical form of WH-clauses.* Ph. D. thesis, University of Massachusetts, Amherst. Also published in Outstanding Dissertations in Linguistics, Garland, 1994.

Blackburn, P. and J. Bos (1999). Representation and inference for natural language: A first course in computational semantics. http://www.coli.uni-sb.de/˜bos/comsem/.

Chierchia, G. (1992). Anaphora and dynamic binding. *Linguistics and Philosophy 15*, 111–183.

Cohen, A. (1996). *Think Generic! The Meaning and Use of Generic Sentences.* Ph. D. thesis, Carnegie Mellon University. Also published in Dissertations in Linguistics, CSLI, 1999.

de Swart, H. (1991). *Adverbs of Quantification: A Generalized Quantifier Approach.* Ph. D. thesis, Rijksuniversiteit Groningen. Also published in Outstanding Dissertations in Linguistics, Garland, 1993.

Diesing, M. (1992). *Indefinites.* MIT Press.

Fodor, J. D. and I. Sag (1982). Referential and quantificational indefinites. *Linguistics and Philosophy 5*, 355–398.

Geurts, B. and R. van der Sandt (1999). Domain restriction. In P. Bosch and R. A. van der Sandt (Eds.), *Focus: Linguistic, Cognitive, and Computational Perspectives*, pp. 268–292. Cambridge University Press.

Heim, I. (1990). E-type pronouns and donkey anaphora. *Linguistics and Philosophy 13*, 137–177.

Jäger, G. (1997). The Stage/Individual Contrast Revisited. In B. Agbayani and S.-W. Tang (Eds.), *Proceedings of WCCFL 15*, pp. 225–239.

Keenan, E. (1971). Two kinds of presupposition in natural language. In C. Fillmore and D. T. Langendoen (Eds.), *Studies in Linguistic Semantics*, pp. 44–52. New York: Holt, Rinehart and Winston, Inc.

Lewis, D. (1975). Adverbs of quantification. In E. L. Keenan (Ed.), *Formal Semantics of Natural Language: Papers from a colloquium sponsored by the King's College Research Centre, Cambridge*, pp. 3–15. Cambridge University Press.

Milsark, G. (1974). *Existential Sentences in English.* Ph. D. thesis, MIT.

Milsark, G. (1977). Toward an explanation of certain peculiarities of the existential construction in English. *Linguistic Analysis 3*, 1–29.

Partee, B. H. (1995). Quantificational structures and compositionality. In E. Bach, E. Jelinek, A. Kratzer, and B. H. Partee (Eds.), *Quantification in Natural Languages*, Volume 2, pp. 541–601. Dordrecht: Kluwer Academic Publishers.

Poesio, M. (1994). *Discourse Interpretation and the Scope of Operators.* Ph. D. thesis, University of Rochester.

Rooth, M. (1985, February). *Association with Focus.* Ph. D. thesis, University of Massachusetts, Amherst.

Schubert, L. K. and F. J. Pelletier (1987). Problems in the Representation of the Logical Form of Generics, Plurals, and Mass Nouns. In E. LePore (Ed.), *New Directions in Semantics*, Chapter 12, pp. 385–451. London: Academic Press.

van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics 9*, 333–377.

van der Sandt, R. A. (1988). *Context and Presupposition.* London: Croom Helm.

von Fintel, K. (1995, February and May). A minimal theory of adverbial quantification. In B. Partee and H. Kamp (Eds.), *Context Dependence in the Analysis of Linguistic Meaning*, IMS Stuttgart Working Papers, Prague and Bad Teinach, pp. 153–193.

Grounding in the Swedish Map-task

HENRIK BERGQVIST
Stockholm University
four.mount@swipnet.se

ABSTRACT. This paper investigates how two conversational moves with grounding properties, acknowledge and reply-yes (Carletta et al. 1997), are distributed in Swedish map task dialogues, and if their respective utterance forms are possible to categorize in order to obtain statistical data which might be useful in an automatic identification system.

1 Introduction

Grounding is the process whereby information between participants in discourse is added to the *common ground.* This common ground, or *common knowledge* is what the participants both believe to be true in some sense in relation to what is being said in the conversation. This process also postulates that there are distinct units in discourse whose function is that of showing understanding or acceptance of an utterance. This phenomenon has been reasonably well investigated by among others, Clark & Schaefer (1989) and David Traum (1994), and is also the subject of this study. In order to clearly identify and categorize something it is often easier to define and restrict the domain of the investigation, which here is limited to Swedish map-task dialogues along the line of Anderson et al. (1991). In the map-task two participants each have a map in front of them. One of the participants is the *giver* and the other is the *follower.* The giver has a route drawn on his map and tries to get the follower to draw the same route, but the maps are not exactly alike which sometimes adds a bit of confusion to the dialogue in trying to determine what exactly the discrepancy is.

The tag set used to categorize utterances in the dialogues was developed by Carletta et al. (1997) and has previously been used in the HCRC-corpus (www.hcrc.ed.ac.uk). Within this tag set there are two separate *conversational moves* which both have grounding properties, namely *acknowledge* and *reply-yes.* These two moves are the main object of this study. They have several traits in common, but it could be argued that one of them, *acknowledge,* is a grounding move in the sense that it only shows understanding

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

and/or acceptance, and that *reply-yes* does this while also contributing semantic information to the conversation. Their lexical forms are often almost identical, but their roles in the discourse are somewhat different. In accordance with Clark & Schaefers (1989) model for grounding in discourse, some utterances can convey semantic content while also functioning as a grounding element. This holds for the *reply-yes* move and it is therefore interesting to investigate from a grounding view point, in the same way as *acknowledge.*

The hypotheses is that it is possible to apply the theory of grounding to the domain of the map-task and that a clear picture concerning the distribution of grounding moves and their surface forms can be obtained. This could give useful clues to an automatic identification system on a statistical basis. Separate utterance forms are expected to be identified consistently for both moves. Can these forms in turn be assembled into a few, easily identifiable categories, and is it possible to make predictions as to the probability of a grounding move after an information bearing move, eg. *instruct, clarify, explain* etc. (Carletta et al. 1997)?

2 Background and Method

The background for this investigation consists of two things, the article by Carletta et al. (1997) which explains and evaluates the tagging model as referred to above, and the work of Clark & Schaefer (1989) and David Traum (1994). Clark & Schaefer introduce a definition of a segment in discourse, which is called the *turn.* This is equivalent to Carletta et al's game level and constitutes a sub-unit in the dialogue. The *turn* is a term which is used throughout the study and in it, it generally encompasses two *moves* in a pair, for example *instruct-acknowledge.* The work by Traum is directed towards forming a computational theory of grounding, explaining how *discourse-units* (equivalent to *moves*) occur in task-oriented dialogue, with empirical information from TRAIN-dialogues. It is a development of the previous article by Clark & Schaefer, since it provides empirically based arguments concerning grounding which the article by Clark & Schaefer lacks.

Briefly, utterances according to Carletta et al's scheme can be divided into three main categories, *initiative, responsive* and *preparative.* Initiative utterances containing a command are coded *instruct.* If they simply consist of a statement, they are coded *explain.* An initiative utterance can also be a query of some kind, *align, check, query-yes/no* or *query-wh.* Possible responses are *acknowledgement,* if only evidence was given that the utterance was successful and *clarify* if the reply contributes information as well. The response category also contains *reply-yes/no,* and *reply-wh* if the initiative move was a *query* or *check.* Preparative utterances are not included in the survey, as they do not pair up with other utterances. This is an example of a turn made up from *instruct-acknowledge.*

Initiator: [...]du siktar du går mitt emellan palmerna du håller dig närmre den undre gruppen av palmer än den övre.[instruct] [you aim for you go between the palm trees you keep closer to the lower group of palm trees than the upper one]

Responder: ja [...][acknowledgement] [yes] (my translation)

The before mentioned difference between *acknowledge* and *reply-yes* is also related to whether a certain move functions on a *conversational-* or *domain level.* The split between *conversational-* and *domain* goal in the tag set is relevant to this investigation since the tagging method contains *moves,* and *turns,* which are by definition conversational and domain oriented. An example of a conversational turn would be *query-reply,* whereas *instruct-acknowledgement,* would be a domain oriented turn. This is not a big thing in the study, but it is a difference that has relevance for how the moves are defined, and it is also something which separates the tag-set in this study from the one used by Traum, which only concerns the domain level.

The Swedish map-task corpus, "the Swedish Corpus of Spontaneous Speech" (SCSS), was used for this investigation. It was assembled by Helgason at the University of Stockholm and was part of an investigation on preaspiration in natural speech. As such its transcription was not with the map-task in mind. Since this was a preliminary investigation, only four out of six available dialogues were used, and several more await transcription. The recordings were made face to face which makes it possible that nods and eye contact alone make up some of the grounding acts. The four dialogues contain approximately 546 utterances as they were segmented in the original material, but this does not coincide with how they were later tagged in accordance with Carletta et al's model. My segmentation of an utterance often resulted in a division of one utterance into two or three moves. This was done with the content of the utterance in mind, and was not related to the separation of utterances in a temporal sense. The material was presented to me both as audiofiles with a division between right and left channel, equivalent to giver-follower, and as textfiles with transcription information and a separation between utterances. When something was unclear to me concerning how the utterances were distributed or how they were to be segmented, I consulted the audiofiles, but otherwise relied on the transcribed material. This had to do with the time frame of the study, which was limited since it was a preliminary investigation. The examples from the dialogues will be presented in a slightly modified version where initiator and responder are marked to signify the participants of the specific turn.
First, an attempt is made to try to establish a few easily identifiable utterance forms that are common to the grounding moves, *acknowledge* and

reply-yes, to see how often they occur after other moves, i.e. *instruct, explain, clarify, check, query-y/n, acknowledge,* and *reply-yes.* At an early stage in the investigation, the *reply-wh* move was also investigated since it also has a grounding function, but the instances of the move in the dialogues were too few and the utterance forms for reply-wh were in no way homogenous. Since this was not what I was looking for, I decided to exclude the results from the investigation. Second, the question as to whether moves which require grounding are actually followed by such moves, is addressed to see if high percentages can be obtained, and when a grounding move is not uttered, what takes its place. Finally an attempt is made at constructing a model, based on Traum´s finite-state machine, (Traum 1994) to display how moves as defined by Carletta et al. can occur in the dialogues as a help in trying to determine which utterance may follow another as far as grounding is concerned in the Swedish map task. I have to add that the empirical evidence for such a model has only superficially been investigated within this study and will need much further investigation if the results are actually going to be used in a functioning computational identification system.

3 Results and Discussion

The hypotheses was that it would be possible to find utterance forms for *acknowledge* and *reply-yes* which were going to be easy to identify and categorize. No expectation existed beforehand as to whether only certain moves, eg. information bearing ones, were going to be followed by grounding moves in the form of *acknowledge.* The categorization had to depend on the results. Seven distinct categories for utterance forms were identified which could be found in all dialogues on multiple occasions. They were: 1) *ja* - 'yes', 2) *mmm* - 'uhuh', 3) *okej* - 'okay', 4) *jaha* - 'oh', 5) *aha* - 'oh' 6) *ja, just det* - 'right', and 7) miscellaneous utterance - for example: *pådet viset* - 'is that so'. The last category not only contained utterance forms which were composed of more than one or two words, but also consisted of short utterances which were too few in numbers to be placed in a separate category. These forms were then counted and assembled into different categories depending on their occurrence after the moves *instruct, explain, clarify, check, query-y/n, acknowledge,* and *reply-yes.* Since *instruct, explain, clarify* and to some extent also *acknowledge,* and *reply-yes,* take *acknowledge* as a grounding move, they must be separated from *check* and *query-y/n* which both take *reply-yes* as their proper reply. This division is based on Carletta et al's scheme (Carletta et al. 1997). The results can be seen below in tables 1 and 2. *Acknowledge* can, -as seen in table 1, occur in most contexts of the dialogues and is also the most common move of all moves, before *reply-yes* in second place.

The results show that there is a tendency for the simple reply forms to

Moves preceding utterance forms	*instruct*	*explain*	*clarify*	*reply-yes,* acknowledge
ja - 'yes'	44	17	13	13
okej - 'okay'	16	8	7	9
mmm - 'uhuh'	24	3	7	10
ja, just det - 'right'	2	3	12	1
aha - 'oh'	1	4	3	-
jaha - 'oh'	5	2	8	3
miscellaneous utterance	20	12	6	4
Total moves:	112	49	56	40
Total sum:257				

Table 2.1: Preceding moves with utterance forms for *acknowledgement*

occur more frequently than do the miscellaneous forms. This is a tendency that I hoped for and which the hypotheses predicted. If this has to do with the settings for the recording of the dialogues, which were in a controlled environment, or if it is a property inherent to the phenomenon of grounding, I can not say, but I personally think that it has to do with how "on-line" the conversation is at a certain moment in the discourse. Sometimes (most of the time) the participants feel that they understand each other, and do not feel the need to be so explicit as to paraphrase the previous utterance to convey that understanding has been achieved. One thing that Clark & Schaefer (1989) discusses is the different *weight* of the utterance forms concerning grounding. They propose that *paraphrasing* or *repetition* of parts of a previous utterance has the most weight, and that replies in the form of *yes, right, uhuh,* etcetera, come just after in terms of conversational weight. Since the simple reply-forms are more common than complex or miscellaneous ones, this might indicate what Clark & Schaefer predict, namely that: "[acknowledgements] ...are backgrounded attempts by partners to create contributions from extended turns, and they almost always succeed" (Clark & Schaefer 1989:283).These results are shown in tables 3 and 4.

The distribution of grounding moves after information bearing moves such as *instruct, explain, clarify, check* and *query-y/n* also needs to be investigated in order for any predictions concerning grounding moves in the dialogue to be useful. The results are displayed in table 5.

What these results imply is that grounding, besides being done with grounding moves such as *acknowledge* and *reply-yes,* can be achieved with

Moves preceding utterance forms	*query-yes/no*	*check*
ja[o], (jepp 2) - 'yes'	12	28
okej - 'okay'	-	1
mmm - 'uhuh'	4	4
ja, just det[precis] - 'right'	3	6
aha - 'oh'	-	-
jaha - 'oh'	-	-
miscellaneous utterance	8	14
Total moves:	27	53
Total sum: 80		

Table 2.2: Preceding moves with utterance forms for *reply-yes*

Utterance forms: ja-'yes', mmm-'uhuh' ,okej-'okay' and ja, just det-'right'	*instruct*	*explain*	*clarify*	*reply-yes,* acknowledge
Percentage of total	77%	63%	70%	83%

Table 2.3: Simple utterance forms for *acknowledge*

almost any move as long as it is relevant to the prior utterance. *Instruct* can for example take *explain* as a possible grounding move, and there are examples of this in the dialogues.

> Initiator: [ ]... och sen så tar vi en snabb väg precis rakt österut och du går norr om dom här bergen. [*instruct*] [and then we take a fast route exactly straight east and you go north of these mountains]
>
> Responder: där har jag krokodiler och så en uppsättning såna här löjliga fåglar som säkert heter lunnefåglar eller nåt annat. [*explain*] [there I have crocodiles and a set of these ridiculous birds which probably are called puffins or something else] (my translation)

Instruct can also be followed by a *query-y/n*, or a *check* move if something is not clear to the follower. This means that the turn is not grounded and

Utterance forms: ja-'yes', mmm-'uhuh' ,okej-'okay' and ja, just det-'right'	*query-yes/no*	*check*
Percentage of total	70%	74%

Table 2.4: Simple utterance forms for *reply-yes*

Moves preceding grounding moves	*instruct*	*explain*	*clarify*	*query-yes/no*	*check*
Instance when followed by grounding move	98	38	39	28	48
Instance when not followed by grounding move	29	11	12	13	22
Sum total:	127	49	51	41	70
Percentage of moves followed by grounding moves	77%	78%	76%	68%	69%

Table 2.5: Distribution and percentage of moves followed by grounding moves

that it will continue in order for there to be agreement, or it will be grounded, at a later stage. There are however cases where a *reply-no* is uttered, which indicates that there is no agreement and that something is wrong without specifying exactly what it is. It seems that a measure of grounding and the only possibility of knowing when an utterance has been grounded, depends on the following segments of discourse. If the information conveyed has been accepted, then the participants move on by discussing other things. What is considered to have entered the common ground is by no means absolute, in the sense that it does not have to do with how the follower has drawn his/her route. Only the dialogue in itself can show if something is considered to have entered the common ground by the participants. Alternative moves occur after all the information bearing moves when a grounding move has not been given, and can perhaps best be explained and displayed by constructing a finite-state model. The model below shows the way a *turn* starting with *instruct,* can evolve on its way to be grounded. The difference between the model in this study and the one presented by Traum (1994), which the model

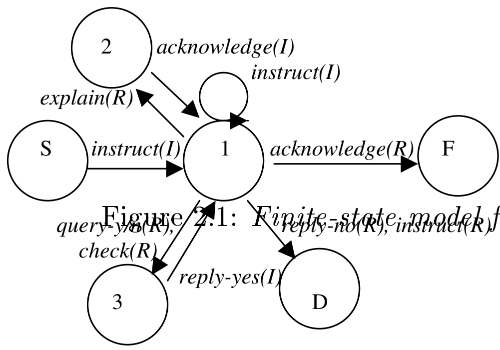


Figure 2.1: *Finite-state model for instruct*

directly builds on, is that the latter encompasses all levels of conversation while mine simply concerns the *turn,* and not the *transaction-level.* These dialogue levels are defined in Carletta et al.´s article. The *transaction-level* would be equivalent to the sub-dialogue level in Traums terms. This means that an entire dialogue or sub-dialogue can not be read through this model, and that it can not be empirically supported by the results of this study. Nevertheless the model of this study can still illustrate how a turn can be constructed and function as far as grounding is concerned. The model should be read as follows: S is the starting-state which leads to state 1. This is the initiative statement given by the initiator. From there *acknowledgement* can be given by the responder so that the turn is grounded and thereby finished, in state F. Alternative moves eg. *explain* (state 2) or *query-yes/no* (state 3) can also be given and they must in turn be *acknowledged* or given a *reply-yes* in order to go back to state 1. State D is also a final state which means the turn is left ungrounded. This can be accomplished by the responder either uttering an opposing *instruct* or simply by replying no. Similar models can be constructed for the other moves as well.

4 Conclusion

What is evident from the investigation is that grounding, as expected, frequently occurs in the Swedish map task dialogues and that it can be categorized into a few utterance forms which in turn are fairly easy to identify. It is very likely, around a 74% chance, that an *instruct, explain* or *clarify* move is followed by an *acknowledge* move, and that one of the four simple utterance forms for *acknowledge* is used, namely *ja-* 'yes', *mmm-* 'uhuh', *okej-*

'okay', or *ja,just det-* 'right'. These simple forms are much more frequent than are the miscellaneous ones with a 73% probability in favor of the four shown forms. The same scenario is valid for *reply-yes,* with a slightly lower probability rate when it comes to predicting grounding moves after infromation bearing ones, ie. after *query-yes/no* (68%) and *check* (69%). What this means is that grounding in the from of *acknowledgement* or *reply-yes* can be identified in task-oriented dialogues and that they have fairly homogenous utterance forms that seem to be particular for those moves. What it does not say anything about is whether the preceding move is information bearing or not, since even an *acknowledgement* can be acknowledged. Lastly the context for grounding moves in the map task and the way grounding can be achieved using alternative moves, other than grounding ones, can be organized into a finite-state model that encompasses the *turn-level* of the discourse (fig.1). The manner in which this way of grounding actually functions in a way similar to the results obtained in this study, is not addressed here and needs an investigation of its own.

Bibliography

Anderson, Anne H., Miles Bader, Ellen Gurman Bard, Elisabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. (1991) *The HCRC Map Task Corpus.* Language and Speech, 34 (4) s. 351-366.

Carletta, Jean, Stephen Isard, Gwyneth Doherty-Sneddon, Amy Isard, Jaqueline C. Kowtko, Anne H. Anderson (1997) *The Reliability of a Dialogue Structure Coding Scheme* Computational Linguistics, vol 23, nr 1: s.13-31,.

Clark, H. Herbert, Schaefer, F. Edward (1989) *Contributing to Discourse,* Cognitive Science 13, s.259-294, Stanford University.

Helgason, Petúr *The Stockholm Corpus of Spontaneous Speech,* University of Stockholm

Stirling, Lesley, Ilana Mushin, Jane Fletcher, Rodger Wales (2000) The Nature of Common Ground Units: *an Empirical Analysis Using Map Task Dialogues,* in Götalog 2000 -workshop on the Semantics and Pragmatics of dialogue Göteborgs Universitet 15-17 June 2000.

Traum, David (1994) *A Computational Theory of Grounding in Natural Language Conversation,* Doctoral Dissertation. `http://www.cs.umd.edu/users/traum/`

Computing the Density of Regular Languages

Manuel Bodirsky
Department Algorithms and Complexity I, Humboldt University Berlin
bodirsky@informatik.hu-berlin.de

Tobias Gärtner
Department of Computer Science, Saarland University
tobias@cs.uni-sb.de

Timo von Oertzen
Department of Computer Science, Saarland University
timovoe@studcs.uni-sb.de

Jan Schwinghammer
Department of Computer Science, Saarland University
jan@ps.uni-sb.de

Abstract. We present an algorithm that computes the limit density of a regular language and prove its correctness. If the language is specified by a finite deterministic automaton $\mathcal{A}$, the algorithm takes time $O(dn^3)$, where n is the number of states of $\mathcal{A}$, and d its periodicity.

1 Introduction

It is a well-known fact that for a given first order formula ϕ without function symbols, the limit density of structures satisfying ϕ is either zero or one ((Fagin 1976; Y. V. Gelbskij and Talanov 1969)). More precisely, if $\mu_m(\phi)$ denotes the fraction of structures of size m that satisfy the formula ϕ, we have that $\lim_{m\to\infty} \mu_m(\phi)$ always exists and is either zero or one. Such 0-1 laws have been established for various extensions of first order logic (Blass et al. 1985; Anuj Dawar 1995; Kolaitis and Vardi 1992; Bars 1998).

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

However, even in cases where the 0-1 laws fail, the class of almost surely true sentences is coherently determined, and it is natural to ask for the algorithmic complexity of deciding this property for a given formula. It turns out that the transition from 'absolute truth' to 'truth with probability one' can reduce the complexity qualitatively. For instance, almost sure validity for first-order logic in the finite is PSPACE-complete (Grandjean 1983), whereas validity in the finite is undecidable, by Trachtenbrot's theorem.

In this paper, we will give an algorithm to determine the limit density $\frac{|L_m|}{|\Sigma|^m}$ as $m \to \infty$ of a regular language L, where $L_m = L \cap \Sigma^m$. The density of regular languages has already been studied in (Berstel 1972), and the methodology using formal power series is standard by now (Salomaa and Soittola 1978). It is known that there are finitely many rational accumulation points. But the algorithmic answer to the above questions is not or only implicitely given. In particular, the exact complexity of computing the limit density remains unclear.

We show that checking the existence of the limit and computing it in the case of existence can be done in time $O(dn^3)$, where n is the number of states of the automaton and d its periodicity. If the underlying graph G of the automaton is strongly connected, the *periodicity* is defined as the greatest common divisor of the length of all cycles in G. For the general case, d is the least common multiple of the periodicities of all strongly connected components of G. The periodicity d may grow exponentially with the size of the automaton, although it should be very small in practice. Nonetheless, our algorithm can be formulated to run in nondeterministic polynomial time.

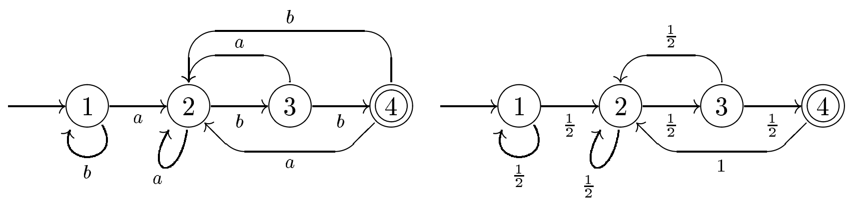


Figure 3.1: Simple automaton (with limit density $\frac{1}{4}$) and the corresponding Markov process.

The task can be seen as a special instance of a convergence problem of Markov chains. We say a matrix $A = (a_{ij})_{i,j}$ with non-negative rational entries is a *DFA-matrix* iff the sum of the elements in each column is one. A corresponds to some complete deterministic finite automaton $\mathcal{A} = (\Sigma, \{1, \ldots, n\}, \delta, F)$ if a_{ij} is the number of edges from j to i, divided by $|\Sigma|$ (see Figure 3.1). When viewed as a Markov process, a stationary distribution would make it easy to compute the percentage of words accepted in the limit. If the initial distribution is given by the vector s (e.g. corresponding to the the initial state of the automaton), we say that $\mathcal{A}$ *converges to* $p \in [0,1]$ if $\lim_{m\to\infty} \sum_{i\in F}(A^m s)_i$ exists and is equal to p.

However, Figure 3.2 shows that the limit density might exist although the Markov chain is not ergodic. In particular, we do not obtain a stationary distribution for any of the leaf components when running the automaton on $(1,0,0,0,0,0)$. A *leaf component* is a strongly connected component where no edge leaves the component.

An implementation of the algorithm is available at `http://fsinfo.cs.uni-sb.de/~timohome/regDensity/`.

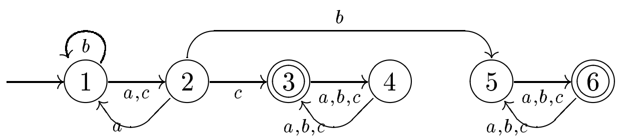


Figure 3.2: Periodic automaton accepting a language of density $\frac{1}{2}$

2 Computing the density of a regular language

Let L be the regular language accepted by a DFA $\mathcal{A}$. Let A be its DFA-matrix and s its starting vector. We describe the algorithm to compute the accumulation points of $\left(\frac{L_m}{|\Sigma|^m}\right)_m$, or, in case of existence, the limit. The first step of the algorithm is a reduction to the case when the periodicity of the automaton is 1.

In order to get rid of a periodicity $d > 1$, consider the matrix A^d. It is easily seen that A^d is again a DFA-matrix, and it can be interpreted as the matrix of the automaton $\mathcal{A}^d$ which is defined as follows: The alphabet of $\mathcal{A}^d$ is Σ^d, the states are the same as those of $\mathcal{A}$. There is an edge $i \to j$ in $\mathcal{A}^d$ with label $w = w_1 \cdots w_d$ iff there is a path in $\mathcal{A}$ from i to j with edge labels $w_1, \ldots, w_d$.

```
DENSITY(𝒜)
1  A ←DFA-matrix(𝒜)
2  s ←StartingVector(𝒜)
3  d ←COMPUTEPERIODICITY(A)
4  for i = 0, ..., d − 1
5  do l_i ←1-DENSITY(A^d, A^i s)
6  return {l_0, ..., l_{d−1}}
```

Figure 3.3: The main procedure

Obviously, the automaton $\mathcal{A}^d$ has periodicity 1. Instead of computing the densities of $\mathcal{A}$ with starting vector s we compute the densities of $\mathcal{A}^d$ with starting vectors $s, As, \ldots A^{d-1}s$. Because of proposition 2.1, the set of accumulation points are the same. The intuition behind this is that running $\mathcal{A}$ k steps of length 1 or $\frac{k}{d}$ steps of length d yields the same. If k is not divisible by d, the starting point of the run has to be postponed.

ComputePeriodicity($\mathcal{A}$)
for each strongly connected component C
do Fix $v \in C$, $L \leftarrow \{v\}$, $p_C \leftarrow 0$
  for $j = 1, \ldots, |C|$
  do $L \leftarrow \bigcup_{l \in L}$ Out $(l) \cap C$;
    if $v \in L$ **then** $p_C = \text{GCD}(p_C, j)$
return $\text{LCM}\{p_C | \, C$ strongly connected component$\}$

Figure 3.4: Computing the periodicity in $O(n^3)$.

Proposition 2.1. *Let A be a DFA-Matrix, and s be a starting vector and $d \in \mathbb{N}$. Let S be the set of accumulation points of $(A^m s)_{m\in\mathbb{N}}$. Let S_i, $0 \leq i \leq d-1$ be the set of accumulation points of $\left((A^d)^m(A^i s)\right)_{m\in\mathbb{N}}$. Then $S = S_1 \cup \ldots \cup S_n$.*

In terms of automata this reads: Given the matrix A of a DFA and a starting vector s, then the accumulation points of the densities $\frac{L_m}{|\Sigma|^m}$ can be obtained by calculating the accumulation points of the densities of A^d with starting vectors $s, As, \ldots, A^{d-1}s$.

Proof. If v is an accumulation point, then there is a subsequence such that $A^{m_j}s \to v$. Write $m_j = k_j d + r_j$, $0 \leq r_j < d$. Then at least one of the numbers $1, \ldots, d$, say k, appears infinitely often as a r_j. This gives a partial sequence of $\left((A^d)^m(A^k s)\right)_m$.

Conversely, the identity $(A^d)^{m_j}(A^k s) = A^{dm_j+k}s$ shows that each accumulation point of $\left((A^d)^m(A^k s)\right)_m$ is also an accumulation point of $(A^m s)_m$. $\square$

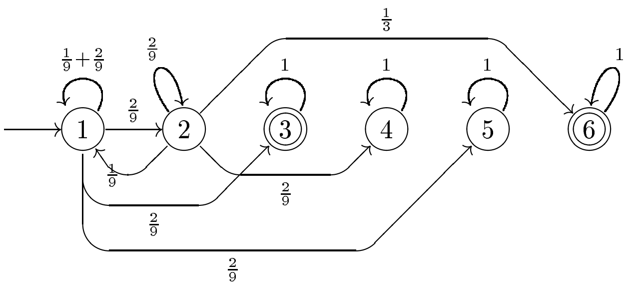


Figure 3.5: A visualisation of the squared transition matrix of the last example

3 Computing the density of an aperiodic DFA-matrix

Let $\mathcal{A}$ be an automaton that is aperiodic. Unfortunately, $\mathcal{A}$ is not necessarily strongly connected, so we can not apply the basic limit theorem (see Chapter 4 on the following page) directly. Instead, we use the following modified version:

Let $J \subseteq \{1, ..., n\}$ be the set of indices of all vertices that are in non-leaf components. Let $I_k \subseteq \{1, ..., n\}$ be the set of vertices in the kth leaf component with $k = 1, ..., K$, where K is the number of different leaf components. Let b_k be the unique convergence vector for each I_k that exists because of the basic limit theorem (filled with zeroes on all components that are not in I_k). Then there exist λ_k such that

$$\lim_{m \to \infty} A^m s = \sum_{k=1}^{K} \lambda_k b_k. \tag{3.1}$$

For a formal proof of this fact see the correctness section. While the b_k are independent of the starting vector, the λ_k must be calculated with respect to s. Considering fig. 3.2, we can easily observe that the sum of the distributional densities in vertices 1 and 2 will approach zero. Still it is not easy to determine which fraction will be led into the leaf component $\{3, 4\}$ and how much to $\{5, 6\}$. To overcome this problem, we apply two reduction rules on non-leaf components that change the automaton but preserve the limit distribution:

> **loop reduction:** If a vertex $j \in J$ has a loop to itself, it is omitted and the other edge weights are scaled such that their sum equals 1 (see fig. 3.7 on the next page).
>
> **edge reduction:** If a vertex $j \in J$ has an edge to $i \in J$ with edge weight a_{ij} and all edges from i end in p vertices $k_1, ...k_p$, then the edge from j to i is replaced by p edges from j to k_l, $l = 1, ..., p$, with edge weight $a_{ij} \cdot a_{k_l i}$. If an edge from j to k_l has already existed, then $a_{ij} \cdot a_{k_l i}$ is added to the edge weight.

It is easily seen that the resulting matrix of a reduction is again a DFA-matrix. The proof that these rules preserve the limit can be found in the correctness section.

If we eliminate the loop to a vertex $j \in J$ and afterwards apply the second reduction rule on all edges to j, there will remain no edge to j. As all edges leading to j that are inserted in the reduction rules lead to target vertices of former edges, further applications of reduction rules will create no edges to j. If we do this for an arbitrary fixed sequence of vertices, it is guaranteed that no further reductions are applicable, and the result is an

1-DENSITY($\mathcal{A}, s$)
for i : i is not in a leaf component
do reduceSelfLoop(i)
    reduceEdgesTo(i)
for $k = 0, \ldots, K$
do $\lambda_k \leftarrow \sum_{i \in I_k} < s, A_i >$
    $b_k \leftarrow$Eigenvector to eigenvalue 1 of I_k
return $\sum_{k=1}^{K} \lambda_k b_k$

Figure 3.6: Computing the density of a strongly connected component.

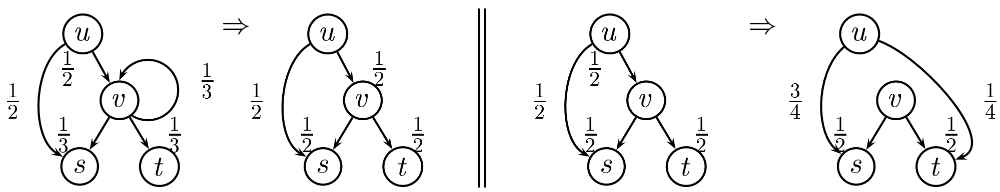


Figure 3.7: Loop reduction and edge reduction.

automaton $\mathcal{A}'$ that has no edges within J. So within one step in $\mathcal{A}'$, the probability of all states in J will be zero. The λ_k can then be obtained by adding up the probability of all vertices in I_k. Now that we have calculated all λ_k with respect to s, it remains to calculate the b_k, i.e. the unique limit distribution of a strongly connected component of $\mathcal{A}$. For this purpose, we restrict A and b_k to the entries of I_k and ignore the rest. $A' \in \mathbb{Q}_+^{|I_k| \times |I_k|}$ and $b'_k \in \mathbb{Q}_+^{|I_k|}$ will denote these objects. The basic limit theorem yields that $\lim_{m \to \infty} A'^m s = b'_k$ for all starting vectors. b'_k is the eigenvector of A' corresponding to eigenvalue 1, scaled to a column sum of 1. So we can easily compute b'_k by solving the homogenous equation system $A' - I = 0$ with the Gaussian elimination algorithm. We do this for all I_k and obtain all b_k, and using λ_k we compute the limit of $A^m s$. The sum of the entries of the accepting states equals the probability that a word will be accepted by $\mathcal{A}$ given the initial probability distribution s.

4 Correctness of the algorithm

We have to prove the statements of Section 3, namely 3.1 on the preceding page and the correctness of loop and edge reductions. We first repeat a standard result for Markov chains, the basic limit theorem. See for instance

(Chang 2000) for a good introduction to Markov processes and a proof of the theorem.

Theorem 4.1 (Basic Limit Theorem). *Let A be a DFA-matrix and the corresponding graph $\mathcal{A}$ be strongly connected and aperiodic. Then $\lim_{m\to\infty} A^m s$ exists for all starting vectors s, and it is independent of s.*

As mentioned above, we cannot apply the theorem directly, since the graph of the automaton is in general not strongly connected. The idea is to concentrate on the leaf components and to apply Theorem 4.1 there. The limit probability of nodes in non-leaf components is zero since for almost all words the automaton will eventually be in a node of a leaf component. This is the subject of the following theorem:

Theorem 4.2. *Let A be a DFA-matrix. Let $I = \{i \mid i \text{ is in a leaf component}\}$ be the set of all nodes which are in leaf components of $\mathcal{A}$, and let $J = \{j \mid j \text{ is not in a leaf component}\}$ be the set of remaining nodes. Consider, for any starting vector s, $(A^m s)_j$, which denotes the j-th entry of the vector $A^m s$. Then*

$$\lim_{m\to\infty} (A^m s)_j = 0.$$

Proof. Let $s_m = \sum_{j\in J}(A^m s)_j$. Then, $(s_m)_{m\in\mathbb{N}}$ is a sequence, which is bounded by 0 and which decreases monotonously. In fact, s_m equals the probability that at step n the automaton is in a node of a non-leaf component. Since there is no path from leaf component nodes to non-leaf component nodes, this probability cannot increase: $\sum_{j\in J}(A^{m+1}s)_j \leq \sum_{j\in J}(A^m)_j$. Therefore $(s_m)_m$ converges to a limit a. We shall show that $a = 0$. Let $\gamma = \min\{a_{kj} \mid j \in J,\, a_{kj} \neq 0\}$ the minimal edge weight which is not zero for all edges starting in a node in J. Let $\epsilon > 0$ and $\delta = \frac{\epsilon\gamma^{|J|}}{|J|}$. Since $(s_m)_{m\in\mathbb{N}}$ converges, there is a $m_0 \in \mathbb{N}$ such that $\forall m \geq m_0 : |s_m - a| \leq \delta$. Let k be any vertex in J that has an edge to a vertex in I. Then $s_{m+1} \leq s_m - (A^{m'}s)_k \cdot \gamma$. Therefore $(A^m s)_k \leq \frac{\delta}{\gamma}$ (because otherwise, s_{m+1} would be smaller than a, which is impossible).
Let k' be an arbitrary node in J. By definition, there is a path from k' to a node in I. Let $r \leq |J|$ be the length of this path, and let k be the last node in J on this path. Then, since each edge weight on this path is at least γ, we have $(A^{m+r}s)_k \geq (A^m s)_{k'} \cdot \gamma^{r-1}$. Therefore, since $r \leq |J|$,

$$(A^m s)_{k'} \leq (A^{m+r}s)_k \;\leq\; \frac{\delta}{\gamma^r} \;\leq\; \frac{\delta}{\gamma^{|J|}}$$
$$\Rightarrow \sum_{j\in J}(A^m s)_j \;\leq\; |J|\frac{\delta}{\gamma^{|J|}} \;=\; \epsilon.$$

□

We can now prove the statement 3.1 on page 27.

Theorem 4.3. *Let A be a DFA-matrix such that the leaf components of the corresponding graph $\mathcal{A}$ are aperiodic. For all leaf components of $\mathcal{A}$, let $I_k := \{i \mid i \text{ is in the } k\text{−th leaf component}\}$. Let $J = \{j \mid j \text{ is not in a leaf component}\}$ denote the remaining subgraph. Let s be any fixed starting vector. Let, for all leaf components I_k, $b^{(k)}$ be the vector that has in its entries with indices from I_k the unique limit for I_k, which exists due to the basic limit theorem, and which is 0 on its other components. Then* $\lim_{m\to\infty} A^m s$ *exists and is a linear combination of the b_k, i.e.*

$$\lim_{m\to\infty} A^m s = \sum_k \lambda_k b^{(k)}.$$

Proof. The idea of the proof is the following: Since the leaf components are aperiodic and strongly connected, the basic limit theorem states that, given a starting vector, they converge to a limit which is independent of this starting vector. The problem here is that the components I_k do not 'run' on a fixed starting vector s. Instead, they get their 'input' from some non-leaf components. Moreover, the initial distributions on the I_k change with the non-leaf components, and the initial distributions are not given by starting vectors, since their sum need not be 1 (remember that a starting vector is characterized by the fact that the sum of its entries equals 1). But we already know that the distributions in the non-leaf component nodes converge to zero, which implies that the initial distributions of the I_k components stabilize. More precisely, consider the following.
Let $\mu_k^{(m)} = \sum_{i\in I_k}(A^m s)_i$. We show that $(\mu_k^{(m)})_{m\in\mathbb{N}}$ converges, and denote the limit by λ_k. Let $\epsilon > 0$ and K be the number of leaf components. Because of the convergence of the distributions of the non-leaf components, there exists a m_1 such that $\forall m > m_1 : |\sum_{j\in J}(A^m s)_j| < \frac{\epsilon}{2K}$. Since I_k is a leaf component, the distributional fraction of I_k remains in I_k. Therefore,

$$\begin{aligned} \mu_k^{(m+m)} &\leq \mu_k^{(m)} + \sum_{i\in J}(A^m s)_j \\ \Rightarrow\ |\mu_k^{(m+m)} - \mu_k^{(m)}| &\leq |\sum_{i\in J}(A^m s)_j|\ \leq\ \epsilon \qquad\qquad \forall m \geq m_1. \end{aligned}$$

This means that $(\mu_k^{(m)})m \in \mathbb{N}$ is a cauchy sequence.
Let $\epsilon > 0$, and m_1 be such that

$$|\mu_k^{(m_1)} - \lambda_k| \leq \frac{\epsilon}{2K|b^{(k)}|}. \tag{3.2}$$

Let P_k be the projection on the indices of I_k. By the basic limit theorem, given a (normalized) starting vector, I_k converges to $b^{(k)}$, which is independent of the starting vector. The vector $P_k A^{m_1} s$ is not necessarily normalized, but the sum of its entries equals $\mu_k^{(m_1)}$. Therefore the distribution of I_k with starting vector $A^{m_1}s$ converges to $\mu_k^{(m_1)}b^{(k)}$. Thus there is a m_2 such that for $m \geq m_2$

$$|P_k A^m (A^{m_1} s) - \mu_k^{(m_1)} b^{(k)}| < \frac{\epsilon}{2K}. \tag{3.3}$$

For $m \geq m_1 + m_2$, this implies

$$\begin{aligned}
\left| A^m s - \sum_k \lambda_k b^{(k)} \right| &= \left| \sum_k \left(P_k A^{m-m_1} A^{m_1} s \right) - \sum_k \lambda_k b^{(k)} \right| \\
&\leq \sum_k \left| P_k A^{m-m_1} A^{m_1} s - \mu_k^{(m_1)} b^{(k)} \right| + \sum_k \left| \mu_k^{(m_1)} b^{(k)} - \lambda_k b^{(k)} \right| \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon, \qquad \text{by (3.2) and (3.3).}
\end{aligned}$$

This completes the proof. □

5 Correctness of the reduction rules

The intuition behind both, loop and edge reductions, is the same: By 'redirecting' the probability distribution via (new) edges of appropriately scaled weights, the sum of the weights along all paths from some i to some j remains the same. Whereas before paths of length two or more were considered, now any such path consists of fewer edges. However, we are interested in the limit distribution only, and so this does not make any difference.

Lemma 5.1 (Minor perturbation). *Let A, B be two DFA-matrices which are equal except for the i-th column. We assume that both $(A^m s)$ and $(B^m s)$ are convergent for all vectors s. Moreover, for both A and B we assume $(\lim_{m\to\infty} A^m s)_j = (\lim_{m\to\infty} B^m s)_j = 0$ for all $j \in J$, the set of all nodes in node leaf components. Then $\lim_{m\to\infty} B^m s = \lim_{m\to\infty} \lim_{k\to\infty} A^k B^m s$, for all s.*

Proof. Since both limits are assumed to exist, all that remains to be shown is equality. Let $\epsilon > 0$, s any vector and v be such that $\lim_m B^m s = v$. Hence, we can choose k such that for all $k' > k$, $\|v - B^{k'} s\|_1 < \frac{\epsilon}{2}$. By assumption, $v_j = 0$ for all $j \in J$, so we obtain $\sum_{j\in J} (B^{k'} s)_j < \frac{\epsilon}{2}$.

Let $u := B^{k'} s$ and let u' be equal to u except for the entries with indices in J, where u' is zero. Since these components in u add up to a number lower or equal ϵ, we know that $||u' - u||_1 \leq \epsilon$. As u' is zero on all components of J, it is obviuos that for arbitrary m, $A^m u' = B^m u'$. Now both A and B preserve the $||.||_1$-norm, in particular $||(u' - u)||_1 = ||A(u' - u)||_1$. Using the triangular inequality we obtain

$$\begin{aligned}||A^m u - B^m u||_1 =& ||(A^m - B^m)u||_1 \\ =& ||(A^m - B^m)(u' + (u - u')||_1 \\ \leq& ||(A^m - B^m)u'||_1 + ||(A^m - B^m)(u - u')||_1 \\ \leq& ||(A^m u' - B^m u'||_1 + ||A^m(u - u')||_1 + ||B^m(u - u')||_1 \\ \leq& 0 + \frac{\epsilon}{2} + \frac{\epsilon}{2} \;=\; \epsilon.\end{aligned}$$

Hence, $\lim_{m\to\infty} \|A^m B^{k'} s - B^m B^{k'} s\|_1 < \epsilon$ for any $k' > k$, which yields

$$\lim_{m\to\infty} B^m s = \lim_{k\to\infty} \lim_{m\to\infty} A^m B^k s.$$

$\square$

For any DFA-matrix A we denote by J_i the set of nodes for which i is reachable in the graph corresponding to A, i.e. $J_i := \{j \mid \text{exists a path from } j \text{ to } i\}$. Correctness of the elimination of self loops follows from

Lemma 5.2. *Let A be a DFA-matrix for $\mathcal{A}$ such that the underlying graph is aperiodic in all of its leaf components. Let i be a node in a non-leaf component, thus in particular $a_{ii} < 1$. Let B be the DFA-matrix obtained from A by replacing a_{ii} by 0 and normalizing the column to 1, i.e. multiplying each entry in column i by $\frac{1}{1-a_{ii}}$. Moreover, let us assume that both $(lim_{m\to\infty} A^m s)_j = (lim_{m\to\infty} B^m s)_j = 0$ for all vectors s and $j \in J_i$. Then B converges for all vectors s to the* same *limit as A does, i.e. $\lim_{m\to\infty} A^m s = \lim_{m\to\infty} B^m s$.*

Proof. First observe that B is again the DFA-matrix of some automaton whose underlying graph is aperiodic in all of its leaf components, since reduction rules are only applied to nodes in non-leaf components. By Theorem 4.3 on page 30 this implies both (A^m) and (B^m) converge on all vectors s.

Let s_i denote the i-th component of a vector s, and $\overline{s_i}$ denote the projection on its i-th component. Then, taking $b = A\overline{s_i}$, $Bs = A(s - \overline{s_i}) + \frac{1}{1-a_{ii}}(b - \overline{b_i})$ by definition of B, and $b = (b - \overline{b_i}) + \overline{b_i} = (b - \overline{b_i}) + a_{ii}\overline{s_i}$. Intuitively, b gives the distribution after one step as obtained by starting in i, and this can be split into the selfloop part and the rest.

We obtain

$$\begin{aligned}\lim_{m\to\infty} A^m \overline{s_i} &= \lim_{m\to\infty} A^{m-1} b \\ &= \lim_{m\to\infty} A^{m-1}((b - \overline{b_i}) + a_{ii}\overline{s_i}) \\ &= \lim_{m\to\infty} A^{m-1}(b - \overline{b_i}) + a_{ii} \lim_{m\to\infty} A^m \overline{s_i}\end{aligned}$$

since, by assumption, A^m is convergent.
Thus, $\lim_{m\to\infty} A^m\overline{s_i} = \frac{1}{1-a_{ii}} \lim_{m\to\infty} A^m(b-\overline{b_i})$. Now

$$\begin{aligned}\lim_{m\to\infty} A^m s &= \lim_{m\to\infty} A^m(s-\overline{s_i}+\overline{s_i}) \\ &= \lim_{m\to\infty} A^{m+1}(s-\overline{s_i}) + \lim_{m\to\infty} A^m\overline{s_i} \\ &= \lim_{m\to\infty} A^{m+1}(s-\overline{s_i}) + \frac{1}{1-a_{ii}}\lim_m A^m(b-\overline{b_i}).\end{aligned}$$

Again, since A is assumed to be convergent this gives

$$\begin{aligned}\lim_{m\to\infty} A^m s &= \lim_{m\to\infty}(A^{m+1}(s-\overline{s_i}) + \frac{1}{1-a_{ii}}A^m(b-\overline{b_i})) \\ &= \lim_{m\to\infty} A^m(A(s-\overline{s_i}) + \frac{1}{1-a_{ii}}(b-\overline{b_i})) \\ &= \lim_{m\to\infty} A^m Bs.\end{aligned}$$

By induction we obtain $\lim_m A^m s = \lim_m A^m B^k s$ for all k. Hence

$$\lim_{m\to\infty} A^m s = \lim_{k\to\infty}\lim_{m\to\infty} A^m B^k s$$

and using 5.1 on page 31 we conclude $\lim_m A^m s = \lim_m B^m$. □

Likewise, the correctness of our rules for the reduction of edges in non-leaf components follows immediately from

Lemma 5.3. *Let A be a DFA-matrix as above, and let $i \neq k$ be two nodes in non-leaf components of the underlying graph. Let B denote the matrix obtained from A by replacing a_{ki} by zero and a_{li} by $a_{li} + a_{ki}\cdot a_{lk}$ for all $l \neq k$. Again, let us assume that $(\lim_{m\to\infty} A^m s)_j = (\lim_{m\to\infty} B^m s)_j = 0$ for all $j \in J_i$. Then B converges for all vectors s to the* same *limit as A does, i.e. $\lim_{m\to\infty} A^m s = \lim_{m\to\infty} B^m s$.*

Proof. As above, we observe that both A and B converge for all vectors s, so it remains to be shown that the limits are equal.
If $A = (a_{ik})$ then let b be the vector with zero entries except for the k-th component, where it is $a_{ki}\cdot s_i$. Let $b' = Ab$, for then $Bs = As - b + b'$, by definition of B. Thus,

$$\begin{aligned}\lim_{m\to\infty} A^m s &= \lim_{m\to\infty} A^m As \\ &= \lim_{m\to\infty} A^m(Bs - b' + b) \\ &= \lim_{m\to\infty} A^m Bs - \lim_{m\to\infty} A^m(Ab) + \lim_{m\to\infty} A^m b \\ &= \lim_{m\to\infty} A^m Bs.\end{aligned}$$

As in the previous proof, $\lim_m A^m s = \lim_m A^m B^k s$ for all k, by induction on k. Hence we can apply Lemma 5.1 on page 31 and obtain $\lim_m A^m s = \lim_m B^m$. □

6 Conclusion and Outlook

In this paper we gave an upper bound for deciding whether a deterministic finite automaton defines a dense regular language. The algorithm we presented can be formulated such that the problem is seen to be in coNP. The nondeterministic version of the algorithm accepts if all the accumulation points equal one. Trading time for space, it might be possible to show that the problem is in PSPACE, even if we are given a *nondeterministic* automaton.

Several other questions remain open: Is there a polynomial time algorithm for the problem, i.e. can we get rid of checking d values? The next step towards computing the limit density of more expressive formalisms might be to look at context free languages. Related to this, we could ask whether finite automata accepting regular languages over finite trees have computable limit densities.

References

Anuj Dawar, E. G. (1995). Generalized Quantifiers and 0-1 Laws. In *Proc. 10th IEEE Sypm. on Logic in Computer Science*, IEEE Computer Society Press, pp. 54–64.

Bars, H.-M. L. (1998). Fragments of existential second-order logik without 0-1 laws. *Logic in Computer Science*, 525–536.

Berstel, J. (1972). Sur la densité de langages formels. *ICALP*, 345–358.

Blass, A., Y. Gurevich, and D. Kozen (1985). A zero-one law for logic with a fixed point operator. *Journal of Information and Conrol 67*, 70–90.

Chang, J. (2000). Stochastic Processes. Lecture notes, `http://pantheon.yale.edu/~jtc5/251/`.

Fagin, R. (1976). Probabilities on Finite Models. *Journal of Symbolic Logic 41*, 50–58.

Grandjean, E. (1983). Complexity of the First-Order Theory of Almost All Finite Structures. *Information and Control 57*, 180–204.

Kolaitis, P. G. and M. Y. Vardi (1992). 0-1 laws and decision problems for fragments of second order logic. *Journal of Information and Computation 98*, 258–294.

Salomaa, A. and M. Soittola (1978). *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag.

Y. V. Gelbskij, D.I. Kogan, M. L. and V. Talanov (1969). Range and degree of realizability of formulas in the restricted predicate calculus. *Kibernetika 2*, 17–28.

On the Logic of Proofs

VLADIMIR BREZHNEV
Department of Mathematical Logic, Faculty of Mechanics and Mathematics,
Moscow State University, 119899 Moscow, Russia
brezhnev@lpcs.math.msu.ru

ABSTRACT. The Logic of Proofs (**LP**) was introduced by S. N. Artemov. It is a system in the propositional language with additional sort of objects – proof terms, and an extra atomic propositions $[t]F$ with intended reading "t is a proof of F". **LP** is an explicit counterpart of modal logic **S4**, since **S4** is the exact term-forgetting projection of **LP**. In the present paper we build sequent system corresponding to fragment of **LP** sufficient to realize **S4**. Using its format we build explicit counterparts of modal logics **K**, **K4**, **D**, **D4** and **T**. Also we define versions of this systems for which the intended interpretation of terms is proof tactics, computable procedures able to find proofs of some formulas, having formula as input.

1 Introduction

In 1933 Gödel informally specified modal logic **S4** as logic for provability and left open the question about its exact intended semantics. The straightforward reading of $\Box F$ as "F is provable in certain formal system" contradicts Gödel's incompleteness theorem. S. N. Artemov gave the solution of this problem in (Artemov 1995; Artemov 2001). In (Artemov 1995) he introduced the Logic of Proofs (**LP**). It is a system in the propositional language with additional sort of objects – proof terms, and an extra atomic propositions $[t]F$ with intended reading "t is a proof of F". Gabbay's Labelled Deductive Systems (cf. (Gabbay 1994)) serves a natural framework for **LP**. **LP** is sound and complete with respect to the the arithmetical proof interpretations. **LP** realizes all the theorems of **S4**, and thus provides the formal provability semantics to it.

S4 has all axioms and rules of propositional logic in the modal language along with the necessitation rule $F \vdash \Box F$ and the modal axiom schemes:

AK. $\Box(F \to G) \to (\Box F \to \Box G)$ "distributivity"
AT. $\Box F \to F$ "reflexivity"
AK4. $\Box F \to \Box\Box F$ "transitivity"

The language of **LP** contains: the usual Boolean connectives and sentence variables; *proof variables* $p_0, p_1, \ldots$; *axiom constants* $a_0, a_1, \ldots$; functional symbols !, + and $\times$; brackets [ ] ( ). Terms and formulas: proof

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

variables and axiom constants are terms; if s and t are terms, then $!t$, $(s+t)$ and $(s \times t)$ are terms; sentence variables and Boolean constants are formulas; Boolean connectives behave in the usual way, and if t is a term and F is a formula, then $[t]F$ is a formula. We write st instead of $(s \times t)$ and skip parentheses when convenient. The system $\mathbf{LP}_{AS}$ has following axiom schemes:

A0. The tautologies in the language of **LP**.
A1. $[t]F \to F$ "verification"
A2. $[t](F \to G) \to ([s]F \to [t \times s]G)$ "application"
A3. $[t]F \to [!t][t]F$ "proof checker"
A4. $[s]F \to [s+t]F$, $[t]F \to [s+t]F$ "choice"
AS. A finite set of formulas of the form $[c]A$, where c is an axiom constant, and A is an axiom A0-A4. "axiom specification"

The only rule is modus ponens. **LP** is the generic name for the systems $\mathbf{LP}_{AS}$ of the various axiom specifications.

Intended semantics of operations ! and $\times$ is quite natural: ! builds the verification of the proof, $\times$ corresponds to modus ponens. Traditional interpretation of $+$ is the operation of concatenation of two multi-conclusion proofs. If we assume that the usual Hilbert style derivations proves all its formulas, we get an example of multi-conclusion proof system.

Axiom specification corresponds to the particular case of necessitation rule. The general case has its counterpart too. If $\mathbf{LP} \vdash A$ then for some term t of axiom consants $\mathbf{LP} \vdash [t]A$ (cf. (Artemov 1995)).

The result of substitution $\Box$ for all occurences of $[\cdot]$ in axioms of **LP** are theorems of **S4**. The same is true for all theorems of **LP**.

By **LP**-*realization* of modal formula F we mean an assignment of proof terms to all occurrences of modality $\Box$ in F. An **LP**-realization is *normal* if all negative occurrences[1] of modality are realized by proof variables. Let F^r be an image of F under a realization r. If $\mathbf{S4} \vdash F$, than $\mathbf{LP}_{AS} \vdash F^r$ for some normal realization r and axiom specification AS (cf. (Artemov 1995)). In this sense **LP** realizes **S4**.

The epistemic meaning of modality suggests that a modal formula $\Box A \to \Box B$ implicitly specifies a function $f(x)$ such that if x is a justification of A then $f(x)$ is a justification of B. **LP** makes this consideration formal. Indeed, negative occurences of modality in the normal realization of the modal formula are lebeled by proof variables and positive ones by **LP**-terms – functions of proof variables.

The logic **LP** is a version of **S4** presented in a more rich operational language, with no information being lost, since **S4** is the exact term-forgetting

[1] Positive and negative occurrences of modality in a formula and a sequent: the outmost occurrence of modality in formula $\Box F$ is positive; any occurrence of modality from F in $G \to F$, $\Box F$, and $\Gamma \vdash \Delta, F, \Phi$ has the same polarity as the corresponding occurrence of modality in F; any occurrence of $\Box$ from F in $F \to G$ and $\Gamma, F, \Pi \vdash \Delta$ has a polarity opposite to that of the corresponding occurrence of $\Box$ in F.

projection of **LP**. Naturally there is an interest in finding explicit counterparts for other modal logics than **S4** and a common procedure of building an explicit counterpart. In this paper we are trying to approach this goal.

We define sequential logic of proofs, system **LSP**(**S4**). This system may be regarded as minimal explicit counterpart of modal logic **S4**. Artemov's realization procedure has a cut-free derivation in the sequent version of **S4** as its input and labels occurences of modality in this derivation. Thus, derivations of **LP** sufficient for realization of **S4** are realizations of cut-free sequent derivations of **S4**. In our system we make this correspondance explicit. Derivations of **LSP**(**S4**) are exactly derivations of sequent cut-free formulation of **S4** with additional labels for occurences of modality.

Using format of **LSP**(**S4**) we easily find explicit counterparts for some modal logics for which cut-free sequent formulations are known. This logics are **K**, **K4**, **D**, **D4** and **T**. All of them are subsystems of **S4**.

In section 4 we define systems $\mathbf{LSP}_T$. Terms of this systems can be interpreted as *proof tactics*. Proof tactics or simply *tactics* are totally computable procedures able to find proofs of some formulas, having formula as input. In section 5 we build arithmetical interpretation of system $\mathbf{LSP}_T(\mathbf{S4})$.

2 Systems LSP

In this systems we derive pairs of the form "*term*:*sequent*", where the *term* encodes the proof of the *sequent*, like terms of typed lambda calulus (cf. (Girard et al. 1989; Sorensen and Urzyczyn 1998)) or justifications of typed logic (cf. (Constable 1998)). We call sequent ψ derivable if there is a term α for which $\alpha : (\psi)$ is derivable.

By capital Greek letters we denote comma-separated lists of formulas. By i, j, k, m, n we denote integer numbers.

The language of **LSP** contains: symbols Ax^i, $i >= -1$, Wl, Wr, Cl, Cr, Tl^j, Tr^j, $j \geq 0$, M, T, D; labels $l_1, l_2, \ldots$; semicolon ";"; proof variables $p_1, p_2, \ldots$; functional symbols !, + (we omit $\times$ in the language); propositional variables $S_1, S_2, \ldots$; Boolean connectives $\rightarrow$, $\perp$; brackets ( ), [ ].

Sequent is an expression of the form $\Gamma \vdash \Delta$, where Γ and Δ are lists of words in the language of **LSP**. We do not define terms and formulas explicitly, instead we state that if $\alpha : (A_1, \ldots, A_n \vdash A_{n+1}, \ldots, A_{n+m})$ is derivable then α is well-formed term, and A_i are well-formed formulas.

Rules. First, we define system corresponding to classical propositional calculus. We call this system **LSP**. Axioms:

$$Ax^{-1} : (\perp \vdash\ ) \qquad Ax^0 : (\perp \vdash \perp) \qquad Ax^i : (S_i \vdash S_i)\,, i \geq 1$$

Weakening:

$$\frac{\alpha : (\Gamma \vdash \Delta) \quad \beta : (A, \Pi \vdash \Phi)}{Wl(\alpha;\beta) : (A, \Gamma \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta) \quad \beta : (\Pi \vdash \Phi, A)}{Wr(\alpha;\beta) : (\Gamma \vdash \Delta, A)}$$

Weakening rules look slightly unusual – they has two premises each. Additional premises appear to guarantee that A is a well-formed formula.

Contraction:

$$\frac{\alpha : (A, A, \Gamma \vdash \Delta)}{Cl(\alpha) : (A, \Gamma \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta, A, A)}{Cr(\alpha) : (\Gamma \vdash \Delta, A)}$$

Transposition:

$$\frac{\alpha : (\Gamma, A, B, \Pi \vdash \Delta)}{Tl^i(\alpha) : (\Gamma, B, A, \Pi \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta, A, B, \Phi)}{Tr^j(\alpha) : (\Gamma \vdash \Delta, B, A, \Phi)},$$

where i is the number of commas before the indicated occurence of A in the premise of the left rule and j is the number of commas after the indicated occurence of B in the premise of the right rule.

Implication introduction rules:

$$\frac{\alpha : (\Gamma \vdash \Delta, A) \quad \beta : (B, \Pi \vdash \Phi)}{Il(\alpha; \beta) : (A \to B, \Gamma, \Pi \vdash \Delta, \Phi)} \qquad \frac{\alpha : (A, \Gamma \vdash \Delta, B)}{Ir(\alpha) : (\Gamma \vdash \Delta, A \to B)}$$

The system **LSP** is defined. Now we construct its extensions corresponding to modal logics. Rule $(\vdash_K \Box)$:

$$\frac{\alpha : (A_1, \ldots, A_n \vdash B)}{M(\alpha; p_1 \ldots p_n; l) : ([p_1]A_1, \ldots, [p_n]A_n \vdash [t]B)},$$

if l is an unique label, $t = t_1 + \cdots + t_k$, every t_i has form $\alpha_i p_{i1} \ldots p_{in_i}$, α_i are well-formed terms, p_{ij} are proof variables, and for some k, $t_k = \alpha p_1 \ldots p_n$. We call expressions of this form *normal proof polynomials* or simply *proof polynomial.* $\mathbf{LSP}(\mathbf{K}) = \mathbf{LSP} + (\vdash_K \Box)$.

In the case of this rule we do not store all the information required to recover the proof of the sequent in the term explicitly. Instead we use unique labels. We do this to allow additional summands in the proof polynomial to contain the term itself.

By *forgetful projection* $\mathcal{D}^0$ of **LSP**-derivation $\mathcal{D}$ we mean derivation obtained by omitting left part terms and substituting $\Box$ for all occurences of labeled modality $[\cdot]$ in sequents. By forgetful projection $\mathcal{L}^0$ of system $\mathcal{L}$ we mean system which derivations are forgetful projections of derivations of system $\mathcal{L}$. Forgetful projection of $\mathbf{LSP}(\mathbf{K})$ corresponds to modal logic **K** (propositional calculus plus axiom AK, plus necessitation rule). We call this system **GK**. There are axioms $S_i \vdash S_i$ and $\bot \vdash \bot$ in this system, but not $A \vdash A$ for any formula A, as usual. One can see that all the traditional axioms are derivable from atomic ones not using cut. $\mathbf{LSP}^0$ is sequential system for classical propositional logic.

Rule $(\vdash_{K4} \Box)$:

$$\frac{\alpha : (A_1, \ldots, A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash B)}{M(\alpha; p_1 \ldots p_n; l) : ([p_1]A_1, \ldots, [p_n]A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash [t]B)},$$

$t = t_1 + \cdots + t_k$, every t_i has form $\alpha_i p_{i1} \ldots p_{in_i}!q_{i1} \ldots !q_{im_i}$, α_i are well-formed terms, p_{ij} and q_{ij} are proof variables, and for some k, $t_k = \alpha p_1 \ldots p_n!q_1 \ldots !q_m$; l is an unique label. $\mathbf{LSP(K4)} = \mathbf{LSP} + (\vdash_{K4}\Box)$. $\mathbf{GK4} = \mathbf{LSP(K4)}^0$. **GK4** corresponds to modal logic $\mathbf{K4} = \mathbf{K}$ plus axiom AS4.

Rule $(\Box\vdash_D)$:

$$\frac{\alpha : (A_1, \ldots, A_n \vdash\ )}{D(\alpha; p_1 \ldots p_n) : ([p_1]A_1, \ldots, [p_n]A_n \vdash\ )}$$

$\mathbf{LSP(D)} = \mathbf{LSP(K)} + (\Box\vdash_D)$. $\mathbf{GD} = \mathbf{LSP(D)}^0$. **GD** corresponds to $\mathbf{D} = \mathbf{K} + \mathrm{AD}$, $\mathrm{AD} = \Box\bot \to \bot$.

Rule $(\Box\vdash_{D4})$:

$$\frac{\alpha : (A_1, \ldots, A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash\ )}{D(\alpha; p_1 \ldots p_n) : ([p_1]A_1, \ldots, [p_n]A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash\ )}$$

$\mathbf{LSP(D4)} = \mathbf{LSP(K4)} + (\Box\vdash_{D4})$. $\mathbf{GD4} = \mathbf{LSP(D4)}^0$. **GD4** corresponds to $\mathbf{D4} = \mathbf{K4} + \mathrm{AD}$.

Rule $(\Box\vdash_T)$:

$$\frac{\alpha : (A, \Gamma \vdash \Delta)}{T(\alpha; p) : ([p]A, \Gamma \vdash \Delta)}$$

$\mathbf{LSP(T)} = \mathbf{LSP(K)} + (\Box\vdash_T)$. $\mathbf{GT} = \mathbf{LSP(T)}^0$. **GT** corresponds to $\mathbf{T} = \mathbf{K} + \mathrm{AT}$. $\mathbf{LSP(S4)} = \mathbf{LSP(K4)} + (\Box\vdash_T)$. $\mathbf{GS4} = \mathbf{LSP(S4)}^0$. **GS4** corresponds to $\mathbf{S4} = \mathbf{K4} + \mathrm{AT}$.

3 Modal logics, LSP and LP

3.1 Realization of Modal Logics

Theorem 3.1. *For every derivation $\mathcal{D}$ of $\mathbf{G}\mathcal{L}$ there is derivation $\mathcal{D}^r$ of $\mathbf{LSP}(\mathcal{L})$ such that $(\mathcal{D}^r)^0 = \mathcal{D}$, where $\mathcal{L}$ is one of the logics* **K**, **K4**, **D**, **D4**, **T** *and* **S4**. *We call $\mathcal{D}^r$ realization of $\mathcal{D}$.*

Proof. We build realization by induction on the modal derivation. During this process we label sequents by terms encoding there proofs and label occurences of modality by proof variables or proof polynomials. Realization of most rules is straighforward.

In the rules introducing modality some new proof variables appear. We use new proof variables each time. In rules $(\vdash\Box)$ we use proof polynomials containing only one essential summand. The only step that remains unclear is contraction. The problem is that corresponding boxes in two occurences of the affected formula may be labeled by different terms.

The occurrences of modality in a derivation are *related* if they are the corresponding occurrences of the related formulas in premises and conclusions of rule; we extend this relation by transitivity. All occurrences of modality in a derivation are split into disjoint *families* of the related ones. In a cut-free derivation the related occurrences of modality have the same polarity, so we define *polarity of a family* as polarity of its members.

Rules $(\vdash \Box)$ introduce positive and negative occurences of modality, rules $(\Box \vdash)$ introduce negative occurences. There are no other sources of modality.

Contraction. From $A, A, \Gamma \vdash \Delta$ infer $A, \Gamma \vdash \Delta$. We already have a derivation of $\alpha : (B, C, \Gamma^r \vdash \Delta^r)$ where $B^0 = C^0 = A$. Let us rebuild the derivation so that $B = C$. Negative occurences of modality are labeled by proof variables, so we substitute proof variables of B for corresponding proof variables of C in the derivation. Positive occurences of modality are introduced by one of the rules $(\vdash \Box)$ for same formula A on the right hand, and labeled by proof polynomials. Let proof polynomials r and s label corresponding occurences of modality in B and C. Let t be the the minimal proof polynomial containing all summands of r and s. Substitute t for all occurences of r and s in the derivation. After this substitutions all rules remain correct, and both B and C became the same formula, so we can apply contraction. $\Box$

Remark 3.2. In **LSP**-derivations positive families of modality are labeled by proof polynomials. Sometimes it is possible to throw out some non-essential summands of these polynomials while keeping the derivation correct. We call $\mathbf{LSP}_R(\mathcal{L})$ subsystem of $\mathbf{LSP}(\mathcal{L})$ which derivations does not contain any non-essential summands in their proof polynomials. Our realization procedure leads to $\mathbf{LSP}_R$-derivation.

Realization procedure applied to the forgetful projection of the $\mathbf{LSP}_R$-derivation $\mathcal{D}$ leads to the derivation very similar to $\mathcal{D}$. Namely, the only possible differences are another proof variables, another numbers of labels, and another order of summands in proof polynomials. In this sense our systems may be regarded as minimal explicit counterparts of corresponding sequent modal systems.

Remark 3.3. Another interesting feature of the systems $\mathbf{LSP}_R$. For every proof polynomial $t = t_1 + \cdots + t_n$ in the $\mathbf{LSP}_R$-derivation there is a formula A such that for some Π_i, sequents $\Pi_i \vdash [t_i]A$ are derivable. On the semantic level it means that for every i under certain circumstances (namely, Π_i) t_i proves A. It gives one a chance to interpret plus as the operation that chose a well-formed proof from two candidates. This operation isn't restricted to multi-conclusion proofs.

3.2 Embedding into LP

Theorem 3.4. *If $A_1, \dots, A_n \vdash B_1, \dots, B_m$ ($A_1, \dots, A_n \vdash$, $\vdash B_1, \dots, B_m$) is derivable in $\mathbf{LSP}(\mathcal{L})$, then there is derivable in* **LP** *formula $\bigwedge C_i \to \bigvee D_j$ ($\bigwedge C_i \to \bot$, $\top \to \bigvee D_j$), where C_i and D_j are obtained from A_i and B_j by substituting* **LSP***-terms by some terms of axiom constants.*

Proof. Induction on the **LSP**-derivation. Formulas corresponding to axioms are propositional tautologies.

Weakening, transposition and rules introducing implication have one of the forms

$$\frac{\Gamma \vdash \Delta}{\Pi \vdash \Phi} \quad \text{and} \quad \frac{\Gamma_1 \vdash \Delta_1 \quad \Gamma_2 \vdash \Delta_2}{\Pi \vdash \Phi},$$

and $(\bigwedge \Gamma \to \bigvee \Delta) \to (\bigwedge \Pi \to \bigvee \Phi)$ or $(\bigwedge \Gamma_1 \to \bigvee \Delta_1) \to (\bigwedge \Gamma_2 \to \bigvee \Delta_2) \to (\bigwedge \Pi \to \bigvee \Phi)$ is a tautology. Take derivations of formulas corresponding to premises, add suitable tautology and derive formula corresponding to the conclusion using modus ponens.

Rule $(\Box\vdash_T)$. Take derivation of formula corresponding to premise, add axiom $[p]A \to A$ and derive the formula corresponding to the conclusion using suitable tautology.

Rule $(\vdash_{K4}\Box)$. Formula $\bigwedge A_i \wedge \bigwedge [q_j]B_j \to A$ is derivable, then for some term α of axiom constants it is derivable $[\alpha](\bigwedge A_i \wedge \bigwedge [q_j]B_j \to A)$ (cf. (Artemov 1995)); using axioms A2 derive $\bigwedge [p_i]A_i \wedge \bigwedge [!q_j][q_j]B_j \to [t]$, $t = \alpha p_1 \dots p_n !q_1 \dots !q_m$; using axioms A3 derive $\bigwedge [p_i]A_i \wedge \bigwedge [q_j]B_j \to [t]$; finally using axioms A4 derive $\bigwedge [p_i]A_i \wedge \bigwedge [q_j]B_j \to [t_1 + \dots + t + \dots + t_k]$, the formula corresponding to the conclusion. **LSP** terms in additional summands t_i are substituted by some unique axiom constants (place holders).

Rules $(\vdash_K\Box)$, $(\Box\vdash_D)$ and $(\Box\vdash_{D4})$ are processed along the same lines as rule $(\vdash_{K4}\Box)$, using axioms $[t]\bot \to \bot$ in the last two cases.

Contraction. Take a derivation of formula $A \wedge B \wedge \bigwedge C_i \to \bigvee D_j$ corresponding to the premise, substitute some place holders in the derivation by terms of axiom constants to make A and B the same formula, add suitable tautology and derive formula corresponding to the conclusion. $\Box$

4 Systems $\mathbf{LSP}_T$

Intended interpretation of terms of this systems is proof tactics, computable functions mapping formulas or sequents to proofs. Both parts of the pair *term:sequent* are necessary and sufficient to build the proof.

The language of $\mathbf{LSP}_T$ proof polynomials contains: symbols At, Ifl, Ifr, Mfl, Mfl, Ax, Wl, Wr, Cl, Cr, Tl^j, Tr^j, M, T, D^j, $j \geq 0$; proof variables p_1, p_2, …; functional symbols ! and +; brackets ( ); semicolon ";". In this and next sections any word in the language of $\mathbf{LSP}_T$ proof polynomials we call proof polynomial. Parts of a proof polynomial separated by + we call its summands.

The language of $\mathbf{LSP}_T$ contains: the language of $\mathbf{LSP}_T$ proof polynomials; propositional variables S_1, S_2, …; Boolean connectives $\bot$ and $\to$; square brackets [ ]; sequent formation symbols $\vdash$, comma "," and $\bullet$.

Rules. Formula construction:

$$At : (\bot \vdash \bullet) \qquad At : (S_i \vdash \bullet) \qquad At : (\bullet \vdash \bot) \qquad At : (\bullet \vdash S_i)\,, i \geq 1$$

$$\frac{\alpha : (\bullet \vdash A) \quad \beta : (B \vdash \bullet)}{Ifl(\alpha;\beta) : (A \to B \vdash \bullet)} \qquad \frac{\alpha : (A \vdash \bullet) \quad \beta : (\bullet \vdash B)}{Ifr(\alpha;\beta) : (\bullet \vdash A \to B)}$$

$$\frac{\alpha : (A \vdash \bullet)}{Mfl(\alpha) : ([p]A \vdash \bullet)} \qquad \frac{\alpha : (\bullet \vdash A)}{Mfr(\alpha) : (\bullet \vdash [t]A)},$$

if p is a proof variable and t is a proof polynomial.
Axioms:

$$Ax : (\bot \vdash\ ) \qquad Ax : (S_i \vdash S_i)\,, i \geq 1$$

Weakening:

$$\frac{\alpha : (\Gamma \vdash \Delta) \quad \beta : (A \vdash \bullet)}{Wl(\alpha;\beta) : (A, \Gamma \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta) \quad \beta : (\bullet \vdash A)}{Wr(\alpha;\beta) : (\Gamma \vdash \Delta, A)}$$

Contraction:

$$\frac{\alpha : (A, A, \Gamma \vdash \Delta)}{Cl(\alpha) : (A, \Gamma \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta, A, A)}{Cr(\alpha) : (\Gamma \vdash \Delta, A)}$$

Transposition:

$$\frac{\alpha : (\Gamma, A, B, \Pi \vdash \Delta)}{Tl^i(\alpha) : (\Gamma, B, A, \Pi \vdash \Delta)} \qquad \frac{\alpha : (\Gamma \vdash \Delta, A, B, \Phi)}{Tr^j(\alpha) : (\Gamma \vdash \Delta, B, A, \Phi)},$$

where i is the number of commas before the indicated occurence of A in the premise of the left rule and j is the number of commas after the indicated occurence of B in the premise of the right rule.

Implication introduction rules:

$$\frac{\alpha : (\Gamma \vdash \Delta, A) \quad \beta : (B, \Pi \vdash \Phi)}{Il(\alpha;\beta) : (A \to B, \Gamma, \Pi \vdash \Delta, \Phi)} \qquad \frac{\alpha : (A, \Gamma \vdash \Delta, B)}{Ir(\alpha) : (\Gamma \vdash \Delta, A \to B)}$$

The system $\mathbf{LSP}_T$ is defined.
Systems $\mathbf{LSP}_T(\mathbf{K})$, $\mathbf{LSP}_T(\mathbf{K4})$, $\mathbf{LSP}_T(\mathbf{D4})$, $\mathbf{LSP}_T(\mathbf{T})$ and $\mathbf{LSP}_T(\mathbf{S4})$:
Rule $(\vdash_K \Box)$:

$$\frac{\alpha : (A_1, \ldots, A_n \vdash B)}{M : ([p_1]A_1, \ldots, [p_n]A_n \vdash [t]B)},$$

if t is a proof polynomial and one of its summands is $\alpha p_1 \ldots p_n$.
Rule $(\vdash_{K4} \Box)$:

$$\frac{\alpha : (A_1, \ldots, A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash B)}{M : ([p_1]A_1, \ldots, [p_n]A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash [t]B)},$$

if t is a proof polynomial and one of its summands is $\alpha p_1 \ldots p_n !q_1 \ldots !q_m$.
Rule $(\Box \vdash_D)$:

$$\frac{\alpha : (A_1, \ldots, A_n \vdash\ )}{D^0(\alpha) : ([p_1]A_1, \ldots, [p_n]A_n \vdash\ )}$$

Rule $(\Box\vdash_{D4})$:

$$\frac{\alpha : (A_1, \ldots, A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash\ )}{D^m(\alpha) : ([p_1]A_1, \ldots, [p_n]A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash\ )}$$

Rule $(\Box\vdash_T)$:

$$\frac{\alpha : (A, \Gamma \vdash \Delta)}{T(\alpha) : ([p]A, \Gamma \vdash \Delta)}$$

Remark 4.1. It is possible to construct the system of $\mathbf{LSP}_T$-tactics corresponding to $\mathbf{LSP}_T$-terms in the following sense: for a word ψ in the language of $\mathbf{LSP}_T$, $\alpha : \phi$ is derivable if and only if tactic corresponding to α gives consistent result on input ψ. Tactics corresponding to symbols At and Ax just check format of its input and if it matches return a derivation containing corresponding axiom. Other tactics are trying to recover premises of the corresponding rule from its input, call tactics corresponding to its parameters and construct the derivation of its input using the corresponding rule of inference. Tactic corresponding to M has no parameters. It uses information stored in the proof polynomial.

5 Arithmetical Interpretation of $\mathbf{LSP}_T(\mathbf{S4})$

We denote $Proof(x, y)$ usual Gödel proof predicate for Peano Arithmetic ($\mathbf{PA}$). $Proof(x, y)$ is a Δ_1-formula. $\mathbf{PA} \vdash Proof(\overline{x}, \overline{y})$ iff y is the code of the last formula in the proof with code x. We denote the empty word Λ. We assume that $\mathbf{PA}$ has all propositional tautologies as its axioms.

Definition 5.1. An arithmetical interpretation $*$ of $\mathbf{LSP}_T(\mathbf{S4})$ has the following parameters: an evaluation of propositional letters by sentences of arithmetic, an evaluation of proof variables by natural numbers.

Proof polynomials are interpreted by tactics. $\bot^* = \bot$, $(A \to B)^* = (A^* \to B^*)$, $(A \vdash \bullet)^* = (A^* \to \top)$, $(\bullet \vdash A)^* = (\bot \to A^*)$, $(\Gamma \vdash\ )^* = (\bigwedge \Gamma^* \to \bot)$, $(\ \vdash \Delta)^* = (\top \to \bigvee \Delta^*)$, $(\Gamma \vdash \Delta)^* = (\bigwedge \Gamma^* \to \bigvee \Delta^*)$ (here conjunctions are grouped to the right and disjunctions are grouped to the left), $([p]A)^* = Proof(\overline{p^*}, \overline{A^*})$, if p is a proof variable, $([t]A)^* = Proof(\overline{n}, \overline{A^*})$, if t is a proof polynomial and n is a natural number, the code of the result of tactic t^* on input A^*.

Interpretation of proof polynomials. Here we use I for input of a tactic.

Tactics Ax, At, $Ifl(\alpha; beta)$, $Ifr(\alpha; beta)$, $Mfl(\alpha)$ and $Mfr(\alpha)$: if I has form $(A \to A)$, $(\bot \to A)$ or $(A \to \top)$ then return proof containing only one formula I (axiom). Otherwise return Λ.

Tactics $Cl(\alpha)$: if input has form $(A \wedge B \to C)$, then call α^* on input $(A \wedge A \wedge B \to C)$, if it returns Λ then return Λ, otherwise add to output of α^* tautology $(A \wedge A \wedge B \to C) \to (A \wedge B \to C)$ and finally derive $(A \wedge B \to C)$ using modus ponens. Otherwise return Λ.

Interpretation of tactics of the forms $Cr(\alpha)$, $Tl^i(\alpha)$, $Tr^i(\alpha)$, $Wl(\alpha; beta)$, $Wr(\alpha; beta)$, $Il(\alpha; beta)$, $Ir(\alpha)$ and $T(\alpha)$ goes along the same lines. While building an interpretation of $T(\alpha)$ we use the fact that $\mathbf{PA} \vdash Proof(\overline{n}, \overline{A}) \to A$. Indeed, consider two possibilities. If $Proof(\overline{n}, \overline{A})$ is true, then n is the proof of A, thus $\mathbf{PA} \vdash A$, and $\mathbf{PA} \vdash Proof(\overline{n}, \overline{A}) \to A$. Otherwise $Proof(\overline{n}, \overline{A})$ is false, it is a Δ_1-formula, thus $\mathbf{PA} \vdash \neg Proof(\overline{n}, \overline{A})$, and again $\mathbf{PA} \vdash Proof(\overline{n}, \overline{A}) \to A$.

Tactics of the form αp, if p is a proof variable, and α does not contain $+$, and the last symbol of α is not !: if p^* is a code of correct proof, then take its last formula A and call α^* on input $A \to I$. If its result is not Λ then add to it the proof with the code p^*, and finally derive I using modus ponens. Otherwise return Λ.

Tactics of the form $\alpha!p$, if p is a proof variable, and α does not contain $+$: if p^* is a code of correct proof, then take its last formula A and call α^* on input $Proof(\overline{p^*}, \overline{A}) \to I$, if its result is not Λ then search for the proof $!p^*$ of the formula $Proof(\overline{p^*}, \overline{A})$ (it can be found because this Δ_1-formula is true, and thus derivable), add it to the result of α^*, and derive I using modus ponens. Otherwise return Λ.

Tactics of the form $\alpha + \beta$: if the result of $\alpha^*(I)$ is not Λ then return it, return $\beta^*(I)$ otherwise.

Tactics of the form M: if input I has other form than $\bigwedge Proof(\overline{p_i}, \overline{A_i}) \to Proof(\overline{n}, \overline{A})$ return Λ. If for all i, p_i^* proofs A_i and n does not proof A return Λ. Otherwise I is a true Δ_1 formula, it is derivable, and proof of it can be found effectively.

Tactics corresponding to proof polynomials of other forms: return Λ.

Theorem 5.2. *If $\alpha : (\Gamma \vdash \Delta)$ is derivable in $\mathbf{LSP}_T(\mathbf{S4})$ and $*$ is an arithmetical interpretation of $\mathbf{LSP}_T(\mathbf{S4})$ then $\mathbf{PA} \vdash (\Gamma \vdash \Delta)^*$ and α^* can find a proof of it.*

Proof. Proof is a straightforward induction on the $\mathbf{LSP}_T$-derivation. Most comments can be found in the definition of the interpretation of proof polynomials. The only step that remains unclear is rule $(\vdash_{K4}\Box)$.

$$\frac{\alpha : (A_1, \ldots, A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash B)}{M : ([p_1]A_1, \ldots, [p_n]A_n, [q_i]C_1, \ldots, [q_m]C_m \vdash [t]B)},$$

is t is a proof polynomial containing summand $t_0 = \alpha p_1 \ldots p_n!q_1 \ldots !q_m$. If one of p_i^* does not proof A_i^* or one of q_i^* does not proof C_i^* then interpretation of the conclusion is a true formula, and M^* is able to find a proof of it. Let p_i^* prove A_i^* and q_i^* prove C_i^*. α^* is able to find a proof of $\bigwedge A_i^* \wedge \bigwedge Proof(\overline{q_i^*}, \overline{C_j^*}) \to B^*$, then $(\alpha p_1 \ldots p_n)^*$ can find a proof of $\bigwedge Proof(\overline{q_i^*}, \overline{C_j^*}) \to B^*$, then t_0^* can find a proof of B^*, t_0 is one of the summands of t, then t^* can find a proof of B^*, thus $Proof(\overline{t^*(B^*)}, \overline{B^*})$ is true, and M^* can find a proof of the interpretation of the conclusion of the rule. $\Box$

References

Artemov, S. (1995). Operational modal logic. Technical Report MSI 95-29, Cornell University. http://www.math.cornell.edu/~artemov/MSI95-29.ps.

Artemov, S. (2001). Explicit provability and constructive semantics. *Bulletin of Symbolic Logic 6*(1). http://www.math.cornell.edu/~artemov/BSL.ps.

Chagrov, A. and M. Zakharyaschev (1997). *Modal Logic*. Oxford Science Publications.

Constable, R. (1998). Types in Logic, Mathematics and Programming. In S. Buss (Ed.), *Handbook of Proof Theory*, Chapter 10. Elsevier Science.

Gabbay, D. M. (1994). *Labelled Deductive Systems*. Oxford University Press.

Girard, J.-Y., Y. Lafont, and P. Taylor (1989). *Proofs and Types*. Cambridge University Press.

Sorensen, M. H. and P. Urzyczyn (1998). Lectures on the Curry-Howard Isomorphism. Technical report, University of Copenhagen. ftp://ftp.diku.dk/diku/semantics/papers/D-368.ps.gz.

Information exchange as reduction

Balder ten Cate
ILLC - University of Amsterdam
b.ten.cate@hum.uva.nl

Abstract. A formal framework is introduced for analysing pragmatic aspects of cooperative information exchange, based on update semantics as well as the theory of abstract reduction systems. The framework provides us with the means to analyse communicative strategies of multiple participants in information exchange dialogues. It is shown how the system can be applied to the analysis of *relevance*.

1 Introduction

The general aim of this paper is to provide a formal framework for analysing *pragmatic aspects of cooperative information exchange*. By *information exchange*, communication is meant involving multiple participants, consisting of the exchange of information, in contrast to, e.g., argumentative discourse. *Cooperative* means that (a) the participants fully trust each other, and (b) they have no interest in lying, misleading or hiding information. The pragmatic aspects of these types dialogues concern goals and strategies of the participants. Clearly, we restrict ourselves to a very specific type of discourse. This will allow us to develop an interesting theory of such dialogues. At the end of the paper, we will address the question to what extent the present theory can be generalised to other types of discourse.

A more specific aim of this paper is to apply the developed framework to the issue of *relevance*. We will see how this can help us to analyse different notions of relevance that have been proposed in the literature.

Our starting point is Stalnaker's theory of assertions. Stalnaker (1978) observed that in making assertions, people take some of their private information and make it common ground. In terms of possible world semantics, they eliminate possibilities from the common ground.

At any point in a dialogue, people make choices concerning which information to exchange (which possibilities to eliminate). Typically, there

[0] I'd like to thank Paul Dekker, Darrin Hindsill, Jan Willem Klop, Rosja Mastop, Marie Nilsenová, Robert van Rooy, Yde Venema and the three anonymous reviewers for their useful comments and discussions.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

are many alternatives to choose from. Notwithstanding this apparent divergence, there is a clear direction in which the conversation proceeds. During the conversation, the information states of the participants are adjusted to each other, converging to a (hypothetical) situation in which there is no information left to communicate. In this situation, all participants have the same information (i.e., the information that used to be distributed between them). Obviously, the situation is never reached in practice.[1]

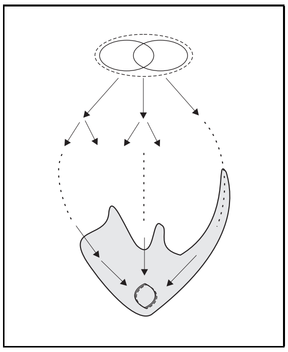


Figure 5.1: Process of information exchange

Assuming for the moment that there are only two participants involved, these considerations lead to a view on the process of cooperative information exchange (CIE) that is depicted in Figure 5.1. In this picture, the top three ovals correspond to the information states of the agents and to the common ground (thought of as sets of possibilities). At the "end state" (at the bottom of the picture) all three circles coincide. This corresponds to the hypothetical state that we discussed. The arrows in the picture correspond to assertions. A dialogue, then, is essentially a walk through this graph.

In CIE dialogues, the participants usually have a specific communicative goal, for instance to resolve a decision problem (van Rooy 2000). In general, each participant can have his or her own goal, but for the moment we will assume that the participants have a common goal. This common goal is represented by the outlined area in the bottom half of the picture (the goal is to reach one of the states in this area).

Relevance can be analysed as a strategy for achieving a goal. Intuitively, irrelevant utterances are utterances that will not help the participants to achieve their goal.

In the rest of this paper, we will formalise the picture that we just sketched. In doing so, we will make use of Abstract Reduction Systems theory. ARS's form a field of research in theoretical computer science that is concerned with the reduction of terms or more abstract objects to a normal form. Combinatory logic and the lambda-calculus are the prime examples, but many other reduction systems have been devised, for instance concerning braids and knots. For a general introduction, the reader is referred to Klop (1992) or Baader and Nipkow (1998).

2 Information states and utterances

We start out by defining Stalnakerian 'contexts', which we will refer to as states. A state specifies the information that each agent has, as well as the

[1]Notice that this is an idealised model of information exchange. In practice, many complications arise (misunderstandings, mistakes, lying, etc.) that we will not address.

information that is common ground. Such states can be modeled in various ways, but for present purposes, the following definition will suffice. Let P be a propositional alphabet and $\mathcal{A}$ a finite set of agents. Let V be the set of all valuations over P. Then we define states as follows (where c stands for the common ground):

Definition 1 (States). *A state is a function* $\sigma : \mathcal{A} \cup \{c\} \to \wp(V)$ *such that* $\forall a \in \mathcal{A} : \sigma(a) \subseteq \sigma(c)$. Σ *is the set of all states.*

In other words, we have a set of agents and the common ground and each of these is assigned a set of valuations representing its information. Notice that it is required that the common ground contains less information than each of the participants, which is only natural.

We will sometimes write $\sigma \cap \tau$ for $\lambda a.(\sigma(a) \cap \tau(a))$. Likewise, we will write $\sigma \subseteq \tau$ instead of $\forall a : \sigma(a) \subseteq \tau(a)$.[2]

Two central notions from update semantics are *support* and *update*. Typically, these notions are defined in terms of individual information states. Here, we will use the same notions, but define them in terms of multi-agent states.

Definition 2 (Support). $\sigma \Vdash_a \phi$ *if* $\forall v \in \sigma(a) : v \models \phi$

This definition says that agent a supports ϕ ("knows that ϕ") in state σ whenever ϕ is true in all the situations a considers possible.

Definition 3 (Update). $\sigma + \phi = \lambda a.\{v \in \sigma(a) \mid v \models \phi\}$

According to this definition, when the agents collectively update with a sentence ϕ, then all possibilities are eliminated in which ϕ is not the case (from the private information sets as well as from the common ground).

In terms of support and update, we can now define the valid "moves of the game", which correspond to assertions. Recall that we focus on *cooperative* information exchange, meaning that the participants (a) trust each other, and (b) have no intent to deceive each other. These two assumptions are reflected in the next definition, using the notions of update and support respectively.

[2]This notation is justified by the fact that type-theoretically, the given definition is equivalent to one in which a state is a relation between agents and possibilities.

Definition 4 (Utterances).

i. $\sigma \xrightarrow{a:\phi} \tau$ *if* $\sigma \Vdash_a \phi$ *and* $\sigma \nVdash_c \phi$ *and* $\tau = \sigma + \phi$

ii. $\sigma \xrightarrow{a} \tau$ *if* $\sigma \xrightarrow{a:\phi} \tau$ *for some* ϕ

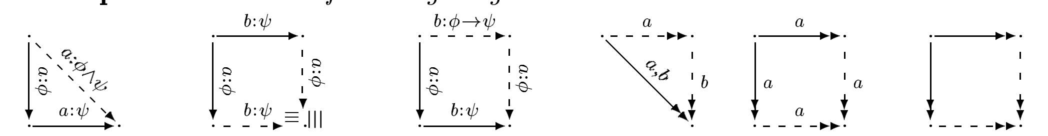
iii. $\sigma \longrightarrow \tau$ *if* $\sigma \xrightarrow{a} \tau$ *for some* $a \in \mathcal{A}$

$\sigma \xrightarrow{a:\phi} \tau$ means that in state σ, agent a can utter the sentence ϕ, resulting in the new state τ. According to this definition, an agent can only utter a proposition if he or she knows the content to be true, and if the information is not already common ground. The effect of the utterance is then a collective update with the proposition uttered.

Notice that the set of all states Σ is closed under $\longrightarrow$, i.e., for all $\sigma \in \Sigma$ and ϕ, $(\sigma + \phi) \in \Sigma$. Also, if $\sigma \longrightarrow \tau$, then $\tau \subset \sigma$, since the effect of an update is always elimination of possibilities. We will refer to the reflexive closure of $\xrightarrow{a}$ by $\xrightarrow{a}^{\equiv}$, and we will refer to the reflexive, transitive closure of $\xrightarrow{a}$ by $\xrightarrow{a}\!\!\!\!\!\rightarrow$.

The next proposition states some properties of the $\rightarrow$'s. The diagrams should be read as follows: whenever the situation occurs that is pictured by the solid arrows, then we can extend this configuration with the dotted arrow. For instance, the *transitivity* diagram says that whenever $\sigma \xrightarrow{a:\phi} \tau$ and $\tau \xrightarrow{a:\psi} \nu$, we have that $\sigma \xrightarrow{a:\phi\wedge\psi} \nu$.

Proposition 1. *The following diagrams hold.*



(transitivity) (subcommutativity) (swap) (PP $\xrightarrow{a}, \xrightarrow{b}$) (CR for $\xrightarrow{a}$) (CR for $\longrightarrow$)

If we conceive of $\longrightarrow$ as a reduction relation, then a normal form is a state that cannot be reduced any further, i.e., a state in which there is no information left to exchange. The last diagram of the above proposition tells us that $\longrightarrow$ is confluent (has the Church-Rosser property). This implies that every state reduces to at most one normal form. Now we can ask ourselves which states have such a normal form, and what it looks like.

If the propositional alphabet P is finite, then the answer to the question is quite simple. In that case, $\longrightarrow$ is Strongly Normalising, which means that the normal form is always reached in a finite number of steps (there are no infinite reduction sequences). Furthermore, $nf(\sigma) = \lambda a. \bigcap_{b \in \mathcal{A}} \sigma(b)$, i.e., the normal form of σ is a state in which each agent has the same information, namely the intersection of the old information states of the agents.[3] It follows that σ is in normal form precisely if $\forall a \in \mathcal{A} : \sigma(a) = \sigma(c)$

In the general case, things are a bit more complicated. In general, we do not have Strong Normalisation (or even Weak Normalisation). This means that if the agents keep on exchanging information, this process might not come to an end. Furthermore, even if a normal form is reached, it is not guaranteed that in the final situation, the agents all have the same information. It might happen that one agent has more information then another, but he cannot communicate this information (for instance if that requires an infinitely long formula). In order to give precise characterisations of the normal forms of our system, we need to introduce the auxiliary notion of *saturation*. This is defined in terms of ultrafilters.

Definition 5 (Saturation). $\hat{\sigma} = \lambda a.\{v \in V \mid \exists u \in Uf(V) : \sigma(a) \in u \;\&\; \forall p \in P : \{w \mid w(p) = v(p)\} \in u\}$

Technically speaking, $\hat{\sigma}$ is the smallest saturated superset of σ. Furthermore, $\hat{\sigma}$ is equivalent to σ, in the sense that it satisfies the same formulae ($\hat{\sigma} \Vdash_a \phi$ iff $\sigma \Vdash_a \phi$).[4] Having the notion of saturation at our disposal, we can generalise our results to the case where P is infinite.

Proposition 2.

i. If σ has a normal form, then $nf(\sigma) = \lambda a.[\sigma(a) \cap \bigcap_{b \in \mathcal{A}} \hat{\sigma}(b)]$.

ii. σ is in normal form iff $\forall a \in \mathcal{A} : \hat{\sigma}(a) = \hat{\sigma}(c)$

In the absence of normal forms, *cofinal reduction sequences* often play a similar role. A reduction sequence $\sigma_1 \longrightarrow \sigma_2 \longrightarrow \ldots$ is called *cofinal* if $\forall \tau : \sigma \twoheadrightarrow \tau \Longrightarrow \exists i : \tau \twoheadrightarrow \sigma_i$. The following proposition tells us that if P is countably infinite, then for every σ (even if σ doesn't have a normal form) there are cofinal reduction sequences starting with σ, and they converge to $\lambda a.[\sigma(a) \cap \bigcap_{b \in \mathcal{A}} \hat{\sigma}(b)]$.

Proposition 3.

i. If $\sigma_1 \longrightarrow \sigma_2 \longrightarrow \ldots$ is a cofinal reduction sequence, then $\bigcap_i \sigma_i = \lambda a.[\sigma_1(a) \cap \bigcap_{b \in \mathcal{A}} \hat{\sigma_1}(b)]$.

ii. If P is countable, then cofinal reduction sequences exist for every σ.

This finishes our discussion of the properties of our ARS. In the next section it will become clear why it makes sense to conceive of $\longrightarrow$ as a reduction relation.

[3]Notice the lambda abstraction over a, which does not occur in the rest of the formula. This means that $nf(\sigma)$ is a constant function, assigning the same information to each agent (and to the common ground).

[4]Cf. Blackburn et al. (To appear) for an introduction to Ultrafilters and saturatedness. For present purposes, one can think of $\hat{\sigma}$ as a state that is almost like σ but not exactly.

3 Goals and strategies

At this point we have formalised most aspects of Figure 5.1. However, what we have left out of consideration so far is the *goal* of the information exchange. We will now reintroduce the notion of a communicative goal.

Definition 6 (Goal). *A goal G is a set of states that is closed under reduction, i.e.* $\sigma \in G \ \&\ \sigma \longrightarrow \tau \Longrightarrow \tau \in G$

Remember that we explicitly assume that the participants have no interest in hiding information from each other. This assumption is reflected by the fact that the set of satisfactory states is closed under further reduction. An example of a goal is $\{\sigma \mid \sigma \text{ is a normal form }\}$, a rather trivial example that illustrates that our notion of a goal is really a generalisation of *reaching the normal form*.[5] A more interesting range of examples is generated by the following definition.

Definition 7 (The goal to resolve an issue). $G_{\phi?} = \{\sigma \mid \sigma \Vdash_c \phi \textit{ or } \sigma \Vdash_c \neg\phi\}$.

$G_{\phi?}$ is the goal to resolve the issue whether ϕ (in other words, it is the goal to reach a state in which it is common ground whether ϕ).

An important question is: what is a good strategy for reaching a particular goal? Given a specific goal, four requirements can be formulated with respect to a strategy.

1. If the goal is feasible (i.e., if it is possible to reach the goal), the goal should be reached if the participants follow the strategy.

2. Second, the goal should be reached as quickly as possible.

3. If the goal cannot be reached, then at some point the strategy should tell you to give up and stop wasting your time.

4. The strategy should be easy to apply (i.e., complexity-wise).

In this paper, we focus on the first of these requirements.

The quest for efficient strategies is not particular to CIE dialogues, but is central to the theory of reduction systems (which is why using ARS's to analyse CIE is useful). It was already established in the previous section that our reduction relations are confluent. The importance of confluence for information exchange can be explained as follows.

Suppose a goal G can be reached from a state σ, which is to say, there is a state τ such that $\sigma \twoheadrightarrow \tau$ and $\tau \in G$. Then confluence says that *no matter what the participants do, the opportunity to reach the goal is never*

[5] The goal of reaching a weak head normal form in lambda calculus also falls under this general notion of a goal, as the set of weak head normal forms is closed under β-reduction.

lost. Formally, for any state σ', if $\sigma \twoheadrightarrow \sigma'$, then there is a state τ' such that $\sigma' \twoheadrightarrow \tau'$.

This does not mean that it is easy to reach the goal. In order to reach the goal, it is still important to have a good strategy. Finding good strategies in the presence of confluence is what ARS theory is concerned with.

Let us start out by defining what strategies are. We are concerned with several agents, and each of them can follow their own strategy. Therefore, we define *strategy profiles* as tuples of reduction strategies, one for each agent (where the individual reduction strategies are defined in the usual way). Also, we must require that the individual strategy of an agent does not make reference to the private information of another agent. This leads to the following definition.

Definition 8 (Strategy profiles). *A strategy profile $\mathcal{F}$ is a tuple $(\mathbb{F}_a)_{a\in\mathcal{A}}$ such that for each agent $a \in \mathcal{A}$,*

- $\mathbb{F}_a$ *is a sequential reduction strategy for* $\xrightarrow{a}$
- $\forall \sigma, \tau : \sigma(a) = \tau(a)$ & $\sigma(c) = \tau(c) \Longrightarrow \mathbb{F}_a(\sigma) = \mathbb{F}_a(\tau)$

Given a state and a strategy profile, we can ask ourselves what will happen if the agents start communicating in accordance to their strategies. In addition to the strategies of the individual players, we then also need to specify a particular system of turn taking: which agent is at turn to say something at a given moment? At this point, it seems most sensible to make only the minimal requirement of *fairness*: every agent should have more or less an equal oppurtunity to say something. We define an $\mathcal{F}$-dialogue to be a reduction sequence that is both in accordance to $\mathcal{F}$ and fair.

Definition 9 (Dialogue). *An $\mathcal{F}$-dialogue is a reduction sequence $\sigma_1 \longrightarrow \sigma_2 \longrightarrow \ldots$ such that*

- $\forall i \exists a \in \mathcal{A} : \sigma_{i+1} = \mathbb{F}_a(\sigma_i)$
- $\forall a \in \mathcal{A}\ \forall i\ \exists j \geq i :$ *either* $\sigma_{j+1} = \mathbb{F}_a(\sigma_j)$ *or* σ_j *is in* $\xrightarrow{a}$*-normal form.*

The first condition demands that the reduction sequence be in accordance with the strategy profile (i.e., every step is in accordance with the strategy of some agent). The second condition expresses fairness: every agent should say something every once in a while, unless he or she has nothing left to say (more precisely: for any participant and at any moment i, the will be a moment $j \geq i$, at which the participant either makes an utterance or has no information left to exchange).

The next definition tells us when a strategy profile is successful with respect to a goal (a G-state being a state which is in G).

Definition 10 (Success). *$\mathcal{F}$ is successful with respect to G if for all $\mathcal{F}$-dialogues $\sigma_1 \longrightarrow \sigma_2 \longrightarrow \ldots$ it holds that if σ_1 can be reduced to a G-state, then a G-state occurs in the sequence.*

This definition formalises the first requirement on strategies that we discussed: *whenever a goal can be reached it will be reached if the participants use the strategy profile.*

4 Relevance

In the previous section, we have developed the necessary vocabulary to talk about goals, strategies, and success of strategies with respect to goals. In the present section, the framework is applied to the analysis of relevance.

Given that information exchange dialogues take place with a specific goal, it seems most natural to relate the relevance of utterances to this goal: an utterance is relevant if it helps to achieve the conversational goal. Van Rooy (2000) argues that the goal of information exchange dialogues is to solve a particular decision problem. Given that decision problems are essentially issues of the form "what should I do?", this means that utterances are relevant to the extent that they address (or help to resolve) this issue.

However, what does it mean for an utterance to address, or help to resolve, an issue? Van Rooy (2000) uses decision theory to explicate this. However, the decision theoretic view is limited by the fact that only single actions and their immediate consequences are evaluated. The effects of utterances with respect to the goal in the long run are not taken into consideration.

The present framework offers a more general view, allowing us to evaluate notions of relevance in terms of successful strategies for achieving a goal. To illustrate this, we will take two formal notions of answerhood and show what happens if we formalise the above discribed view on relevance using that notion. In particular, when considering a potential notion of relevance we will look at the implications of being relevant (in the proposed sense) with respect to reaching the goal of the conversation.

Before we proceed, we give a general result concerning the existence of successful strategies. Let us define *cofinal* strategy profiles as follows.

Definition 11 (Cofinality of strategy profiles). *$\mathcal{F}$ is cofinal if all $\mathcal{F}$-dialogues are cofinal.*

Cofinal strategy profiles satisfy the following properties.

Proposition 4.

i. Cofinal strategies are succesful with respect to every goal.

ii. If P is finite, then every strategy profile is cofinal.

iii. If P is countably infinite, then there are cofinal strategy profiles.

From the above result we can conclude that it is not unreasonable to make our requirements on notions of relevance in terms of *successfulness*.

Let us consider again what it means for an utterance to address the decision problem. Groenendijk and Stokhof (1984) have distinguished and formalised four ways in which an assertions can address an issue: being a complete answer, being a partial answer, giving a complete answer and giving a partial answer. The last of these four notions of answerhood is the most liberal one: an assertion gives a partial answer to a question if it eliminates at least one possible answer, which is to say, if it eliminates at least one block of the partition generated by the question.

For convenience, let us assume that the decision problem corresponds to a yes/no question, and let $\phi?$ denote the issue whether ϕ is the case. In the previous section, we already defined $G_{\phi?}$ to be the goal of resolving $\phi?$. Now a proposition ψ gives a partial answer to the issue $\phi?$ just in case updating with ψ results in a state in which $\phi?$ is resolved.[6] In other words, ψ gives a partial answer to $\phi?$ in a state σ precisely if $(\sigma + \psi) \in G_{\phi?}$.

Let us define ψ to be **relevant**$_1$ with respect to $\phi?$ if ψ gives a partial answer to $\phi?$. Unfortunately, it turns out that, in order to resolve $\phi?$, it can be necessary for the participants to make assertions that are not **relevant**$_1$ with respect to $\phi?$. Consequently, any successful strategy profile will sometimes prescribe utterances that are not **relevant**$_1$ with respect to the decision problem.

There are weaker notions of answerhood than that of *giving a partial answer*. Ten Cate (2000) has developed a notion of relatedness of issues, based on composition of equivalence relations (essentially, two issues are related if their corresponding partitions are not orthogonal). Using this, let us define ψ to be **relevant**$_2$ with respect to $\phi?$ if $\psi?$ is related to $\phi?$.[7]

Relevance$_2$ does not have the problem described above: there are succesful strategies with respect to any goal that prescribe only **relevant**$_2$ utterances. Although **relevance**$_2$ behave better than **relevance**$_1$ in this respect, it has another problem: given any sentence ψ and any issue $\phi?$, we can split up ψ into two formulae, $(\phi \to \psi)$ and $(\neg\phi \to \psi)$, which are both **relevant**$_2$ with respect to $\phi?$. Consequently, in terms of our ARS, the reduction graph generated by **relevant**$_2$ reductions is cofinal in the entire reduction graph.

A natural language example that illustrates this problem is the following. Suppose the issue is whether Amsterdam is the capital of the Netherlands and consider the proposition "it rains". Then the latter proposition can be split into two parts, "if Amsterdam is the capital of the Netherlands then

[6] In the case of a yes/no question, eliminating at least one alternative means resolving the question completely.

[7] As a matter of fact, a proposition ψ is **relevant**$_2$ with respect to an issue $\phi?$ precisely if either ψ or $\neg\psi$ is **relevant**$_1$ with respect to $\phi?$.

it rains" and "if Amsterdam is not the capital of the Netherlands then it rains", such that both propositions are **relevant**$_2$ with respect to the issue (i.e., whether Amsterdam is the capital of the Netherlands).

Consequently, **relevance**$_2$ does not significantly decrease the complexity of the reduction graph. Ideally we therefore need a notion of relevance that is weaker than **relevance**$_1$, but stronger than **relevance**$_2$. We do not have such a notion at present. Nevertheless, the discussion illustrates how the present framework can contribute to analysis of notions of relevance.

5 Conclusion and discussion

It has been shown that ARS theory can be used for analysing pragmatic aspects of cooperative information exchange. In particular, it was shown how such an approach to pragmatics can contribute to the analysis of *relevance*. There are a number of directions for further research.

First of all, a number of assumptions were made with respect to the type of communication. The present approach depends crucially on the confluence property (most rewriting literature assumes that the reduction relations are either confluent or have some infinitary variant of the confluence property). When we look at other types of communication than information exchange (e.g., argumentation) then there seems to be no confluence.

However, when we stay with information exchange, it is still possible to generalise over the assumptions that were made. In particular, we have looked only at the case in which the participants share a common goal, but we could also assign all participants their own goal. In this way, it is possible to model less cooperative types of communication.

In this paper, we have focussed only on one aspect of information exchange strategies, namely that they must be succesful with respect to the goal of the conversation. However, we mentioned three other requirements on strategies. These requirements should be examined in more detail, and formalised where possible. Note that the requirements on strategies we discussed are not specific to the case of information exchange but are relevant for many rewriting applications.

Another direction of further research would be in enriching the models of information. We took a simple update semantics based on propositional logic, where the worlds are propositional valuations, but various alternatives present themselves. In ten Cate (2000) a start is made with a similar system on the basis of Groenendijk (1999)'s QL, which is a predicate logical language with questions. Also, recently attention has been directed to the dynamic epistemic semantics of public announcements (Gerbrandy 1999). It seems natural to look at these epistemic models of information exchange from the perspective of ARS's.[8]

[8] Although the resulting ARS is in general not confluent, van Benthem (2000) has shown

Finally, it is interesting to see that the link between information exchange and rewriting becomes more apparent when we shift from the set-theoretic perspective of dynamic semantics to a more representational view. In the latter case, information states are no longer sets of possible worlds, but rather formulae, files or discourse representation structures. If we would proceed in this way, then our ARS would look more like a *term* rewriting system.

References

Baader, F. and T. Nipkow (1998). *Term rewriting and all that.* Cambridge, UK: Cambridge University Press.

Blackburn, P., M. de Rijke, and Y. Venema (To appear). *Modal logic.* Cambridge, UK: Cambridge University Press.

Gerbrandy, J. (1999). *Bisimulations on planet Kripke.* Ph. D. thesis, ILLC, University of Amsterdam.

Groenendijk, J. (1999). The logic of interrogation: classical version. In T. Matthews and D. Strolovitch (Eds.), *Proceedings of the ninth conference on semantics and linguistics theory (SALT-9), Santa Cruz.* CLC Publications.

Groenendijk, J. and M. Stokhof (1984). *Studies on the semantics of questions and the pragmatics of answers.* Ph. D. thesis, University of Amsterdam.

Klop, J. W. (1992). Term Rewriting Systems. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum (Eds.), *Handbook of Logic in Computer Science*, Volume 2, Chapter 1, pp. 1–117. Oxford: Oxford University Press.

Stalnaker, R. (1978). Assertion. In P. Cole (Ed.), *Syntax and semantics*, Volume 9, *Pragmatics*, pp. 315–332. New York: Academic Press.

ten Cate, B. (2000). Dynamic and epistemic semantics of questions: the logic of consultation. Master's thesis, Vrije Universiteit, Amsterdam. Available from `http://home.student.uva.nl/balder.tencate/`.

van Benthem, J. (2000). Philosophy 298: Information update from a logical perspective (lecture notes). Available from `http://turing.wins.uva.nl/~johan/298-2000.html`.

van Rooy, R. (2000). Decision Problems in Pragmatics. In D. Traum and M. Poesio (Eds.), *Proceedings of GÖTALOG 2000.*

that if we restrict ourselves no finite Kripke structures, we do have confluence, as well as strong normalisation.

Fuzzy Hedges: a Next Generation

MARTINE DE COCK
Department of Applied Mathematics and Computer Science
Ghent University, Krijgslaan 281 (S9), B-9000 Ghent, Belgium
Martine.DeCock@rug.ac.be

ABSTRACT. Fuzzy set theory provides a framework in which natural language expressions can be modelled mathematically. In fuzzy systems a linguistic term (e.g. fast) is represented by a fuzzy set, while a linguistic modifier (e.g. very) is modelled by an operator (also known as *hedge*) transforming a fuzzy set into another. In the paper first we give a short overview of the traditional fuzzy set theoretical approaches to this problem. We point out that these "hedges of the first generation" are in general technical tools without meaning of their own, which explains their most important shortcomings. To overcome this, we present two new approaches in which the representation of linguistic modifiers is endowed with an inherent semantics ("hedges of the second generation"): the framework of fuzzy modifiers based on fuzzy relations (recently developed by us) and the so-called horizon approach (an extension of the research initiated by Novák).

1 Introduction

During millenia yet people tend to express real-life information by means of natural language; it allows them to reason about everyday issues within a certain (tolerated) degree of imprecision. Hence it is not surprising that the introduction of fuzzy set theory (FST) in the sixties (Zadeh 1965) as a framework for the mathematical representation of linguistic concepts, has given rise to an important evolution in the field of computer science. The importance of computing with words has increased tremendously over the last years (Zadeh and Kacprzyk 1999a; Zadeh and Kacprzyk 1999b) and it will grow even more in the years to come (Zadeh 2000). Nowadays the fuzzy set theoretical representations of linguistic terms are already applied succesfully in an impressive amount of systems, ranging from control and reasoning systems over preference and decision making systems to database systems. We refer to (Babuška 1998; Ruan and Kerre 2000; Yazici and George 1999) for some examples.

The key idea is to model linguistic terms such as "cheap" or "more or less cheap" by a $X \to [0,1]$ mapping for X a suitable universe (e.g. the universe of prices). When developing an application, the design of such a $X \to [0,1]$

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

mapping for each term involved is therefore a fundamental task. Quite early in the history of FST, researchers (Lakoff 1973; Zadeh 1972) started studying the possibility to partially automate this not so trivial task in the following way: given a $X \rightarrow [0,1]$ mapping for a term (e.g. "cheap"), how can we automatically generate a suitable $X \rightarrow [0,1]$ mapping for the modified term (e.g. "more or less cheap")? All their efforts resulted in an extensive collection of technical operators transforming one $X \rightarrow [0,1]$ mapping into another (see (Kerre and De Cock 1999) for an overview).

Although already useful, these so-called hedges have important shortcomings, which are in our opinion due to the fact that they are designed simply to perform a technical transformation, but have no further meaning of their own. In our paper we will present two new approaches with a clear inherent semantics, and we will show that they overcome the shortcomings of the hedges of the first generation. However we will start with a short discussion on the representation of (modified) linguistic terms in general, focussing on the linguistic modifiers "very" and "more or less", and a brief overview of the hedges of the first generation and their shortcomings.

2 Representing linguistic terms

In fuzzy systems, each linguistic term is represented by a fuzzy set on a suitable universe X. A fuzzy set A on X is a $X \rightarrow [0,1]$ mapping, also called the membership function of A. For all x in X, $A(x)$ is called the membership degree of x in A. The class of all fuzzy sets on X is denoted $\mathcal{F}(X)$. Furthermore for A and B in $\mathcal{F}(X)$ we say that $A \subseteq B$ if and only if $A(x) \leq B(x)$, for all x in X. When representing a linguistic term by a fuzzy set, the most important question is of course: which membership function should we choose?

2.1 Two interpretations

In the *inclusive interpretation* (Vanden Eynde 1996) it is assumed that semantic entailment (Lakoff 1973) holds : for A in $\mathcal{F}(X)$ and x in X

$$x \text{ is very } A \Rightarrow x \text{ is } A \Rightarrow x \text{ is more or less } A$$

These kind of assumptions are often made in the literature (see e.g. (Kerre 1993; Novák and Perfilieva 1999; Zadeh 1972)). Representing linguistic terms by fuzzy sets, they correspond to:

$$\text{very } A \subseteq A \subseteq \text{more or less } A \tag{6.1}$$

The underlying semantics is that every object that is very A is also A and that every object that is A is also more or less A. In this interpretation the membership degree of x in A corresponds to the degree to which x satisfies

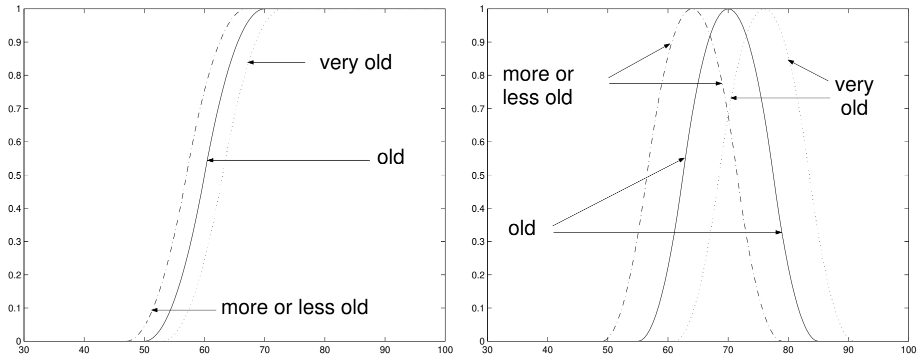


Figure 6.1: Modifiers based on fuzzy relations: a) inclusive. b) non-inclusive.

the term modelled by A: indeed the degree to which an object x is more or less A is always greater than or equal to the degree to which the same object x is A. The same holds for A w.r.t. very A. Figure 6.1a depicts possible membership functions for the linguistic terms old, more or less old, and very old in the universe of ages expressed in years. Note that such kind of membership functions are dependent on context and observer and may vary likewise depending on the application and the applicant. However since this is not the central issue of our paper, we will not go into further detail about it and assume that the given membership functions are acceptable to a particular observer in a given context.

Psycholinguistic research (Hersh and Caramazza 1976; Vanden Eynde 1996) showed that there is also a *non-inclusive interpretation*. In this interpretation, which is often used in fuzzy control applications (e.g. (Babuška 1998)), a term modified by "more or less" or "very" doesn't denote a subset neither a superset of the original term. The terms denote different (possibly overlapping) categories. In this case the membership degree of x in A corresponds rather to the degree to which x is representative for the term modelled by A. Figure 6.1b depicts possible membership functions for old, more or less old, and very old in the non-inclusive interpretation.

2.2 Fuzzy modifiers

As already indicated in the introduction, the design of a suitable membership function for every term involved in an application is a fundamental task. Furthermore saving all the membership functions in memory can be quite memory consuming. For both these reasons it is desirable to generate the membership function for a modified linguistic term automatically from the membership function of the original term. In practice this is done by representing every linguistic modifier by a fuzzy modifier, i.e. a $\mathcal{F}(X) \to \mathcal{F}(X)$ mapping. In FST linguistic/fuzzy modifiers are commonly called linguistic/fuzzy hedges. Most of the fuzzy modifiers proposed in the literature (we call them "hedges of the first generation") are (r,t)-decomposable. For r a $[0,1] \to [0,1]$ mapping and t a $X \to X$ mapping, a fuzzy modifier m on X is called (r,t)-decomposable if and only if for all A in $\mathcal{F}(X)$, for all x in X: $m(A)(x) = r(A(t(x)))$. t and r are called the pre- and the post-modifier of m respectively. Often either r is the identical mapping $\mathbb{I}_{[0,1]}$ on $[0,1]$ (i.e. $r(x) = x$, for all x in $[0,1]$) or t is the identical mapping $\mathbb{I}_X$ on X (i.e. $t(x) = x$, for all x in X). In these cases we talk about *pure premodification* and *pure postmodification* respectively.

Pure postmodification By far the most popular postmodifiers are the mappings $.^\alpha$, for α in $[0,+\infty[$. The associated $(.^\alpha, \mathbb{I}_X)$-decomposable fuzzy modifiers — also called powering modifiers — are originally introduced by Zadeh (Zadeh 1972). When m is a $(r, \mathbb{I}_X)$-decomposable fuzzy modifier then either $(\forall x \in X)(A(x) = 1 \Rightarrow m(A)(x) \neq 1)$ or $(\forall x \in X)(A(x) = 1 \Rightarrow m(A)(x) = 1)$ Suppose we use $m(\mathsf{tall})$ to model very tall in the universe of heights of men, then in the first case no height can be very tall to degree 1, while in the second all heights that are tall to degree 1, are also very tall to degree 1. According to our intuition however a height of 2m00 is clearly very tall to degree 1, while a height of 1m80 is tall to degree 1 but very tall only to a lower degree. This example show that a representation of "very" by a $(r, \mathbb{I}_X)$-decomposable fuzzy modifier leads to counter-intuitive results. Similar remarks can be made for "more or less." Furthermore it's not possible to use a $(r, \mathbb{I}_X)$-decomposable fuzzy modifier for the non-inclusive interpretation.

Pure premodification Premodifiers act on the objects of the universe X. They are mainly studied for X a subset of $\mathbb{R}$, the set of real numbers. The most popular premodifiers are the mappings T_α defined by $T_\alpha(x) = x+\alpha$, for all x in $\mathbb{R}$, α in $\mathbb{R}$. The resulting $(\mathbb{I}_{[0,1]}, T_\alpha)$-decomposable modifiers $S_{-\alpha}$ — also called shifting modifiers — are informally suggested by Lakoff (Lakoff 1973) and more formally developed later on (Bouchon-Meunier 1993; Hellendoorn 1990; Kerre 1993). Since they shift the membership function α units to the left or the right they are suitable to model "more or less" and "very$''$ in the non-inclusive interpretation. In the inclusive interpretation shifting modifiers can be used to represent more or less A and very A provided that A is increasing or decreasing, and that X is a subset of $\mathbb{R}$ of course. If A is partially increasing and partially decreasing however (e.g. about 8 p.m.) a shifted version can never be a superset of A and can therefore not be used to model more or less A. In this case an artificial and quite complicated solution to the problem can be found by dividing the membership function into its increasing and decreasing parts and applying a different shift to each part.

A suggestion to use both a non-trivial pre- and postmodifier at once for the inclusive interpretation was made by Novák (Novák 1992). Although it is an improvement on the solutions discussed in the previous paragraphs, it can only be applied for a special kind of membership functions and it also involves a process of division of the membership function in increasing and decreasing parts. For a detailed overview of the hedges of the first generation and their shortcomings, we refer to (De Cock 1999; Kerre and De Cock 1999).

3 The next generation

The fuzzy modifiers of the first generation are merely artificial operators that transform the membership function of a term A into a membership function that is kind of acceptable (to the designer) for very A or more or less A, but they do not have an inherent semantics. Hence it is no surprise that they are afflicted with the above mentioned shortcomings. We feel that fuzzy modifiers should be endowed with a clear inherent semantics (see also (De Cock and Kerre 2000)).

3.1 Fuzzy modifiers based on fuzzy relations

Inclusive interpretation We suggest to do this by taking the context, i.e. mutual relationships in the universe into account. More specifically to compute the degree to which an object y is more or less A or very A, we will take a look at the objects that resemble to y. Indeed: according to our intuition a person can be called more or less old if he resembles to someone who is old. Furthermore he is very old if everybody whom he resembles to is old.

Resemblance can be modelled by means of a fuzzy relation R on X, i.e. a fuzzy set on $X \times X$. It is clear that such a resemblance relation should be reflexive and symmetrical. For a more detailed study on how it should look like, we refer to (De Cock and Kerre 2001). In Figure 6.1a we used the fuzzy relation R defined by $R(x, y) = min(1, max(2.5 - 0.5|x - y|), 0))$, with $R(x, y)$ being the degree to which x and y are approximately equal, for all x and y in X. The so-called R-foreset of y is the fuzzy set on X denoted by Ry and defined by $(Ry)(x) = R(x, y)$. If R models resemblance, then Ry is the fuzzy set of objects resembling y, in other words the context of y that we wish to consider.

Furthermore we want to express that y is more or less A if y resembles to an object that is A, in other words if the intersection of A and Ry is not

Term	*Fuzzy set*	*Conditions*
very A	$R^{\heartsuit}(A)$	R is a resemblance relation
more or less A	$R^{\clubsuit}(A)$	R is a resemblance relation

Table 6.1: Modifiers based on fuzzy relations - inclusive interpretation

empty. Likewise y is very A if all the objects that resemble to y are A, i.e. if Ry is included in A. Therefore we suggest to model more or less A and very A by the fuzzy sets $R^{\clubsuit}(A)$ and $R^{\heartsuit}(A)$ respectively, with for all y in X:

$$R^{\clubsuit}(A)(y) = \sup_{x \in X} \max(A(x) + (Ry)(x) - 1, 0)$$

$$R^{\heartsuit}(A)(y) = \inf_{x \in X} \min(1 - (Ry)(x) + A(x), 1)$$

The meaning of these formulas becomes very clear when A and Ry are crisp sets. A crisp set B in a universe X only has membership degrees 0 and 1; $B(x) = 1$ is denoted "$x \in B$" while $B(x) = 0$ corresponds to "$x \notin B$". Indeed if A and Ry are crisp, it can be verified that

$$\begin{array}{lll} y \in R^{\clubsuit}(A) & \text{iff} & (\exists x \in X)(x \in A \wedge x \in Ry) \\ y \in R^{\heartsuit}(A) & \text{iff} & (\forall x \in X)(x \in Ry \Rightarrow x \in A) \end{array}$$

Furthermore if A is a crisp singleton, i.e. $A(z) = 1$ for some z in X, and $A(x) = 0$ for all other x in X, then, regardless whether Ry is crisp or not:

$$R^{\clubsuit}(A)(y) = R(z, y)$$

In the chosen representation this corresponds to "y is more or less $\{z\}$ to the degree to which z and y resemble," which is according to our intuition. To show that the presented framework respects the semantic entailment needed in the inclusive interpretation (cfr. Formula (6.1)), we will proof the following proposition which holds for an arbitrary reflexive fuzzy relation. A fuzzy relation R is called reflexive if and only if $(\forall x \in X)(R(x, x) = 1)$.

Proposition 3.1. *If R is a reflexive fuzzy relation on X then for all A in $\mathcal{F}(X)$:*

$$R^{\heartsuit}(A) \subseteq A \subseteq R^{\clubsuit}(A)$$

Proof. For all y in X:

$$\begin{array}{ll} R^{\heartsuit}(A)(y) & \leq \min(1 - R(y, y) + A(y), 1) \\ & \leq \min(A(y), 1) \\ & \leq A(y) \\ & \leq \max(A(y), 0) \\ & \leq \max(A(y) + R(y, y) - 1, 0) \\ & \leq R^{\clubsuit}(A)(y) \end{array}$$

	beautiful	average	ugly
snowwhite	1.00	0.00	0.00
witch	0.00	0.30	0.70
wolf	0.00	0.00	1.00
dwarf	0.10	0.70	0.20
prince	0.80	0.20	0.00
red-hood	0.50	0.50	0.00

Table 6.2: Fuzzy sets in the universe of fairy-tale characters

R	snowwhite	witch	wolf	dwarf	prince	red-hood
snowwhite	1.00	0.00	0.00	0.00	1.00	0.50
witch	0.00	1.00	1.00	0.50	0.00	0.00
wolf	0.00	1.00	1.00	0.00	0.00	0.00
dwarf	0.00	0.50	0.00	1.00	0.00	0.88
prince	1.00	0.00	0.00	0.00	1.00	1.00
red-hood	0.50	0.00	0.00	0.88	1.00	1.00

Table 6.3: Resemblance relation on the universe of fairy-tale characters

According to the definition of fuzzy set inclusion given at the beginning of Section 2, this proofs the proposition.

As stated above it is natural to assume reflexivity for a fuzzy relation modelling resemblance; indeed every object resembles to itself to the highest degree. Therefore our framework garantuees semantic entailment. Furthermore it imposes no restrictions on the membership functions and it can be applied to all kinds of universes (numerical as well as non-numerical), as we will illustrate in the following example:

Example 3.1. *Table 6.2 shows the membership degrees of the fuzzy sets* beautiful, average *and* ugly *in the universe X of 6 fairytale characters, while Table 6.3 defines a resemblance relation R on X. For more details on the construction of R we refer to (De Cock and Kerre 2001). The modified fuzzy sets obtained using the representational scheme of Table 6.1 are given in Table 6.4. Comparing our technique to the traditional fuzzy modifiers discussed in Section 2, we notice that there is no clear shifting operation on the non-numerical universe of fairy-tale characters, hence no straightforward way to use shifting hedges. The application of powering hedges is technically possible, but can never yield the results obtained in Table 6.4. In fact the* prince *who is* beautiful *only to degree 0.80, is considered* more or less beautiful *to degree 1 in our representation. Likewise snowwhite who is* average *to degree 0 is promoted a bit to being* more or less average *to degree 0.20. This is due to her resemblance to other fairy-tale characters who are already* average *to some degree greater than 0. Since modifiers with pure postmodification, such*

	more or less beautiful	more or less average	more or less ugly	very beautiful	very average	very ugly
snowwhite	1.00	0.20	0.00	0.80	0.00	0.00
witch	0.00	0.30	1.00	0.00	0.00	0.70
wolf	0.00	0.30	1.00	0.00	0.00	0.70
dwarf	0.38	0.70	0.20	0.10	0.62	0.12
prince	1.00	0.50	0.00	0.50	0.00	0.00
red-hood	0.80	0.58	0.08	0.22	0.20	0.00

Table 6.4: Modified fuzzy sets

*as powering modifiers, can not make the difference between objects belonging to degree 1 to the original fuzzy set and belonging to degree 1 to the modified fuzzy set (and analogously for the degree 0) they can not yield such results. Similar remarks can be made for "*very*".*

Non-inclusive interpretation For the non-inclusive interpretation suitable fuzzy relations R can easily be chosen such that $R^{\clubsuit}(A)$ imitates the behaviour of shifting as discussed in Section 2.2. Hence for such suitable relations R_1 and R_2 we can use ${R_1}^{\clubsuit}(A)$ to model more or less A and ${R_2}^{\clubsuit}(A)$ to model very A. Furthermore for quite similar fuzzy relations not only a shift but also a change of shape can be obtained, making the representational scheme even more flexible. Note that this allows the use of the same powerful framework for both the inclusive and the non-inclusive interpretation. However since no further inherent semantics is yet assigned to this technique in the non-inclusive interpretation, we omit the details here. An example is given in Figure 6.1b.

3.2 The horizon approach

Another approach to give inherent semantics to modifiers is the horizon idea developed by Novák (Novák 1996; Novák and Perfilieva 1999) for the inclusive interpretation. We extend it to the non-inclusive interpretation. For both interpretations we also integrate a mathematical notion of distance, namely a pseudo-metric, so the technique can be used not only in numerical universes but in pseudo-metric spaces in general. We recall that an $\mathcal{M}^2 - [0, +\infty[$ mapping d is called a pseudo-metric on $\mathcal{M}$ if and only if $d(x,x) = 0$, $d(x,y) = d(y,x)$ and $d(x,y) + d(y,z) \geq d(x,z)$, for all x, y, and z in $\mathcal{M}$. $(\mathcal{M}, d)$ is then called a pseudo-metric space.

Let (X,d) be a pseudo-metric space and let Ω be an element of X called the *observer*. In real life situations the objects that are close to Ω (closer than some distance α) are clearly visible to him (visible to degree 1). The visibility of an object x of X to Ω drops with the distance $d(\Omega, x)$ between Ω and that object. Somewhere in the far distance, behind the horizon (say, at distance γ), the objects aren't visible at all anymore (visible to degree 0). α and γ characterize the vision of the observer; therefore we call them the *quality of vision parameters.* It is natural to assume that $\alpha \in \mathbb{R}$, $\gamma \in \mathbb{R}$ and $\alpha < \gamma$. The visibility of objects in X to an observer Ω in X with quality of vision parameters α and γ can now be expressed by a $X \to [0,1]$ mapping $\mathbb{H}(\Omega, (\alpha, \gamma))$ with the following characteristics:

(H.1) $d(\Omega, x) \leq \alpha \Rightarrow \mathbb{H}(\Omega, (\alpha, \gamma))(x) = 1$
(H.2) $\gamma \leq d(\Omega, x) \Rightarrow \mathbb{H}(\Omega, (\alpha, \gamma))(x) = 0$
(H.3) $d(\Omega, x) \leq d(\Omega, y) \Rightarrow \mathbb{H}(\Omega, (\alpha, \gamma))(x) \geq \mathbb{H}(\Omega, (\alpha, \gamma))(y)$

Furthermore for two couples of quality of vision parameters (α_1, γ_1) and (α_2, γ_2) such that $\alpha_1 \leq \alpha_2$ and $\gamma_1 \leq \gamma_2$, it should hold that

(H.4) $\mathbb{H}(\Omega, (\alpha_1, \gamma_1))(x) \leq \mathbb{H}(\Omega, (\alpha_2, \gamma_2))(x)$

which expresses that the visibility is better if the quality of vision parameters are higher. An example of such a mapping is $f = \mathbb{H}(\Omega, (\alpha, \gamma))$ defined by, for all x in X,

$$f(x) = \begin{cases} 1 & \text{if } d(\Omega, x) \leq \alpha \\ 1 - \frac{d(\Omega, x) - \alpha}{\gamma - \alpha} & \text{if } \alpha < d(\Omega, x) < \gamma \\ 0 & \text{otherwise} \end{cases}$$

Inclusive interpretation To model a term A in X (e.g. the term "small" in the universe of heights), we place the observer Ω with quality of vision parameters (α, γ) in the element of X that satisfies the term A the best (the smallest height, i.e. 0). The visibility of an object x of X to Ω corresponds to the degree to which x satisfies A. Then we make the observer take off his glasses, which changes the quality of his vision to the worse: it is now characterized by two new quality of vision parameters (α_1, γ_1) such that $\alpha_1 \leq \alpha$, $\gamma_1 \leq \gamma$. The objects he can still see without his glasses must be very close to him, so they must be very A. Finally we give the observer back his glasses and a telescope as well, thereby improving the quality of his vision. Now he can see all objects that are more or less A. This is summarized in Table 6.5 in which Ω is the element of X that satisfies A the best. Note that the characteristics imposed on $\mathbb{H}$ and the conditions on the quality of vision parameters guarantee that Formula (6.1) is respected. The membership functions in Figure 6.2a are constructed according to this scheme, using the mapping f defined above with $d(x, y) = |x - y|$ for all x and y in X (the universe of ages).

Term	*Corresponding fuzzy set*	*Conditions*
A	$\mathbb{H}(\Omega, (\alpha, \gamma))$	
very A	$\mathbb{H}(\Omega, (\alpha_1, \gamma_1))$	$\alpha_1 \leq \alpha$, $\gamma_1 \leq \gamma$
more or less A	$\mathbb{H}(\Omega, (\alpha_2, \gamma_2))$	$\alpha \leq \alpha_2$, $\gamma \leq \gamma_2$

Table 6.5: Horizon approach - inclusive interpretation

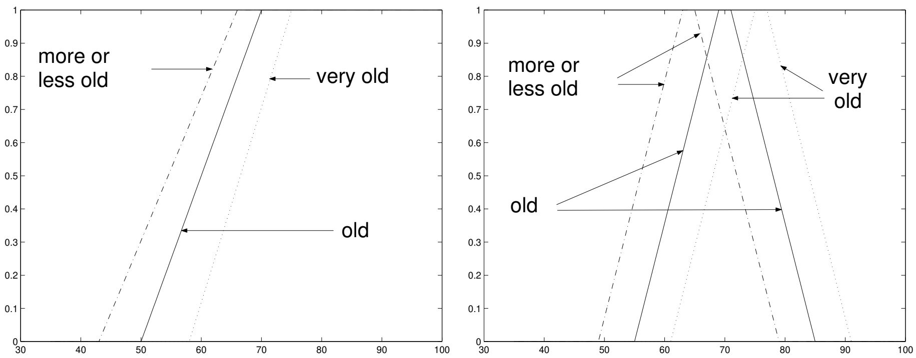


Figure 6.2: Horizon approach: a) inclusive. b) non-inclusive.

Non-inclusive interpretation For the inclusive interpretation we have kept the observer Ω fixed on some object of X, and we have changed his quality of vision parameters. In the non-inclusive interpretation we will do exactly the opposite: we keep the quality of vision fixed, but we make the observer walk through the universe. More in particular, to model a term, we will place the observer on the object of X that is most representative for that term. In Table 6.6 Ω_1, Ω_2, and Ω_3 are the elements of X most representative for A, very A, and more or less A respectively. See figure 6.2b for an example of membership functions generated according to this scheme

Term	*Corresponding fuzzy set*
A	$\mathbb{H}(\Omega_1, (\alpha, \gamma))$
very A	$\mathbb{H}(\Omega_2, (\alpha, \gamma))$
more or less A	$\mathbb{H}(\Omega_3, (\alpha, \gamma))$

Table 6.6: Horizon approach - non-inclusive interpretation

using the mapping f and d as defined above.

Conclusion and future work

After indicating the shortcomings of traditional fuzzy modifiers, we have presented two new approaches. Unlike the hedges of the first generation, these new kinds of fuzzy modifiers are not merely technical operators, but are endowed with clear inherent semantics. For the fuzzy modifiers based on fuzzy relations this is achieved by taking the context into account (more specifically: mutual relationships between objects). In the horizon approach (an extension of the work initiated by Novák) the modified terms are characterized by a change of quality of vision parameters or a change of the observer, giving them a semantics smoothly integrated in the surrounding horizon idea. Since this new generation of hedges has a strong semantical ground, they lead to membership functions which are closer to human understanding. Furthermore they can be used on all kinds of membership functions with (hardly) no restriction. Our further work will focus on the application of these new kinds of fuzzy modifiers in fuzzy systems.

Acknowledgements The author would like to thank the Fund For Scientific Research-Flanders (FWO) for funding the research reported on in this paper, as well as the anonymous referees for their helpful comments.

References

Babuška, R. (1998). *Fuzzy Modeling for Control.* Boston/Dordrecht/London: Kluwer Academic Publishers.

Bouchon-Meunier, B. (1993). *La Logique Floue.* Que sais-je? Paris.

De Cock, M. (1999). Representing the Adverb Very in Fuzzy Set Theory. In A. Todirascu (Ed.), *Proceedings of the ESSLLI 1999 Student Session*, Utrecht, the Netherlands.

De Cock, M. and E. E. Kerre (2000). A New Class of Fuzzy Modifiers. In *Proc. ISMVL 2000*, pp. 121–126. IEEE Computer Society.

De Cock, M. and E. E. Kerre (2001). On (un)suitable Fuzzy Relations to Model Approximate Equality. *Fuzzy Sets and Systems*. Accepted for publication.

Hellendoorn, H. (1990). *Reasoning with fuzzy logic.* Ph. D. thesis, T. U. Delft, Delft.

Hersh, H. M. and A. Caramazza (1976). A Fuzzy Set Approach to Modifiers and Vagueness in Natural Language. *Journal of Experimental Psychology : General 105*(3), 254–276.

Kerre, E. E. (1993). *Introduction to the Basic Principles of Fuzzy Set Theory and Some of its Applications.* Gent: Communication and Cognition.

Kerre, E. E. and M. De Cock (1999). Linguistic Modifiers: An Overview. In G. Chen, M. Ying, and K.-Y. Cai (Eds.), *Fuzzy Logic and Soft Computing*, pp. 69–85. Kluwer Academic Publishers.

Lakoff, G. (1973). Hedges : a Study in Meaning Criteria and the Logic of Fuzzy Concepts. *Journal of Philosophical Logic 2*, 458–508.

Novák, V. (1992). *The alternative mathematical model of linguistic semantics and pragmatics*, Volume 8 of *International Series on Systems Science and Engineering.* Plenum Press New York and London.

Novák, V. (1996). A horizon shifting model of linguistic hedges for approximate reasoning. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, New Orleans, pp. 423–427.

Novák, V. and I. Perfilieva (1999). Evaluating Linguistic Expressions and Functional Fuzzy Theories in Fuzzy Logic. In L. A. Zadeh and J. Kacprzyk (Eds.), *Computing with Words in Information/Intelligent Systems 1 (Foundations)*, pp. 383–406. Physica-Verlag.

Ruan, D. and E. E. Kerre (Eds.) (2000). *Fuzzy If-Then Rules in Computational Intelligence; Theory and Applications.* Kluwer Academic Publishers, Boston/Dordrecht/London.

Vanden Eynde, C. (1996). A very difficult problem : modelling modification using VERY; A semantic–pragmatic approach. (in Dutch) FKFO–project, private communication.

Yazici, A. and R. George (Eds.) (1999). *Fuzzy Database Modeling*, Volume 26 of *Studies in Fuzziness and Soft Computing.* Physica-Verlag.

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control 8*, 338–353.

Zadeh, L. A. (1972). A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges. *Journal of Cybernetics 2*(3), 4–34.

Zadeh, L. A. (2000). Toward an Enlargement of the Role of Natural Languages in Information Processing, Decision and Control. In *Proceedings 6th Int. Conf. on Soft Computing (IIZUKA2000), CD-ROM*, Fukuoka, Japan, pp. 7–10.

Zadeh, L. A. and J. Kacprzyk (Eds.) (1999a). *Computing with Words in Information/Intelligent Systems 1 (Foundations)*, Volume 33 of *Studies in Fuzziness and Soft Computing.* Physica-Verlag.

Zadeh, L. A. and J. Kacprzyk (Eds.) (1999b). *Computing with Words in Information/Intelligent Systems 2 (Applications)*, Volume 34 of *Studies in Fuzziness and Soft Computing.* Physica-Verlag.

One is Enough: The Case Against Aspectual Proliferation

Bridget Copley
Massachusetts Institute of Technology
copley@mit.edu

Abstract. This paper defends the "naive view" that English simple past tense sentences have only one temporal-aspectual element. The view that there are two temporal-aspectual elements in such sentences has been proposed by proponents of various improvements on the Reichenbachian system of tense, but their arguments are shown to be inconclusive, and other arguments are presented in favor of the naive view. If correct, this result casts serious doubts on the use of the Reichenbachian temporal points as primitives in theories of temporal-aspectual semantics.

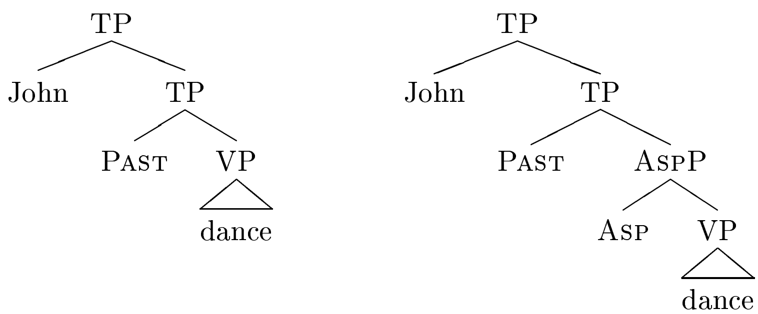
1 Introduction

Consider (1).

(1) John danced.

The naive view of the logical form of (1) is that there is one temporal element (operator, or predicate, or whatever you believe tenses are[1]), namely Past, as in (2a). The proposal has been made, however, that (1) actually involves two temporal-aspectual elements, identified as Past and Asp below in (2b):

(2) a. Naive view b. Less naive view



[1] I will use the word "operator" for convenience even though I later assume tenses of the type $\langle i, t\rangle$; neither the terminology nor the later assumption is crucial to the argument.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

This proposal has been made in the service of two somewhat different, though related, ideas.

The first begins ultimately with Reichenbach (1947), with additional input from Hornstein (1990) and Klein (1997) along the way. This idea takes as primitive three temporal times (S, R, and E). Each temporal-aspectual construction is distinguished by different instantiations of two relations: between S and R on the one hand and R and E on the other. Two relations means two temporal-aspectual elements in the morphosemantics (or ought to, even if Reichenbach and Hornstein don't put it that way; Klein does).

The second idea leading to (2b), independently of the Reichenbachian line of reasoning, is the idea that (1) has a perfective aspectual operator. Proponents of this view include Klein (for whom this perfective operator is the second operator alluded to above).

Despite both of these reasons to adopt (2b) as the correct view of (1), however, I will argue that the naive view in (2a) is the correct one. First, I will argue that the arguments given in support of the first idea, i.e., the Reichenbachian line of reasoning, are not convincing and make wrong predictions. Then I will make the case for (2a) using two new arguments against the first idea. Finally, I will address the second idea in the light of these results and show that (2b) makes sense in light of the semantic facts.

This result, if correct, is significant for any theory which seeks to eliminate the use of the Reichenbachian temporal points S, R, and E as primitives in temporal-aspectual semantics. (C.f., e.g., (Stowell 1996) for one theory which explicitly does this.) If it can be shown that the presence of R is not needed and indeed makes incorrect predictions, then the designation of these points as primitives is called seriously into doubt.

2 The Reichenbachian line: two operators

Though Hornstein does not espouse a view in which each relation has its own morpheme, Klein does, as mentioned above. However, since Hornstein's arguments for three points are more syntactically oriented than Klein's, and can be carried over to Klein's structural view, in this section I will for the most part be arguing against Hornstein's reasons for having three temporal points implicit in (1), with the view of arguing for (2a). First, however, I will provide some background on the original motivation for the Reichenbachian system, and Hornstein's and Klein's improvements to it.

2.1 The Reichenbachian system

Reichenbach's system is a formal characterization of tense constructions. Each tense construction consists of relations among three times. Two of these times, which are relevant for any tense construction are speech time (S), the time at which the sentence is uttered, and event time (E), the time at which the event or state mentioned in the sentence is to have happened, be happening, or be going to happen. For simple sentences such as those in (3), seemingly this is all that is needed. (I will be arguing, in effect, that for simple sentences it really *is* all that is needed.) In the Reichenbach notation, "E_S" means that E is before S, and "E,S" means that E and S are located at the same time.

(3) a. John danced. E_S
b. John is dancing. E,S
c. John will dance. S_E

This is fine, but the perfect apparently needs more than just S and E. Consider a past perfect construction as in (4):

(4) At 6:00, John had finished his paper.

How should the perfect be represented? E is before S, but that is also true for the simple past, as in (4a). We would like to be able to distinguish the two. Furthermore, assuming that temporal variables are only allowed to pick out times that have these "temporal roles," what is *6:00* doing there? The sentence in (4) has a reading in which E is not located at 6:00, but before 6:00, so 6:00 is neither E nor S. This is why Reichenbach introduces a third time: the "reference time" R. The past perfect he then characterizes as E_R_S.

Casting all temporal constructions (even the simple tenses) as combinations of E, R, and S allows Reichenbach to make a list of 24 possible temporal combinations, which is nicely restrictive. Hornstein points out, however, that we can get an even more restrictive set of possibilities, since there are numerous combinations that English apparently does not distinguish between. For example, the future perfect seems to correspond to E_S_R, E,S_R, and S_E_R; the sentence in (5) is silent about whether the finishing event has taken place before the speech time, is taking place as we speak, or will take place between now and noon.

(5) John will have finished his paper by noon.

What is important here is that the relation between E and S is unspecified. Thus Hornstein concludes that the only two relations that matter are the relation between E and R and the relation between R and S. This also reduces the number of possible temporal constructions to 16.

Hornstein still has to make the case that all tense combinations have all three times and both relations, a case which is outlined below. But note that one would only want to make this argument if these times — or more accurately, the "roles" they instantiate (speech time, reference time/time of topic, etc.) — and the relations between them are really primitives in the system. If there is a structural or pragmatic justification for the roles, then the number of roles and relations in the sentence should simply follow from the structural or pragmatic considerations. Likewise, if we find the latter to be the case, that undermines the idea that the temporal roles are primitives. The term "roles" I mean to be reminiscent of theta roles, but a closer analogy would be to grammatical relations such as subject and object. Some theories of syntax (e.g. Perlmutter and Postal (1983); see also other papers in that volume) have taken these as primitives, but there has been much fruitful work that derives subjecthood or objecthood from structural facts. My approach to the Reichenbachian temporal roles is in the spirit of the latter.

Klein, in fact, adds a structural component to the theory, to the effect that morphemes are responsible for the relations. (A more elaborated structural theory is also given by Thompson (1994), to be discussed below.) In the case of the present perfect, for example, the present tense is responsible for the R-S relation, and the perfect aspect (or tense) is responsible for the R-E relation. Not only that, but any other aspect, such as imperfective or perfective, can also affect the R-E relation. To achieve this, Klein treats the Reichenbachian points in terms of intervals, not instants. Thus imperfectivity and perfectivity can be stated in terms of E containing or overlapping R (or in Klein's terms, Time of Situtation containing overlapping Time of Topic).

Once you agree with Hornstein's story that E, R, and S are in all tense constructions, and Klein's story that the relations between E and R on the one hand and R and S on the other are mediated structurally by morphemes in the right positions, (2b) follows as a null hypothesis (with the Reichenbach-Hornstein-Klein representation being something like E,R_S). So to understand more about the motivation for (2b), let's proceed to three arguments of Hornstein's, and one of Klein's, that the simple past should have all three of the Reichenbachian temporal roles.

2.2 Conceptual argument for two operators

The conceptual argument for (2b) as a null hypothesis is as follows. The perfect requires all three time primitives (and hence, on a structural view like Klein's, two operators to mediate between them). We would like all constructions to use all three primitives, to avoid introducing other constraints which explain why not all constructions require all three time primitives. An advantage of such a system is that the number of possible tense-aspect constructions is limited in nature, consisting of all possible combinations of relations between S-R and R-E, making tense-aspect learnable for the child.

However, the fact that (1) has a single piece of temporal-aspectual morphology and the past perfect *(John had danced)* has (at least) two is certainly enough to motivate a theory in which (1) involves a single operator. As for the learnability issue, we cannot insist that the child needs all tense-aspect constructions to have two operators, or three Reichenbachian temporal roles just because the perfect does; to continue the parallel with grammatical relations, that would be something like insisting that all verbs are underlyingly ditransitive just because some are. Instead, presumably, Universal Grammar includes one-operator tense-aspect constructions (with two times implicated, with pragmatics determining which gets the role of S and which gets the roles of E and R). The child would know that sometimes just one tense-aspect operator is needed, just as she would have to know that some verbs are transitive. She would be led towards such a hypothesis by the fact that such constructions only have one temporal-aspectual morpheme.

Hornstein's concern that this state of affairs would lead to the iteration without end of temporal-aspectual operators – if one, then why not three or four? – should be tempered by the fact that there do exist constructions in various languages with three or four (see Copley (forthming) for a few of these). And in fact, there is some evidence that sometimes even *none* is enough, in the case of some present tenses (Enç (1996), Ogihara (1989)) and languages like Haitian Creole (Dechaine 1991).

2.3 Two temporal adverbials implies two operators

Hornstein further argues that the presence of both R and E in the simple past can be detected by the possibility for two temporal adverbials in examples such as (6).

(6) Yesterday, John left a week ago. (Hornstein's judgment)

I agree with the validity of Hornstein's test, but would like to argue that he gets some of the facts wrong. I find (6) marginal, certainly worse than the perfect version (*Yesterday, John had left a week ago,*) and also worse than, say, a progressive past futurate, as in (7a):

(7) a. Yesterday, John was leaving next week.
b. *? Yesterday, John left next week.

Since progressives clearly have two temporal-aspectual operators – i.e., tense and progressive aspect – they would be expected to have a full complement of Reichenbachian temporal points, according to Klein. Thus it's not surprising that they should be able to have two temporal adverbials. On the other hand, the simple past as in (7b) does not have this ability, which also suggests that there are not two temporal-aspectual operators.

2.4 Temporal connectives need R

Hornstein further argues that a theory which aims to explain the behavior of temporal connectives such as *when, before,* and *after* requires that the simple past have an R point. Our modus operandi here to argue against his claim is to consider in turn the connectives that can occur between the simple past and a construction such as the perfect, which on anyone's theory has to have two operators, and see if any explanatory power is lost by treating the simple past as having a single tense operator.

Suppose then that tenses and sentences are predicates of times, type $\langle i, t\rangle$. (For concreteness, though it could be handled other ways, I assume a semantic framework as in Heim and Kratzer (1998), and also assume that tenses combine with tenseless sentences by predicate modification) For space reasons let us consider here only the connective *after*, which might have the denotation below, where $\exists_c$ stands for a contextually restricted existential quantifier. Both p and q stand for sentences, type $\langle i, t\rangle$.

(8) $[\![\text{after}]\!]$(p)(q) = 1 iff $\exists_c$ i: p(i) and $\exists_c$ j: q(j) and j is after i

For Hornstein, temporal connectives relate R points to each other. In this system, however, *after* relates the time points that are arguments of the component sentences p and q. A further assumption I will make, following Stowell (1996) and Iatridou (2001), is that the perfect has a stative feature in it, spelled out as *have*. Semi-formally, the perfect can be partially broken down as in (9):

(9) $[\![\text{John had left at 5}]\!]$(i) = 1 iff at i John was in the state of having left at 5

Using these assumptions, there are no problems accounting for the meaning of (10a,b). I include the second temporal argument *at 5* in the case of the perfect to show that the argument *i* really can be (the same time as) the R point, as desired.

(10) a. John left after Harry arrived.
$\exists_c$ i: $[\![\text{Harry arrived}]\!]$(i) and $\exists_c$ j: $[\![\text{John left}]\!]$(j) and j is after i

b. John left after Harry had arrived at 5.
$\exists_c$ i: $[\![\text{Harry had arrived at 5}]\!]$(i) and $\exists_c$ j: $[\![\text{John left}]\!]$(j) and j is after i
$\exists_c$ i: at i Harry was in the state of having arrived at 5 and $\exists_c$ j: $[\![\text{John left}]\!]$(j) and j is after i

Hornstein's theory of temporal connectives rules out (11a), although I find it to be fairly good. Even better is (11b).

(11) a. John had left after Harry arrived. (Hornstein's judgment)

b. John had insulted Mary$_i$ after she$_i$ left.

Why is (11b) better? The temporal connective *after she left* attaches relatively low, which is a possibility with the perfect, according to Thompson (1994); in (11b) that position is favored because of the coreference. The structure would be in part as in (12):

(12) John had [insulted Mary after she left]

According to Thompson, there is a higher attachment point for the adverbial (we saw this with the two temporal arguments for the perfect above, as well). Perhaps it is this higher attachment that yields Hornstein's judgment on (11), stemming from a conflict between the stativity of the perfect and a revised meaning of *after*, reflecting the idea that statives are generally degraded where they appear with *after*-clauses (Dowty 1979; Smith 1991). Supposing this argument holds for other temporal connectives as well, temporal connectives provide no rationale for including three temporal points (and two operators) in the semantics of the simple past.

2.5 Topic Time is real

Here we return to Klein, for whom "Time of Topic" (TT) is the analogue to R. TT is "the time span to which the speaker's claim on this occasion is confined." (Of course there are utterances which do not make any claim, but the idea of how to extend this definition will be clear enough that we may justifiably abstract away from those cases.) Klein's example to illustrate this uses a discourse between a judge and a witness, given in (13).

(13) a. What did you notice when you looked into the room?

b. There was a book on the table. It was in Russian.

Klein's point here is that the statements in (13b) can be true without saying anything about whether the situations still hold at the time of utterance, or indeed at any time other than the one provided by context. Perhaps the book has since been removed from the table; the "Time of Situation" (TSit, analogous to E), corresponding to the interval that the book was on the table, in any case was not limited to the interval over which the witness looked into the room. This point is even more clear with the second statement in (13b). Presumably the book remains in Russian. Yet the past tense is nonetheless felicitous, and indeed is much more so than the present tense in this discourse (at least in English).

Still, it is not at all clear that these facts need, or even should, be addressed via a temporal primitive R, as has been pointed out by Kamp and Reyle (1993); in addition, ter Meulen (1995) attributes such usages as (13b) in part to the idea that statives piggyback on a temporal node in the discourse without creating one of their own. So again, it may not be the semantics that must account for the fact in (13b), but rather the pragmatics.

There are a number of ways to put contextual information into the tense, either by treating it as a referential entity (see for instance Partee (1984)), or by putting a contextual restriction on a temporal existential quantifier, as we did above. Thus Klein's argument for the reality of Topic Time cannot be taken as an argument for R as a semantic primitive.

3 Arguments for only one operator

So far we have seen that there is no reason to posit a third temporal point R in the English simple past. On a structural view such as Klein's, in which operators associated with tense and aspectual atoms correlate with the Reichenbach/Hornstein relations, this would mean that there is no reason to posit an aspectual operator between T and the VP. But there are two further reasons why it would be preferable not to posit an operator there[2].

3.1 Condition C and adverbials

As alluded to above, (Thompson 1994), investigating the perfect, finds that different attachment sites for temporal adverbials in perfect constructions come with different possibilities for modification of R or E. For example, (14) has two readings, but (15) has only one:

(14) Mary had seen John at 3.

a. Reading 1: The seeing time was at 3. (adv. modifies E)
b. Reading 2: The seeing time was before 3 (adv. modifies R)

(15) Mary had seen him$_i$ at the time John$_i$ presented his paper.

a. *Reading 1: The seeing time was at the time of the presentation. (adv. modifies E)
b. Reading 2: The seeing time was before the time of the presentation. (adv. modifies R)

(15) is bad because the lower position for the adverbial is low enough to be c-commanded by the object (assumed to have undergone object shift), violating Condition C of the binding theory, which states that a referring expression such as *John* must not be c-commanded by anything that refers to the same entity (in this case, John). The R-modifying reading remains good, however, because it is high enough not to be c-commanded by the object.

[2] These are not, however, arguments against an extra perfective operator within the VP as proposed by, e.g., (Travis 1991), or aspect associated with the direct object as in (Tenny 1994).

If there really is both an E and an R in the simple past, we would expect them both to be available to adverbial modification. Given Klein's assumption that temporal-aspectual heads correspond to temporal relations, this means that we should be able to get adverbials to be at both heights in the simple past. And even if R and E in the simple past happen to be cotemporaneous, so that we cannot distinguish two different readings as we did for the perfect in (14), we should still be able to distinguish two different readings with the condition C test as in (15). However, only the lower position, modifying E, seems to be available, since there is a Condition C violation:

(16) * Mary saw him_i at the time of John_i's presentation

We may conclude that the fact that the adverbial can only be at the E-modifying height means that there is no R-modifying position available, and therefore no morpheme in the field between T and VP responsible for relating R to E.

3.2 Tense must be near the verb for morphological reasons

The analysis in (2b) is also a problem in view of the morphological closeness of past tense and the verb in English (Alec Marantz, p.c.). The existence of ablauted forms (*sing-sang*) suggests that the past tense is local enough to the verb (in some sense of "local enough"), so that the relevant morphological processes can occur conditioned on the identity of the verb root. Whatever the precise statement of locality that is needed here, the affixation of a null morpheme between tense and the verb would be expected to keep the tense feature far enough from the verb that the correct form of the verb could not be inserted.

3.3 "Perfectives" without two operators?

I have argued so far against the first idea that motivated the structure in (2b): the idea that an intervening operator is needed to relate R to E in the simple past. But what about the second idea, Klein's claim that there is a contentful aspectual operator there, in addition to the tense operator, namely the perfective operator?

Actually, Smith (1991) finds the semantics of the English "perfective" to be extremely bland, unlike perfectives in other languages. It essentially passes up the aspectual semantics given by the Aktionsart ("situation type") of the verb:

> For English [the perfective] has a consistent yet variable meaning: the perfective presents in its entirety the temporal schema associated with each situation type, as in (Smith 1991).

If that is the case, the semantics of English "perfectives" are no obstacle to choosing (2a) as the structure of (1).

However, Kratzer (1998) brings up some differences between the English simple past and the German simple past that are supposed to show that the English simple past is not an (anaphoric) past tense but rather can also be the reflex of a structure with present tense and a Klein-style aspectual operator as well. Her argument is made in the wake of the main point of the paper, that tense can be anaphoric in nature. Her concern is that there are some cases in which the English past tense seems not to behave anaphorically. For example, the English past tense, but not the German past tense, is possible in situations where no past time is contextually salient, as in (17).

(17) a. Who built this church? Borromini built this church.

b. * *Wer baute diese Kirche? Borromini baute diese Kirche.*
Who built this church? Borromini built this church.
'Who built this church? Borromini built this church.'

The English past tense, but not the German past, can also be used in embedded contexts in which similarly, no past time is contextually salient (so in (18), for instance, the relevant context would be one in which the letters were not received at some previous salient time before the utterance time):

(18) a. We will answer every letter we got.

b. # *Wir werden jeden Brief beantworten, den wir bekamen.*
We will every letter answer that we received
'We will answer every letter that we received.'

Kratzer accounts for this abberrant behavior of the English past tense by positing an ambiguity between the anaphoric past and a present tense plus anterior aspect combination. She calls the anterior aspectual element responsible for the non-anaphoric pastness the "perfect" (acknowledging that this is certainly not the same perfect as the English perfect construction which uses *have*).

But does this proposal commit us to the structure in (2b), for this aspectual version of the simple past tense? Not if we are to believe Enç (1996) and Ogihara (1989) (whose arguments I unfortunately lack space to discuss here) that present tense is no tense at all, but rather the absence of tense. In that case the one temporal-aspectual operator I have been arguing for could be Kratzer's "perfect" aspect. More research is called for here, but Kratzer's idea does not *a priori* pose a problem for maintaining the structure in (2a) as the logical form of (1).

4 Conclusion

I hope to have shown that the arguments made by Hornstein and Klein for three times in the English simple past to be unconvincing. On a structural view this argues for (2a) over (2b) as the null hypothesis, since only one temporal-aspectual operator is needed; I supported this hypothesis with evidence that there is only one operator there. Finally, there are some indications that (2b) looks strange semantically as well. All of these are evidence that the Reichenbachian temporal roles should not be taken as primitives of a theory of temporal-aspectual semantics.

References

Copley, B. (forthcoming). Ph. D. thesis, MIT.

Dechaine, R. (1991). Bare sentences. In *SALT I*, Ithaca. Cornell University.

Dowty, D. (1979). *Word meaning and Montague Grammar.* Dordrecht: Reidel.

Enç, M. (1996). Tense and Modality. In S. Lappin (Ed.), *The Handbook of Contemporary Semantic Theory*, pp. 345–358. Oxford, England: Blackwell.

Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar.* Blackwell.

Hornstein, N. (1990). *As Time Goes By.* MIT Press.

Iatridou, S. (2001). Temporal Existentials? MIT unpublished ms.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic.* Dordrecht: Kluwer.

Klein, W. (1997). *Time in Language.* New York: Routledge.

Kratzer, A. (1998). More Structural Analogies Between Pronouns and Tenses. In D. Strolovitch and A. Lawson (Eds.), *Semantics and Linguistic Theory VIII*, Ithaca, NY, pp. 92–128. Cornell Linguistics Circle.

Ogihara, T. (1989). *Temporal Reference in English and Japanese.* Ph. D. thesis, Ann Arbor, MI.

Partee, B. (1984, August). Nominal and Temporal Anaphora. *Linguistics and Philosophy 7*(3), 243–286.

Perlmutter, D. M. and P. M. Postal (1983). The Relational Succession Law. In *Studies in Relational Grammar*, Chapter 2, pp. 30–80. Chicago: University of Chicago Press.

Reichenbach, H. (1947). *Elements of Symbolic Logic*. New York: MacMillan.

Smith, C. (1991). *The parameter of aspect*. Dordrecht: Kluwer.

Stowell, T. (1996). The Phrase Structure of Tense. In J. Rooryck and L. Zaring (Eds.), *Phrase Structure and the Lexicon*, Volume 33 of *Studies in Natural Language and Linguistic Theory*. Kluwer.

Tenny, C. (1994). *Aspectual Roles and the Syntax-Semantics Interface*. Dordrecht: Kluwer.

ter Meulen, A. (1995). *Representing Time in Natural Language*. Cambridge, MA: MIT Press.

Thompson, E. (1994). The Structure of Tense and the Syntax of Temporal Adverbs. In R. Aranovich, W. Byrne, S. Preuss, and M. Senturia (Eds.), *Proceedings of the Thirteenth West Coast Conference on Linguistics*, Palo Alto, CA, pp. 499–514. CSLI.

Travis, L. (1991). Derived Objects, Inner Aspect, and the Structure of VP. McGill University ms.

The Compositional Rule of Inference in an Intuitionistic Fuzzy Logic Setting

CHRIS CORNELIS
Department of Mathematics and Computer Science, Ghent University
chris.cornelis@rug.ac.be

GLAD DESCHRIJVER
Department of Mathematics and Computer Science, Ghent University
glad.deschrijver@rug.ac.be

ABSTRACT. The incorporation of imprecise, linguistic information into logical deduction processes, as opposed to the practice of traditional two–valued propositional logic and set theory, continues to be a predominant feature of fuzzy expert systems. Throughout the literature, we can find all sorts of intelligent inference schemes acting under imprecision; common to most approaches is their reliance on if–then rules of the kind "IF X is A THEN Y is B", where A and B are fuzzy sets in given universes U and V. Intuitively, fuzzy sets (FSs) can be used to model elastic constraints on the values a variable may assume. While the theory of FS–based approximate reasoning is surely a well–established and commonly applied one, there is still a demand for further expanding the expressiveness of the formalism. One such improvement can be obtained by using Atanassov's (Atanassov 1983) intuitionistic fuzzy sets (IFSs), of which FSs are specific instances, and which highlight the fundamental importance of negation: the degree to which a proposition is false, or equivalently to which an object does not belong to a set, is given an independent status here. In this paper we will contribute to the further development of this relatively young theory, by generalizing the well-known Compositional Rule of Inference (CRI) to IFSs. We also deal with the related problem of checking the validity of the inference, as motivated in (Atanassov and Gargov 1998).

1 Introduction and preliminaries

Inference is defined as a procedure for deducing new facts out of existing ones on the basis of formal deduction rules. Classical paradigms like two–valued propositional and predicate logic, exhibit some important drawbacks (lack of expressivity in describing incomplete and/or imprecise knowledge, high computational complexity) that make them unsuitable for application in automated deduction systems (e.g. for medical diagnosis). To alleviate these difficulties, Zadeh in 1973 introduced a formalism called approximate reasoning to cope with problems which are too complex for exact solution

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

but which do not require a high degree of precision. (Zadeh 1975) His work is centered around the notions of a fuzzy set and a fuzzy restriction.

In his seminal 1965 paper (Zadeh 1965), Zadeh generalized ordinary sets to fuzzy sets (FSs, for short), allowing an element $u \in U$ to belong to any degree of membership in [0,1] (denoted $A(u)$) to a fuzzy set A in U. It is clear that the extension equivalently gives rise to a continuum of truth values between 0 and 1 for a logical proposition P.

Definition 1.1. (Fuzzy set) *A fuzzy set A in a given universe U is a mapping from U into the unit interval $[0, 1]$. The class of fuzzy sets in U is denoted $\mathcal{F}(U)$.*

To define the intersection and union of fuzzy sets (equivalently, conjunction and disjunction of fuzzy propositions), so–called t–norms and t–conorms are used: a t–norm is any symmetric, associative, increasing $[0, 1] \times [0, 1] \to [0, 1]$ mapping T satisfying $T(1, x) = x$ for every $x \in [0, 1]$, whereas for a t–conorm S the last property is replaced by $S(0, x) = x$ for every $x \in [0, 1]$. t–norms give rise to fuzzy intersections, in the sense that $A \cap_T B(u) = T(A(u), B(u))$ for every $u \in U$ and T a t–norm. An analogous result holds of course for t–conorms and unions.

Now consider the statement: “Paul is very old”. Modelling “very old” as a fuzzy set on a suitable range of ages, this statement constitutes a so–called fuzzy restriction on the possible values of Paul’s age rather than an assertion about the membership of Paul in a class of individuals. (Zadeh 1975) From a logical perspective, it is interesting to see how people are able to combine such information efficiently in a Modus Ponens–like fashion to allow for inferences of the following kind:

IF	bath water is “too hot”	THEN	I’m apt to get burnt
	bath water is “really rather hot”		
			I’m quite apt to get burnt

The technique used above is in fact less restrictive than the actual MP from propositional logic since it doesn’t require the observed fact (“really rather hot”) and the antecedent of the rule (“too hot”) to coincide to yield a meaningful conclusion. The need emerges for a flexible, qualitative scale of measuring to what extent the antecedent is fulfilled, on the basis of which we could obtain an approximate idea (stated under the form of another fuzzy restriction) of the value of the consequent variable.

With his introduction of a calculus of fuzzy restrictions (Zadeh 1975), Zadeh paved the way towards a reasoning scheme called Generalized Modus Ponens (GMP) to systematize deductions like the example we presented. Since his pioneering work, many researchers have sought for efficient realizations[1] of this approximate inference scheme. In section 2 we will formally

[1]By “realization”, we mean any computational procedure unambiguously defining the output in terms of the inputs

define the GMP and survey its most common realization, the Compositional Rule of Inference (CRI). Section 3 introduces the notion of Intuitionistic Fuzzy Sets (IFSs) and their connectives. In section 4, we proceed to extend the CRI to the IFS setting. In section 5, we address a common validation procedure based on the notion of Intuitionistic Fuzzy Tautology (IFT) and discuss how it affects our reasoning processes. Finally, section 6 offers some options for future research.

2 FS–based Compositional Rule of Inference

We start by recalling from (Cornelis et al. 2000) the definition of the main concept that we are concerned with:

Definition 2.1. (Generalized Modus Ponens, GMP) *Let X and Y be variables assuming values in U, resp. V. Consider then a fuzzy rule "IF X is A, THEN Y is B" and a fuzzy fact (or observation) "X is A'" ($A, A' \in \mathcal{F}(U), B \in \mathcal{F}(V)$). The GMP allows deduction of a fuzzy fact "Y is B'", with $B' \in \mathcal{F}(V)$.*

Expressing this under the form of an inference scheme, we get:

$$\begin{array}{ll} \textit{IF } X \textit{ is } A, \textit{ THEN} & Y \textit{ is } B \\ X \textit{ is } A' & \\ \hline & Y \textit{ is } B' \end{array}$$

Definition 2.1 does not state what the fuzzy restriction B' should be when A, A' and B are given. A lot of approaches have been proposed for this purpose ((Cornelis 2000), among others, gives a survey), the most common one relying on the so–called Compositional Rule of Inference, a convenient mechanism for calculating with fuzzy restrictions introduced by Zadeh in (Zadeh 1975).

Definition 2.2. (Compositional Rule of Inference, CRI) *(Cornelis et al. 2000) Let X and Y be defined as in definition 2.1. Consider also fuzzy facts "X is A'" and "X and Y are R", where $A' \in \mathcal{F}(U), R \in \mathcal{F}(U \times V)$ (R is a fuzzy relation between U and V). The CRI allows us to infer the fuzzy fact: "Y is $R \circ_T A'$", in which the direct image of A' under R, denoted[2] $R \circ_T A'$, is defined as, for $v \in V$:*

$$R \circ_T A'(v) = \sup_{u \in U} T(A'(u), R(u, v))$$

Expressing this under the form of an inference scheme, we get:

[2]Some people prefer to speak of the "composition of R with A'" hence the appearance of the composition symbol.

$$\begin{array}{l} X \textit{ is } A' \\ X \textit{ and } Y \textit{ are } R \\ \hline Y \textit{ is } R \circ_T A' \end{array}$$

For definition 2.2 to be a realization of the GMP, R must be a relational representation of a fuzzy implicator, an extension of the classical implication operator:

Definition 2.3. (Fuzzy implicator) *(Ruan and Kerre 1993) A fuzzy implicator is any* $[0,1]^2 \to [0,1]$ *mapping* $\mathcal{I}$ *for which the restriction to* $\{0,1\}^2$ *coincides with classical implication:* $\mathcal{I}(0,0) = 1$, $\mathcal{I}(1,0) = 0$, $\mathcal{I}(0,1) = 1$, $\mathcal{I}(1,1) = 1$. *Moreover,* $\mathcal{I}$ *should satisfy the following monotonicity criteria:*

$$(\forall y \in [0,1])(\forall (x,x') \in [0,1]^2)(x \leq x' \Rightarrow \mathcal{I}(x,y) \geq \mathcal{I}(x',y)) \tag{8.1}$$

$$(\forall x \in [0,1])(\forall (y,y') \in [0,1]^2)(y \leq y' \Rightarrow \mathcal{I}(x,y) \leq \mathcal{I}(x,y')) \tag{8.2}$$

Given $\mathcal{I}$ and A and B, the fuzzy sets used in definition 2.1, R is defined as, for $(u,v) \in U \times V$: $R(u,v) = \mathcal{I}(A(u), B(v))$. The two most important classes of fuzzy implicators are called $\mathcal{S}$– and $\mathcal{R}$–implicators, and are defined as follows[3]:

Definition 2.4. ($\mathcal{S}$–implicator) *(Bouchon-Meunier et al. 1999) Let* S *be a t–conorm. The* $\mathcal{S}$*–implicator generated by* S *is the mapping* $\mathcal{I}_S$ *defined as:*

$$\begin{array}{rccl} \mathcal{I}_S: & [0,1]^2 & \to & [0,1] \\ & (x,y) & \mapsto & S(1-x,y), \quad \forall (x,y) \in [0,1]^2 \end{array}$$

Definition 2.5. ($\mathcal{R}$–implicator) *(Bouchon-Meunier et al. 1999) Let* T *be a t–norm. The* $\mathcal{R}$*–implicator generated by* T *is the mapping* $\mathcal{I}_T$ *defined as:*

$$\begin{array}{rccl} \mathcal{I}_T: & [0,1]^2 & \to & [0,1] \\ & (x,y) & \mapsto & \sup\{\gamma \in [0,1] | T(x,\gamma) \leq y\}, \quad \forall (x,y) \in [0,1]^2 \end{array}$$

3 Intuitionistic Fuzzy Sets

IFSs, first introduced by Atanassov (Atanassov 1983) in 1983, generalize Zadeh's fuzzy sets. While FSs merely give the degree of membership of an element in a set, IFSs also involve a degree of non–membership.

Definition 3.1. *An intuitionistic fuzzy set in a universe* U *is any object* A *of the form* $A = \{(u, \mu_A(u), \nu_A(u)) | u \in U\}$, *where the membership function* μ_A *and the non–membership function* ν_A *are* $U \to [0,1]$ *mappings satisfying* $(\forall u \in U)(\mu_A(u) + \nu_A(u) \leq 1)$. *The class of all IFSs in* U *is denoted* $\mathcal{IF}(U)$.

[3] It is easily verified that they are indeed fuzzy implicators. (Cornelis 2000)

Clearly any FS $A \in \mathcal{F}(U)$ has an IFS representation where for any $u \in U$ the degree of non–membership equals one minus the degree of membership. There also exist straightforward extensions of the FS union and intersection to IFSs. Let T be a t-norm and S a t-conorm. Then the generalized intersection $A \cap_{T,S} B$ of two IFSs A and B in U can be defined as $A \cap_{T,S} B = \{(u, T(\mu_A(u), \mu_B(u)), S(\nu_A(u), \nu_B(u))) | u \in U\}$. The resulting object is again an IFS provided $T \leq S^*$, where S^* denotes the dual t-norm of S, defined as $S^*(x,y) = 1 - S(1-x, 1-y)$ for all x and y in $[0,1]$. Indeed, from $T \leq S^*$ and using the increasing property of the t-conorm S, we obtain $T(\mu_A(u), \mu_B(u)) \leq 1 - S(1-\mu_A(u), 1-\mu_B(u)) \leq 1 - S(\nu_A(u), \nu_B(u))$. Putting $\nu_A(u) = 1 - \mu_A(u)$ and $\nu_B(u) = 1 - \mu_B(u)$, it is clear that the condition $T \leq S^*$ is also necessary. A similar result can be obtained for the IFS union $\cup_{S,T}$, under the condition $S \leq T^*$, which is equivalent to $T \leq S^*$.

4 IFS–based Compositional Rule of Inference

As discussed in the previous section, IFSs offer a more general framework than FSs do, thus allowing representation of relations between variables that could previously not be described (see e.g. (Atanassov 1999) (De et al. 2001) for some real–world examples). Also in (Atanassov 1999), a first attempt is made to endow fuzzy expert systems with concepts from IFS theory, indicating the real interest excited by these structures. It would therefore be nice to find some suitable IFS adaptation of the CRI, by far the most common means of deduction in the FS setting.

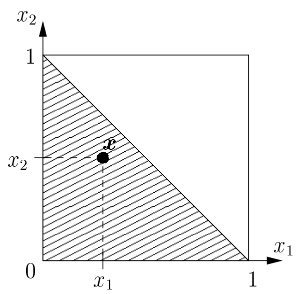
Before we can generalize the GMP and the CRI, we have to introduce some preliminary concepts.

Definition 4.1. (Lattice $(L^*, \leq_{L^*})$) *Define a lattice $(L^*, \leq_{L^*})$ such that:*

$$L^* = \{(x_1, x_2) \in [0,1]^2 \mid x_1 + x_2 \leq 1\}$$

$$(x_1, x_2) \leq_{L^*} (y_1, y_2) \Leftrightarrow x_1 \leq y_1 \wedge x_2 \geq y_2$$

The shaded area in the figure is the set of elements $x = (x_1, x_2)$ belonging to L^*.



The lattice $(L^*, \leq_{L^*})$ is a complete lattice: for each $A \subseteq L^*$,

$$\begin{aligned} \sup A &= (\sup\{x_1 \in [0,1] \mid (\exists x_2 \in [0,1])((x_1,x_2) \in A)\}, \\ &\quad \inf\{x_2 \in [0,1] \mid (\exists x_1 \in [0,1])((x_1,x_2) \in A)\})\,, \\ \inf A &= (\inf\{x_1 \in [0,1] \mid (\exists x_2 \in [0,1])((x_1,x_2) \in A)\}, \\ &\quad \sup\{x_2 \in [0,1] \mid (\exists x_1 \in [0,1])((x_1,x_2) \in A)\})\,. \end{aligned}$$

Equivalently, this lattice can also be defined as an algebraic structure $(L^*, \wedge, \vee)$ where the meet operator $\wedge$ and the join operator $\vee$ are defined as follows, for $(x_1,x_2),(y_1,y_2) \in L^*$:

$$(x_1,x_2) \wedge (y_1,y_2) = (\min(x_1,y_1), (\max(x_2,y_2))$$

$$(x_1,x_2) \vee (y_1,y_2) = (\max(x_1,y_1), \min(x_2,y_2))$$

For our purposes, we will also consider a generalized meet operator $\wedge_{T,S}$ and join operator $\vee_{S,T}$ on $(L^*, \leq_{L^*})$:

$$(x_1,x_2) \wedge_{T,S} (y_1,y_2) = (T(x_1,y_1), S(x_2,y_2))$$

$$(x_1,x_2) \vee_{S,T} (y_1,y_2) = (S(x_1,y_1), T(x_2,y_2))$$

for given t–norm T and t–conorm S satisfying $T \leq S^*$. Again, it can be shown that the condition $T \leq S^*$ is necessary and sufficient for these operators to be well-defined.

Lastly, we define an order–reversing mapping $\mathcal{N}$ by $\mathcal{N}(x_1,x_2) = (x_2,x_1)$, $\forall (x_1,x_2) \in L^*$.

We now propose the following extensions for intuitionistic fuzzy implicators, as well as the special instances of $\mathcal{S}$– and $\mathcal{R}$–implicators.

Definition 4.2. (Intuitionistic Fuzzy Implicator) *An intuitionistic fuzzy implicator is any $(L^*)^2 \to L^*$–mapping $\mathcal{I}$ satisfying the border conditions*

$$\begin{aligned} \mathcal{I}(0_{L^*}, 0_{L^*}) &= 1_{L^*}, \\ \mathcal{I}(1_{L^*}, 0_{L^*}) &= 0_{L^*}, \\ \mathcal{I}(0_{L^*}, 1_{L^*}) &= 1_{L^*}, \\ \mathcal{I}(1_{L^*}, 1_{L^*}) &= 1_{L^*}, \end{aligned}$$

where $0_{L^*} = (0,1)$ and $1_{L^*} = (1,0)$ are the identities of $(L^*, \leq_{L^*})$. Moreover we require $\mathcal{I}$ to be decreasing in its first, and increasing in its second component, i.e.

$$(\forall y \in L^*)(\forall (x,x') \in (L^*)^2)(x \leq_{L^*} x' \Rightarrow \mathcal{I}(x,y) \geq_{L^*} \mathcal{I}(x',y)) \tag{8.3}$$

$$(\forall x \in L^*)(\forall (y,y') \in (L^*)^2)(y \leq_{L^*} y' \Rightarrow \mathcal{I}(x,y) \leq_{L^*} \mathcal{I}(x,y')) \tag{8.4}$$

Definition 4.3. ($\mathcal{S}$–implicator) *Let T be a t-norm, S a t-conorm satisfying $T \leq S^*$, and $\mathcal{N}$ an involutive*[4] *order–reversing operator on L^*. The $\mathcal{S}$–implicator generated by T, S and $\mathcal{N}$ is the mapping $\mathcal{I}_{S,T,\mathcal{N}}$ defined as, for $(x,y) \in (L^*)^2$:*

$$\mathcal{I}_{S,T,\mathcal{N}}(x,y) = \mathcal{N}(x) \vee_{S,T} y$$

Definition 4.4. ($\mathcal{R}$–implicator) *Let T be a t-norm, S a t-conorm satisfying $T \leq S^*$. The $\mathcal{R}$–implicator generated by T and S is the mapping $\mathcal{I}_{T,S}$ defined as, for $(x,y) \in (L^*)^2$:*

$$\mathcal{I}_{T,S}(x,y) = \sup\{\gamma \in L^* \mid x \wedge_{T,S} \gamma \leq_{L^*} y\}$$

Theorem 4.1. *The mappings $\mathcal{I}_{S,T,\mathcal{N}}$ and $\mathcal{I}_{T,S}$ are intuitionistic fuzzy implicators.*

Proof. It is easy to verify that the defined operators satisfy the border conditions. We now prove that they satisfy the hybrid monotonicity properties.

(i) Since each t-norm T and each t-conorm S are increasing in both components, it can easily be seen that, for all $(a_1,a_2),(b_1,b_2),(c_1,c_2)$, $(d_1,d_2) \in L^*$, $(a_1,a_2) \leq_{L^*} (b_1,b_2)$ and $(c_1,c_2) \leq_{L^*} (d_1,d_2)$ implies $(S(a_1,c_1),T(a_2,c_2)) \leq_{L^*} (S(b_1,d_1),T(b_2,d_2))$, i.e. $(a_1,a_2) \vee_{S,T} (c_1,c_2)$ $\leq_{L^*} (b_1,b_2) \vee_{S,T} (d_1,d_2)$. Hence $\vee_{S,T}$ is increasing in both components. Similarly $\wedge_{T,S}$ is increasing in both components.

Since $\mathcal{N}$ is order–reversing, we obtain successively, for $x,x',y \in L^*$,

$$\begin{array}{rcl} x & \leq_{L^*} & x' \\ \mathcal{N}(x) & \geq_{L^*} & \mathcal{N}(x') \\ \mathcal{N}(x) \vee_{S,T} y & \geq_{L^*} & \mathcal{N}(x') \vee_{S,T} y \\ \mathcal{I}_{S,T,\mathcal{N}}(x,y) & \geq_{L^*} & \mathcal{I}_{S,T,\mathcal{N}}(x',y) \end{array}$$

It is equally obvious that $\mathcal{I}_{S,T,\mathcal{N}}$ is increasing in its second component.

(ii) Now we prove that an arbitrary $\mathcal{R}$–implicator $\mathcal{I}_{T,S}$ is hybrid monotonous. Let $x = (x_1,x_2), x' = (x'_1,x'_2), y = (y_1,y_2) \in L^*$ such that $x \leq_{L^*} x'$. Then

$$\begin{aligned} &\{(\gamma_1,\gamma_2) \in L^* \mid (T(x_1,\gamma_1),S(x_2,\gamma_2)) \leq_{L^*} (y_1,y_2)\} \\ &\qquad \supseteq \{(\gamma_1,\gamma_2) \in L^* \mid (T(x'_1,\gamma_1),S(x'_2,\gamma_2)) \leq_{L^*} (y_1,y_2)\} \end{aligned}$$

since $T(x_1,\gamma_1) \leq T(x'_1,\gamma_1)$ and $S(x_2,\gamma_2) \geq S(x'_2,\gamma_2)$. Hence

$$\begin{aligned} &\sup\{(\gamma_1,\gamma_2) \in L^* \mid (T(x_1,\gamma_1),S(x_2,\gamma_2)) \leq_{L^*} (y_1,y_2)\} \\ &\qquad \geq \sup\{(\gamma_1,\gamma_2) \in L^* \mid (T(x'_1,\gamma_1),S(x'_2,\gamma_2)) \leq_{L^*} (y_1,y_2)\} \end{aligned}$$

Analogously, monotonicity in the second component is obtained.

[4] An $X \to X$–mapping f is involutive iff, for all $x \in X, f(f(x)) = x$

□

A generalization of the CRI will require us to define the direct image of an IFS under an intuitionistic fuzzy relation (IFR).

Definition 4.5. *Let $A \in \mathcal{IF}(U)$, $R \in \mathcal{IF}(U \times V)$, T a t–norm and S a t–conorm satisfying $T \leq S^*$. The direct image $R \circ_{T,S} A$ is defined as:*

$$R \circ_{T,S} A = \left\{ \left(v, \sup_{u \in U} T(\mu_A(u), \mu_R(u,v)), \inf_{u \in U} S(\nu_A(u), \nu_R(u,v)) \right) \mid v \in V \right\} \tag{8.5}$$

Now we have all the necessary tools to generalize the GMP and the CRI. Definition 2.1 can be maintained if we replace the word "fuzzy" by "intuitionistic fuzzy", and $\mathcal{F}(U)$ and $\mathcal{F}(V)$ by $\mathcal{IF}(U)$ and $\mathcal{IF}(V)$ respectively. We call this pattern Intuitionistic Generalized Modus Ponens (IGMP).

Just as in the fuzzy case, a realization of the IGMP can be obtained by defining the output B' as the direct image of the input A' under an intuitionistic fuzzy relation R representing the intuitionistic fuzzy rule. This gives rise to the following generalization of the CRI.

Definition 4.6. (IFS-based Compositional Rule of Inference, ICRI) *Let X and Y be variables assuming values in U, resp. V. Consider intuitionistic fuzzy facts "X is A'" and "X and Y are R", where $A' \in \mathcal{IF}(U), R \in \mathcal{IF}(U \times V)$ (R is an intuitionistic fuzzy relation between U and V). The ICRI allows us to infer the fuzzy fact: "Y is $B' = R \circ_{T,S} A'$", where (T,S) is a given pair of a t–norm and a t–conorm satisfying $T \leq S^*$.*
Expressing this under the form of an inference scheme, we get:

$$\begin{array}{l} X \text{ is } A' \\ (X,Y) \text{ is } R \\ \hline Y \text{ is } B' = R \circ_{T,S} A' \end{array}$$

We use an IF implicator $\mathcal{I}$ to define R. Given IFSs A and B and $\mathcal{I}$, we calculate, for $(u,v) \in U \times V$,

$$(\mu_R(u,v), \nu_R(u,v)) = \mathcal{I}((\mu_A(u), \nu_A(u)), (\mu_B(v), \nu_B(v))),$$

thus defining R. Using this definition, it is clear that the ICRI is an extension of the fuzzy-based CRI.

5 Validity of the Modus Ponens

As pointed out in (Atanassov and Gargov 1998), validity of the modus ponens (MP) is essential if one is interested in passing from hypothesis to conclusions without loss in the degree of truth. Before we introduce the definition of validity, we first give the following definition.

Definition 5.1. (Intuitionistic fuzzy tautology, IFT) *(Atanassov and Gargov 1998) Let $a = (a_1, a_2) \in L^*$, then a is said to be an "intuitionistic fuzzy tautology" if and only if $a_1 \geq a_2$.*

Definition 5.2. (Validity of the modus ponens) *(Atanassov and Gargov 1998) We say that the MP is valid for an IF implicator $\mathcal{I}$ iff, for $a = (a_1, a_2), b = (b_1, b_2) \in L^*$, we have*[5]

$$a_1 \geq a_2 \wedge pr_1(\mathcal{I}(a, b)) \geq pr_2(\mathcal{I}(a, b)) \Rightarrow b_1 \geq b_2$$

This amounts to: "if a is an IFT and $\mathcal{I}(a, b)$ is an IFT, then b is an IFT".

Unfortunately, the implicators defined in the previous section, do not satisfy the validity of the modus ponens, as shown in the following theorem.

Theorem 5.1.

- *If for the mapping $\mathcal{N}$ there exists an $x = (x_1, x_2) \in L^*$ such that $x_1 \geq x_2$ and $pr_1(\mathcal{N}(x)) \neq 0$, then the modus ponens in not valid for the $\mathcal{S}$–implicator $\mathcal{I}_{S,T,\mathcal{N}}$.*
- *The modus ponens is not valid for any $\mathcal{R}$–implicator.*

Proof.

- Let $x = (x_1, x_2) \in L^*$ with $x_1 \geq x_2$ be such that $\mathcal{N}(x) = (x'_1, x'_2)$ with $x'_1 \neq 0$, and let $y = (0, x'_1)$. Then

$$\mathcal{I}_{S,T,\mathcal{N}}(x, y) = \mathcal{N}(x) \vee_{S,T} y = (S(x'_1, 0), T(x'_2, x'_1)) = (x'_1, T(x'_2, x'_1)),$$

  with $x'_1 \geq T(x'_2, x'_1)$. This shows that the modus ponens is not valid for $\mathcal{I}_{S,T,\mathcal{N}}$.

- Let $x = (x_1, x_2) \in L^*$ and $y = (y_1, y_2) \in L^*$ such that $y_2 > y_1 \geq x_1 \geq x_2 = 0$. Then $T(x_1, \gamma_1) \leq x_1 \leq y_1$, $\forall \gamma_1 \in [0, 1]$ and $S(x_2, \gamma_2) = \gamma_2$, hence

$$\sup\{(\gamma_1, \gamma_2) \in L^* \mid (T(x_1, \gamma_1), S(x_2, \gamma_2)) \leq_{L^*} y\} = (1 - y_2, y_2).$$

  If $y_2 \leq \frac{1}{2}$, then $1 - y_2 \geq y_2$. For the MP to be valid, we need:

$$(\forall (a_1, a_2) \in L^*)(\forall (b_1, b_2) \in L^*)(a_1 \geq a_2 \wedge \\ pr_1(\mathcal{I}((a_1, a_2), (b_1, b_2))) \geq pr_2(\mathcal{I}((a_1, a_2), (b_1, b_2))) \Rightarrow b_1 \geq b_2)$$

  The chosen x and y satisfy $x_1 \geq x_2$ and $pr_1(\mathcal{I}(x, y)) \geq pr_2(\mathcal{I}(x, y))$, but have $y_1 < y_2$. Hence the MP is not valid.

[5]The projection mappings pr_1, pr_2 are defined, for any $(x_1, x_2) \in L^*$, as: $pr_1(x_1, x_2) = x_1, pr_2(x_1, x_2) = x_2$

$\square$

In (Atanassov and Gargov 1998) Atanassov defines a number of alleged implicators for which the modus ponens is valid. Unfortunately, none of his proposed mappings is an IF implicators in the sense of definition 4.2 (either the border conditions, or the hybrid monotonicity, or both, are violated).

One may wrongly conclude from the above discussion that the implicators defined in section 4 are of minor interest. Indeed, in the fuzzy case validity is defined as, for $a, b \in [0,1]$ and $\mathcal{I}$ a fuzzy implicator, "a is a fuzzy tautology (FT) and $\mathcal{I}(a,b)$ is a FT implies b is a FT", where $x \in [0,1]$ is said to be a FT if and only if $x \geq \frac{1}{2}$ (cfr. (Atanassov and Gargov 1998)). The modus ponens is not valid for fuzzy $\mathcal{S}$–implicators either.[6] However, most of the commonly used fuzzy implicators belong to this class.

Set apart from these considerations, we still find it useful to look for intuitionistic fuzzy implicators for which the modus ponens does hold. This will be the subject of a future paper.

6 Conclusion and Future Work

The interest for Intuitionistic Fuzzy Sets from the perspective of logical deduction will continue to grow as more people become familiar with their straightforward semantics and flexible operations. As it turns out, we could find very useful results not previously established, notably the extension of a very wide range of fuzzy implicators to IFS's and their application in the intuitionistic CRI. The consistency of the reached approach still needs to be looked into systematically. The procedure of checking for validity lined out in section 5, along with several other criteria that the inference should satisfy, provides us with various yardsticks for evaluating the performance of our inference scheme. For FS–based GMP, extensive studies have been carried out for this purpose (Baldwin and Pilsworth 1980) (Fukami et al. 1981).

Acknowledgements

Chris Cornelis would like to thank the Fund for Scientific Research Flanders (FWO) for funding the research elaborated on in this paper.

[6]Specifically, if there exists an $x \in [0,1]$ such that $x \geq \frac{1}{2}$ and $\mathcal{N}(x) = \frac{1}{2}$, then $S(\mathcal{N}(x), y) \geq \mathcal{N}(x) = \frac{1}{2}$ for any $y \in [0,1]$

References

Atanassov, K. T. (1983). Intuitionistic fuzzy sets. (in Bulgarian) VII ITKR's Session, Sofia (deposed in Central Sci.–Technical Library of Bulg. Acad. of Sci., 1697/84.

Atanassov, K. T. (1999). *Intuitionistic fuzzy sets.* Heidelberg/New York: Physica–Verlag.

Atanassov, K. T. and G. Gargov (1998). Elements of intuitionistic fuzzy logic. Part I. *Fuzzy Sets and Systems 95*, 39–52.

Baldwin, J. F. and B. W. Pilsworth (1980). Axiomatic approach to implication for approximate reasoning using fuzzy logic. *Fuzzy Sets and Systems 3*, 193–219.

Bouchon-Meunier, B., D. Dubois, L. Godo, and H. Prade (1999). Fuzzy sets and possibility theory in approximate and plausible reasoning. In J. Bezdek, D. Dubois, and H. Prade (Eds.), *Fuzzy sets in approximate reasoning and information systems*, pp. 15–190. Kluwer Academic Publishers.

Cornelis, C. (2000). Approximate reasoning in fuzzy set theory. Master's thesis (in Dutch), Ghent University.

Cornelis, C., M. De Cock, and E. E. Kerre (2000). The Generalized Modus Ponens in a Fuzzy Set Theoretical Framework. In D. Ruan and E. E. Kerre (Eds.), *Fuzzy IF-THEN Rules in Computational Intelligence, Theory and Applications*, pp. 37–59. Kluwer Academic Publishers.

De, S. K., R. Biswas, and A. R. Roy (2001). An application of intuitionistic fuzzy sets in medical diagnosis. *Fuzzy Sets and Systems 117*, 209–213.

Fukami, S., M. Mizumoto, and T. Tanaka (1981). Some considerations on fuzzy conditional inference. *Fuzzy Sets and Systems 4*, 243–273.

Ruan, D. and E. E. Kerre (1993). Fuzzy implication operators and generalized fuzzy method of cases. *Fuzzy Sets and Systems 54*, 23–37.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control 8*, 338–353.

Zadeh, L. A. (1975). Calculus of fuzzy restrictions. In L. A. Zadeh, K.-S. Fu, K. Tanaka, and M. Shimura (Eds.), *Fuzzy sets and their applications to cognitive decision processes*, pp. 1–40. New York: Academic Press.

Movement as well-formedness conditions

Ralph Debusmann
University of Saarland
rade@coli.uni-sb.de

Abstract. We introduce a new formulation of dependency grammar recently suggested in (Duchier and Debusmann 2001) (henceforth DD). DD shares with GB (Chomsky 1986) a notion of *movement*. In GB, movement is carried out by tree transformations. In DD, it is the effect of well-formedness conditions on dependency trees and does not involve transformations. We illustrate both kinds of movement by showing how both theories analyze German verb-second clauses. Then, we point out the similarities between GB and DD, and raise the question whether GB's transformational notion of movement could be replaced by DD's constraint-based account of movement.

1 Introduction

In this article, we introduce a new formulation of dependency grammar recently suggested by (Duchier and Debusmann 2001) (henceforth DD). One of the key assets of DD is that its axiomatization can be (and has been) turned into efficient constraint-based parsers. DD analyses consist of two tree structures: a *syntactic dependency tree* (*ID tree*) and a *topological dependency tree* (*LP tree*). The ID tree is a dependency tree in the spirit of (Tesnière 1959) whose edges are labeled by grammatical roles. ID trees are unordered (and hence in a sense non-projective), as opposed to LP trees which are ordered and projective. The LP tree is inspired by topological fields theory. Its shape is essentially a flattening of the ID tree, allowing us to handle discontinuous constructions in free word order languages such as German.

The approach taken in DD is very similar to other recent dependency-based approaches tackling discontinuity, such as (Bröker 1998), (Kahane et al. 1998) and (Gerdes and Kahane 2001). It is also reminiscent of HPSG-based theories by (Reape 1994), (Kathol 2000) and (Müller 1999). DD's strong resemblance with Government-Binding (GB) theory (Chomsky 1986) is probably less obvious. We will show that, above all, DD shares with GB a notion of *movement*. In GB, movement is carried out by tree transformations, which have properties undesirable for parsing. In DD, movement is

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

the effect of well-formedness conditions on finite labeled trees and does not involve transformations.

The outline of this article is as follows. We start out by presenting the essentials of topological fields theory in section 2. In section 3, we introduce the dependency-based theory proposed in (Duchier and Debusmann 2001), and show how this theory handles German verb-second clauses. In section 4, we explain an analysis of German verb-second clauses in GB, as put forward by (Grewendorf 1988). Then, in section 5, we compare GB and DD, and hint at the possibility to reformulate parts of GB, movement in particular, in a constraint-based way. The conclusion in section 6 rounds up the article.

2 Topological fields theory

Both GB and DD use ideas from *topological fields theory*. Topological fields theory has a long tradition in German linguistics reaching back to the works of (Herling 1821) and (Erdmann 1886). As an example, consider the sentence below:

Einen	Mann	hat	Maria	geliebt	
A	man(acc)	has	Maria	loved	(9.1)

"A man, Maria has loved."

Topological fields theory divides (9.1) into four parts, which are called *fields*: Vorfeld, left parenthesis, Mittelfeld and right parenthesis:[1]

Vorfeld	left parenthesis	Mittelfeld	right parenthesis
Einen Mann	*hat*	*Maria*	*geliebt.*

where the finite verb *hat* in the left parenthesis and its verbal complement *geliebt* (right parenthesis) form a *bracket* around the non-verbal material in the *Mittelfeld*. The *Vorfeld*, left of the left parenthesis, can be occupied by at most one topicalized constituent, whereas the *Mittelfeld* can host any number of non-verbs. The order of the material in the Mittelfeld is almost arbitrary.

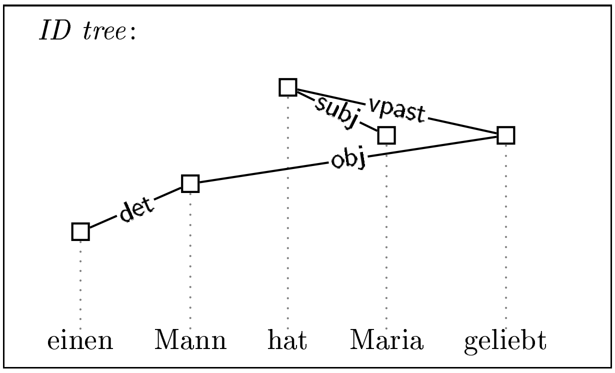
3 Verb-second clauses: A DD analysis

3.1 ID and LP trees

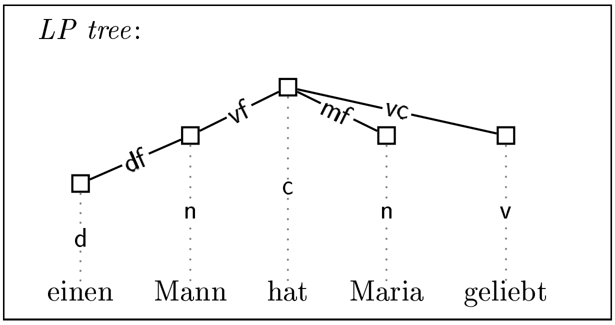
DD distinguishes two tree structures: the unordered ID tree and the partially ordered and projective LP tree. ID and LP trees share the same set of nodes, which correspond one-to-one with words, but have different edges. Below, we give an ID tree analysis of (9.1). Since ID trees are unordered, we can

[1] Actually, topological fields theory postulates one more field which is called *Nachfeld* and hosts material right of the right parenthesis.

pick an arbitrary linear arrangement for display purposes. In the picture below, we stick to the word order given in sentence (9.1).



ID edges are labelled by grammatical functions like subj (for a nominative subject), obj (for an accusative object) vpast (for a past participle complement) and det (for a determiner). The mother of a node in the ID tree is called *head* and its daughters *syntactic dependents.* Here is the corresponding topological dependency (LP tree) analysis:



The mother of a node in the LP tree is called *host* and its daughters *topological dependents.*

3.2 Ordering words in the LP tree

DD employs a set $\mathcal{F}$ of *fields* to determine the licensed linearizations. $\mathcal{F} = \mathcal{F}_{\text{ext}} \uplus \mathcal{F}_{\text{int}}$, where $\mathcal{F}_{\text{ext}} = \{\mathsf{df}, \mathsf{vf}, \mathsf{mf}, \mathsf{vc}\}$ is the set of *external fields* or LP edge labels. $\mathcal{F}_{\text{int}} = \{\mathsf{d}, \mathsf{n}, \mathsf{c}, \mathsf{v}\}$ is the set of *internal fields* or LP node labels.[2] $\mathcal{F}$ is totally ordered, which induces a partial order on LP trees:

1. The topological dependents of each node are ordered by their edge labels or external fields.

2. Each node is positioned with respect to its topological dependents by its node label or internal field.

[2]Note that for expository reasons, $\mathcal{F}$ only includes the fields needed to account for our example. For a full account of German, more fields are required.

The order is partial and not total because if two words land[3] in the same external field, they remain unordered with respect to each other.

The set $\mathcal{F}$ of fields is essentially motivated by topological fields theory. For example vf models the Vorfeld, c the left parenthesis, mf the Mittelfeld and vc the right parenthesis (or *verb cluster*). df and n are used to determine word order within noun phrases: e.g. df stands for *determiner field* and n for *noun field*. The total order on $\mathcal{F}$ is given below:

$$\mathsf{d} \prec \mathsf{df} \prec \mathsf{n} \prec \mathsf{vf} \prec \mathsf{c} \prec \mathsf{mf} \prec \mathsf{vc} \prec \mathsf{v} \tag{9.2}$$

The global total order on $\mathcal{F}$ can be decomposed into local total orders. For instance the local order $\mathsf{df} \prec \mathsf{n}$ requires determiners to precede their corresponding nouns. Conversely, the sequence $\mathsf{vf} \prec \mathsf{c} \prec \mathsf{mf} \prec \mathsf{vc}$ requires the Vorfeld (vf) to precede the left parenthesis (c) to precede the Mittelfeld (mf) to precede the verb cluster or right parenthesis (vc).

In our example, the desired linearization is attained as follows:

1. *Mann* lands in the vf, *Maria* in the mf and *geliebt* in the vc of *hat*. Since $\mathsf{vf} \prec \mathsf{mf} \prec \mathsf{vc}$ in (9.2), *Mann* must precede *Maria* and *Maria* must precede *geliebt*.

2. The internal field of *hat* is c. Because $\mathsf{vf} \prec \mathsf{c} \prec \mathsf{mf}$ in (9.2), it must be placed between the *Mann* in the vf and *Maria* in the mf.

3.3 Example lexicon

DD states well-formedness conditions for LP trees in a lexicalized fashion: a lexical entry stipulates which external fields are *offered* for topological dependents to land in and which are *accepted*. A node w' can land in an external field f of host w iff w *offers* f and w' *accepts* f. A lexical entry also assigns a set of possible internal fields to each word. Here are the lexical entries for our example[4]:

	offers	accepts	internal
einen	{}	{df}	{d}
Mann	{df}	{vf, mf}	{n}
Maria	{}	{vf, mf}	{n}
hat	{vf?, mf∗, vc?}	{}	{c}
geliebt	{}	{vc}	{v}

The set of fields offered by a node is given in *wildcard notation*: e.g. $\mathsf{vf}?$ indicates that there can be at most one topological dependent in the vf (as stated by topological fields theory), and $\mathsf{mf}*$ that any number of daughters can land in the mf.

[3] A node is said to *land* in external field f iff its incoming edge is labeled with f.

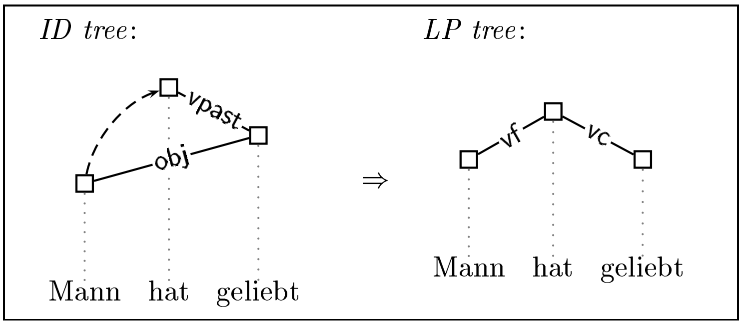
[4] We display only the LP tree part of each lexical entry. Full lexical entries also comprise ID tree information such as subcategorization and agreement.

3.4 Climbing

The well-formedness conditions for LP trees are further constrained by a *grammatical principle*[5]:

Principle 1. *a node must land on a transitive head*

which states that the host w of a node w' in the LP tree must be above w' in the ID tree. If a node lands above of its syntactic head, it is said to have *climbed*. Below, we illustrate how *Mann* climbs into the vf of *hat* (as indicated by the dashed arrow):



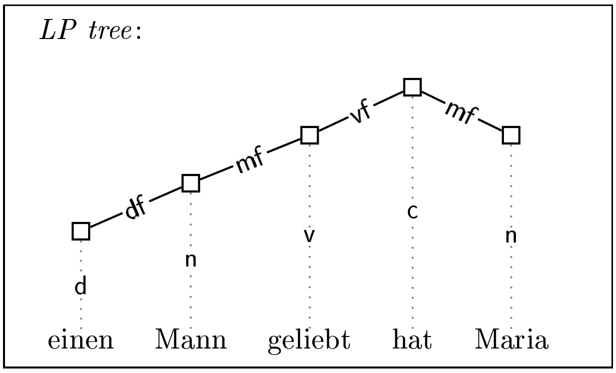
Mann is forced to climb by the well-formedness conditions stated in the lexicon: it cannot land on *geliebt* because *geliebt* offers no field.

3.5 Extending coverage

Note that the example lexicon has been designed to be as simple as possible, and is therefore not representative of DD's coverage. In fact, we have developed a German grammar for our prototype parser which also covers e.g. verb-last sentences, partial verb phrase fronting and relative clauses. In order to also handle verb-last sentences, we introduce separate lexical entries for finite verbs that have internal field v and do not offer a vf. The introduction of additional lexical entries does not really hamper the efficiency of our approach as the parser uses a novel constraint-based treatment of lexical ambiguity as introduced in e.g. (Duchier 1999).

In order to accomodate partial verb phrase fronting in the example lexicon, we would need an additional lexical entry for *geliebt*. If it lands in the Vorfeld, i.e. if it accepts $\{\mathsf{vf}\}$, then it should offer $\{\mathsf{mf}*\}$. An LP tree analysis of the partial verb phrase fronting sentence *Einen Mann geliebt hat Maria* would then look like the one depicted below:

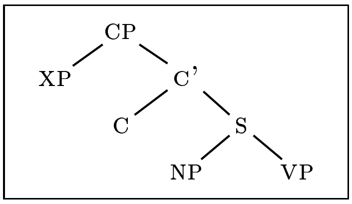
[5] We only mention the first of the three principles given in (Duchier and Debusmann 2001).



4 Verb-second clauses: a GB analysis

4.1 German sentence structure

There are two approaches to analyzing verb-second sentences in GB. We choose the approach taken in (Grewendorf 1988), as originally suggested in (Chomsky 1986). It assumes the following sentence structure for German:[6]

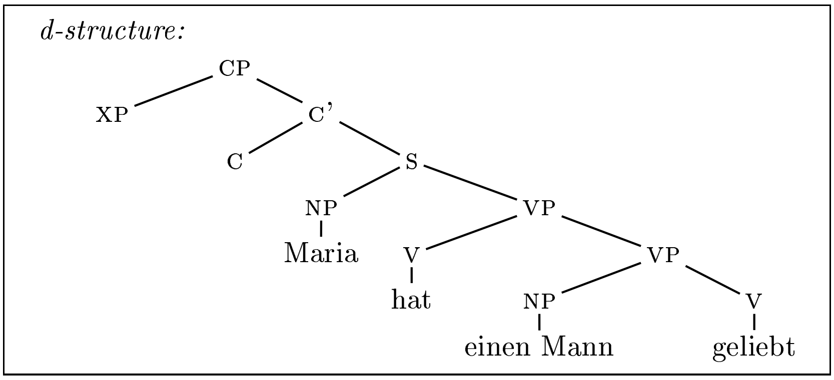


The theory of topological fields can be mapped to the proposed GB structure as follows: the Vorfeld corresponds to the [CP,XP]-position and the left parenthesis to [C',C]. The right parenthesis and the Mittelfeld have no equivalents in the above tree configuration.

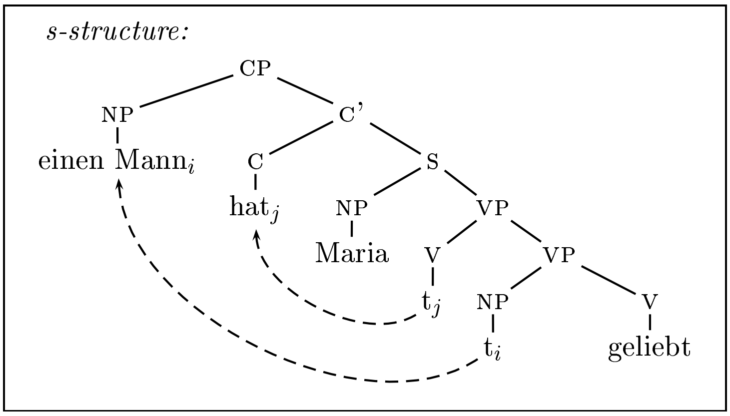
4.2 d- and s-structure

Two of the four levels of analysis that GB posits are of interest for our purposes here: d- and s-structure. Here is a d-structure analysis of (9.1):

[6]For simplicity, we do not depict the IP- and I'-nodes from the original tree shown in (Grewendorf 1988), p. 49.



GB uses an operation called move-α to mediate between d- and s-structure. Move-α moves nodes into *landing sites*, which are positions available for movement to take place to. The number of landing sites is restricted by constraints such as the θ-criterion and the Case-Filter. In the example, [CP,XP] and [C',C] function as landing sites. The XP-position [CP,XP] is a landing site only for maximal projections, whereas [C',C] is a head position and therefore only available to heads. In the s-structure shown below, move-α takes place into both landing sites: the NP *einen Mann* moves to [CP,XP] (Vorfeld), and the finite verb *hat* to [C',C] (left parenthesis):

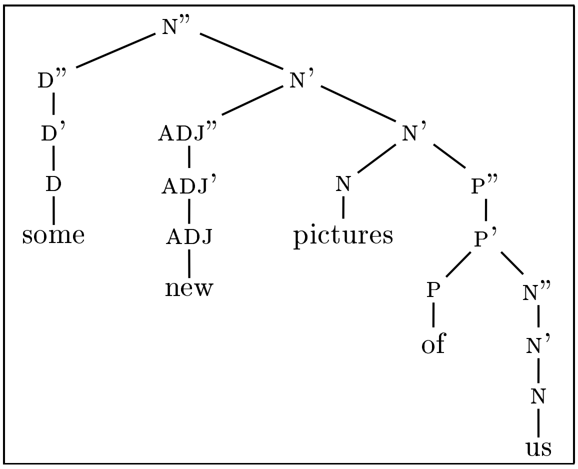


5 GB and DD: a comparison

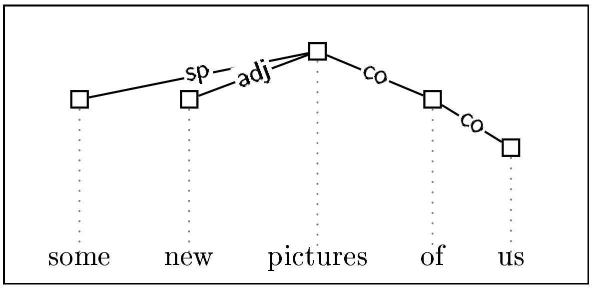
5.1 Dependency

An obvious difference between GB and DD is that GB is a phrase structure-based theory and DD dependency-based. But is no crucial difference: since GB is based on X-bar theory (Jackendoff 1977), it also incorporates the notion of a *head*: X-bar theory requires that every phrase has a head which is a single word. (Covington 1990) argues that if a phrase structure analysis (1) picks out one node as the head of each phrase and (2) has no labels or features on non-terminal nodes (unless of course copied unchanged from terminal nodes), it can be regarded as being equivalent to a dependency analysis that specifies word order.

(Covington 1992) even goes one step further by attempting to simplify GB theory by recasting it into a dependency formalism. He shows how to convert GB's X-bar-based phrase structure trees into equivalent dependency trees and then redefines government in terms of dependency. As an example, consider the GB phrase structure tree below:



The head of the phrase is *pictures*, which has a specifier (*some*), an adjunct (*new*) and a complement (*of*). The complement of *of* is *us*. Here is an equivalent dependency tree:



where *sp* stands for *specifier*, *adj* for *adjunct* and *co* for *complement*. With respect to a dependency tree, GB's notion of government is now much easier to define:

Definition 1. *A governs B iff B is an immediate dependent of A.*

By using dependency trees rather than phrase structure trees, not only principles such as government become easier to define. From a computational point of view, dependency grammar also has the advantage of postulating fewer nodes, i.e. exactly one node per word. Fewer nodes lead to improved performance because the trees to be processed are smaller.

5.2 Valency

GB and DD and in fact most linguistic theories to date share a notion of *valency*. Both GB and DD state valency requirements in the lexicon: GB uses subcategorization frames to specify the required θ-roles, and a DD lexicon includes ID tree and LP tree valency. ID tree valency is encoded by stating which syntactic roles a word *offers* and is very similar to GB's subcategorization frames. For example, a finite transitive verb offers subj and obj. On the opposite, LP tree valency specifies which fields a word offers.

5.3 Constituency

While GB includes the notion of dependency as a derived notion only, constituency is incorporated as a first class citizen. Constituents or phrases in GB are contiguous substrings of the analyzed sentence. DD includes constituency as a derived notion in both the ID and LP tree. In the ID tree, the set of nodes equal or below a head can be viewed as a constituent, but one which is not required to be contiguous (since the ID tree is unordered and non-projective). In the LP tree, the set of nodes equal or below a host forms a contiguous substring constituent (because LP trees are ordered and projective).

5.4 Movement

Both theories use a notion of *movement* to relate a deep or syntactic structure (d-structure in GB, ID tree in DD) to a surface or topological structure (s-structure, LP tree). But while in GB, move-α is a primitive operation modelled by tree transformations, climbing is a derived notion in DD: it describes the effect of well-formedness conditions. These well-formedness conditions are axiomatized in a constraint-based fashion, as outlined in (Duchier 1999) and (Duchier 2000), and can be easily turned into a parser.

GB's tree transformational account of movement severely restricts the computational usability of the theory. As (Covington 1990) argues, because transformations are tree-to-tree mappings, a parser can only undo a transformation if it has already parsed the tree structure that represents the output of the transformation. That is, the only way to parse a transformed sentence is to undo the transformation — but the only way to undo the transformation is to parse the sentence first.

GB restricts the applicability of move-α by providing a fixed set of kinds of movement, including *wh*- and NP-movement. Movement is further constrained by general principles such as the Case Filter and the θ-criterion. For instance, only $\overline{\theta}$-positions[7] may function as *landing sites* for movement

[7] A $\overline{\theta}$-position is a position which is not assigned a θ-role.

in GB. XP-positions are landing sites for maximal projections only (e.g. [CP,XP]) and head-positions for heads.

DD constraints movement in a lexicalized way. Only a small number of grammatical principles are postulated and the remaining work is done in the lexicon: a lexical entry stipulates which fields are *offered* and which are *accepted.* Making the connection to GB again, the notion of offered fields is very similar to GB's *landing sites.*

6 Conclusion

The new dependency grammar-based framework described in DD employs concepts which are strikingly similar to concepts in GB theory. Above all, both GB and DD use a notion of *movement* to mediate between levels of syntax and linear precedence. But while GB models movement as tree transformations, movement in DD is the consequence of well-formedness conditions.

We demonstrated with analyses of verb-second clauses that on a descriptive level, the notions of movement in GB and DD are yet very similar. This suggests that GB's approach to movement could be reformulated in a way similar to DD's constraint-based approach. A non-transformational account of movement based on well-formedness conditions would make GB much more attractive from a computational point of view, and could make use of techniques developed for DD, including an efficient treatment of ambiguity using finite-set constraints.

References

Bröker, N. (1998). Separating Surface Order and Syntactic Relations in a Dependency Grammar. In *COLING-ACL 98 - Proc. of the 17th Intl. Conf. on Computational Linguistics and 36th Annual Meeting of the ACL.*, Montreal/CAN.

Chomsky, N. (1986). *Barriers.* Linguistic Inquiry Monograph 13. Cambridge/MA: MIT Press.

Covington, M. A. (1990). A Dependency Parser for Variable-Word-Order Languages. Research Report AI-1990-01, Artificial Intelligence Programs, University of Georgia, Athens/GA.

Covington, M. A. (1992). GB Theory as Dependency Grammar. Research Report AI-1992-03, Artificial Intelligence Program, University of Georgia, Athens/GA.

Duchier, D. (1999). Axiomatizing Dependency Parsing Using Set Constraints. In *Sixth Meeting on the Mathematics of Language*, Orlando/FL.

Duchier, D. (2000). Configuration Of Labeled Trees Under Lexicalized Constraints And Principles. To appear.

Duchier, D. and R. Debusmann (2001). Topological Dependency Trees: A Constraint-based Account of Linear Precedence. In *39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse/FRA. To appear.

Erdmann, O. (1886). *Grundzüge der deutschen Syntax nach ihrer geschichtlichen Entwicklung dargestellt.* Stuttgart/FRG: Erste Abteilung.

Gerdes, K. and S. Kahane (2001). Word Order in German: A Formal Dependency Grammar Using a Topological Hierarchy. In *39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse/FRA. To appear.

Grewendorf, G. (1988). *Aspekte der deutschen Syntax. Eine Rektions-Bindungs-Analyse.* Studien zur deutschen Grammatik 33. Tübingen/FRG: Gunter Narr.

Herling, S. (1821). Über die Topik der deutschen Sprache.

Jackendoff, R. (1977). *$\bar{X}$ Syntax: A Study of Phrase Structure.* Number 2 in Linguistic Inquiry Monographs. Cambridge/MA: MIT Press.

Kahane, S., A. Nasr, and O. Rambow (1998). Pseudo-projectivity: a polynomially parsable non-projective dependency grammar. In *36th Annual Meeting of the Association for Computational Linguistics (ACL 1998)*, Montreal/CAN.

Kathol, A. (2000). *Linear Syntax.* Oxford University Press.

Müller, S. (1999). *Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche.* Linguistische Arbeiten 394. Tübingen/FRG: Max Niemeyer Verlag.

Reape, M. (1994). Domain Union and Word Order Variation in German. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, pp. 151–197. Stanford/CA: CSLI.

Tesnière, L. (1959). *Eléments de Syntaxe Structurale.* Paris/FRA: Klincksiek.

On Interpolation and Model Theoretic Characterization of Logics

MARTA GARCIA MATOS
University of Helsinki
garciam@mat-92.pc.helsinki.fi

ABSTRACT. We present a generalized form of Lindström's Theorem that characterizes the fragment of first order sentences in which a particular predicate occurs only positively. It has as corollaries Lyndon's Interpolation and Preservation theorems. The aim of the general project is to analyze the proof of Lindström's Theorem and try to extend it, first by relating it to interpolation theorems, and later by extending the study to non-classical logics, such as systems of modal logics.

1 Introduction.

Two famous properties of first order logic are the Compactness Theorem: *'A set of sentences Σ has a model if and only if every finite subset of Σ has a model'*, and the Downward Löwenheim-Skolem Theorem: *'If a countable set of sentences has a model of an infinite cardinality, then it has a countable model'*. Per Lindström (5) characterized first order logic as the maximal logic which satisfies these model theoretic properties. The study of extensions of first order logic is well established (1), but practically no other logic than first order logic has a model theoretic characterization. The few results in this area can be found in (3) and (10).

A close examination of the ingredients of Lindström's Theorem reveals that it is actually an application of some basic properties of back-and-forth systems. When the proof is broken into its parts, a proof of Craig Interpolation Theorem: *'If $\psi \models \phi$, there is θ such that $\psi \models \theta$ and $\theta \models \phi$, and every non-logical symbol which occurs in θ also occurs in both ψ and ϕ'* emerges. This has been known in the folklore of the subject (3), but has not been systematically exploited. Two recent works (2), (7) study the relations between back and forth systems, interpolation, and characterization theorems.

In this work we show that a generalized Lindström's Theorem can be proved for the fragment of first order logic consisting of formulas in which a particular predicate occurs positively (Theorem 4). This characterization immediately implies Lyndon Interpolation and Preservation Theorems

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

(Corollaries 10 and 11). In (2) there is a similar result: a generalized theorem (in their case an interpolation theorem) with respect to a back and forth system. The strategy for proving interpolation theorems begins therefore by finding a back and forth system for the logic. A preservation theorem related to the nature of the back and forth system is immediately obtained as a corollary. However, the existence of back and forth systems does not guarantee the success in finding interpolation theorems, as happens in the case of $L_{\infty\kappa}$ for $\kappa > \omega$.

In our project we analyze the proof of Lindström's Theorem and try to extend it, first by relating it to interpolation theorems, and later by extending the study to non-classical logics. In this paper we report the results of the first stage only. We plan to go further, pursuing the connection between model theoretic characterizations and definability results further.

When one looks at the spectrum of extensions of first order logic (1), it is noticeable that precisely the extensions that have a model theoretic characterization are the ones that also satisfy the Interpolation Theorem. This is so with first order logic, the infinitary language $L_{\omega_1\omega}$, and the admissible fragments of $L_{\omega_1\omega}$. On the other hand, a lot of effort has been put in obtaining model theoretic characterizations of extensions of first order logic by generalized quantifiers (9). These attempts have failed, and remarkably, all known such extensions also fail to satisfy Interpolation theorems of any kind.

Let S be a finite and relational vocabulary. An *S-structure* $\mathfrak{A}$ is a pair (A, σ), where A is a nonempty set and σ is a map that assigns to every n-ary relation symbol R in S, a n-ary relation on A. A *logic* is a pair $(\mathfrak{L}, \models_{\mathfrak{L}})$ where $\mathfrak{L}$ is a mapping defined on vocabularies S such that $\mathfrak{L}[S]$ is a class (the class of *$\mathfrak{L}$-sentences* of vocabulary S) , and $\models_{\mathfrak{L}}$ (the *$\mathfrak{L}$-satisfaction relation*) is a relation between $\mathfrak{L}$-sentences and structures. To define a logic we must also give its closure properties. Usually these properties are:

(*i*) If $S_0 \subseteq S_1$, then $\mathfrak{L}(S_0) \subseteq \mathfrak{L}(S_1)$.

(*ii*) If $\mathfrak{A} \models_{\mathfrak{L}} \psi$, then $\psi \in \mathfrak{L}[S_{\mathfrak{A}}]$. Where $S_{\mathfrak{A}}$ is the vocabulary of $\mathfrak{A}$.

(*iii*) *Isomorphism property.* If $\mathfrak{A} \models_{\mathfrak{L}} \psi$, and $\mathfrak{A} \cong \mathfrak{B}$, then $\mathfrak{B} \models_{\mathfrak{L}} \psi$

(*iv*) *Reduct Property.* If $\psi \in \mathfrak{L}(S)$ and $S \subseteq S_{\mathfrak{A}}$, then $\mathfrak{A} \models_{\mathfrak{L}} \psi$ iff $\mathfrak{A} \restriction_S \models_{\mathfrak{L}} \psi$.

(v) *Renaming Property.* Let $\rho : S_0 \to S_1$ be a renaming. Then, for each $\psi \in \mathfrak{L}(S)$ there is a sentence ψ^ρ, from $\mathfrak{L}[S_1]$ such that for all S_0-structures $\mathfrak{A}$, $\mathfrak{A} \models_{\mathfrak{L}}$ iff $\mathfrak{A}^\rho \models_{\mathfrak{L}} \psi^\rho$.

First order logic is a logic in the sense of this definition, if we define $\mathfrak{L}^S$ to be the set of first order sentences of vocabulary S. Lindström (5) defines an *abstract logic* as a logic with the above properties plus closure under negation and conjunction. We'll see that, for the purpose of this paper, we have to give different closure properties to define our logic.

We call a formula of vocabulary S an *S-formula.* We will use $\mathfrak{L}^S$ to denote the set first order S-formulas. If P is a predicate symbol in S, $\mathfrak{L}^{S,P}$ denotes the set of sentences in $\mathfrak{L}^S$ in which P occurs only positively. $\mathfrak{L}_r^{S,P}$ is the set of formulas in $\mathfrak{L}^S$with variables among $x_1, \ldots, x_r$ in which P occurs only positively.

2 A Lindström Theorem

Next definition introduces a relation that is an isomorphism with respect to every symbol in the language except for a particular predicate P. This is related to the particular back-and-forth system (Definition 2) that we need for proving Lindström's Theorem for the P-positive fragment of first order logic.

Definition 1. *Let $\mathfrak{A}, \mathfrak{B}$ be S-structures, and $\bar{a} \in A$ an r-tuple. A bijection $\pi : \mathfrak{A} \to \mathfrak{B}$ is a P-isomorphism if*

$$\mathfrak{A} \models P(\bar{a}) \Rightarrow \mathfrak{B} \models P(\bar{\pi a})$$

and

$$\mathfrak{A} \models R_i(\bar{a}) \iff \mathfrak{B} \models R_i(\bar{\pi a}) \text{ for } R_i \in S \setminus \{P\}.$$

Definition 2. *A P-Back-and-Forth sequence for $(\mathfrak{A}, \mathfrak{B})$ is a sequence $P_n \subseteq P_{n-1} \subseteq \ldots \subseteq P_{n-k}$ that satisfies the following conditions:*

(i) *Each P_i is a set of partial P-isomorphisms.*

(ii) $\emptyset \in P_n$

(iii) *If $p \in P_m$ and $a \in A$, then there is $b \in B$ such that $p \cup \{\langle a, b\rangle\} \in P_{m-1}$.*

(iv) *If $p \in P_m$ and $b \in B$, then there is $a \in A$ such that $p \cup \{\langle a, b\rangle\} \in P_{m-1}$.*

If $A = \{a_n : n \in \omega\}$, and $B = \{b_n : n \in \omega\}$, and there is a P-back-and-forth sequence of length ω for them, then they are P-isomorphic. For suppose we have defined, for $n \leq k$, $c_{2n} = a_n$, and $d_{2n+1} = b_n$, and a sequence $\langle c_n\rangle P_n \langle d_n\rangle$. If $k = 2r$, set $c_k = a_r$. By (iii), there is a least index s such that $\langle c_0, \ldots, c_k\rangle P_{k+1} \langle d_0, \ldots, d_{k-1}, b_s\rangle$. Set $d_k = b_s$. Similarly if k is odd. By (i), $f = \langle c_n, d_n\rangle : n \in \omega$ is a P-isomorphism on $\mathfrak{A}$ onto $\mathfrak{B}$.

We adapt the following technical notion from (5). It is introduced to avoid redundancy in the use of quantifiers or boolean connectives. This prevents formulas from being arbitrarily long, and the number of them becoming infinite.

Definition 3. *An (r, r)-condition is any atomic or negated atomic formula in $\mathfrak{L}_r^{S,P}$. A complete (r, i)-condition is any disjunction of conjunctions of (r, i)-conditions. An $(r, i-1)$-condition is $\exists x_i \chi$ or $\forall x_i \chi$, where χ is a complete (r, i)-condition.*

We will see in Lemmas 7 and 8 that any $\phi \in \mathfrak{L}_r^{S,P}$ of quantifier rank $\leq n$ is equivalent to some $(n + r, r)$-condition. For each back-and-forth system we have a different definition of the (r, i)-conditions. Lemma 9 (which is the central part of the proof of Theorem 4) shows the dependency between back-and-forth systems and formulas. In particular, in the proof of $(ii) \to (i)$ of this theorem is essential that the set of formulas is finite.

The following theorem is a generalization of Lindström's Theorem with respect to back-and-forth systems. In this particular case, the logic to be characterized being the fragment of first order logic in which a given predicate occurs only positively, the closure property needed is that of a P-isomorphism, and the back-and-forth system needed is that of Definition 2. This generalization enables us to have Lindström-like theorems characterizing other than first order logic. This theorem can be also stated in a Lindström like form: $\mathfrak{L}^{S,P}$ is the strongest logic with the compactness and Löwenheim-Skolem properties that is closed under negation of all predicates in S except P. We obtain this result by making $\mathfrak{L}^*$ in the Theorem 4 closed under negation of all predicates in S except P.

Theorem 4 (Lindström's Theorem *(Generalized Form)*). *Let S be a finite and relational vocabulary and P be a predicate symbol in S. Let $\mathfrak{L}^*$ be a logic with the Compactness and Downward Löwenheim-Skolem properties, satisfying the above properties $(i)-(v)$, except (ii), which is replaced by closure under P-isomorphisms. Let $\mathfrak{L}^*$ also be closed under $\vee, \wedge, \exists, \forall$ (but not necessarily under negation).*

Let ϕ, ψ be S-sentences in $\mathfrak{L}^*$ such that $Mod(\phi) \cap Mod(\psi) = \emptyset$. Then there is $\theta \in \mathfrak{L}^{S,P}$ such that $Mod(\psi) \cap Mod(\theta) = \emptyset$, and $Mod(\phi) \subseteq Mod(\theta)$.

Definition 5. *A formula is said to be in negation normal form (nnf) if it is built up from atomic formulas and their negations using $\vee, \wedge, \exists, \forall$.*

Every first order formula is equivalent to an nnf formula. We assume throughout this paper that first order formulas are in nnf.

Lemma 6. *Every P-positive formula is preserved by P-isomorphisms.*

Proof. By induction on the formation of P-positive formulas. The case for a P-positive atomic or negated atomic formulas comes directly form Definition 1, and the case for connectives is easy.

Now suppose $\phi = \exists x \psi(x)$, and $\mathfrak{A} \models \phi$. Then, $\mathfrak{A} \models \exists x \psi(x) \iff \exists a \in A$ s.t. $\mathfrak{A} \models \psi(a) \Rightarrow$ (by induction hypothesis) $\mathfrak{B} \models \psi(\pi a) \iff$ (since π is

a surjection) $\exists b \in B$ s.t.
$\mathfrak{B} \models \psi(b) \iff \mathfrak{B} \models \exists x \psi(x)$. The case for $\forall$ is similar. $\dashv$

Lemma 7. *Let $\Phi = \{\phi_1, \ldots, \phi_n\}$ be a finite set of S-formulas, and $\langle\Phi\rangle$ the least set that contains Φ and is closed under $\vee, \neg$. Then any formula in $\langle\Phi\rangle$ is logically equivalent to some disjunction of conjunctions of the formulas in $\{\phi_1, \ldots, \phi_n, \neg\phi_1, \ldots, \neg\phi_n\}$. In addition, there are only finitely many pairwise logically nonequivalent formulas in $\langle\Phi\rangle$.*

Proof. Suppose each ϕ_i has free variables among $x_1, \ldots, x_r$. Given a model $\mathfrak{A}$, and a tuple $\bar{a} = (a_1, \ldots, a_r) \in A$, take the conjunction $\xi_{(\mathfrak{A},\bar{a})}$ of formulas from $\{\phi_1, \ldots, \phi_n, \neg\phi_1, \ldots, \neg\phi_n\}$ such that $\mathfrak{A} \models \xi_{(\mathfrak{A},\bar{a})}(a_1, \ldots, a_r)$. It is clear that the set of satisfiable conjunctions is finite, with cardinality at most 2^n. For any $\psi \in \langle\Phi\rangle$, take the disjunction $\chi = \bigvee\{\xi_{(\mathfrak{A},\bar{a})}(a_1, \ldots, a_r) : \mathfrak{A} \models \psi(a_1, \ldots, a_r)\}$. Suppose $\mathfrak{B} \models \chi(b_1, \ldots, b_r)$. Then there is $\xi_{(\mathfrak{A},\bar{a})}$ with $(a_1, \ldots, a_r) \in A$ such that $\mathfrak{B} \models \xi_{(\mathfrak{A},\bar{a})}(b_1, \ldots, b_r)$ and $\mathfrak{A} \models \psi(a_1, \ldots, a_r)$. So, we have is $\mathfrak{A}$ such that $\mathfrak{A} \models \phi_i(a_1, \ldots, a_r) \iff \mathfrak{B} \models \phi_i(b_1, \ldots, b_r)$. Given that $\psi \in \langle\Phi\rangle$, and $\mathfrak{A} \models \psi(a_1, \ldots, a_r)$, we conclude that $\mathfrak{B} \models \psi(b_1, \ldots, b_r)$. Now suppose $\mathfrak{B} \models \psi(b_1, \ldots, b_r)$. Then $\xi_{\mathfrak{B},\bar{b}}$ belongs to the disjunction χ. Since $\mathfrak{B} \models \xi_{\mathfrak{B},\bar{b}}(b_1, \ldots, b_r)$, it follows that $\mathfrak{B} \models \chi$. $\dashv$

The representatives of each of these equivalence classes can be taken to be the (r, i)-conditions.

Lemma 8. *For any $n \in \mathbb{N}$ there are, up to logical equivalence, only finitely many P-positive formulas of quantifier rank $\leq n$.*

Proof. By induction on n.

$n = 0$: Let $\psi \in \mathfrak{L}_r^{S,P}$ be an atomic or negated atomic formula. Since S is finite and relational, the set of (r, r)-conditions is finite. The case for quantifier free formulas follows from Lemma 7 if we take Φ to be the set of (r, r)-conditions.

Induction step: Suppose we have proved that there are formulas $\{\phi_i, \ldots, \phi_h\} \in \mathfrak{L}_r^{S,P}$ of quantifier rank $\leq n$, and formulas $\{\chi_j, \ldots, \chi_k\} \in \mathfrak{L}_{r+1}^{S,P}$ of quantifier rank $\leq n$ such that every formula in $\mathfrak{L}_r^{S,P}$ (respectively in $\mathfrak{L}_{r+1}^{S,P}$) of quantifier rank $\leq n$ is logically equivalent to some ϕ_i (respectively χ_j).

Given a formula in $\psi \in \mathfrak{L}_r^{S,P}$ of quantifier rank $\leq n+1$, it can be proved by induction on ψ that it is contained in

$$\langle\Phi_1\rangle = \langle\{\phi \in \mathfrak{L}_r^{S,P} : \mathrm{qr}(\phi) \leq n\} \cup \{\exists x \phi \in \mathfrak{L}_r^{S,P} : \mathrm{qr}(\phi) \leq n\}\rangle$$

We prove: $(*)$ Any such ψ is logically equivalent to a formula in

$$\langle\Phi_2\rangle = \langle\{\phi_1, \ldots, \phi_h\} \cap \{\exists x_r \chi_1, \ldots, \exists x_r \chi_k\}\rangle.$$

But then, by Lemma 7, $\langle\Phi_2\rangle$ contains only finitely many formulas which are pairwise logically non-equivalent.

Proof of $(*)$: By the induction hypothesis, every $\psi \in \mathfrak{L}_r^{S,P}$ of quantifier rank $\leq n$ is logically equivalent to some ϕ_i. Now, if $\exists x\phi \in \mathfrak{L}_r^{S,P}$ and $\mathrm{qr}(\phi) \leq n$, then $\mathfrak{A} \models \exists x\phi(a_1, ..., a_r, x)$ iff there is $a \in A$ $\mathfrak{A} \models \phi(a_1, ..., a_r, a_{r+1})$. $\phi \in \mathfrak{L}_{r+1}^{S,P}$ and has quantifier rank $\leq n$, hence is logically equivalent to some χ_j; thus $\exists x\phi$ is equivalent to $\exists x_r\chi$. Finally, it is easy to verify that if every formula in Φ_1 is logically equivalent to a formula in Φ_2, then every formula in $\langle\Phi_1\rangle$ is logically equivalent to a formula in $\langle\Phi_2\rangle$. $\dashv$

Lemma 9. *The following two facts are equivalent:*

(i) For every $\psi \in \mathfrak{L}_{r+1}^{S,P}$ of quantifier rank $\leq n$, $\mathfrak{A} \models \psi \Rightarrow \mathfrak{B} \models \psi$.

(ii) There is a P-back-and-forth sequence $P_n, \ldots, P_0$ for $(\mathfrak{A}, \mathfrak{B})$.

Proof. We prove by induction on n the equivalence of

(i') For all $\psi \in \mathfrak{L}_{r+1}^{S,P}$ of quantifier rank $\leq n$, $(a_1, \ldots, a_r) \in A, (b_1, \ldots, b_r) \in B$, $\mathfrak{A} \models \psi(a_1, \ldots, a_r) \Rightarrow \mathfrak{B} \models \psi(b_1, \ldots, b_r)$.

(ii') There is a P-back-and-forth sequence $P_n, \ldots, P_0$ for $(\mathfrak{A}, \mathfrak{B})$ such that $\{\langle a_1, b_1\rangle, \ldots, \langle a_r, b_r\rangle\} \in P_n$.

$i' \Rightarrow ii'$: By induction hypothesis we have $P_n \subseteq P_{n-1} \subseteq \ldots \subseteq P_m$. Suppose $p \in P_m, \mathrm{dom}(p) = \{a_1, \ldots, a_r\}$, and $a \in A$. Take $\xi(\mathrm{dom}(p), a)$ to be the $(r+m, r+1)$-condition such that $\mathfrak{A} \models \xi(\mathrm{dom}(p), a)$. $\exists x\xi(\mathrm{dom}(p), x)$ has quantifier rank m. By the induction hypothesis $\mathfrak{B} \models \exists x\xi(\mathrm{rng}(p), x)$. Thus there is $b \in B$ such that $\mathfrak{B} \models \xi(\mathrm{rng}(p), b)$. Let $b = p(a)$. Thus we have $p \cup \{\langle a, b\rangle\} \in P_{m-1}$. The back condition is proved in a similar way. The induction hypothesis in this case uses $\mathfrak{B} \not\models \phi \Rightarrow \mathfrak{A} \not\models \phi$.

$ii' \Rightarrow i'$: By induction on the complexity of formulas ψ. Suppose $\psi \in L_r^{S,P}$, $\mathrm{qr}(\psi) \leq m$, $p \in P_m$, and $a_{1'}, \ldots, a_r \in \mathrm{dom}(p)$. The atomic case comes from Definition 1. The connective cases are clear. Suppose $\psi = \exists x\phi$, and $\mathfrak{A} \models \psi(a_1, \ldots, a_r)$. Then there is an element $a \in A$ such that $\mathfrak{A} \models \phi(a_1, \ldots, a_r, a)$. Using the forth condition of P_m, take the element such that there is $q \in P_{m-1}$, $q \supset p$, $a \in \mathrm{dom}(q)$, and $\mathfrak{A} \models \phi(a_1, \ldots, a_r, a)$. By induction hypothesis $\mathfrak{B} \models \phi(p(a_1), \ldots, p(a_r), q(a))$. By the back condition, there is $b \in B$ such that $q(a) = b$. Thus $\mathfrak{B} \models \phi(p(a_1), \ldots, p(a_r), b)$, and therefore $\mathfrak{B} \models \psi(p(a_1), \ldots, p(a_r))$. $\dashv$

Proof of Theorem 4. For each $m \in \omega$, construct $\theta_m \in L^{S,P}$ such that $Mod(\phi) \subseteq Mod(\theta_m)$ as

$$\theta_m = \bigvee_{\mathfrak{A} \in Mod(\phi)} \bigwedge \{\chi : \chi(m, 0) - \text{condition, and } \mathfrak{A} \models \chi\}$$

That θ_m is in $\mathfrak{L}^{S,P}$ follows from the fact that there are only finitely many $(m,0)-$conditions. (Lemma 8). Clearly, if $\mathfrak{A} \models \phi$, then $\mathfrak{A} \models \theta_m$ for all m. So $Mod(\phi) \subseteq Mod(\theta)$ for all m.

If there is an $m \in \omega$ such that $Mod(\theta_m) \cap Mod(\psi) = \emptyset$, take $\theta = \theta_m$. Suppose to the contrary that there is no such m. In this case, there is $\mathfrak{B}_m \in Mod(\psi)$ such that $\mathfrak{B}_m \models \bigwedge\{(m,0) - \text{conditions } \chi : \mathfrak{A} \models \chi\}$, for some $\mathfrak{A}_m \in Mod(\phi)$, for each $m \in \omega$. So, we have $\mathfrak{A}_m$ and $\mathfrak{B}_m$ such that $\mathfrak{A}_m \models \chi \Rightarrow \mathfrak{B}_m \models \chi$ for each m. By compactness, we get models $\mathfrak{A} \in Mod(\psi)$ and $\mathfrak{B} \in Mod(\phi)$, such that for all $(m,0)$-conditions χ, $\mathfrak{A} \models \chi \Rightarrow \mathfrak{B} \models \chi$. This gives us by Lemma 9 a P-back-and-forth sequence of length ω for $(\mathfrak{A}, \mathfrak{B})$. By Löwenheim-Skolem theorem, we can get $(\mathfrak{A}, \mathfrak{B})$ countable. By going back and forth we get a bijection $\pi : \mathfrak{A} \to \mathfrak{B}$ which is a P-isomorphism. Since $Mod(\phi)$ is closed under P-isomorphisms, $\mathfrak{A} \models \phi$ implies $\mathfrak{B} \models \phi$, but then $\mathfrak{B} \in Mod(\phi) \cap Mod(\psi)$, a contradiction. $\dashv$

Let R be any relational vocabulary and P a predicate symbol in R. Let $\psi, \phi, \theta \in \mathfrak{L}^R$. Theorem 4 has as corollaries the following known results:

Corollary 10 (Lyndon's Interpolation Theorem.). *Let ψ, ϕ be such that $\psi \models \phi$. Then there is a sentence θ such that:*

(i) $\psi \models \theta$ and $\theta \models \phi$.

(ii) θ contains only those predicate symbols that occur in both ψ and ϕ.

(iii) If ψ is P-positive, then so is θ, and if ϕ is P-negative, then so is θ.

(iv) The same as in (iii), changing 'positive' to 'negative'.

Proof. Take $\mathfrak{L}^*$ in Theorem 4 to be the logic of classes $\mathbf{K}$ such that for some $\chi \in L^{R,P}$, $\mathfrak{A} \in \mathbf{K}$ iff there is $\mathfrak{B}$ such that $\mathfrak{B} \models \chi$ and $\mathfrak{B} \upharpoonright_R = \mathfrak{A}$. Let R_1 be the vocabulary of ψ, and R_2 that of ϕ, and suppose P occurs only positively in ψ. Let $\mathbf{K}_1, \mathbf{K}_2 \in \mathfrak{L}^*$ be such that:

$$\mathfrak{A} \in \mathbf{K}_1 \text{iff there is } \mathfrak{B} \text{ such that } \mathfrak{B} \models \psi \text{ and } \mathfrak{B} \upharpoonright_{R_1} = \mathfrak{A}.$$

$$\mathfrak{A} \in \mathbf{K}_2 \text{ iff there is } \mathfrak{B} \text{ such that } \mathfrak{B} \models \neg\phi \text{ and } \mathfrak{B} \upharpoonright_{R_2} = \mathfrak{A}.$$

Then $\mathbf{K_1} \cap \mathbf{K_2} = \emptyset$. By Theorem 4, there is a $\theta \in \mathfrak{L}^{S,P}$ such that $\mathbf{K_1} \subseteq Mod(\theta)$ and $\mathbf{K_2} \cap Mod(\theta) = \emptyset$. Then $\psi \models \theta$ and $\theta \models \phi$. $\dashv$

Corollary 11 (Lyndon's Preservation Theorem.). *For any formula ϕ, the following conditions are equivalent:*

(i) Mod(ϕ) can be characterized by a set of positive formulas.

(ii) Mod(ϕ) is closed under homomorphisms.

Proof. The trivial direction is $(i) \Rightarrow (ii)$. Assume (i). Let ξ be any P-positive S-formula, $\mathfrak{A}, \mathfrak{B}$ be R-structures, $\bar{x} \in A$, and $\pi : \mathfrak{A} \to \mathfrak{B}$ a P-homomorphism and $\mathfrak{A} \models \xi$. To show (ii) it suffices to prove that $\mathfrak{B} \models \xi$. This follows from Lemma 6, by induction in the formation of formulas.

$(ii) \Rightarrow (i)$ Make $\mathcal{L}^*$ in the Theorem 4 Closed under negation of all predicates in R except P. Take $\psi = \neg\phi$. Then $\psi \equiv \theta$. $\dashv$

We believe that such generalized forms of Lindström's Theorem consitute a good general methodology for proving preservation and interpolation theorems.

3 Further Directions

For non-classical logics, such as systems of modal logic, various Interpolation Theorems are known (4), (6). Our purpose is to extend this work in the direction of modal logics and study model theoretic characterizations of these logics. There is a preliminary result in this area proved by de Rijke (8) but our goal is to prove a more conclusive result. The interesting thing about Lindström's theorem is that the Löwenheim-Skolem and Compactness theorems establish a limit in the expressive power of first order logic, in such a way that if we want this expressive power to be wider, precisely those properties are the first to be given up. The work of de Rijke did not get a real model theoretic characterization of propositional modal logic in the sense that it proposes a property of basic model logic which is lost in any extension of it, keeping therefore the characterization aspect of Lindström's theorem, but not its semantic implications.

Acknowledgements I want to acknowledge anonymous referees for helpful suggestions and advises, and Professor Jouko Väänänen for constant support, and help in making me see the meaning of all this.

References

[1] J. Barwise and S. Feferman, Model-theoretic logics, Springer-Verlag, New York, 1985.

[2] J. Barwise and J. van Benthem, Interpolation, Preservation, and Pebble Games, Journal of Symbolic Logic, 64:2, 881-903, 1999.

[3] J. Flum, Characterizing Logics, in J. Barwise and S. Feferman, Model-theoretic logics, Springer-Verlag, New York, 1985.

[4] D. Gabbay, Craig's Interpolation Theorem for Modal Logics, Conference in Mathematical Logic, 111-127 London, 1970, Lecture Notes in Math., Vol. 255, Springer, Berlin, 1972.

[5] P. Lindström, On Extensions of Elementary Logic, Theoria, 35, 1–11, 1969.

[6] L. L. Maksimova Interpolation Theorems in Modal Logics. Sufficient Conditions. (Russian) Algebra i Logika 19:2, 194–213, 250–251, 1980. English translation: Algebra and Logic 19:2, 106–120, 1981.

[7] M. Otto, An Interpolation Theorem, Bulletin of Symbolic Logic, 6:2, 447-462, 2000.

[8] M. de Rijke, A Lindström Theorem for Modal Logic, In: A. Ponse, M. de Rijke, and Y. Venema, editors, Modal Logic and Process Algebra, Lecture Notes 53, CSLI Publications, Standford, 217-230, 1995.

[9] S. Shelah and J. Väänänen, A Note on Extensions of Infinitary Logic, www.math.helsinki.fi/ logic/jvaaftp.html.

[10] J. Väänänen, Set Theoretic Definability of Logics, in J. Barwise and S. Feferman, Model-theoretic logics, Springer-Verlag, New York, 1985.

A Dynamic Approach to Referent Tracking and Pronoun Resolution

ELSI KAISER
University of Pennsylvania
ekaiser@ling.upenn.edu

ABSTRACT. This paper presents a referent-tracking and pronoun resolution system for Finnish, a free-word-order, articleless language that poses a challenge for algorithms designed for languages like English that have definite/indefinite articles. To track referents and interpret pronouns in Finnish, this algorithm uses the pragmatically-motivated word order tendencies of Finnish to create an ordered register of pegs (where each peg is associated with an entity in the discourse), ranked according to salience. Pronouns are interpreted as referring to the topmost peg in the register. The algorithm aims to extend and adapt notions from Dynamic Semantics ((Groenendijk et al. 1996)) and Centering Theory ((Grosz et al. 1995)) to a typologically different language.

1 Introduction

In this paper,[1] I present an entity-tracking and pronoun interpretation system which extends and elaborates some of the methods of Centering Theory (e.g. (Grosz et al. 1995)) and Dynamic Semantics (e.g. (Groenendijk et al. 1996)). In order to track referents and interpret pronouns in Finnish, a free word order language without articles, this algorithm uses the pragmatically-motivated word order tendencies of Finnish to create an ordered register of pegs (where each peg is associated with an entity in the discourse), ranked according to salience. In this algorithm, pronouns are interpreted as referring to the topmost (most salient) peg in the register.

The structure of this paper is as follows. First, Section 2 is a review of some previous work, and in Section 3, I present the basics of the algorithm. I discuss why languages like Finnish pose interesting problems for entity-tracking systems in Section 4, and Section 5 addresses the details of the current version of the algorithm. Section 6 presents some additional issues

[1]Thanks are due to Robin Clark for many comments and ideas. Thanks also to Ellen Prince, Misha Becker, Tonia Bleam, Na-Rae Han, Eleni Miltsakaki, Kimiko Nakanishi and audiences at Penn and Amsterdam for their feedback on earlier versions of this paper. I would also like to thank three anonymous ESSLLI reviewers for very useful comments.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

and Section 7 contains the conclusion, as well as directions for future work. The algorithm presented here is still work in progress, and is best viewed as a first step rather than the final answer.

2 A look at some previous work

The algorithm presented in this paper makes crucial use of the notion of 'pegs' ((Groenendijk et al. 1996)), which can be thought of as "addresses in memory" and are used to keep track of entities in a discourse. In this model, as in dynamic semantics, when a new (previously unmentioned) entity enters the discourse model for the first time, it is mapped onto a new peg. Pronouns refer to entities that are already linked to pegs.[2] More specifically, in this algorithm, a pronoun is interpreted as an instruction to find the entity linked to the highest-ranked featurally-compatible peg (The method of ranking the pegs is discussed below in Sections 3 and 5).

The notion that a pronoun 'points to' the topmost peg of an ordered register of pegs is based on Centering Theory ((Grosz et al. 1995))[3]. According to Centering Theory, which is a model of the local-level component of attentional state in discourse (Grosz et al. (1995:4-5)), the entities ('centers') evoked by a sentence are ordered in terms of their discourse salience The most salient entity is the one that is at the center of attention at that particular point in the discourse. The ranking can depend on a number of criteria, including syntactic, morphological and prosodic factors. It is usually assumed to be *subject* > *object* > *others* for English (Walker et al. (1998:7)). Each utterance thus has a single highest-ranked center. In addition, most utterances have a 'backward-looking center', i.e. a center that refers to an entity mentioned in the preceding utterance. According to Centering Theory, if any of the centers expressed in a sentence is a pronoun, then the backward-looking center must be a pronoun ('Pronoun Rule').

3 Introduction to the algorithm

The algorithm, in its current incarnation, is designed to track the entities mentioned in a discourse and to resolve the pronouns that refer to them. It has three main components; (1) A supply of pegs; (2) a discourse register (an array) to act as a storehouse for the pegs that are 'in use' in the discourse, and (3) a current register (an array) for pegs in the current sentence (cf. (Hintikka and Sandu 1997)). The pegs are dumped from the current

[2] As one of the reviewers noted, this is somewhat of a simplification, since there are some exceptions. For example, after talking about 'a couple', one can then use the pronoun 'she' to refer to the woman in the couple.

[3] See (Walker et al. 1998) for a more in-depth look at Centering.

register to the discourse register at the end of every utterance, and their order relative to one another is retained. The current version of the algorithm is not intended to deal with (discourse) deixis. Moreover, at the current stage, only sentences with neutral intonation are considered.

3.1 Ranked pegs

A central aspect of this algorithm is the idea that the pegs are ranked in the registers, with the topmost one being more accessible[4] than the ones lower down. When a new discourse entity is introduced, it is associated with a peg, and that peg is added to the current register. The algorithm treats subjects and objects asymmetrically to capture the fact that subjects tend to be more salient than objects (as noted for English by (Grosz et al. 1995), (Hudson-D'Zmura and Tanenhaus 1998), *inter alia*; for crosslinguistic work, see (Walker et al. 1998)). To reflect this, the peg for the subject of a sentence is located above (i.e. is more accessible than) the peg for the object. This ranking could easily be extended to include other grammatical functions beyond subject and (direct) object. However, for reasons of brevity, this paper only discusses subjects and objects.

It is worth noting that this algorithm ranks entities in terms of their grammatical functions, not their linear position in the sentence. As we will see in Section 5, it seems that, in Finnish, subjects are ranked higher (i.e. are more accessible) than objects regardless of whether the word order is SVO or OVS. Using grammatical relations instead of word order to rank referents is supported by preliminary psycholinguistic results for Finnish (Kaiser in preparation), as well as crosslinguistic research in other languages with flexible word order (see e.g. (Hoffman 1998) and (Turan 1998) on Turkish, and (Prasad and Strube 2000) on Hindi. However, there seems to be some crosslinguistic variation in what factors determine the ranking (see e.g. (Rambow 1993) and (Strube and Hahn 1996), (Strube 1998) on German).

3.2 Resolving pronouns

According to this algorithm, a pronoun is interpreted as an instruction to point to the topmost peg in the discourse register (actually, to the entity that the topmost peg is connected to) and to bring it into the current register. If the entity linked to the topmost peg is featurally incompatible with the pronoun (e.g. the pronoun is *he* but the entity is feminine), then the algorithm checks the next highest peg in the discourse register.

This algorithm is more 'global' than Centering Theory, which focuses on the local relationship between two adjacent utterances within a discourse

[4]The term 'accessible' is used here to mean 'salient,' where the most salient entity is the one at the center of attention.

segment. Within Centering Theory, it is not clear how one should deal with a pronominal 'backward-looking center' that does not have an antecedent in the immediately preceding utterance (e.g. *Peter visited his mother yesterday. She had recently moved to New York. He enjoyed his trip, especially since the weather was so nice.*). In contrast, in the current algorithm, the search for the suitable peg is not limited to the pegs for entities in the immediately preceding utterance. Along similar lines, (Walker 1998) argues in favor of a Cache Model, according to which centers can remain accessible even two to three sentences after they occur.

3.3 When to add pegs, when to look for existing pegs

The algorithm needs to ensure that the number of pegs in use is correct. It needs to know whether a given NP, say *cat*, refers to a cat that was already mentioned and has a peg in the discourse register, or whether it is introducing a new discourse entity. In English, the articles *a* and *the* provide useful clues for this task. In contrast, Finnish has neither a definite nor an indefinite article, and thus the distinction between a noun phrase that is 'pegless' and one that already has a peg cannot be made simply on the basis of morphology. Thus, it is not clear how a system, e.g. Dynamic Semantics, that aims to track the referents mentioned in a discourse would deal with Finnish. The algorithm discussed here deals with this complication by using information from word order patterns. The next section addresses the relationship between word order and information status in Finnish.

4 Basics of Finnish word order

As mentioned above, the articleless nature of Finnish poses problems for referent-tracking systems that use the definite/indefinite article distinction to distinguish new entities from previously mentioned ones.[5] However, the lack of articles does not mean that Finnish fails to mark the distinction between old and new information. Finnish word order is flexible, and can be used to make many of the distinctions that other languages accomplish by using articles (see e.g. (Chesterman 1991))[6].

Although Finnish is canonically SVO, all six permutations of these elements are grammatical in the appropriate contexts (Vilkuna (1995:245)). Different word orders convey different kinds of pragmatic information. For

[5](Kruijff-Korbayová 1997) discusses the application of File Change Semantics to Czech, another articleless language, and shows how word order marks the novelty/familiarity distinction. However, she does not propose a reference resolution algorithm. Thanks to one of the reviewers for bringing this paper to my attention.

[6]A kind of optional 'definite article' is evolving in colloquial Finnish from the demonstrative pronoun *se* 'it' ((Laury 1997)). However, this phenomenon is not a part of standard Finnish.

example, in orders with more than one preverbal argument (SOV, OSV), as well as verb-initial orders (VOS, VSO), the initial constituent is interpreted as contrastive[7] (see e.g. (Vallduví and Vilkuna 1998), (Vilkuna 1995)). When trying to ascertain the discourse status of an entity, knowing whether it is contrastive is not necessarily informative because, as (Vallduví and Vilkuna 1998) note, the notions of 'rheme' (roughly speaking, 'new' information) and 'kontrast' are distinct and should not be conflated.[8] On the other hand, SVO and OVS orders tell us about the discourse status of the subject and object – i.e. whether they are new to the discourse or have already been mentioned. Thus, this paper focuses on SVO and OVS orders.

Let us now take a closer look at the pragmatics of SVO and OVS orders. In Finnish, a preverbal subject NP is usually interpreted as being 'old' information, i.e., as referring to an entity already mentioned in the discourse (see (Prince 1992) for a discussion of 'discourse-old') (ex. (1)). If an SVO sentence occurs at the very beginning of a discourse, however, the preverbal subject can be 'new' information. Moreover, postverbal NP subjects are 'new' information (ex. (2)). When an NP object is preverbal, it is interpreted as old (ex. (2)). However, NP objects in their canonical postverbal position can be new or old (ex. (1)). These patterns are summarized in Chart 1 (see (Chesterman 1991) for further discussion).

(1) *Mies* *luki* *kirjan.* *(SVO order)*
Man-NOM read book-ACC
'A/the man read a/the book.'

(2) *Kirjan* *luki* *mies.* *(OVS order)*
Book-ACC read man-NOM
'The book, a man read.'

Chart (1)

	subject-old	subject new
object-old	SVO	OVS
object-new	SVO	SVO

In the next section we will see how the entity-tracking algorithm can take these ordering patterns into account, in order to avoid misinterpreting previously-mentioned entities as new, or new entities as already mentioned.

[7] putting aside the finer details of the various ways of definining the term 'contrastive'

[8] See (Kaiser 2000a) for further discussion concerning the functions of OSV order in Finnish and the distinction between discourse status and contrast.

5 Algorithm for Finnish

5.1 Full NP subjects and objects

First, we will consider what happens when the algorithm encounters a subject[9]. As illustrated in the simple example below, when the algorithm comes across a preverbal NP subject ('man' in ex. (3)) at the very *beginning* of a discourse, it adds a peg to the top of the current register. (Curly brackets { } are used to denote registers, 'd.r.' means discourse register, and 'c.r.' stands for current register. The rightmost elements in the registers are the most salient – i.e. 'topmost.') When the sentence ends, the peg is dumped from the current register into the discourse register. The next sentence begins with the (gender-neutral) pronoun *hän* 's/he', which the algorithm interprets as an instruction to look for the highest-ranked featurally-compatible peg. After this peg is located, it is moved to the current register, and the referent linked to this peg is interpreted as the subject of the predicate 'smiled'. At the end of the second sentence, all pegs in the current register are dumped into the discourse register.

(3)	*Mies*	*käveli*	*sisään.*	*Hän*	*hymyili.*
	Man-NOM	walked	in.	S/he-NOM	smiled.
	(discourse-initial)				

'A man walked in. He smiled.'

$\{\ \ldots\}_{d.r.}\ \{\text{man}\}_{c.r.}$	[peg from first sentence]
$\{\text{man}\}_{d.r.}\ \{\ \ldots\}_{c.r.}$	[peg dumped into d.r. at end of sentence]
$\{\ldots\}_{d.r.}\ \{\text{man}\}_{c.r.}$	[pronoun *hän* 's/he' encountered, points to top peg, which is moved to c.r.]
$\{\text{man}\}_{d.r.}\ \{\ldots\}_{c.r.}$	[peg back to d.r. after second sentence is over]

Recall now that a preverbal NP subject 'inside' a discourse usually refers to an already-mentioned entity. Thus, in such a case, the algorithm searches for the peg that already exists for this entity, and brings it to the top of the current register. If the algorithm cannot find a suitable peg, an accommodation process is presumably triggered; perhaps, as a last resort, the algorithm 'repairs' the situation by adding a new peg. An utterance requiring accommodation is presumably felt to be infelicitous.

When dealing with a postverbal NP subject (new information), the algorithm adds a new peg to the top of the current register. This is illustrated in ex. (4). This example can also be used to illustrate the algorithm's treatment of preverbal objects. When the algorithm comes across a preverbal

[9]The examples discussed in this paper involve only bare nouns or pronouns. The issues involved with more complex noun phrases such as *joku nainen* 'some woman' and *toinen nainen* 'another woman' merit further research, but cannot be addressed here for space reasons.

NP object (old information), it searches for an existing peg in the discourse register, and brings it to the second-highest slot in the current register. At the end of the first sentence, the pegs for the subject and object are thus ranked such that the subject peg is above the object peg. As mentioned earlier, the subject peg is more accessible than the object peg, regardless of the word order (SVO or OVS). In other words, the algorithmn ranks the pegs by grammatical function, not word order or information status. This arrangement predicts that a pronoun in the subsequent sentence will tend to refer to the subject 'man', and not the object 'woman.' This is supported by preliminary psycholinguistic findings for Finnish (see (Kaiser in preparation)).[10]

(4)	*...Naisen*	*näki*	*mies.*	*Hän*	*oli*	*iloinen.*
	...woman-ACC	saw	man-NOM.	S/he-NOM	was	happy.

'...The woman, a man saw. He was happy.'

$\{\ldots\}_{d.r.}$ $\{\text{woman}\}_{c.r.}$	[peg for sentence-initial object moved from d.r. to c.r.]
$\{\ldots\}_{d.r.}$ $\{\text{woman, man}\}_{c.r.}$	[peg for postverbal subject added on top of object peg in c.r.]
$\{\text{woman, man}\}_{d.r.}$ $\{\ldots\}_{c.r.}$	[pegs dumped into d.r. at end of first sentence]
$\{\text{woman} \ldots\}_{d.r.}$ $\{\text{man}\}_{c.r}$	[pronoun *hän* 's/he' encountered, peg that pronoun points to is moved to c.r.]
$\{\text{woman, man}\}_{d.r.}$ $\{\ldots\}_{c.r.}$	[pegs back to d.r. after second sentence is over]

A postverbal NP object is less straightforward than a preverbal one. In the case of a postverbal object, it is unclear whether it already has a peg or whether is new to the discourse. Thus a search procedure is used to check if a matching peg is present in the discourse register. The algorithm checks the seven highest pegs[11] to see if one of them is linked to a referent than matches the postverbal NP object. If it finds one, this peg is moved to the second-highest position in the current register. If no peg is found, a new peg is added to the top of the current register, below the subject peg. The details of the matching process are left as questions for future research.

5.2 Pronominal subjects and objects

The asymmetry in the ranking of subjects over objects also holds for pronominal arguments. When the algorithm encounters a pronominal subject, it looks for the topmost compatible peg in the discourse register and brings it to the top of the current register. However, when the algorithm comes across a pronominal object, it brings the topmost compatible peg from the

[10] As one of the reviewers points out, this does not hold for parallel constructions (e.g. *Lisa saw Anne and Mary saw her too.*). It is worth noting that these constructions are 'special' in that they are often marked by a special kind of intonation and/or use of *too/also*.

[11] The number seven was chosen because human short-term/working memory can contain approximately seven items. See e.g. (Miller 1956).

discourse register to the current register, but does not raise this peg above the peg for the subject. Example (5) illustrates this. According to informant judgments, the gender-neutral subject pronoun *hän* 's/he' (in the third sentence) shows a preference for the preceding subject even if an object pronoun is present (in the second sentence). If this object pronoun raised its peg to very top, we would expect a subsequent pronoun to refer to it.

(5) *Pekka keittää kahvia. Liisa katselee häntä*
Pekka cooks coffee-PART. Liisa looks-at s/he-PART
samalla kun vesi kiehuu. Hän tykkää
same-time as water-NOM boils. S/he likes
Pekasta/???Liisasta.
Pekka-ELA/Liisa-ELA.
'Pekka is making coffee. Liisa looks at him while the water is boiling. She likes Pekka/???He likes Liisa.'

These examples also illustrate the advantage of having a separate discourse register. The distinction between the current register and the discourse register explains why the object pronoun *häntä* 'him/her' in the second sentence cannot refer to the subject of its sentence, Liisa. Pronouns are interpreted by the algorithm as instructions to point to the topmost compatible peg in the discourse register, and in ex. (5), the peg for Liisa is still in the current register and thus it is not a possible antecedent for the pronoun (See Section 6 for a discussion of reflexives). Another advantage of having both a discourse register and a current register is the ease with which sentences with two pronouns (e.g. *Peter saw John. He kicked him*) can be interpreted (see the end of Section 5.3).

5.3 Demonstrative anaphors

Let us now tackle a sentence where both the subject and object are pronouns, such as *Peter saw John. He kicked him.* In Finnish, such sentences are somewhat marked, due to the existence of another alternative, namely the demonstrative *tämä* 'this,' which can be used as an anaphor (e.g. (Hakulinen and Karlsson 1988), (Sulkala and Karjalainen 1992)). It has often been noted that this anaphoric demonstrative tends to refer to non-subject arguments (e.g. (Saarimaa 1949)). Corpus studies support this observation ((Halmari 1994), (Kaiser 2000b)). In a two-pronoun sentence, the anaphoric demonstrative is usually used instead of a second pronoun (ex. (6)).

(6) *Pekka huomasi Matin pihalla. Hän*
Pekka-NOM noticed Matti-ACC yard-ADE. S/he-NOM
tervehti tätä.
greeted this-PART.
'Pekka noticed Matti in the yard. He$_{Pekka}$ greeted this$_{Matti}$.'

Given the tendency of *tämä* to refer to objects, one might hypothesize that, when the algorithm encounters an anaphoric demonstrative, it points to the second highest peg in the discourse register. However, (6) shows that this is insufficient, because *tämä* can be used even when only one peg is left. At the start of the second sentence in (6), the algorithm encounters *hän* 's/he' and pulls the peg for Pekka into the current register. Then the anaphoric demonstrative *tämä* 'this' is encountered – but at this point, only the peg for Matti remains in the discourse register, and it is referred to with *tämä*. One might conclude that the anaphoric demonstrative is *preferably* interpreted as an instruction to point to the second-highest peg but, if no other alternative is available, it can also be interpreted as pointing to the top peg. This approach, however, runs into trouble with ex. (7).

(7)	*Liisa*	*nukkuu*	*kotona.*	*Hän*	/	*???Tämä*	*on*	*sairas.*
	Liisa	sleeps	at-home.	S/he	/	This	is	sick.
	'Liisa is sleeping at home. She is sick.'							

Even though the peg for Liisa is the only available peg in the discourse register by the time the algorithm gets to the second sentence, *tämä* 'this' cannot be used to refer to Liisa. Instead of trying to define the anaphoric demonstrative *tämä* by the ranking of the peg it points to, maybe we should treat it as having preference to refer to objects, i.e. to be associated with referents that have a certain grammatical relation/semantic role. While this solution is not very elegant, it seems, so far, to offer the best account of the empirical data. This area would clearly benefit from future work.

On a more positive note, the algorithm works smoothly for 'double-pronoun' sentences of the English type. Consider a sentence such as *Peter saw John. He hit him*, which most people tend to interpret as meaning 'Peter hit John.'[12] When the algorithm encounters the subject pronoun 'he' in the second sentence, it pulls the peg for Peter into the current register. When it reaches the object pronoun 'him,' it points to the top peg in the discourse register which, at this point, is the peg for John. Thus, due to the presence of two registers, the algorithm resolves the pronouns successfully, without the need for any addition stipulations concerning the second pronoun.

6 Some other issues

6.1 Pronouns vs. reflexives

There are a number of possible approaches to binding-theoretic phenomena. One option is to posit that pronouns refer to entities linked to pegs in the discourse register, and reflexives to entities whose pegs are in the

[12] assuming neutral intonation.

current register ('divided registers'). Another option would be to use coreference restrictions to prevent certain NPs from referring to the same peg (see (Hintikka and Sandu 1997)). In order to account for some of the details of binding domains (*picture of NP's* etc.), we could add further partitions (embedded registers) or impose coindexation restrictions derived from syntactic binding theories. It is not clear whether a proliferation of registers is the best approach, given that this algorithm is not intended to replace Binding Theory. It seems reasonable to conclude that coindexation restrictions should play a role in the algorithm.

6.2 Main vs. Subordinate clauses

An important question that we have sidestepped so far is: What is the size of the utterance? I.e., when are pegs dumped from the current utterance register to the discourse register? This question is especially relevant when it comes to sentences that contain subordinate clauses (see e.g. (Kameyama 1998) and (Miltsakaki 1999)). The distribution of anaphoric demonstratives in Finnish texts ((Kaiser 2000b)) suggests that at least some kinds of embedded clauses should be treated as subparts of the main clause.

In addition to referring to objects (Section 5), the anaphoric demonstrative *tämä* can also be used to refer to an embedded subject - especially when a 'competitor' antecedent is present in the main clause (ex.(8)).

(8) *Matti sanoi, että Liisa on sairas. Tämä oli saanut*
Matti$_i$ said that Liisa$_j$ is sick. This$_j$ was gotten
flunssan hiihtomatkalla.
flu-ACC ski-trip-ADE
'Matti said that Liisa is sick. She had caught the flu during a skiing trip.'

If a subordinate clause behaves just like a main clause, the regular pronoun *hän* should be used to refer to the subjects of all kinds of sentences equally, regardless of whether they are matrix or subordinate. The preference to use *tämä* for referents in embedded clauses (ex.(8)) suggests that embedded clauses are best treated as subparts of the main clause. We can encode this in the algorithm by introducing a partition in the current register, such that subordinate sentences have their own separate current register alongside the current register of the main sentence. When the algorithm has processed the main clause, it dumps the current register of the main clause into the discourse register, and then after processing the subordinate clause, it dumps its current register into the discourse register below that of the main clause, as if this second current register were an object in some sense.

7 Conclusion

In this paper, I have presented a preliminary referent-tracking and pronoun resolution system for Finnish, an articleless, free-word-order language that poses a challenge for entity-tracking systems designed for languages like English or German. The algorithm described in this paper uses the word order patterns of Finnish to distinguish 'old' and 'new' referents. In addition, by mapping the entities in a discourse onto pegs that are ranked with respect to their salience, the algorithm also aims to function as a pronoun resolution system.

Many issues remain open for future research, including the interpretation of plural pronouns, the role of intonation and the treatment of more complex noun phrases. Moreover, so far, the algorithm has only been tested on very small amounts of text. In the future, it should be tested on larger corpora and also compared to/combined with existing algorithms, in order to determine how to best tackle referent tracking and resolution in a language of this type.

References

Chesterman, A. (1991). *On definiteness*. Cambridge University Press.

Groenendijk, J., M. Stokhof, and F. Veltman (1996). Coreference and Modality. In S. Lappin (Ed.), *The Handbook of Contemporary Semantic Theory*, pp. 179–213. Blackwell.

Grosz, B., A. Joshi, and S. Weinstein (1995). Centering: A Framework for Modelling the Local Coherence of Discourse. Technical report, Instute for Research in Cognitive Science, University of Pennsylvania.

Hakulinen, A. and F. Karlsson (1988). *Nykysuomen lauseoppia*. Suomalaisen Kirjallisuuden Seura.

Halmari, H. (1994). On accessibility and coreference. *Nordic Journal of Linguistics 17*, 35–59.

Hintikka, J. and G. Sandu (1997). Game-Theoretical Semantics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 361–410. Elsevier Science.

Hoffman, B. (1998). Word Order, Information Structure and Centering in Turkish. In M. Walker, A. Joshi, and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 251–272. Clarendon Press.

Hudson-D'Zmura, S. and M. Tanenhaus (1998). Assiging Antecedents to Ambiguous Pronouns: The Role of the Center of Attention as the Default Assignment. In A. J. M.A. Walker and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 199–226. Clarendon Press.

Kaiser, E. (2000a). The discourse functions and syntax of OSV word order in Finnish. In *Proceedings of the 36th Annual Meeting of the Chicago Linguistics Society.*

Kaiser, E. (2000b). Pronouns and demonstratives in Finnish: Indicators of Referent Salience. In *Proceedings of the Third International Conference on Discourse Anaphora and Anaphor Resolution (DAARC).*

Kaiser, E. (2001). Word order and pronoun interpretation in Finnish. In preparation, University of Pennsylvania.

Kameyama, M. (1998). Intrasentential Centering: A case Study. In A. J. M.A. Walker and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 98–112. Clarendon Press.

Kruijff-Korbayová, I. (1997). Czech Noun Phrases in File Change Semantics. In G. K. A. Drewery and R. Zuber (Eds.), *Proceedings of the Student Session at the 10th ESSLLI*, pp. 107–118.

Laury, R. (1997). *Demonstratives in Interaction - The Emergence of a Definite Article in Finnish.* John Benjamins.

Miller, G. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review 3*, 81–97.

Miltsakaki, E. (1999). Locating Topics in Text Processing. In *Proceedings of Computational Linguistics in the Netherlands (CLIN'99).*

Prasad, R. and M. Strube (2000). Discourse Salience and Pronoun Resolution in Hindi. *UPenn Working Papers in Linguistics 6*, 189–208.

Prince, E. (1992). The ZPG letter: subjects, definiteness, and information-status. In S. Thompson and W. Mann (Eds.), *Discourse description: diverse analyses of a fund raising text*, pp. 295–325. John Benjamins.

Rambow, O. (1993). Pragmatic Aspects of Scrambling and Topicalization in German. Paper presented at the Workshop on Naturally-Occurring Discourse, IRCS, University of Pennsylvania.

Saarimaa, E. (1949). Kielemme käytäntö. Pronominivirheistä. *Virittäjä 49*, 250–257.

Strube, M. (1998). Never Look Back: An Alternative to Centering. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the ACL.*

Strube, M. and U. Hahn (1996). Functional Centering. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics.*

Sulkala, H. and M. Karjalainen (1992). *Finnish.* Routledge.

Turan, U. D. (1998). Ranking Forward-Looking Centers in Turkish: Universal and Language-Specific Properties. In A. J. M.A. Walker and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 139–160. Clarendon Press.

Vallduví, E. and M. Vilkuna (1998). On rheme and kontrast. In P. Culicover and L. McNally (Eds.), *The Limits of Syntax Syntax and Semantics 29*, pp. 79–108. Academic Press.

Vilkuna, M. (1995). Discourse Configurationality in Finnish. In K. E. Kiss (Ed.), *Discourse Configurational Languages*, pp. 244–268. Oxford University Press.

Walker, M. (1998). Centering: Anaphora Resolution and Discourse Structure. In A. J. M.A. Walker and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 401–435. Clarendon Press.

Walker, M., A. Joshi, and E. Prince (1998). *Centering Theory in Discourse*. Clarendon Press.

Towards a comprehensive meaning of German *doch*

ELENA KARAGJOSOVA
Computerlinguistik, Universität des Saarlandes, Saarbrücken, Germany
elka@coli.uni-sb.de

ABSTRACT. In this paper, an overall abstract meaning is assumed to cover all syntactic instantiations of German *doch* (i.e., as a discourse connector, modal particle, sentence adverb and response particle). It is claimed that the overall meaning is a contrastive coherence relation. Three kinds of contrast relations are considered and it is argued that they can hold between relata that have different status in discourse: in its use as a discourse connector, *doch* can express all three different kinds of contrast relations that hold between two propositions, whereas as a sentence adverb, modal particle and response particle, it marks the contrast relation of denied expectation between the proposition expressed by the sentence containing *doch* and some implication arising from it.

1 Introduction

In the examples below, the word *doch* fulfills different syntactic functions corresponding to the different categories of speech it belongs to: discourse connective (1)-(2), sentential adverb (3), modal particle (4)-(5), and response particle (6).

(1) *Maria ist verreist, doch Peter ist da geblieben.*
Maria has left, but Peter has stayed.

(2) *Peter hatte gesagt, daß er da bleibt, doch er ist weg.*
Peter had said that he stays, but he has left.

(3) *Peter ist also doch verreist.*
Peter has left after all.

(4) (a) *A: Peter kommt auch mit.*
A: Peter is also coming.

(b) *B: Er ist doch verreist.*
B: But he has left.

(5) *Peter ist verreist. Dabei hat er doch gesagt, er bleibt.*
Peter has left. Although he said that he will stay.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

(6) (a) *A: Ist Peter nicht verreist?*
A: Hasn't Peter left?

(b) *B: Doch.*
B: He has.

It has been argued that the different functions *doch* fulfills are historically and semantically connected (Hentschel 1986), but very few attempts have been made to define an overall core meaning underlying them (Abraham 1991; Graefen 2000; König et al. 1990).

Abraham (1991) defines the meaning of the conjunction *doch* as an adversative *implicational relation* between propositions where different implications may be involved in the interpretations of the propositions. He defines the meaning of the adversative conjunction as $doch(y) =_{def} \exists x(x \rightarrow \neg y)$ or $doch(y) =_{def} \exists x \exists y \exists z(x \rightarrow z)\&(y \rightarrow \neg z)$[1], where x, y and z are propositions and $\rightarrow$ is some semantic entailment denoting probability or normality. In the case where just one proposition is involved, e.g., *Es regnet doch* (where *doch* is not accentuated, a modal particle in our account), a proposition x (and z for the second formula) has to be found such that the same two-place relation holds as for the coordinating conjunction. For *doch* as a sentence adverb[2] he suggests the scheme

$doch(x) =_{def} \{A : \exists x(x \rightarrow \neg y)\}, \{B : \neg x\}, A : x$

a variant of which is that *doch* may fail to imply the first part. This scheme he exemplifies by the dialogue *(A: One cannot go out today. It is raining.) (B: It does n o t*[3] *rain!) A: Es regnet d o c h! (It d o e s rain, too!).*

We find this approach erroneous in several aspects. First, it need not be another speaker who negates an implication from the previous utterance of the *doch*-speaker, but it can be an old belief or expectation of the same speaker, e.g., in (3). Besides, the first step in the scheme for the response particle does not seem to have anything to do with the semantics of *doch.* E.g., we can imagine the first utterance in the example dialogue to be *A: We can go out today. The sun is shining.* The dialogue can take still a similar course: *B: The sun is not shining, A: Doch.* Here, no implication of the type $x \rightarrow \neg y$ is needed for or invoked by the use of the adverb *doch.* We also believe that the formalization provided for the modal particle *doch* is inadequate since the modal particle does not denote a relation between propositions but between propositional attitudes (Karagjosova 2000).

Graefen (2000) claims that the overall (functional) meaning of *doch* is to cancel a negation and to process a contradiction. She derives this meaning from the function of *doch* as a response particle which she considers to be primary. In her approach, all other uses follow the schema typical for the

[1] The two formulas correspond to the notion of denied expectation and concession, respectively, that will be addressed further below.

[2] Abraham uses the term *discourse modal particle under contrastive accent.*

[3] The wide spacing indicates a word under contrastive accent.

response particle. This scheme consists of three steps: (i), the speaker S introduces a new (knowledge) element into the discourse; (ii), questioning: one of the participants points out an objection to it; (iii), S rejects the objection. The following example dialogue, and more closely the part $B - A - B$, is supposed to exemplify these steps: *A: Die 'BMW' ist gut. B: Die 'HONDA' ist besser. A: Nein. B: Doch. (A: The 'BMW' is good. B: The 'HONDA' is better. A: No, it isn't. B: Yes, it is.)*

In our view, it is not adequate to take the use of *doch* as a response particle to be basic and to project its meaning to other *doch*-uses. First of all, it does not seem possible to extend this scheme to cover all uses. Parts (ii) and (iii) of it can be seen to apply to the modal particle *doch* only after a slight modification, e.g., if we said that the speaker rejects (part (iii)) an implication of the hearer which is not in accord with the common ground already established between the two conversants (part (ii)). The adverbial use of *doch* covers part (iii), where the speaker rejects an earlier opposite assumption (this would be part (ii)), which however is only implicit and not contextually present. Furthermore, the scheme does not apply at all to the conjunction *doch*: although the speaker points to an objection, he does not reject it, on the contrary, he affirms it. Consider (1). Here, with *doch*, not the proposition that Peter has stayed is rejected, but the implicit expectation to the opposite: that he would also leave.

Secondly, it can be argued from a typological point of view that it would be more plausible to start from the meaning of the diachronically original use, i.e., the coordinative adversative conjunction *doch* which is synonymous with German *aber* and English *but*. And thirdly, Graefen's claim that using *doch*, the speaker blocks an expectation of the hearer, does not apply to all uses since in the case of the conjunction *doch* it is an expectation of the speaker himself that is "blocked".

So we need a wider abstraction that covers adequately all uses of *doch*. This is what is proposed in König et al. (1990), where the basic meaning of the word *doch* is specified in terms of its use as a conjunction from which the other uses are derived. The meaning of the different senses of *doch* is defined as fulfilling the function of (i) objecting to an assumption/statement of the conversational partner (response particle) or (ii) pointing at (a) a contrast between two circumstances (conjunction), (b) a denied expectation (sentence adverb), (c) a conflict between the knowledge that the hearer should have and his actual assumptions/behavior (modal particle).

The assumptions in this account are closely related to the ones underlying the present account, but they still need some elaboration and specification, e.g., how do the notions of contrast, denied expectation and conflict relate to each other, what unifies the different uses, and what entities are the "circumstances" that are related by *doch*?

The present approach, while based on assumptions about the comprehensive meaning of the form *doch* similar to those underlying the accounts described above, attempts to overcome the drawbacks of the previous approaches. Our claim is that what is common to all uses of *doch* is that a contrastive coherence relation is being expressed. In our approach, the notion of contrast is based on a taxonomy of discourse relations used in Lagerwerf (1998). We shortly introduce three types of contrastive relations–denial of expectation, semantic opposition and concession–and propose an analysis of examples with different *doch*-uses in terms of the type of contrastive relation they express. It turns out that in most of its uses, *doch* expresses the relation of denied expectation. The relation is furthermore shown to hold between discourse entities of different kind depending on the particular use: for the conjunction *doch*, it is a relation between two propositions[4] (1),(2), for the adverb a relation between a proposition (propositional content) and an implicit (not overtly stated) propositional attitude (3). In the case of the modal particle, it is a relation between two propositional attitudes one of which can be implicit (4)-(5), and for the response particle a relation between a proposition and a propositional attitude of the hearer (6).

2 Contrast as a coherence relation

In theories of discourse, coherence relations are used to account for the fact that a meaningful text is more that just a sentence concatenation. Discourse relations are usually expressed by connectives, but can also hold implicitly between the propositions expressed by the respective clauses. Examples of theory-independent coherence relations are *contrast* (usually indicated by *but, although*) and *causality* (*because*).

According to a classification proposed in Sanders et al. (1992), coherence relations can be (i) causal or additive, (ii) negative or positive, and (iii) interpreted either semantically or pragmatically[5]. Contrastive relations are negative: in all of them, a negation of a clause is involved. They may also be additive or causal, and either semantic or pragmatic. Contrastive connectives can be characterized in terms of these features, e.g., *although* is causal, *but* can have both additive and causal interpretations.

A semantic (or content) interpretation applies to contrast relations that are based on world knowledge, as in *Connors didn't use Kevlar sails, although he expected little wind*[6], where the contrastive relation is based on the world knowledge implication *Normally, if one expects little wind, one uses Kevlar sails.* Pragmatic interpretation can either be an epistemic (a conclusion of the speaker is involved) or a speech act interpretation. E.g., in *Theo was not*

[4]As a conjunction, *doch* occurs only in declarative sentences.

[5]There is a fourth feature, "representing its segments in basic order or not", which will not be considered here.

[6]All examples in this section are taken from (Lagerwerf 1998).

exhausted, although he was gasping for breath, a speculation of the speaker is involved, so the interpretation is epistemic: *Normally, if someone is gasping for breath, I conclude*[7] *that he is exhausted.* The interpretation of *Mary loves you very much, although you already know that* is based on the consideration of the speaker *If x already knows p, I need not say: p*, and therefore it is a speech act interpretation.

Lagerwerf (1998) distinguishes three basic kinds of contrastive relations– **denial of expectation**, **semantic opposition**, and **concession**–which will be shortly introduced below. All of them are typically expressed by the discourse connective *but* which corresponds to the German connectives *doch* and *aber*.

2.1 Denial of expectation

Denial of expectation is defined by Lagerwerf as a causal relation between two propositions. It can be both semantic or pragmatic and is typically expressed by the discourse connector *although*. *but* can also have a denial of expectation interpretation. This relation is classified as causal in the framework of the above taxonomy since a causal relation between two propositions, related to the connected clauses, should be available as an expectation to be denied. Consider *Although Greta Garbo was called the yardstick of beauty, she never married* and *Greta Garbo was called the yardstick of beauty but she never married.* Both lead to the expectation *if a woman is beautiful, she will marry.* This expectation is denied by the follow-up clause *she never married.*

Moreover, Lagerwerf argues that the expectation that is denied is a presupposition triggered by connectives like *although* and *but*. The presupposition has the form of a defeasible implication (denoted by $>$): *although* p, q presupposes $p' > \neg q'$, where p,q are clauses and p', q' propositions. E.g., for (2): *Normally, if Peter tells me that he stays, I conclude that he will not leave*[8]. The expectation is derived by negating one of the clauses. In cases where no sensible expectation can be derived, another interpretation has to be found.

2.2 Semantic opposition and concession

The relation of **semantic opposition** holds between two clauses with parallel structure where the contrast is induced by two incompatible predicates:

(7) Greta was single, but Prince was married.

According to Spooren (1989), semantic opposition *is about two entities in the domain of discussion.* This explains also why *Greta was single, but*

[7]since in the real world there may be also other reasons for gasping for breath.

[8]This is also an example of an epistemic interpretation of a contrastive relation.

she was married is unacceptable: the two predicates in the domain, *being single* and *being married*, cannot enter into a contrast relation since there is only one entity in the domain available for application, *Greta*: this leads to a contradiction. However, denial of expectation and concession can also be about two entities: *Although Greta was a beauty, Prince married another woman* (Ivana Kruijff-Korbayovà, p.c.). The crucial difference seems rather to be that in semantic opposition, a causal connection between the two propositions related is not available which makes it impossible to arrive at a (plausible or immediately evident) defeasible rule. As Lagerwerf points out, semantic opposition is additive and not causal. It may sometimes be implicational (e.g., *if someone has left he is not here* for (1)). However, this is not a causal connection between propositions but an antonymous relation between predicates. This is also why it is not immediately plausible for the two clauses in (7) to be connected by *although* (?*Although Greta was single, Prince was married*) without having some additional information that helps to derive an expectation, e.g., that there is an agreement between Greta and Prince that they would never marry.

Concession is always defined via a contextually determined 'tertium comparationis': a proposition for which both a positive and a negative argument are provided by the two clauses connected by the concession relation:

(8) A: Shall we take this room?
B: It has a beautiful view, but it is very expensive.

In (8), B's answer presents arguments both for and against the propositional content of A's question. Experiments have shown that the *but*-clause is given more importance by speakers (Spooren 1989), so that B's answer in (8) is negative after all.

Whereas there are cases where only concessive interpretation is possible (e.g., *The weather is nice, but I am tired*), it is not difficult to provide concessive interpretations also for cases of denial of expectation or semantic opposition if a 'tertium comparationis' can be found, e.g., (1) can be interpreted as concession if we imagine it as an answer to the question *Have your flatmates left?*.

Concession is negative, pragmatic and additive. No causal relation between the two clauses is inferred, but *a conjunction on the basis of causal inferences (namely, from clause to tertium comparationis)*, (Lagerwerf 1998, 40).

3 Analyzing the examples

In this section, we provide an analysis of the different senses of *doch* based on the taxonomy of coherence relations and attempt to integrate the basic principles of meaning unification mentioned in section **1**.

3.1 *doch* as a discourse connector

In this function, *doch* corresponds to English *but* in that it can express all three kinds of contrast already mentioned in section **2**. (1) has a semantic opposition interpretation since two contrasting predicates are applied to two different arguments. However, as pointed out in the previous section, a concession interpretation for this example can also be found provided that a tertium comparationis is contextually present (see (8)). Finally, the conjunction *doch* can also have a denial of expectation interpretation as in (2). Here, the speaker has expected that Peter has not left since he told him so: *if Peter tells me that he will not leave, then he will not leave* which has the causal implication *Peter has not left because he told me he would not do so.*

3.2 *doch* as a sentence adverb

Example (3) can be interpreted only if one imagines a second, implicit proposition standing in a contrast relation to the asserted one. The sentence can be thus paraphrased as *I had reasons*[9] *to believe that Peter would not leave, but he has left.* Only a denial of expectation interpretation is immediately plausible (a semantic opposition interpretation would require two parallel structures, a concession interpretation a tertium comparationis). The appropriate paraphrase reconstructs an implicit denial of the expectation of the speaker that Peter has not left:[10] *Although I had reasons to believe that Peter would not leave, he has left.* The underlying causal relation (the expectation that is being denied by *doch*) can then be said to be *Peter has not left, because I have reasons to believe that he has not left*, and from this, the presupposition can be derived: *Normally, if I have reasons to believe that Peter has not left, I conclude*[11] *that he has not left.*

Alternatively, *doch* could be said to trigger the simpler presupposition *Peter has not left* (Zeevat 1999). But a problem we see with this interpretation is that this presupposition, when added to the context created by the utterance, would make it inconsistent with what is asserted (that Peter has left). Even if we take it to be a pragmatic or speaker presupposition that the speaker believes that Peter has not left, it would be inconsistent with an inference based on Grice's quality maxim according to which the speaker believes what he says: that Peter has left. On our account, the

[9]The paraphrase is formulated this way since the proposition that is opposed to the asserted one can be either believed by the speaker, or by the hearer, or it can be some information provided by Peter himself or another person, or any other circumstance that speaks against the truth of the asserted proposition.

[10]In this and the following examples, the connector *although* is used for the denial of expectation paraohrases instead of the English *doch*-counterpart *but*, since *although* is typically expressing denial of expectation and as such it is partially synonymous with *but*.

[11]This is the epistemic interpretation of the contrast relation.

expected consequence is rather overridden in the beliefs of the speaker by the unexpected fact the speaker witnesses.

3.3 *doch* as a modal particle

In (4), *doch* serves to express the objection of the speaker to an (according to the speaker) incorrect implication made by the other conversant which violates the common knowledge already established between both interlocutors[12]. The sentence can be paraphrased as *We both know that Peter has left, but you assert that he will come along.* The contrast expressed here cannot be interpreted as semantic opposition: although there is an opposition (in the broader sense) between two predicates (coming along and being absent), they apply to the same entity (Peter) which leads to a contradiction (**Peter is coming, but he has left*). Nor can a concessive interpretation be established (due to the lack of a tertium comparationis). Thus, as a modal particle, *doch* can be interpreted only as a case of denial of expectation: the expectation of the speaker that his conversational partner will behave in accord with the common knowledge between them is not met. Thus, the expectation denied here is not associated barely with the fact that Peter has left as in (3), but with the common knowledge between the interlocutors and their conversational behaviour. This can be paraphrased as: *Although we both know that Peter has left, you assert that he will come along.* The underlying expectation that is being denied by *doch* can be then said to be *Normally, if we both know that Peter has left, you do not assert that he will come along.*

(5) is very similar to (2). However, the contrast relation holds not only between the fact that Peter has left and what he previously said, but also between the old belief which the speaker shared with the hearer (that Peter will stay), and the new belief of the speaker (that he has left): *Although we both believed that Peter will stay, he has left* which is based on the causal relation *Since we both believe that he will stay, he will not leave* from which the presupposition can be derived: *Normally, if we both believe that he will stay, I conclude that he has not left.*

3.4 *doch* as a response particle

The response particle *doch* serves to negate a negative question or a negative statement. In these cases, it can be paraphrased as *on the contrary*:

(9)	*A:*	*Peter*	*ist*	*nicht*	*verreist.*
	A:	Peter	has	not	left.
	B:	*Doch,*	*(er*	*ist*	*verreist).*
	B:	On	the	contrary,	he has.

[12] *doch* as modal particle triggers additionally the attitude of the speaker that the proposition in its scope is common knowledge between the conversants, see Karagjosova (2000).

But there are also cases where it can serve to confirm a positive statement[13] where this paraphrase does not apply:

(10)	(a)	*A:*	*Das*	*war*	*sehr*	*freundlich*	*von*	*ihm.*
		A:	This	was	very	friendly	of	him.
	(b)	*B:*	*Doch,*	*das*	*muss*	*man*	*schon*	*sagen.*
		B:	Yes,	admittedly.				

Thus, another paraphrase is needed. Consider (6). As a response to *A*'s utterance, *doch* is intuitively synonymous with the clause *this is not true*. Thus, (6b) means in the context of (6a) *That Peter has not left is not true*. As an answer to (10a), however, *doch* is synonymous with *this is true*. This gives the impression that the responce particle means two diametrically opposed things. On the other hand, it seems that, since *doch* after a negative statement is equal to *not p is not true*, one would expect for *doch* after a positive statement to mean *p is not true*. But this is not the case as shown in (12) on p. 140 where *doch* cannot be used to negate a positive question.

The key to that problem lies in our view in explicating *doch*'s meaning in relational terms, as a marker of contrast, rather than in treating it as a simple negation or confirmation. E.g., in (6), its meaning can be paraphrased as *You believe that Peter has not left, but he has*. Here, of the three kinds of contrast, only the denial of expectation interpretation seems plausible (no parallel structure, no tertium comparationis). The response particle rather serves to deny the expectation of conversant *A* that the proposition in (6a) is true. The denial of expectation relation can be explicated by the paraphrase *Although you believe that Peter has not left, it is not true that Peter has not left*. The underlying defeasible implications for (6) will be then *Normally, if you believe something, it is true*[14].

Example (10) can be also explained[15] in conformity with the interpretation provided for (6) using the familiar denial of expectation technique: it can be paraphrased as *Although you do not (seem to) believe that I believe that this was friendly of him, I do*[16]. The underlying defeasible implication for the denied expectation will be then *Normally, if you do not believe that I believe what you convey, you conclude that I do not believe it*[17].

In the above examples, *doch* serves to deny not only a proposition (as in (9)), but also inferences from utterances (as in (6) and (10)). This indicates

[13]This example is taken from (Helbig 1988).

[14]This implication is problematic as an axiom of epistemic logic, but as a defeasible rule it seems to be passable.

[15]Otherwise we would be forced to permit a word to mean two so diametrically opposed things as a negation and a confirmation.

[16]This paraphrase is based on the implicature that can be derived from assertions provided the maxim of quantity is observed: that the speaker believes the content of his utterance to be new to his addressee.

[17]This (a bit baroque) paraphrase is also not tenable in epistemic logic, but can be accepted as a defeasible rule.

that it is not a simple negation synonymous with *on the contrary*, but a denial of an expectation to the opposite of what the respective utterance expresses.

As already mentioned, *doch* cannot be used to negate a positive question:

(11)	*A:*	*Willst*	*Du*	*keinen*	*Zucker*	*in*	*den*	*Tee?*	*B:*	*Doch.*	
	A:	Don't	you	want	sugar	in	your	tea?	B:	Yes,	I do.

(12)	*A:*	*Willst*	*Du*	*Zucker*	*in*	*den*	*Tee?*	*B:*	*Doch*	*(≠*	*Nein).*
	A:	Do you	want	sugar	in	your	tea?	B:	Doch	(≠	No).

Here, it could be argued that, against the background of what has been stated so far, it is not clear why in (12), *doch* cannot serve to refute the expectation of A (that B wants sugar), whereas in (11) it does serve to refute $A's$ expectation. On a second thought, however, *doch* does refute both the expectation in (11) and the one in (12). They are simply not the same expectation: *Although you seem to believe that I do not want sugar in my tea, I do* is the one in (11) and *Although you do not seem to know that*[18] *I want sugar in my tea, I do* the one in (12). From a logical point of view, there should not be any difference in representing both questions[19]. Pragmatically, however, there seems to be a difference in terms of the implicatures that the hearer of the two questions can draw from them: if A didn't know that B wanted sugar, he would utter rather a question like (12), but if he had the suspicion that B does not want sugar in his tea, he whould ask (11).

4 Conclusion

In this paper, an attempt was made to offer an explanation about what unifies the different syntactic instantiations of the word *doch*. It was claimed that it expresses a contrastive coherence relation between various discourse entities. Except for its use as a conjunction, where it can express three different kinds of contrast between two propositions (denial of expectation, semantic opposition and concession), *doch* in its other uses marks only the relation of denied expectation which holds between a proposition and some implication arising from it[20].

[18]If someone asks a positive yes/no question, he does not know **whether** the propositional content of the question is true, but from the point of view of the *doch*-speaker it is true **that** he wants sugar in his tea (in *"Although you do not know that p, q"*, p is presupposed), and the *doch*-paraphrase reflects the speaker's point of view. Alternative paraphrase and rule: *Although you do not know whether p, it is true*; *Normally, if you do not know whether something is true, you do not conclude that it is true.*

[19]Both negative and positive questions imply that the speaker knows neither p nor $\neg p$.

[20]I would like to thank Ivana Kruijff-Korbayovà and four anonymous reviewers for their comments on earlier drafts of this paper.

References

Abraham, W. (1991). Discourse particles in German: How does their illocutive force come about? In W. Abraham (Ed.), *Discourse particles in German*, pp. 203–252. Amsterdam/Philadelphia: John Benjamins.

Graefen, G. (2000). Ein Beitrag zur Partikelanalyse - Beispiel: *doch*. *Linguistik online 6*(2).

Grice, H. (1975). Logic and Conversation. In P. Cole and J. L. Morgan (Eds.), *Syntax and Semantics*, Volume III. Orlando: Academic Press.

Helbig, G. (1988). *Lexikon deutscher Partikeln*. Leipzig: Verlag Enzyklopädie.

Hentschel, E. (1986). *Funktion und Geschichte deutscher Partikeln: ja, doch, halt und eben*. Reihe germanistische Linguistik; 63. Tübingen: Niemeyer.

Karagjosova, E. (2000). A unified approach to the meaning and function of modal particles in dialogue. In C. Pilière (Ed.), *Proceedings of the ESSLLI 2000 Student Session*, University of Birmingham.

König, E., D. Stark, and S. Requardt (1990). *Adverbien und Partikeln. Ein deutsch-englisches Wörterbuch*. Heidelberg: Julius Groos.

Lagerwerf, L. (1998). *Causal Connectives have Presuppositions*. Ph. D. thesis, Tilburg.

Sanders, T. J. M., W. Spooren, and L. Noordman (1992). Toward a taxonomy of coherence relations. *Discourse Processes 15*(1), 1–35.

Spooren, W. P. M. S. (1989). *Some Aspects of the Form and Interpretation of Global Contrastive Coherence Relations*. Ph. D. thesis, University of Nijmegen.

Zeevat, H. (1999). Explaining presupposition triggers. Manuscript.

Dialogue Act Modelling Using Bayesian Networks

Simon Keizer
Parlevink Language Engineering Group
University of Twente
The Netherlands
skeizer@cs.utwente.nl

Abstract. A probabilistic approach to interpretation of natural language utterances in terms of dialogue acts is proposed. It is illustrated how using Bayesian Networks, partial information obtained from an NLP component can be combined with knowledge the agent has about the state of the dialogue and about the user, in order to find the most probable dialogue act made.

1 Introduction

In this paper, we are dealing with a conversational agent (the 'SERVER'), which participates in a dialogue with another agent (the 'CLIENT'). This conversational agent perceives utterances of the client and tries to react to these utterances in an appropriate way. In Figure 13.1, a possible architecture for the conversational agent is given.

The dialogue manager coordinates the various steps involved in the interpretation of an incoming user utterance and the planning of what action to take next. First of all, an incoming utterance is submitted to the components of speech recognition (in case of spoken dialogue), and syntactic/semantic analysis. Next, the pragmatic aspect of identifying what *communicative action* was performed by the user in uttering the sentence is dealt with. This part is what we will be concentrating on in this paper. Finally, the resulting interpretation will have to lead to a decision on what actions to take, in particular, what communicative action to be addressed to the user.

In identifying the communicative action performed in an utterance, the server has to deal with uncertainty. This uncertainty arises because of the incompleteness of the information that is provided by the speech recognition and syntactic/semantic analysis components. In general, such components cannot provide all linguistic information that is contained in an utterance,

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

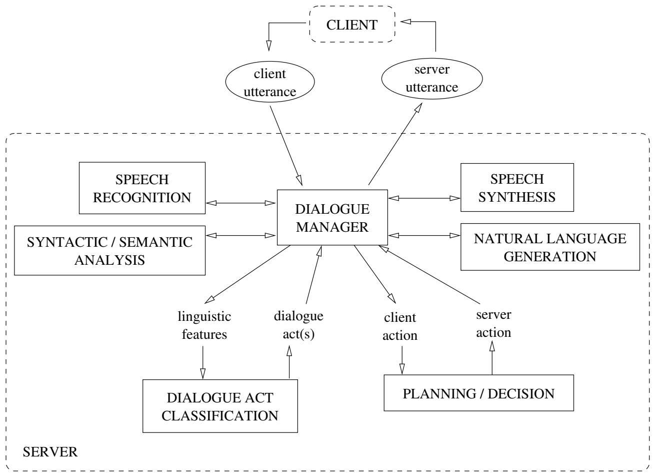


Figure 13.1: An architecture for a conversational agent

especially because we are dealing with human speakers that may produce utterances that are partly unrecognised through bad pronunciation, sentences that are ungrammatical, or that contain unknown words. This is especially the case in mixed-initiative dialogues, where a client may tend to use more complex utterances.

In order to deal with the uncertainty, the server will have to make *educated guesses* in the interpretation of the client's utterances. Therefore, we will take a probabilistic approach to dialogue modelling, in the form of Bayesian networks.

In Section 2, we will describe our approach to identifying communicative actions in terms of *dialogue acts*. In Section 3, we will introduce the notion of Bayesian networks and how they can be used in *dialogue act classification*. In Section 4, we discuss some related work and finally, in Section 5 some conclusions are drawn and an indication of further research is given.

2 Dialogue Act Modelling

The notion of dialogue acts is originated in the work of Austin and Searle (Austin 1962; Searle 1969). They observed that utterances are not merely sentences which can be either true or false, but should be seen as (communicative) actions. Searle introduced the theory of *speech acts*, which he defined as the whole act of uttering a sentence. He gave a categorisation of types of speech acts, including e.g. REQUEST, ASSERTION, and ADVICE.

When we speak of *dialogue acts* however, we emphasise the importance of looking beyond the boundaries of an utterance itself when analysing that utterance (see also (Traum 1999)). Besides linguistic information of the utterance in isolation (prosodic features, syntactic/semantic features, surface patterns, keywords, etc.), the meaning of dialogue utterances may also be determined by information concerning two other aspects:

1. the state of the dialogue: e.g. the current topic or the communicative act(s) performed in the previous utterance(s);
2. the state of the hearer's model of the speaker (e.g. the state of the model that our server has of the the client, the so-called *user model*).

Based on these three aspects, a categorisation of different dialogue act types can be made. The dialogue act hierarchy we use, is based on an existing dialogue act hierarchy underlying an annotation scheme, called DAMSL (Allen and Core 1997). This scheme has been developed as a standard for annotating task-oriented dialogues. We will not go into the details of this hierarchy here, but just mention some of the dialogue act types that are relevant in the remainder of this paper:

- `q_ref`: the speaker requests the hearer for information in the form of references satisfying some specification also given by the speaker (e.g. a list of theatre performances scheduled on a specified date: "what operas are on next week?").
- `q_if`: the speaker asks the hearer if something is the case or not ("do you want to make reservations?").
- `req`: the speaker requests the hearer for a non-communicative action ("two tickets please").
- `neg_ans`: often in response to a `q_ref` previously performed by the hearer, the speaker indicates that no references satisfy the given specification ("there are no operas scheduled for next week").

Because of the need to be able to reason under uncertainty as explained in Section 1, we propose the use of probability theory in modelling the various relationships involved in interpreting dialogue utterances. Our model will consist of three interrelated components:

1. the Belief State of the server: this state is determined by beliefs concerning:
   (a) the course of the dialogue, and
   (b) the beliefs, desires and intentions of the client.
2. the Dialogue Act(s) performed in the client's utterance;

RV	meaning	component
PSN PCQ	the previous dialogue act of S was a `neg_ans`. the previous dialogue act of C was a `q_ref`.	Belief State
CQ CR	C performed a `q_ref` in the current utterance. C performed a `req` in the current utterance.	Dialogue Acts
CONT QM	the current utterance shows a continuation pattern, e.g. in Dutch, if it starts with the word "en". the current utterance contains a question mark.	Linguistic Features

Table 13.1: The RVs of the example network of Figure 13.2.

3. the relevant Linguistic Features that the client's utterance may contain.

Using this model, the server can calculate what most probably must have been the dialogue act performed in an utterance of the client, given new information w.r.t. linguistic features of that utterance, obtained from the speech recognition and syntactic/semantic components. We will now describe how this can be done, using probabilistic inference in a Bayesian network.

3 Bayesian Dialogue Act Classification

The model introduced in Section 2 is described by a set of discrete *Random Variables* (RVs), i.e. variables that describe events with different possible outcomes. In Table 13.1 some two-valued, Boolean RVs in this set are given with their meaning and which of the components, as indicated in Section 2, they are associated with. It should be noted that it is not a requirement that the all RVs are Boolean; we could also have a RV that represents the previous dialogue act of the server S, its values ranging over all possible dialogue act types.

To illustrate the influences that exist between the RVs, consider the following dialogue passage, where we are interested in the dialogue act performed in utterance (3):

(1) C: Wat gebeurt er komend weekend 19 maart in de schouwburg? (What is happening in the theatre next weekend March 19?)

(2) S: Op deze datum is er geen uitvoering. (On this date no performances have been scheduled.)

(3) C: En op 18 maart? (What about March 18?)

Our judgement that a `q_ref` was performed in (3) may be determined by the observation that C previously performed a `q_ref` in (1), and that S's previous dialogue act in (2) was a `neg_ans`. In this context, C has decided to continue his previous dialogue act, but with a different specification accompanying that `q_ref`. Note that all this cannot be concluded by just analysing the linguistic features of utterance (3) only.

A probabilistic model is completely specified by a *joint probability distribution* (jpd), which assigns a number between 0 and 1 to every instantiation of the RVs. However, the number of probabilities to be assessed in order to specify the jpd increases exponentially with the number of variables. This problem can be overcome by identifying *conditional independencies* between RVs, reducing the number of probabilities needed for specifying the jpd. For example, we may indicate that if we know the value of CQ and CR, then learning the value of PCQ gives us no information on QM: CQ and QM are conditionally independent, given CQ and CR.

These conditional independencies can be specified by means of a *Bayesian Network*. A Bayesian network (Pearl 1988) is a DAG (Directed Acyclic Graph) in which the nodes represent RVs and the arcs reflect the informational dependencies between these variables. In Figure 13.2, a Bayesian Network is depicted containing the RVs given above. It reflects a number of conditional independencies, including the one indicated above.

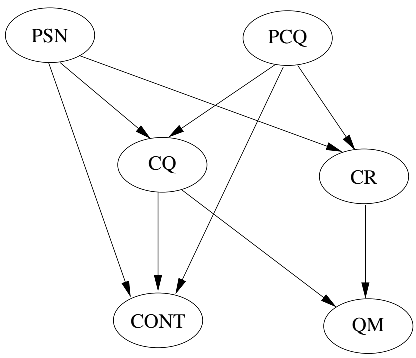


Figure 13.2: Simple Bayesian Network for utterance interpretation.

Associated with each RV is a *conditional probability distribution* (cpd) given its parents in the network. This means that for CQ a cpd is specified, given its parents PSN and PCQ. In the cpd's we should have numbers which reflect the qualitative relationships between the RVs involved. In Table 13.2, some of the chosen distributions are given. From the distribution of CQ, one can see that if *PSN=true* and *PCQ=true*, then CQ is more probably true (0.7) than false (0.3). For this example, the numbers have been assessed by using 'expert' knowledge; for more accurate and realistic models, statistical analysis on data using an annotated dialogue corpus is needed for the assessment (see for example (Heckerman 1995)).

PSN	
true	0.75
false	0.25

PCQ	
true	0.8
false	0.2

CQ				
PSN	true		false	
PCQ	true	false	true	false
true	0.7	0.2	0.2	0.5
false	0.3	0.8	0.8	0.5

Table 13.2: Some of the probability distributions specifying the Bayesian network of Figure 13.2.

The RVs concerning the Belief State, *PSN* and *PCQ*, have no parents, so their cpd's are not conditional, but 'prior' distributions (also given in Table 13.2). Although we have specified all prior distributions in the network as fixed, the distributions of RVs like these actually depend on the Belief State at the time-step in which the previous utterance was processed, and is therefore subject to updating. However, in order to keep our story within limits, we will not go into this dynamic extension of our model, but have based the prior distributions intuitively on the course of the dialogue passage given before. Here, the previous act by S, performed in utterance (2), was probably a `neg_ans` (probability 0.75) and the previous act of C, performed in utterance (1) a `q_ref` (probability 0.8).

This network can now be used for *probabilistic inferences*: we can determine the probability that e.g. a `q_ref` was performed, given the *partial* information that e.g. C's utterance contained a question mark, like in (3) of the dialogue passage given before. This process of determining the *posterior probability distribution* is called *belief updating*. In this process, the formula for the joint probability distribution (jpd) over all RVs in the network plays a central role. By making use of the conditional independencies implicitly given by the network structure, the jpd is given by the product of the specified cpd's:

$$\begin{aligned} P(PSN, PCQ, CQ, CR, CONT, QM) &= P(PSN) \cdot P(PCQ) \cdot \\ \cdot P(CQ|PSN, PCQ) &\cdot P(CR|PSN, PCQ) \cdot \\ \cdot P(CONT|PSN, PCQ, CQ) &\cdot P(QM|CQ, CR) \end{aligned} \tag{13.1}$$

Suppose the server gets to know that there was a question mark in the utterance. He will be interested in the updated probability that the dialogue act performed is a `q_ref`, given this new information. This probability can be calculated as follows:

$$P(CQ = true|QM = true) = \frac{P(CQ = true, QM = true)}{P(QM = true)} \tag{13.2}$$

Both numerator and denominator can be obtained from the jpd (13.1) by summing over all possible configurations of the other RVs in the network. Let $S = s$ and $T = t$ denote instantiations of the RVs in $\{PSN$, PCQ, CR, $CONT\}$ and $\{PSN$, PCQ, CR, $CONT$, $CQ\}$ respectively. Then we get our posterior probability from 13.3 and 13.4.

$$P(CQ = true, QM = true) = \sum_s P(CQ = true, QM = true, S = s) \tag{13.3}$$

$$P(QM = true) = \sum_t P(QM = true, T = t) \tag{13.4}$$

We will now show some results of probabilistic inferences in the network. For three different cases of particular information (which will be called the 'evidence' $\mathcal{E}$), we have calculated the posterior probability distribution of CQ and CR, given that information. This has been done with two different choices for the prior distributions of PSN and PCQ. From the results in Table 13.3, one can observe that with the original priors (upper row of the table), the probability that `q_ref` was performed changes as the available information varies. Especially the case where S gets to know that the utterance shows a continuation pattern (starting with the word "en"), clearly reflects the correctness of classifying the utterance as a `q_ref` (with probability 0.804) and not `req` (with probability 0.313).

In the case of 'uniform' distributions for the priors (lower row of the table), i.e. the probabilities of both values $true$ and $false$ are 0.5 for both PSN and PCQ, one can observe that there is much more indifference between CQ and CR than before. This illustrates how the role of beliefs concerning the course of the dialogue (as part of the Belief State) in dialogue act classification can be taken into account.

4 Related Work

Other work on dialogue modelling which is based on the notion of dialogue acts includes e.g. Dynamic Interpretation Theory (DIT) and the Information State model (Poesio and Traum 1998). In DIT (Bunt 1995), dialogue acts are defined as functional units used by the speaker to change the context. They consist of a *semantic content* and a *communicative function*, so a

priors	$\mathcal{E}$	$P(CQ\|\mathcal{E})$	$P(CR\|\mathcal{E})$
$P(PSN) = 0.75$	(none)	0.515	0.298
and	(QM=true)	0.562	0.380
$P(PCQ) = 0.8$	(QM=true, CONT=true)	0.804	0.313
$P(PSN) = 0.5$	(none)	0.400	0.413
and	(QM=true)	0.521	0.537
$P(PCQ) = 0.5$	(QM=true, CONT=true)	0.624	0.485

Table 13.3: Results of Probabilistic Inferences.

dialogue act changes the context in a way that is given by the communicative function, using the semantic content as a parameter. According to the Information State model, both of the dialogue participants keep track of the *Conversational Information State* (CIS), in which *grounded conversational acts* are recorded and also *ungrounded contributions.* A CIS is characterised by a feature structure, containing embedded feature structures for both dialogue participants.

Concerning the classification of communicative actions, various research has been done. In plan-based approaches (Perrault and Allen 1980), the communicative acts (speech acts) are predicted on the basis of recognition of plans that the speaker has. Therefore, the interpretation of utterances is extended from identifying direct speech acts from linguistic features, to indirect speech acts, taking into account the course of the dialogue in terms of a speaker's plans. This rule-based approach may lead to difficulties when dealing with uncertainty.

In other approaches, statistical methods are used to model dialogue, see for example (Nagata and Morimoto 1994; Mast et al. 1996; Stolcke et al 2000). In (Stolcke et al 2000), a probabilistic model obtained from statistical analyses of a dialogue corpus is presented. Dialogues are modelled in a Hidden Markov Model, with states corresponding to dialogue acts and observations corresponding to utterances (in terms of word sequences, acoustic evidence and prosodic features). The transition probabilities are obtained from n-gram analysis of dialogue acts and the observation probabilities are given by local utterance-based likelihoods.

Also the use of Bayesian networks for interpreting utterances in a dialogue has been proposed before. In (Pulman 1996), Stephen Pulman proposes a framework for classifying communicative actions, very similar to our approach, in which a framework of conversational games and moves is used, in stead of dialogue acts. A more interesting difference with our approach however, is in the structure of the Bayesian network used. While we have chosen for arcs from nodes representing dialogue acts to nodes representing linguistic features (like the arc from CQ to QM), Pulman has chosen arc in the opposite direction. One could say that in our approach a model of the speaker's behaviour is given, which is used to derive the most probable dialogue act he performed. Pulman's network however, models the hearer, using various sources of information as 'inputs' to derive the most probable conversational move made by the speaker. This difference is an interesting topic for further research.

Other work on the use of Bayesian networks in dialogue systems includes research, where the emphasis is on the *user modelling* part within a specific (task-)domain (Akiba and Tanaka 1994), in stead of the aspect of dealing with partial linguistic information in understanding communicative behaviour.

5 Conclusion

In this paper we have shown a probabilistic approach to interpreting natural language utterances in a dialogue. We have described how Bayesian networks can be used to interpret partial information from natural language analysis in terms of dialogue acts. Not only the linguistic information from utterances is taken into account, but also knowledge about the course of the dialogue and about the mental state of the speaker, the dialogue participant that performed the utterance. Using Bayesian networks, these various sources of information can be integrated into one probabilistic model.

Current research questions are related to the construction of a Bayesian network for dialogue act classification. This means that we have to find a set of random variables representing all relevant aspects of identifying a dialogue act type, identify a sufficient number of conditional independencies among these variables (i.e. finding the network structure), and assess the conditional probability distributions associated with the network. One way of doing that is using an annotated dialogue corpus to train a Bayesian network from data. Ongoing work includes the annotation of a corpus of dialogues, which was obtained from Wizard of Oz experiments. The corpus contains mixed-initiative dialogues between a human user typing utterances and a system also producing textual utterances. The dialogues concern the theatre domain, in which the user can get information about performances and ticket reservations can be made if required. The annotation currently concerns dialogue acts (from a dialogue act hierarchy based on the DAMSL scheme, as mentioned in Section 2) and superficial linguistic features like sentence type and punctuation.

References

Akiba, T. and H. Tanaka (1994). A Bayesian approach for user modeling in dialogue systems. Technical Report TR94-0018, Tokyo Institute of Technology.

Allen, J. and M. Core (1997, October). Draft of DAMSL: Dialog Act Markup in Several Layers. Dagstuhl Workshop.

Austin, J. L. (1962). *How to Do Things with Words.* Harvard University Press.

Bunt, H. C. (1995). Dynamic Interpretation and Dialogue Theory. In M. Taylor, D. G. Bouwhuis, and F. Neel (Eds.), *The Structure of Multimodal Dialogue*, Volume 2. A'dam: John Benjamins.

Heckerman, D. (1995). A Tutorial on Learning With Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research.

Mast, M., H. Niemann, E. Noth, E. Guenter, and E. Schukat-Talamazzini (1996). Automatic Classification of Dialog Acts with Semantic Classification Trees and Polygrams. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, Volume 1040 of *Lecture Notes in Artificial Intelligence*, pp. 217–229. Springer-Verlag.

Nagata, M. and T. Morimoto (1994). First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication 15*, 193–203.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Perrault, C. R. and J. Allen (1980). A Plan-based Analysis of Indirect Speech Acts. *American Journal of Computational Linguistics 6*(3-4), 167–182.

Poesio, M. and D. Traum (1998). Towards an Axiomatization of Dialogue Acts. In J. Hulstijn and A. Nijholt (Eds.), *TwenDial'98: Formal Semantics and Pragmatics of Dialogue*, Number 13 in TWLT.

Pulman, S. G. (1996). Conversational Games, Belief Revision and Bayesian Networks. In K. v. D. Jan Landsbergen, Jan Odijk and G. V. van Zanten (Eds.), *Computational Linguistics in the Netherlands.* SRI Technical Report CRC-071.

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press.

Stolcke et al, A. (2000). Dialogue Act Modelling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics 26*(3), 339–374.

Traum, D. R. (1999). Speech Acts for Dialogue Agents. In M. Wooldridge and A. Rao (Eds.), *Foundations of Rational Agency*, pp. 169–201. Kluwer.

On the Semantics of Concurrent Programming Languages: An Automata-Theoretic Approach

ROMAN V. KONCHAKOV
Faculty of Computational Mathematics and Cybernetics,
Moscow State University
roman@cs.msu.su

ABSTRACT. In this paper we present an approach to developing formal semantics of programming languages with concurrency and real time. The main idea is to combine Denotational Semantics and the concept of Timed Automata. We adopt general timed automata and define some operations on them. These operations make it possible to define a compositional semantics of MM language. This approach can be applied to programming languages based on message passing (as MM language) and shared variables as well. It also provides promising opportunities of well-known verification techniques application.

Introduction

A problem of developing reliable software gets more and more actual since software systems capture wider application areas whereas software systems in their own turn become more complicated. In particular, this problem is very important for distributed and embedded systems that control traffic, communications, industrial processes, etc. Distributed systems are the systems comprised of several communicating processes executed in parallel. Due to concurrency these systems as a rule are highly nondeterministic. Modeling, testing and other similar techniques for program correctness checking are not well-suited for distributed systems, since these techniques can handle only a small part of all possible executions.

Formal methods provide an alternative approach to the problem. The key idea of formal methods is as follows. One builds a formal model of the program under consideration and then proves that the model meets the requirements (program specification). The benefits of formal methods are well-known: checking procedure can be done automatically and it always gives mathematically correct answer to the question. Usually formal methods are applied to models of computation that are abstractions of real

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

computer programs. Therefore one has to translate an analyzed program into a formal model to achieve benefits of formal verification techniques. When turning from a real computer program to its abstract formal model one should also guarantee the correctness of the translation. It is possible only in the case when a programming language is provided with a formal semantics that is suitable for application of formal methods.

In this paper we present an approach to developing formal semantics of programming languages with real time and concurrency. The key idea of the approach is to handle concurrency by means of timed automata. As an example of a programming language we choose MM (Model with Messages) language of DYANA environment (Bakhmurov et al. 1999).

MM language is intended for design and development of real-time distributed systems. We will show that both key features of MM programs – real time and true concurrency – are captured by timed automata. A run of an MM program operating in real time may be thought of as a sequence of observable events marked with timestamps. We demonstrate below that timed observation sequences of this kind are generated by timed automata. Thus a real-time behaviour of an MM program can be described by a timed automaton.

On the other hand, the concept of timed automata admits an introduction of some operations (parallel composition, sequential composition, choice and iteration) in a natural manner. These operations make it possible to present a sequential process in terms of timed automata associated with individual statements this process is composed of. Furthermore, a true concurrent behaviour of several communicating processes can be presented as a parallel composition of timed automata corresponding to processes and communication channels.

Thus the notion of timed automata is suitable for defining a compositional semantics of programming languages with real time and concurrency. In the section 1 we introduce the concept of timed automata. Timed automata we use are slightly different from those are presented in (Alur and Henzinger 1991; Alur and Dill 1994): in our automata time progresses in a non-strictly monotone sense, i.e. these automata admits several subsequent state transitions within the same time instant. In the section 2 we adopt the general concept of timed automata for purposes of designing formal specification of programming languages semantics and define some useful operations on timed automata. Syntax and informal description of MM language are briefly presented in the section 3. In the last section we demonstrate how introduced timed automata can be applied for developing the formal semantics of MM language.

1 Timed automata

Intuitively, a timed automaton operates in real time by taking transitions from one location into another. Executing a transition takes no time. If no transitions are taken, time progresses by an arbitrary real number. An automaton can stay in a location as long as timing constraints are satisfied. An observable behaviour of a timed automaton is indicated by its propositional variables. Formally, a *timed automaton* A is a tuple

$$(P, Q, Q_0, C, \mu, \nu, E)$$

such that P is a set of basic propositions, Q is a finite set of locations, $Q_0 \subseteq Q$ is a subset of initial locations, C is a finite set of clocks, μ and ν are labeling functions, and $E \subseteq Q \times Q \times 2^C$ is a set of transitions. *Clocks* are real-valued variables, whose values either ascend uniformly as time elapses or reset to 0 on some transitions. Labeling function ν assigns to each location in Q a *timing constraint* over the clocks. Each timing constraint is a boolean combination of atomic timing constraints of the form $\tau \circ c$, where $c \in \mathbb{R}_+$, τ is a clock, and $\circ$ is a binary relation, e.g. $<$, $=$ or $\leq$. Labeling function μ assigns to each location q in Q a boolean formula over the set P of propositions. The formula $\mu(q)$ is called a *propositional constraint*. Each triple $(q, q', \lambda) \in E$ identifies a transition from q into q' with a subset λ of clocks to be reset on the transition.

Any time instant during a run, a state of the timed automaton A is completely characterized by a triple (q, σ, γ), where $q \in Q$ is a location in which the control resides, $\sigma \subseteq P$ is an observation and $\gamma \colon C \to \mathbb{R}_+$ is a clock interpretation. By an *observation* we mean a subset of propositions that are evaluated with true in a location. A *clock interpretation* determines the values of the clocks. At any state (q, σ, γ) the observation σ and the clock interpretation γ should satisfy the propositional $\mu(q)$ and timing $\nu(q)$ constraints. This is denoted by $\sigma \models \mu(q)$ and $\gamma \models \nu(q)$ respectively.

As time goes on, locations and observations may remain unchanged within some intervals. An *interval* is a convex subset of non-negative real numbers $\mathbb{R}_+$. The left end-point of an interval I is denoted by $l(I)$, and the right end-point is denoted by $r(I)$. An *interval sequence* is a finite or infinite sequence of intervals $I_0, I_1, \ldots$ such that for all $t \in \mathbb{R}_+$ there is an interval I_k with $t \in I_k$, and $l(I_{k+1}) = r(I_k)$ for all intervals I_k and I_{k+1}.

A *run* of the timed automaton A is a finite or infinite sequence

$$r\colon \quad \xrightarrow{\gamma_0} (q_0, \sigma_0, I_0) \xrightarrow[\lambda_1]{\gamma_1} (q_1, \sigma_1, I_1) \xrightarrow[\lambda_2]{\gamma_2} (q_2, \sigma_2, I_2) \xrightarrow[\lambda_3]{\gamma_3} \ldots$$

where $q_k \in Q$ are locations, $\sigma_k \subseteq P$ are observations, I_k are intervals, $\lambda_k \subseteq C$ are clock sets, and $\gamma_k \colon C \to \mathbb{R}_+$ are clock interpretations, if the following conditions hold for all $k \geq 0$ and $\tau \in C$

- *initialization*: $q_0 \in Q_0$, $\gamma_0(\tau) = 0$;

- *consecution*: $I_0, I_1, I_2 \ldots$ is an interval sequence,
$$(q_k, q_{k+1}, \lambda_{k+1}) \in E, \quad \gamma_{k+1}(\tau) = \begin{cases} 0, \ \tau \in \lambda_{k+1} \\ \gamma_k(\tau) + r(I_k) - l(I_k), \text{ otherwise} \end{cases}$$
- *invariance*: $\sigma_k \models \mu(q_k)$ and $\gamma_k + t - l(I_k) \models \nu(q_k)$ hold for all $t \in I_k$.

Each run r of the timed automaton A produces a *timed observation sequence*

$$\rho(r): \quad (\sigma_0, I_0) \longrightarrow (\sigma_1, I_1) \longrightarrow (\sigma_2, I_2) \longrightarrow \ldots$$

The set of timed observation sequences generated by a timed automaton is called a *language* of the automaton, or timed language. There is an obvious definition of a parallel composition of timed languages (see (Alur and Dill 1994)). Also a parallel composition of timed automata can be introduced such that the language of parallel composition on automata is, in fact, a parallel composition of their languages.

2 Timed automata of fixed signature

The general concept of timed automata should be adopted to suit our purpose of defining formal semantics of sequential processes. Namely, we associate a timed automaton with each statement of a process and introduce some operations on timed automata to capture the composition of process statements.

Timed automata $(P, Q_i, Q_{0i}, C, \mu_i, \nu_i, E_i)$ corresponding to statements of the same process have many common components. They share a set of clocks C and a set of propositions P. Also their sets of locations Q_i have the similar structure $Q_i = D \times L_i$, where D is a set of process data states and L_i are sets of control flow positions (*labels*) occurred in the i^{th} statement of the process.

These common components form a *signature* π which is a triple (P, D, C), where P is a set of propositions, D is a set of data states, C is a set of clocks. A *timed automaton* A_π *of the signature* π is a tuple

$$(V, L_0, L_1, E),$$

where $V = \{(\ell, \hat{\mu}, \hat{\nu}) \mid \ell \in L\}$ is a set of *location classes*, $L_0 \subseteq L$ is a set of initial labels, $L_1 \subseteq L$ is a set of final labels, and $E \subseteq D \times L \times D \times L \times 2^C$ is a set of transitions. A location class $(\ell, \hat{\mu}, \hat{\nu})$ consists of the label $\ell \in L$ and functions $\hat{\mu}(d)$ and $\hat{\nu}(d)$ that specify propositional and timing constraints of the locations of the form $(d, \ell) \in D \times L$ for every data state $d \in D$. The pair (L_0, L_1) of sets of initial and final labels is called a *context* of the timed automaton A_π.

It is obvious that any timed automaton A_π gives rise to a general timed automaton A in a natural way.

Basic operations Now we are at the point of introducing some basic operations on timed automata of fixed signature. These operations have usual automata-theoretic treatments.

Union Consider timed automata A'_π and A''_π with disjoint sets of labels L' and L''. Then the automaton $A_\pi = (V, L_0, L_1, E)$ such that $V = V' \cup V''$, $L_0 = L'_0 \cup L''_0$, $L_1 = L'_1 \cup L''_1$, $E = E' \cup E''$ is called a *union* of automata and is denoted by $A'_\pi \cup A''_\pi$.

Factorization Let A_π be a timed automaton and $R \subseteq L \times L$ be an equivalence relation on the set of its labels L. Denote by $[\ell]_R$ the equivalence class of a label ℓ w.r.t. R and by $L/_R$ the set of equivalence classes of L induced by R. Then the automaton $A'_\pi = (V', L'_0, L'_1, E')$ such that $V' = \{(\ell', \bigwedge_{\ell\in\ell'} \hat{\mu}, \bigwedge_{\ell\in\ell'} \hat{\nu}) \mid \ell' \in L/_R\}$, $L'_0 = L_0/_R$, $L'_1 = L_1/_R$, and $E' = \{(d_1, [\ell_1]_R, d_2, [\ell_2]_R, \lambda) \mid (d_1, \ell_1, d_2, \ell_2, \lambda) \in E\}$ is called a *factorization of A_π induced by R* and is denoted by $A_\pi/_R$.

Replacing of a context Let $A_\pi = (V, L_0, L_1, E)$ be a timed automaton and (L'_0, L'_1) be a context such that $L'_0, L'_1 \subseteq L$. Then denote by $\left[A_\pi\right]_{(L'_0, L'_1)}$ the timed automaton (V, L'_0, L'_1, E).

Compositional operations Consider timed automata A'_π and A''_π. By $R(L', L'')$ denote the total equivalence on $L' \cup L''$ for any pair of label sets L' and L''. Then sequential composition ($\triangleright$), nondeterministic choice ($+$), and infinite iteration (*) are defined by the following equations.

$$\begin{aligned} A'_\pi \triangleright A''_\pi &= \left(\left[A'_\pi \cup A''_\pi\right]_{(L'_0, L''_1)}\right) \Big/_{R(L'_1, L''_0)} \\ A'_\pi + A''_\pi &= \left(A'_\pi \cup A''_\pi\right) \Big/_{R(L'_0, L''_0) \cup R(L'_1, L''_1)} \\ {A'_\pi}^* &= \left(\left[A'_\pi\right]_{(L'_0 \cup L'_1, \emptyset)}\right) \Big/_{R(L'_0, L'_1)} \end{aligned}$$

These operations give us a possibility to define a compositional semantics of MM statements, so that composing statements corresponds to composing operations on timed automata.

3 Overview of DYANA and MM language

DYANA (DYnamic ANAlyser) is an integrated software environment for design and analysis of distributed system behaviour without hardware prototyping. DYANA deals with models of distributed systems that consist of three main parts: hardware components, software components and communications.

Processors, timers, memory units and other devices are considered as hardware components, or *executors*. Hardware components are specified by their architecture and clock speed. Architecture of an executor provides time estimation subsystem of DYANA with necessary information on duration of basic operations of the executor. It gives DYANA a possibility to estimate execution time of entire computation blocks of processes.

Software components, called *processes*, runs on executors. A process may perform some computation, suspend its own execution or interact with other processes. MM language provides *message passing* for an interprocess communication. A process sends messages into *output ports* and receives messages from *input ports*. There are also intermidiates between output and input ports called *channels*. They are responsible of storing and transmitting of messages. Each channel connects several output ports with an input port, thus a process can receive messages coming from several processes through the same input port.

Below a grammar that describes the syntax of MM processes and statements is presented. In square brackets [ ] we put optional elements and by $(\,)^*$ repeating elements are denoted.

$$P ::= \textbf{process}\ \text{name}\textbf{()}\ \textbf{<input}\ in_0\big(,\ in_j\big)^*\textbf{;}\ \textbf{output}\ out_0\big(,\ out_i\big)^*\textbf{;>}\quad S$$

$$\begin{array}{ll} S ::= & T\ v\textbf{;} \quad | \\ & v = E\textbf{;} \quad | \\ & \textbf{\{}\ S^*\textbf{\}} \quad | \\ & \textbf{complex}\ S \quad | \\ & \textbf{if(}E\textbf{)}\ S_1\ \big[\ \textbf{else}\ S_2\ \big] \quad | \\ & \textbf{while(}E\textbf{)}\ S_1 \quad | \\ & \textbf{stop;} \quad | \\ & \textbf{delay(}E\textbf{);} \quad | \\ & \textbf{send(}v,\ out\textbf{);} \quad | \\ & \textbf{receive(}v,\ in\textbf{);} \quad | \\ & \textbf{select(}\text{nm}\textbf{)}\ \textbf{\{}\ \big(\textbf{case}\ in_j\textbf{:}\ S_j\big)^*\ \big[\ \textbf{timeout}\ E\textbf{:}\ S_0\ \big]\ \textbf{\}} \end{array}$$

Table 1. Syntax of MM process declaration.

In the table 1 and hereafter S is a statement, in is an input port name, out is an output port name, T is a type (**int**, **bool**, **msg**, etc.), v is a name of a variable and E is an expression. An informal meaning of MM statements is as follows:

- $T\ v$; declares a new variable of name v and type T;
- $v = E$; evaluates the expression E and assigns the value to v;
- $\{\ S^*\}$ simple groups statements and presents the statement sequence S^* as a single statement;

- **complex** S indicates a computation block that is processed by the time estimation subsystem;
- **if(**E**)** S_1 [**else** S_2] and **while(**E**)** S_1 have conventional meaning: they are conditional and loop statements respectively;
- **stop;** immediately terminates execution of the process;
- **delay(**E**);** suspends execution of the process for the specified amount of time; time is measured in processor ticks;
- **send(**v**,** out**);** sends a message containing in the variable v to the output port out;
- **receive(**v**,** in**);** receives a message from the input port in and assigns the message to the variable v; if the channel contains no messages then the process is suspended until a message arrives;
- **select { ... }** statement gives a possibility to receive a message from one of the specified input ports in_j; an optional section **timeout** E**:** S_0 sets an upper bound on time that the process waits while all the channels are empty.

4 Semantics of MM language

In order to describe the semantics of MM language we define the mapping of MM models into timed automata in such a way that the timed automaton of a distributed system model produces timed observation sequences of all possible executions of the model. These timed automata are constructed as parallel compositions of timed automata of channels and timed automata of processes, i.e.

$$\Big\Vert_{p} \left(P(p, B(p)) \Big\Vert_{in} C(p, in, K(p, in)) \right),$$

where p is a process name, $B(p)$ is an executor of the process p, $P(p, B(p))$ is a timed automaton of the process p running on the executor $B(p)$, in is an input port name, $K(p, in)$ is a set of names of output ports that are connected to the input port in of the process p, and $C(p, in, K(p, in))$ is a timed automaton of the channel. All functions mentioned above are constructed from a distributed system model. For the sake of brevity much details of second importance will be omitted. We focus mostly on the principles of derivation of timed automata by statements and processes.

Observable variables In order to describe an observable behaviour of processes and communication channels we introduce the following propositional variables:

- $\mathsf{send}(p.out, m)$ and $\mathsf{full}(p.out)$ for each output port out of a process p. $\mathsf{send}(p.out, m)$ holds whenever the process p is at the state of sending a message m into the output port out. $\mathsf{full}(p.out)$ holds whenever the channel is capable of taking a message from the output port out.

- $\mathsf{receive}(p.in, m)$ and $\mathsf{empty}(p.in)$ for each input port in of a process p. $\mathsf{receive}(p.in, m)$ holds whenever the process p is at the state of receiving a message m from the input port in. $\mathsf{empty}(p.in)$ holds whenever the channel of the input port in contains no messages.

We use predicate-like notation for names of propositional variables. For example, $\mathsf{send}(p.out, m)$ denotes an individual propositional variable for a message $m \in \mathsf{M}$, process p and output port out.

Within the timed automata framework interaction of automata is modeled by means of sharing observable propositional variables. Namely, consider a parallel composition of two timed automata that share some observable variables, i.e. there are variables of the same name that belong to both automata. Then, along each run of the parallel composition any shared variable has the same value in both automata.

Each propositional variable associated with an input port in or an output port out is shared by a timed automaton of the process p and a timed automaton of the channel connected to the port.

Semantics of process declaration Consider a process declaration. As we mentioned above all timed automata associated with statements of the process have the same signature. This signature $\pi = (P, \mathsf{D}, \{\tau\})$ is constructed according to the header of process declaration. The set P of propositional variables consists of all observable variables describing behaviour of the input and output ports. Data space D is a set of all possible evaluations of variables declared in the process. The set of clocks consists of a single clock τ which is used to measure duration of actions performed by the process.

Semantics of statements In the table 2 the semantics of some compound MM statements is defined.

For each statement S the semantic function $\mathbb{S}[\![S]\!](\pi, T)$ maps a signature $\pi \in \Sigma$ and a time unit $T \in \mathsf{T}$ onto a timed automaton of the signature π that behaves like the statement S. We see that the semantic definitions can be derived pretty straightforwardly using the introduced operations on timed automata.

$$
\begin{aligned}
&\mathcal{S}\colon \textsc{Statement} \longrightarrow \Sigma \times \mathsf{T} \longrightarrow \mathsf{TA}\\
&\mathcal{S}[\![S_1\ S_2]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.\quad \Big(\mathcal{S}[\![S_1]\!](\pi, T) \triangleright \mathcal{S}[\![S_2]\!](\pi, T)\Big)\\
&\mathcal{S}[\![\mathbf{if(}E\mathbf{)}\ S_1\ \mathbf{else}\ S_2]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.\\
&\qquad\qquad \Big(\mathcal{C}[\![E]\!](\pi, \mathbf{true}) \triangleright \mathcal{S}[\![S_1]\!](\pi, T)\Big) + \Big(\mathcal{C}[\![E]\!](\pi, \mathbf{false}) \triangleright \mathcal{S}[\![S_2]\!](\pi, T)\Big)\\
&\mathcal{S}[\![\mathbf{while(}E\mathbf{)}\ S_1]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.\\
&\qquad\qquad\qquad\qquad \Big(\mathcal{C}[\![E]\!](\pi, \mathbf{true}) \triangleright \mathcal{S}[\![S_1]\!](\pi, T)\Big)^* + \mathcal{C}[\![E]\!](\pi, \mathbf{false})
\end{aligned}
$$

Table 2. Semantics of compound MM statements.

Semantic function $\mathcal{C}[\![E]\!](\pi, v)$ gives a timed automaton that evaluates the expression E at the initial location ℓ and then immediately passes to the final location ℓ_c if the value is equal to the parameter v. In the table 3 and hereafter we assume that ℓ, ℓ_c, etc. are newly introduced labels. Propositional constraint N of the location ℓ or ℓ_c specifies that the process does not send or receive any message at the location.

Semantic function $\mathcal{E}[\![E]\!]$ specifies a denotation of the expression E. It maps an valuation of variables onto an expression value (domain EV). The definition of $\mathcal{E}[\![E]\!]$ is quite standard and omitted here for brevity.

$$
\begin{aligned}
&\mathcal{E}\colon \textsc{Expression} \longrightarrow \mathsf{D} \longrightarrow \mathsf{EV}\\
&\mathcal{C}\colon \textsc{Expression} \longrightarrow \Sigma \times \mathsf{EV} \longrightarrow \mathsf{TA}\\
&\mathcal{C}[\![E]\!] = \lambda\pi \in \Sigma.\ \lambda v \in \mathsf{EV}.\\
&\qquad\qquad \left[\begin{array}{l}(\ell, \mathsf{N}, \tau = 0)\\(\ell_c, \mathsf{N}, \tau = 0)\end{array}, \{\ell\}, \{\ell_c\}, \Big\{\ (d, \ell, d, \ell_c, \tau)\ \Big\}_{d \in \mathsf{D}\colon\ \mathcal{E}[\![E]\!](d) = v}\right]_\pi
\end{aligned}
$$

Table 3. Semantics of expressions.

Finally, in the table 4 we demonstrate how to define the formal semantics of some typical MM statements.

In the table 4 we use the following abbreviations. A propositional constraint $\mathsf{E}(in)$ specifies that the input port in is empty, and a propositional constraint $\mathsf{R}(m, in)$ specifies that the process is receiving the message m from the input port in. By $d[v \mapsto e]$ we denote a mapping that coincides with the mapping d everywhere except v, where it gives e.

As an example we consider the semantics equation of **receive** statement. A timed automaton realizing behaviour of **recieve** statement acts as follows. It starts at the initial location ℓ and passes to the location ℓ_w if the channel of the input port in is empty. Otherwise it passes to the location ℓ_a, where the process receives an incoming message. At the location ℓ_w the automaton can stay for an arbitrary long time (timing constraint is **true**) and it leaves the location only when a message arrives to the channel. ¿From the location ℓ_a the automaton immediately gets to the final location $\hat{\ell}$.

The semantics of other MM statements is defined in a similar way.

$$\mathcal{S}[\![v = E;]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.$$
$$\left[\begin{array}{l} (\ell, \mathsf{N}, \tau = 0) \\ (\hat{\ell}, \mathsf{N}, \tau = 0) \end{array}, \{\ell\}, \{\hat{\ell}\}, \left\{ \left(d, \ell, d[v \mapsto \mathcal{E}[\![E]\!](d)], \hat{\ell}, \tau\right) \right\}_{d \in \mathsf{D}} \right]_\pi$$
$$\mathcal{S}[\![\mathbf{delay}(E);]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.$$
$$\left[\begin{array}{l} (\ell, \mathsf{N}, \tau = 0) \\ (\ell_w, \mathsf{N}, \tau \le T * \mathcal{E}[\![E]\!]) \\ (\ell_a, \mathsf{N}, \tau = T * \mathcal{E}[\![E]\!]) \\ (\hat{\ell}, \mathsf{N}, \tau = 0) \end{array}, \{\ell\}, \{\hat{\ell}\}, \left\{ \begin{array}{l} (d, \ell, d, \ell_w, \tau) \\ (d, \ell_w, d, \ell_a, \emptyset) \\ (d, \ell_a, d, \hat{\ell}, \tau) \end{array} \right\}_{d \in \mathsf{D}} \right]_\pi$$
$$\mathcal{S}[\![\mathbf{receive}(v, in);]\!] = \lambda\pi \in \Sigma.\ \lambda T \in \mathsf{T}.$$
$$\left[\begin{array}{l} (\ell, \mathsf{N}, \tau = 0) \\ (\ell_w, \mathsf{N} \wedge \mathsf{E}(in), \mathbf{true}) \\ (\ell_a, \mathsf{R}(\mathcal{E}[\![v]\!], in), \tau = 0) \\ (\hat{\ell}, \mathsf{N}, \tau = 0) \end{array}, \{\ell\}, \{\hat{\ell}\}, \left\{ \begin{array}{l} (d, \ell, d, \ell_w, \tau) \\ (d, \ell_w, d, \ell, \tau) \\ (d, \ell, d[v \mapsto m], \ell_a, \tau) \\ (d, \ell_a, d, \hat{\ell}, \tau) \end{array} \right\}_{\substack{m \in \mathsf{M} \\ d \in D}} \right]_\pi$$

Table 4. Semantics of MM statements.

Conclusions

In this paper we show how it is possible to define formally a semantics of programming language with real-time and concurrency by combining Denotational Semantics (Mosses 1990) and the concept of Timed Automata (Alur and Dill 1994; Alur and Henzinger 1991). As an example we consider MM language based on message passing, but this approach can be adopted as well to programming languages that provide shared variables for synchronization. Our techniques captures the main characteristic features of real-time parallel execution of communicated processes and provides promising opportunities of well-known formal verification techniques (Alur et al. 1993; Henzinger et al. 1994) application.

References

Alur, R., C. Courcoubetis, and D. L. Dill (1993, May). Model-Checking in Dense Real-time. *Information and Computation 104*(1), 2–34.

Alur, R. and D. L. Dill (1994, 25 April). A theory of timed automata. *Theoretical Computer Science 126*(2), 183–235.

Alur, R. and T. A. Henzinger (1991). Logics and Models of Real Time: A Survey. *Lecture Notes in Computer Science 600*, 74–106.

Bakhmurov, A. G., A. P. Kapitonova, and R. L. Smeliansky (1999). DYANA: an Environment for Embedded System Design and Analysis. In W. R. Cleaveland (Ed.), *Proceedings of TACAS '99*, Volume 1579 of *LNCS*, pp. 390–404. Springer.

Henzinger, T. A., X. Nicollin, J. Sifakis, and S. Yovine (1994, June). Symbolic Model Checking for Real-Time Systems. *Information and Computation 111*(2), 193–244.

Mosses, P. D. (1990). Denotational Semantics. In J. van Leewen (Ed.), *Handbook of Theoretical Computer Science*, Volume B: Formal Models and Semantics, Chapter 11, pp. 575–631. Amsterdam: North-Holland.

Learning Phonotactics Using ILP

STASINOS KONSTANTOPOULOS
Alfa-Informatica, Rijks*universiteit* Groningen
konstant@let.rug.nl

ABSTRACT. This paper describes experiments on learning Dutch phonotactic rules using Inductive Logic Programming, a machine learning approach based on the notion of inverting resolution. Different ways of approaching the problem are experimented with, and compared against each other as well as with related work on the task. Further research is outlined.

1 Introduction

The *Phonotactics* of a given language is the set of rules that identifies what sequences of phonemes constitute a possible word in that language. The problem can be broken down to the *syllable structure* (i.e. what sequences of phonemes constitute a possible syllable) and the processes that take place at the syllable boundaries (e.g. assimilation).

Previous work on the syllable structure of Dutch includes hand-crafted models, like the ones described in (van der Hulst 1984) and (Booij 1995), but also machine-learning approaches: abduction in (Tjong Kim Sang and Nerbonne 2000) and neural networks in (Stoianov and Nerbonne 1999) and (Stoianov 2001, ch. 4).

This paper describes experiments on the task of constructing from examples a model of Dutch monosyllabic words. The reason for restricting the domain is to avoid the added complexity of handling syllable boundary phonological processes. Furthermore by not using polysyllables no prior commitment is made to any one particular syllabification (and thus syllable structure) theory.

2 Inductive Logic Programming and Aleph

Inductive Logic Programming (ILP) is a machine learning discipline. It is logic programming in the sense that the target concept to be learned is a logic program, i.e. a set of Horn clauses. It is inductive because the core operator used is that of *induction*.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

Induction can be seen as the inverse of deduction. For example from the clauses 'All humans die' and 'Socrates is human' the fact that 'Socrates will die' can be deduced. Inversely, induction uses background knowledge (e.g. 'Socrates is human') and a set of observations (training data) (e.g. 'Socrates died') to search for a hypothesis that, in conjunction with the background knowledge, can deduce the data. In more formal terms, given a logic program B modelling the background knowledge and a set of ground terms D representing the training data, ILP constructs a logic program H, such that $B \wedge H \models D$.

If the deductive operator used is resolution (as defined by Robinson 1965), then the inductive operator necessary to solve the equation above is the inverse resolution operator, as defined by Muggleton and De Raedt (1994).

Aleph is an ILP system implementing the Progol algorithm (Muggleton 1995). This algorithm allows for single-predicate learning only, without background theory revision or predicate invention. It incrementally constructs the clauses of a single-predicate hypothesis that describes the data, by iterating through the following basic algorithm:

- Saturation: pick a positive example from the training data and construct the most specific, non-ground clause that entails it. This is done by repeated application of the inverse resolution operator on the example, until all its ground terms have been replaced by variables which are appropriately bound by the body of the clause, so that the original ground positive and only that is covered by the bottom clause. This minimally generalised clause is called the *bottom clause.*

- Reduction: search between the maximally general, empty-bodied clause and the maximally specific bottom clause for a 'good' clause. The space between the empty clause and the bottom clause is partially ordered by θ-subsumption, and the search proceeds along the lattice defined by this partial ordering. The 'goodness' of each clause encountered along the search path is evaluated by an *evaluation function.*

- Cover removal: add the new clause to the theory and remove all examples covered by it from the dataset.

- Repeat until all positive examples are covered.

The *evaluation function* quantifies the usefulness of each clause constructed during the search and should be chosen so that it balances between overfitting (i.e. covering the data too tightly and making no generalisations that will yield coverage over unseen data) and overgeneralising (i.e. covering the data too loosely and accepting too many negatives.) It can be simple coverage (number of positive minus number of negative examples covered by the clause) or the Bayes probability that the clause is correct given the data (for learning from positive data only) or any function of the numbers of examples covered and the length of the clause (for implementing bias towards shorter clauses.)

Syntactic bias is applied during reduction to either prune paths that are already known to not yield a winning clause, or to enforce restrictions on the constructed theory, for example, conformance to a theoretical framework.

3 Setting up the Experiments

As a starting point, a rough template matching all syllables is assumed. This template is $\mathcal{C}_3\mathcal{V}\mathcal{C}_5$, where $\mathcal{C}_n$ represents any consonant cluster of length up to n and $\mathcal{V}$ any vowel or diphthong. The problem can now be reformulated as a single-predicate learning task where the target theory is one of acceptable prefixes to a given vowel and partial consonant cluster. The rules for prevocalic and postvocalic affixing are induced in two separate learning sessions.

The training data is derived from 5095 monosyllabic words taken from the Dutch section of the CELEX Lexical Database, with an additional 597 reserved for evaluation.

The positive examples are constructed by breaking the phonetic transcriptions down to three parts: a prevocalic and a postvocalic consonant cluster (consisting of zero or more consonants) and a vowel or diphthong. The consonant clusters are treated as 'affixes' to the vowel, so that syllables are constructed by repeatedly affixing consonants, if the *context* (the vowel and the pre- or post-vocalic material that has been already affixed) allows it. So, for example, from the word /maːkt/ the following positives would be generated:

```
prefix(m,[],[a,:]).    suffix(k,[],[:,a]).
prefix(^,[m],[a,:]).   suffix(t,[k],[:,a]).
                       suffix(^,[tk],[:,a]).
```

Note that the context lists in suffix rules is reversed, so that the two processes are exactly symmetrical and can use the same background predicates.

The caret, `^`, is used to mark the beginning and end of a word. The reason that the affix termination needs to be explicitly licenced is so that it is not assumed by the experiment's setup that all partial sub-affixes of a valid affix are necessarily valid as well.

In Dutch, for example, a monosyllable with a short vowel has to be closed, which means that the null suffix is not valid. The end-of-word mark allows for this to be expressable as a theory that does not have the following clause: `suffix(^,[],[V]).`

The positives are, then, all the prefixes and suffixes that must be allowed in context, so that all the monosyllables in the training data can be constructed: 11067 amd 10969 instances of 1428 and 1653 unique examples, respectively.

The negative data is randomly generated words that match the $\mathcal{C}_3\mathcal{V}\mathcal{C}_5$ template and do not appear as positives. The random generator is also designed so that approximately equal numbers of examples at generated at each at each affix length.

The negative data is also split into evaluation and training data, and the negative examples are derived from the training negative data by the following deductive algorithm:

1. For each example, find the maximal substring that is provable by the positive `prefix/3` and `suffix/3` clauses in training data. So, for example, for /mtratk/ it would be `trat` and for /mlat/, `lat^`.

2. Choose the clause that should be a negative example, so that this word is not accepted by the target theory. Pick the inner-most one on each side, i.e. the one immediately applicable to the maximal substring computed above. For /mlat/ that would be `suffix(m,[l],[a])`. /mtratk/, however, could be negative because either `prefix(m,[tr],[a])` or `suffix(k,[t],[a])` are unacceptable . In such cases, pick one at random. This is bound to introduce false negatives, but no alternative that does not presuppose at least part of the solution could be devised.

3. Iterate, until enough negative examples have been generated to disprove all the words in the negative training data.

Since the problem is, in effect, that of identifying the sets of consonants that may be prefixed or suffixed to a partially constructed monosyllable, the clauses of the target predicate must have a means of referring to various subsets of $\mathcal{C}$ and $\mathcal{V}$ in a meaningful and intuitive way. This is achieved by defining a (possibly hierarchical,) linguistically motivated partitioning of $\mathcal{C}$ and $\mathcal{V}$. Each partition can then be referred to as a feature-value pair, for example LAB+ to denote the set of the labials or VOIC+ for the set of voiced consonants. Intersections of these basic sets can then be easily referred to by feature-value vectors; the intersection, for example, of the labials and the voiced consonants (i.e. the voiced labials) is the feature-value vector [VOIC+,LAB+].

The *background knowledge* is, as seen in section 2, playing a decisive role in the quality of the constructed theory, by implementing the theoretic framework to which the search for a solution will be confined. In more concrete terms, the background predicates are the building blocks that will be used for the construction of the hypothesis' clauses and they must be defining all the relations necessary to discover an interesting hypothesis.

For the purposes of this task, they have been defined as relations between individual phones and feature values, e.g. `labial(m,+)` or `voiced(m,+)`.

Feature-value vectors can then be expressed as conjunctions like, for example, `labial(C,+)` $\wedge$ `voiced(C,+)` to mean the voiced labials.

Except for the linguistic features predicates, the background knowledge also contained the `head/2` and `rest/2` list access predicates. This approach was chosen over direct list access with the `nth/3` predicate, as bias towards rules with more local context dependencies.

The experiments described in sections 3.1, 3.2 and 3.3 below, were conducted with background knowledge that encodes increasingly more information about Dutch phonology as well as Dutch phonotactics: for the experiment in 3.1 the learner has access to the way the various symbols are arranged in the IPA table, whereas for the experiment in 3.2 a classification that is sensitive to Dutch phonological processes was chosen. And, finally, in section 3.3 the *sonority level* feature is implemented, which has been proposed with the explicit purpose of solving the problem of Dutch syllable structure.

The quantitative evaluation shown for three experiments was done using the 597 words and the part of the randomly generated negative data that have been reserved for this purpose.

3.1 The IPA segment space

In this experiment the background knowledge reflects the way that the IPA table is organised: the phonetic inventory of Dutch consists of two disjoint spaces, one of consonants and one of vowels, with three and four orthogonal dimensions of differentiation respectively.

The consonant space varies in *place* and *manner of articulation,* and *voicing.* The manner of articulation can be *plosive, nasal, lateral approximant, trill, fricative* or *approximant.* The place can be *bilabial, alveolar, velar, labiodental, postalveolar* or *palatal.* Voicing can be present or not.

Similarly for vowels, where there are four dimensions: place (front, centre, back) and manner of articulation (open, mid-open, mid-closed, closed), length and roundedness.

The end-of-word mark has no phonological features whatsoever and it does not belong to any of the partitions of either $\mathcal{C}$ or $\mathcal{V}$.

This schema was implemented as one background predicate per dimension relating each phone with its value along that dimension, for example:

```
manner( plosive, p ).   place( bilabial, p ).
manner( nasal, m ).
```

etc.

The evaluation function used was the *Laplace function* $\frac{P+1}{P+N+2}$, where P and N is the coverage of positive and negative examples, respectively.

Since the randomly generated negatives must also contain false negatives, it cannot be expected that even a good theory will fit it perfectly. In order to avoid overfitting, the learning algorithm was set to only require an accuracy of 85% over the training data.

The resulting hypothesis consisted of 199 prefix and 147 suffix clauses and achieved a recall rate of 99.3% with 89.4% precision. All the false negatives were rejected because they couldn't get their onset licensed, typically because it only appears in a handful of loan words. The /ʤ/ onset necessary to accept 'jeep' and 'junk', for example, was not permitted and so these two words were rejected.

The most generic rules found were:

```
prevoc(A,B,C) :- A= '^'.
prevoc(A,[],C).

postvoc(A,B,C) :-  A= '^'.
postvoc(A,[],C).
```

meaning that (a) the inner-most consonant can be anything, and (b) all sub-prefixes (-suffixes) of a valid prefix (suffix) are also valid.

Other interesting rules include pairs like these two:

```
prevoc(A,B,C) :-
   head(B,D), manner(trill,D), head(C,E), length(short,E),
   manner(closed,E), manner(plosive,A).
prevoc(A,B,C) :-
   head(B,D), manner(trill,D), head(C,E), length(short,E),
   manner(open_mid,E), manner(plosive,A).
```

and

```
prevoc(A,B,C) :-
   head(B,D), manner(approx,D), head(C,E), length(short,E),
   place(front,E), voiced(minus,A).
prevoc(A,B,C) :-
   head(B,D), manner(approx,D), head(C,E), length(short,E),
   place(front,E), manner(plosive,A), place(alveolar,A).
```

that could have been collapsed if a richer feature system would include features like 'closed or mid-open vowel' and 'devoiced consonant or plosive alveolar', respectively. These particular disjunctions might be un-intuitive or even impossible to independently motivate, but they do suggest that a redundant feature set might allow for more interesting theories than the minimal, orthogonal one used for this experiment. This is particularly true for a system like Aleph, that performs no predicate invention or background theory revision.

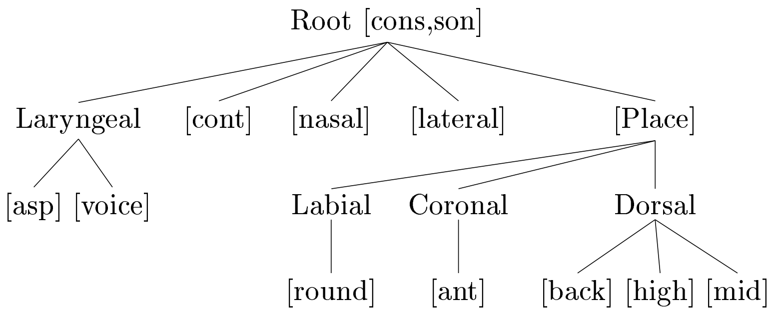


Figure 15.1: The feature geometry of Dutch

3.2 Feature Classes

For this experiment a richer (but more language-specific) background knowledge was made available to the inductive algorithm, by implementing the feature hierarchy suggested by Booij (1995, ch. 2) and replicated in figure 15.1.

The most generic features are the *major class features* (Consonant and Sonorant) that are placed on the root node and divide the segment space in vowels [Cons-,Son+], obstruents [Cons+,Son-] and sonorant consonants [Cons+,Son+]. Since all vowels are sonorous, [Cons-,Son-] is an invalid combination.

The features specifying the continuants, nasals and the lateral /l/ are positioned directly under the root node, with the rest of the features bundled together under two *feature classes,* those of the *laryngeal* and the *place* features. These classes are chosen so that they collect together features that behave as a unit in phonological processes of Dutch. The class of laryngeal features is basically making the voiced-voiceless distinction, while the Aspiration feature is only separating /h/ from the rest. The *place* class bundles together three subclasses of place of articulation features, one for each articulator. Furthermore some derived or redundant features such as Glide, Approximant and Liquid are defined. The vowels do not include the schwa, which is set apart and only specified as Schwa+.

Using the Laplace evaluation function and this background, the constructed theory consisted of 13 prefix and 93 suffix rules, accepting 94.2% of the test positives and under 7.4% of the test negatives.

Among the rejected positives are loan words ('jeep' and 'junk' once again), but also all the words starting with perfectly Dutch /s/ - obstruent - liquid clusters. The prefix rule with the widest coverage is:

```
prefix(A,B,C) :-
```

```
   head(C,D), sonorant(D,plu), rest(B,[]).
```

or, in other words, 'prefix anything before a single consonant before a nucleus other than the schwa'.

The suffix rules were less strict, with only 3 rejected positives, 'branche', 'dumps' and 'krimpst' (the first two of which are loan words) that failed to suffix /ʃ/, /s/ and /s/ respectively. Some achieve wide coverage (although never quite as wide as that of the prefix rules,) but some are making reference to individual phonemes and are of more restricted application. For example:

```
suffix(A,B,C) :-
   rest(C,D), head(D,E), rest(B,[]), A=t.
```

or, 'suffix a /t/ after exactly one consonant, if the nucleus is a long vowel or a diphthong'.

Of some interest are also the end-of-word marking rules (see in section 3 above about the ^ mark), because of the fact that open, short monosyllables are very rare in Dutch (there are four in CELEX: 'schwa', 'ba', 'h?', and 'joh'). This would suggest that the best way to treat those is as exceptions, and have the general rule disallow open, short monosyllables. What was learned instead was a whole set of 29 rules for suffixing ^, the most general of which is:

```
postvoc(A,B,C) :-
   head(B,t), larynx(t,E), rest(B,F),
   head(F,G), larynx(G,E), A= '^'.
```

or 'suffix an end-of-word mark after at least two consonants, if the outer-most one is a /t/ and has the same values for all the features in the LARYNGEAL feature class as the consonant immediately preceding it'.

A final note that needs to be made regarding this experiment, is one regarding its computational complexity. Overlapping and redundant features might be offering the opportunity for more interesting hypotheses, but are also making the search space bigger. The reason is that overlapping features are diminishing the effectiveness of the inverse resolution operator at keeping uninteresting predicates out of the bottom clause: the more background predicates can be used to prove the positive example on which the bottom clause is seeded, the longer the latter will get.

3.3 Sonority Scale

This experiment implements and tests the syllabic structure model suggested in (van der Hulst 1984, ch. 3). The Dutch syllable is there analysed as having three prevocalic and 5 postvocalic positions, (some of which may be empty) and constraints are placed on the set of consonants that can occupy each.

phoneme	obstruents	m	n	l	r	glides	vowels
sonority	1	2	2.25	2.5	2.75	3	4

Table 15.1: The Sonority Scale

The most prominent constraint is the one stipulating a high-to-low *sonority* progression from the nucleus outwards. Each phoneme is assigned a sonority value (table 15.1) based not only on language-independent features such as it being a SONORANT or an OBSTRUENT, but also because of syllable structure of Dutch itself. Especially the fine tuning done with respect to the sonority values of the nasals and the liquids is explicitly justified on filtering out impossible consonant clusters that would otherwise be predicted by the simpler model. It must, therefore, be noted that the background knowledge for this experiment is not only language-specific, but is also directly aimed at solving the very problem that is being investigated.

In addition to the high-to-low sonority level progression from the nucleus outwards, there are both *filters* and explicit licensing rules. The former are restrictions referring to sonority (e.g. 'the sonority of the three left-most positions must be smaller than 4') or other phonological features (e.g. the 'no voiced obstruents in coda' filter in p. 92) and are applicable in conjunction to the sonority rule. The latter are typically restricted in scope rules, that take precedence over the sonority-related constraints mentioned so far. The left-most position, for example, may be /s/ or empty, regardless of the contents of the rest of the onset.

Implementing the basic sonority progression rule as well as the most widely-applicable filters and rules[1] yielded impressive compression rates matched with results lying between those of the two previous experiments: 93.1% recall, 83.2% precision.

4 Conclusions and Further Work

The quantitative results from the machine learning experiments presented above are collected in table 15.2, together with those of Tjong Kim Sang and Nerbonne (2000)[2] and the results from the sonority scale experiment. Those last ones in particular, are listed for comparison's sake and as the logical

[1]Some were left out because they were too lengthy when translated from their fixed-position framework to the affix licensing one used here, and were very specifically fine tuning the theory to individual onsets or codas.

[2]From experiments on phonetic data in the 'Experiments without linguistic Constraints' section.

	(Tjong 2000)	IPA	Feat. Classes	Sonority
Recall	99.1%	99.3%	94.2%	93.1%
Precision	74.8%	79.8%	92.6%	83.2%
Num. Clauses	577 + 577	145 + 36	13 + 93	3+8

Table 15.2: Results

end-point of the progression towards more language- and task-specific prior assumptions.

The first two columns are directly comparable, because they are both only referring to phonetic primitives with no linguistically sophisticated background knowledge. Furthermore the fact that the $\mathcal{C}_3\mathcal{V}\mathcal{C}_5$ template assumed in this work is not taken for granted in (Tjong Kim Sang and Nerbonne 2000) is taken in account in terms of compactness as well as performance, since (a) in (Tjong Kim Sang and Nerbonne 2000) there are 41 extra rules besides the 1154 prefix and suffix rules that describe the "basic word" on which the latter operate, and (b) the precision in (Tjong Kim Sang and Nerbonne 2000) is measured on random strings, whereas in this work only strings matching the $\mathcal{C}_3\mathcal{V}\mathcal{C}_5$ template are used.

As can be seen, then, the ILP-constructed rules compare favourably (in both performance and hypothesis compactness) with those constructed by the deductive approach employed in (Tjong Kim Sang and Nerbonne 2000).

What cat be also seen by comparing the two ILP results with each other, is that the drop in recall between the the second and third column is compensated by higher precision and compression, suggesting a direct correspondence between the quality of the prior knowledge encoded in the background theory and that of the constructed hypothesis.

One interesting follow-up to these experiments would be attempting to expand their domain to that of syllables of multisyllabic words and, eventually, full word-forms. In the interest of keeping the problems of syllabic structure and syllable-boundary phonology apart, a way must be devised to derive from the positive data (i.e. a corpus of Dutch word-forms) examples for a distinct machine learning session for each task.

Furthermore, it would be interesting to carry out the same (or rather the equivalent) experiments on other parts of the CELEX corpus (e.g. English or German) and see to which extend the results-to-background relation follows the same patterns.

On the purely computational aspect of the problem, and as has already been mentioned in section 3.2, including overlapping and redundant features in the background might be interesting, but is also implying a very fast growth of the search space. It would, therefore, be useful to employ the search path pruning facilities of Aleph to avoid directions along the search path that would yield inconsistent or unlikely feature combinations.

References

Booij, G. (1995). *The Phonology of Dutch.* The Phonology of the World's Languages. Oxford: Clarendon Press.

Muggleton, S. and L. De Raedt (1994). Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming 19*(20), 629–679. Updated version of technical report CW 178, May 1993, Department of Computing Science, K.U. Leuven.

Muggleton, S. H. (1995). Inverse Entailment and Progol. *New Generation Computing 13*, 245–286.

Robinson, J. A. (1965). A Machine-Oriented Logic based on the Resolution Principle. *Journal of the ACM 12*(1), 23–41.

Stoianov, I. (2001). *Connectionist Lexical Processing.* Ph. D. thesis, Rijksuniversiteit Groningen.

Stoianov, I. and J. Nerbonne (1999). Exploring Phonotactics with Simple Recurrent Networks. In van Eynde, Schuurman, and Schelkens (Eds.), *Computational Linguistics in the Netherlands.*

Tjong Kim Sang, E. and J. Nerbonne (2000). Learning the Logic of Simple Phonotactics. In J. Cussens and S. Dzeroski (Eds.), *Learning Language in Logic*, Volume 1925 of *Lecture Notes in Artificial Intelligence.* Springer Verlag.

van der Hulst, H. (1984). *Syllable Structure and Stress in Dutch.* Ph. D. thesis, Rijksuniversiteit Leiden.

Self Embedded Relative Clauses in a Corpus of German Newspaper Texts

CHRISTIAN KORTHALS
University of the Saarland
cnkortha@coli.uni-sb.de

ABSTRACT. The distribution of center self-embeddings and extrapositions in German is assumed to minimize memory load during parsing. Self-embedded relative clauses were semi-automatically analysed in a treebank of German newspaper texts. Clause length and especially extraposition distance are found as the most important parameters distinguishing center embeddings from extrapositions. The internal structure of superordinated and subordinated clause is not important.[1]

1 Introduction

The opposition of center self-embedding constructions and extrapositions is an interesting phenomenon of syntax from two points of view: from the perspective of parsing and automata theory, center embeddings force the parser to possess context free power at least. From a psycholinguistic perspective, center embeddings and extrapositions are assumed to differ in their memory load, because the processing of one phrase may have to be delayed until intervening material has been processed.

This paper examines German relative clause self-embeddings as an example of these kinds of constructions.[2] In section 2, we define and classify relative clauses and demonstrate that the choice of center embedding vs. extraposition arises in German syntax due to verb final position. Hypotheses on the acceptability of different constructions are formulated primarily on the basis of Hawkins' performance theory, which is introduced in section 3 together with other preliminaries. The hypotheses were tested against data

[1]This work was funded within the NEGRA project, which is part of the *Sonderforschungsbereich* 378, funded by the *Deutsche Forschungsgemeinschaft* at the University of the Saarland. I wish to thank Thorsten Brants, Reinhard Köhler, Lars Konieczny, Valia Kordoni, Daniela Kurz, Oliver Plähn, Christoph Scheepers and Hans Uszkoreit for their comments and support.

[2]Note that the term "self-embedding" is used to refer to all cases where a RC (or, defining more weakly, also an S) contains another RC. We use the terms "center embedding", "right embedding" and "extraposition" to subclassify self-embeddings.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

from a treebank of German newspaper texts. This corpus is described in section 4 together with a method of deriving description parameters of each relative clause in the corpus fully automatically. In section 5, we present the results of the evaluation of three hypotheses: a) Longer relative clauses should tend to be extraposed, while shorter ones should tend to be embedded. This turns out to be weakly supported only (Section 5.1). b) The potential extraposition distance (the distance between relative pronoun and antecedent) should decide whether a clause is extraposed or not (Section 5.2). This factor will turn out to be the most relevant one. c) Superordinated and subordinated RC should differ in their internal structure. This turns out to be false (Section 5.3).

2 Types of Relative Clauses

A relative clause is a sub-clause that contains at least one relative pronoun. The TIGER annotation scheme (Brants et al. 2000) and the Stuttgart Tübingen tagset (Schiller et al. 1999) classify relative pronouns into four classes based on two criterions: substituting vs. attributive, and proper relative pronoun vs. wh-pronoun. Especially when wh-pronouns are involved, there is gradiance between relative clauses and other clause types (e.g. place holder phrase, clausal complement).

In German, main declarative clauses in the present or past tense have verb second position. If a post-verbal NP contains a relative clause, it may turn out to be situated at the end of the sentence, directly following its antecedent NP. Such a case of *right embedding* (RE) yields no difficulty at all, as in example (1).

(1) *Er kannte den Termin, der für die Konferenz festgesetzt war.*
He knew the deadline which for the conference set was.
"He knew the deadline set for the conference."

In contrast, perfect tense declarative clauses require an auxiliary verb and an infinite verb in final position. Also subordinate clauses (and among them relative clauses) have the (finite) verb in final position. This means that whenever a relative clause is embedded in another relative clause, at the very least the verb of the superordinated relative clause must follow after the subordinated relative clause, if the antecedent NP is to be realized continuously, i.e. without intervening material. These cases will be called *center embeddings* (CE). (2) is a complex (artificial) example, where a relative clause is center embedded in another relative clause, which is again center embedded in the main clause. If center embedding is to be avoided, the subordinated relative clause must be extraposed over the verb (and maybe some of its arguments). This case is illustrated in example (3) and will be called an *extraposition* (EX).

However, it is possible to extrapose even more material than the relative clause alone, as illustrated in example (4). This again results in a continuous realization, and thus is considered a type of RE, too.

(2) [$_S$*Er hat den Termin*$_{\mathsf{NN}}$, [$_{RC}$*der*$_{\mathsf{PREL}}$ *für die Konferenz*$_{\mathsf{NN}}$, [$_{RC}$*die*$_{\mathsf{PREL}}$
He has the deadline which for the conference which
er besuchen$_{\mathsf{VINF}}$ *wollte*$_{\mathsf{VFIN}}$,] *festgesetzt*$_{\mathsf{VPART}}$ *war*$_{\mathsf{VFIN}}$,] *gekannt*$_{\mathsf{VFIN}}$.]
he attend wanted set was known
"He knew the deadline that was set for the conference he wanted to attend."

(3) [$_S$Er hat den Termin, [$_{RC}$der für die Konferenz festgesetzt war, [$_{RC}$die er besuchen wollte,]] gekannt.]

(4) [$_S$Er hat den Termin, [$_{RC}$der festgesetzt war für die Konferenz, [$_{RC}$die er besuchen wollte,]] gekannt.]

Throughout the paper, we will investigate relative clauses of embedding depth 2, that is the most inner RCs in sentences of the structure $[S\ \alpha\ [RC\ \beta\ [RC]\ \gamma]\ \delta]$. Note that the dominance relation between the clauses will in general be indirect: Normally, at least an NP, the antecedent NP of the RC will occur in α or β; there may also be more material, e.g. coordinated structures or VPs. If there is no antecedent NP, we speak of a *free relative clause* (Eisenberg 1986).

3 Theoretical Assumptions

Standard context free phrase structure grammars would indeed generate center self-emedded structures like (2) and render them grammatical. It was early noted, though, e.g. by Chomsky (1959), that humans have difficulty understanding center embedded structures. The existence of alternative syntactic constructions like right extrapositions can therefore be viewed as emerging from a universal principle of preventing center embeddings or restricting them to a certain depth (Köhler 1999).

Many alternative explanations have been provided to explain humans' parsing difficulties with center embedding. Lewis (1996) discusses some of them. In his model, center embedded relative clauses are predicted to be hard due to memory interference effects between stacked preverbal NPs that cannot be integrated into a coherent parse tree before the verb is encountered, and therefore have to be stored in short term memory. He predicts center embeddings are incomprehensible only in those cases where more than two NPs before the occurence of the verb receive the same grammatical function.

In contrast, Hawkins (1994) does not consider the internal structure of embedded material at all but reduces all the factors that may contribute to difficulties in understanding to one simple variable, namely phrase length in terms of the number of word forms. The quality of a preterminal node is simply the number of terminal nodes from the beginning of the phrase to the first terminal of the last immediate constituent, divided by the number of immediate constituent nodes of the phrase. The overall quality of the sentence is the sum of the quality of its nodes. For relative clauses, Hawkins's principle of Early Immediate Constituents (EIC) predicts a) that shorter center embedded relative clauses should be more acceptable than longer ones and b) that extrapositions should be more acceptable if their extraposition distance is small. Thus, a) and b) are competing principles of locality that predict a high memory load when the processing of a phrase has to be delayed until intervening material has been processed.

4 Data and Methods

In order to test hypotheses on complex syntactic constructions, syntactically annotated corpora (treebanks) are needed. The NEGRA Millennium corpus (Negra) is a treebank comprising 20,571 sentences semi-manually annotated at the University of the Saarland. Word forms are POS-tagged according to the Stuttgart Tübingen tagset (Schiller et al. 1999). The syntactic structure is annotated according to an annotation scheme that recognizes formal node tags (phrase labels like *sentence*, *noun phrase*) and functional edge labels (like *subject*, *post-modification*, *relative clause*, *clausal complement*). The annotation scheme allows for crossing phrase-structure edges to encode discontinouous constituents. The Negra corpus is a sub-corpus of the 2.4 million sentence *Frankfurter Rundschau* newspaper corpus (Rundschau).

Since center embedded relative clauses are quite rare a phenomenon, the data from Negra was not sufficient. Therefore, we heuristically searched a statistically POS-tagged version of the *Frankfurter Rundschau* corpus for possible center embeddings and annotated the findings according to the TIGER annotation scheme. The heuristics searched for at least two occurences of commas followed by an optional preposition and a potential relative pronoun in the sentence. Since word forms that are potential relative pronouns are highly ambiguous between determiner and relative pronoun in German, manual filtering was necessary. At the present point of time, we cannot yet estimate recall and precission accurately, but it is clear that further work is necessary to increase the recall.

The resulting data was automatically analysed with C, Perl and Awk tools operating on the data structures representing the syntactic trees. For each sentence, a characterization of all the relative clauses it contained was generated fully automatically. The parameters were then filtered and statistically evaluated. Among the parameters were the **embedding depth** within the top level S (0, 1, or 2), the **relative pronoun** together with a detailed description of its tag and its function within the clause, the kind of **antecedent phrase** (NP, PP or *free relative clause*), the kind of **embedding** (RE, CE, EX), the **length** of superordinated and subordinated clause and the **(potential) extraposition distance** (cf. section 5.2).

5 Results

Of 20,571 sentences in Negra, 2352 (11.4%) contained at least one relative clause, and there were 2389 relative clauses in total. 61 sentences (0.3%) contained a relative clause embedding, i.e. two or more relative clauses of which at least two were embedded into each other. 1 of these sentences contained a double RC self-embedding, 1 sentence contained a conjunction of two embedded RCs clauses within another RC. These two instances will be ignored in the rest of the paper. Of the remaining 59 RCs of embedding depth 2, 39 (66%) were extraposed from their matrix RC, 13 (22%) were center embedded, and 7 (12%) were "right embedded", i.e. involved constructions where more material than the relative clause was extraposed. Table 16.1 shows the type of embedding of the superordinated RC in the matrix clause (*super*), and of the subordinated RC in the superordinated RC (*sub*), and provides the counts for each class in detail.

	super			Σ
	RE	EX	CE	59
RE	4	2	1	7
sub EX	25	11	3	39
CE	7	3	3	13

Table 16.1: Types of all RC self-embeddings in Negra

The heuristical search in the unannotated *Frankfurter Rundschau* corpus yielded 62 center embedded RCs. 14 of these were CE in CE embeddings, 28 CE in RE embeddings and 17 CE in EX embeddings. There were 3 more complex constructions involving CE relative clauses.

A corpus of all CE relative clauses was compiled by merging the CE relative clauses in Negra with those from *Frankfurter Rundschau*, which yielded 68 CE relative clauses in total. Since we do not yet have more data from the *Rundschau* corpus, we ignored the type of embedding of the self-embedded RC within the top level S for the compilation of the corpus of center embeddings.

In analogy, a corpus of non-center embedded RCs (nonCE) was computed. It included the 39 extrapositions and the 7 right embeddings from Negra. Figure 16.1 shows the structure of the data.

5.1 Clause Length

Based on the principle of Early Immediate Constituents (EIC, Hawkins (1994)), we expected to find a preference for the extraposition of longer phrases, while shorter phrases are expected to appear center embedded more often, because they would not delay the processing of their matrix clause for very long.

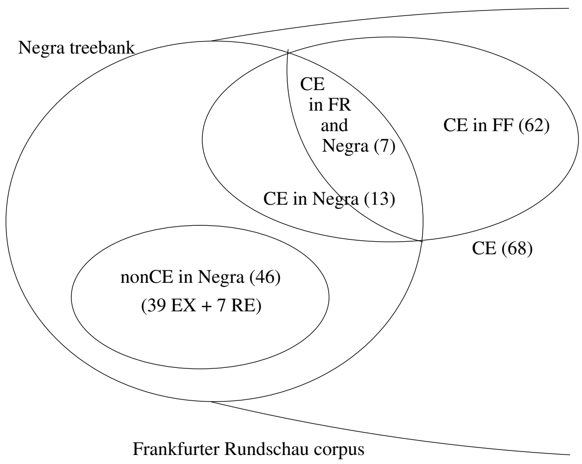


Figure 16.1: Relations between corpora used

As a comparison, we computed the overall distribution of relative clause length in the Negra corpus, i.e. the length of all 2389 relative clauses in the Negra corpus, be they embedded in other clauses or not. Within Negra, we found an average relative clause length of 10.2. We compared these results to all 68 center embedded relative clauses in the corpus of center embeddings (CE sub-corpus in figure 16.1) as well as with all 46 non-center embedded relative clauses from Negra (nonCE sub-corpus). Note that extrapositions and "right embeddings" were treated as one class on the base of the arguments under section 2.

Table 16.2 shows the frequencies of relative clauses as a function of length in the corpus of center embeddings (CE) and in the corpus of extrapositions and right embeddings (nonCE) together with the overall distribution of relative clause length in Negra.

length	2	3	4	5–6	7–10	11–14	15–24	25–47	Total	∅
Negra	.00	2.2	5.0	19.3	36.7	20.4	13.3	2.9	2389	10.2
nonCE	0	2.1	2.1	23.4	38.3	17.0	17.0	0	46	9.4
CE	0	7.4	10.3	35.3	25.0	20.6	1.5	0	68	7.5

Table 16.2: Distribution of RC length for all RCs in Negra, and of center embedded RCs and non center embedded RCs of depth 2 (in %)

There is no significant difference between the distribution of non-center embedded RCs we investigated and the overall distribution in Negra ($\chi^2_{(df=7)} = 5.09$). That is, extraposed or right embedded RCs are quite representative in their length of RCs in total. However, there is a significant difference between the distributions of center embedded RCs and the distribution in Negra ($\chi^2_{(df=10)} = 34.0$; $p < 0.001$). There is also a significant difference between the distribution of CE and nonCE ($\chi^2_{(df=6)} = 68.9;\ p < 0.001$).

We conclude that a subordinated RC in a relative clause extraposition is not significantly shorter than the average of relative clauses in total, namely around 10 word forms. In contrast, a subordinated RC center embedded within another relative clause is significantly shorter, namely about 8 word forms.

5.2 Extraposition Distance

One of the parameters that were fully automatically generated from the corpus data is the *(potential) extraposition distance* of a relative clause. In the case of an extraposition, this is the number of word forms between the end of the annotated antecedent NP and the relative clause (*ex*). In example (3) above, e.g., the RC is extraposed over the two word forms *festgesetzt war* ($ex = 2$). In the case of a center embedding, it is the number of remaining word forms of the superordinated clause after the end of the subordinated clause (*rd*). Note that potentially every center embedding can be transformed into an extraposition and vice versa, and that always $rd = ex$, as in example (2).

A hypothesis following from Hawkins (1994) is that extrapositions over a small number of word forms are more acceptable than those over a far distance. There is striking evidence for this hypothesis. In fact, extrapositions over a distance of more than three word forms do not occur in subordinated embedded relative clauses at all. In contrast, there is massive data for center embeddings whose potential extraposition distance is between 4 and 17. Table 16.3 shows the distributions in detail. Note that RE embeddings were ignored in this analysis.

distance	1	2	3	4	5	6	7	8	9	10–13	14-17	total
CE rd	11	13	8	12	6	6	2	2	2	4	2	68
EX ex	24	11	4									39

Table 16.3: Distribution of (potential) extraposition distance of center embedded and extraposed RCs of embedding depth 2

As can be seen from the table, relative clause embeddings with a potential extraposition distance between 1 and 3 do occur center embedded as well as extraposed. If center embedding is assumed to be the preferred option for relative clauses with a potential extraposition distance greater than 3, but center embeddings and extrapositions are both possible for RCs with small potential extraposition distance, one would expect length differences not only between EX and CE but also between “close” CEs and “far” CEs.

The analysis confirmed this hypothesis. While extrapositions are 9.4 word forms long on average (cmp. above, table 16.2), “far” center embeddings (whose potential extraposition distance rd is greater than 3) are 8.6

word forms long, and "close" center embeddings ($rd \leq 3$) are 5.7 word forms long (The χ^2 values reached p¡0.001 for EX vs. close CE, $p < 0.02$ for EX vs. far CE, and $p = 0.004$ for far CE vs. close CE).

The data may predict that not all center embeddings lead to problems, but that especially those center embeddings are acceptable that would lead to large extraposition distances if the relative clause was extraposed. A corpus example of an acceptable double center embedding is (5). Sentences with a small (potential) extraposition distance, on the other hand, occur center embedded in, as well as extraposed from their matrix RC. Examples for both cases are provided in (6) and (7).

(5) [$_S$ *Der Regierungschef*$_{\mathsf{NN}}$, [$_{RC}$ *der*$_{\mathsf{PREL}}$ *sich wegen Pensions- und*
The head of government who himself due to pensions and
Ausgleichszahlungen$_{\mathsf{NN}}$, [$_{RC}$ *die*$_{\mathsf{PREL}}$ *sein Gehalt jahrelang monatlich um*
compensations which his salary for years monthly by
einige tausend Mark aufstockten$_{\mathsf{VFIN}}$], *seit zwei Wochen unangenehmen*
some thousand marks increased for two weaks embarrassing
Fragen stellen$_{\mathsf{VINF}}$ *muß*$_{\mathsf{VFIN}}$], *forderte*$_{\mathsf{VFIN}}$ *den Jubel und die*
questions answer must called for the cheering and the
unkritische Solidarität seiner Genossen im Oskar-Land ein].
uncritical solidarity of his comrades in Oskar land (verb particle)
"The head of government, who has been exposed for two weaks to anwering embarrassing questions about pensions and compensation money that have been increasing his monthly salary by some thousand marks for years called for his comrades' applause and uncritical solidarity in the 'Oskar' territory." *(far CE under far CE)*

(6) [$_S$Im selben Zuge soll der Rabatt$_{\mathsf{NN}}$ von 20 Pfennig pro Kubikmeter, [$_{RC}$der$_{\mathsf{PREL}}$ bisher denjenigen Großverbrauchern$_{\mathsf{NN}}$, [$_{RC}$die$_{\mathsf{PREL}}$ mehr als 10.001 Kubikmeter im Jahr verbrauchen$_{\mathsf{VFIN}}$], zugestanden$_{\mathsf{VPART}}$ wurde$_{\mathsf{VFIN}}$], entfallen$_{\mathsf{VFIN}}$]. *(short CE under short CE)*

(7) [$_S$Dieser Fonds wird$_{\mathsf{VFIN}}$ beispielsweise mit dem Flaschenpfand$_{\mathsf{NN}}$ gefüllt$_{\mathsf{VPART}}$, [$_{RC}$das$_{\mathsf{PREL}}$ Annemarie Roth für die Flaschen$_{\mathsf{NN}}$ erlöst, [$_{RC}$die$_{\mathsf{PREL}}$ sie bei den regelmäßigen Spaziergängen im Grüneburgpark findet]]]. *(short EX under short EX)*

5.3 Structural Parallelism or Anti-Parallelism

If more complex categories than S or NP are assumed for syntactic descriptions, the concept of self-embedding becomes unclear: there might be differences between an object RC embedding a subject RC, for example. Also Lewis (1996) would predict differences in acceptability depending on the noun phrases involved. As a matter of fact, Lewis (1996) predicts an influence of *all* noun phrases occuring within the matrix clause and all embedded clauses. For reasons of simplicity, we concentrated on two factors only: the antecedent NPs and the phrase containing the relative pronoun. We considered these variables as possible criterions of structure similarity of the clauses involved and hypothesized there might be a trend towards anti-parallelism. There is still not enough data for more variables to be investigated.

Within the Negra corpus, there is a ratio of roughly 70% antecedent NPs and 30% antecedent PPs of all relative clauses in total. Unexpectedly, embedded relative clauses have 57% antecedent NPs and 43% antecedent PPs, be they center embedded or not ($\chi^2_{(df=1)} = 9.34$; $p = 0.002$). There is no significant difference between CE and nonCE.

The hypothesis that the distribution of the antecedent NP of the subordinated RC might depend on the antecedent NP of the superordinated RC could be falsified: There are four possible classes (RC with NP or PP antecedent embedded within RC with NP or PP antecedent), whose distribution does not differ significantly between CE and nonCE.

Our second criterion of internal clause structure was the function of the relative pronoun within the RC. Again, the distribution of RC pronoun functions within the entire Negra corpus was automatically computed as a comparison. Table 16.4 shows descriptions of the most frequent relative pronoun functions in the RC. Note that subject RCs are by far the most frequent class of relative clauses in German, and that object RCs are quite rare.

64%	subject	3%	modifying wh-pronoun
11%	within adverbial PP	2%	attributive pronoun in NP
10%	accusative object	10%	other

Table 16.4: Function of relative pronoun within all relative clauses in Negra

Significance tests were possible for the main classes *subject relative, object relative, others* only. There is only a marginal difference in their distributions between nonCE and Negra ($\chi^2_{(df=2)} = 5.1$; $p = 0.07$). In contrast, there are significant differences between the distributions of CE and of Negra ($\chi^2_{(df=2)} = 21.9$; $p < 0.0001$), and between nonCE and CE ($\chi^2_{(df=2)} = 43$; $p < 0.0001$). Center embedded RCs were significantly more often subject RCs (74%) than nonCEs (45%) or RCs in general (58%).

While center embedded RCs may have a tendency to be subject RCs in general, it could be shown that it does not have any influence on the type of subordinated relative clause whether the superordinated relative clause is a subject RC, an object RC or any other type of RC. The matrix clauses of subject RCs are also subject RCs in 63% of the cases and others in the remaining 37%. This is true of center embedded RCs as well as of non-center embedded RCs ($\chi^2_{(df=1)} = 2.9 \cdot 10^{-4}$).

In summary, the hypotheses that there might be a dependency between the internal structure of the superodinated RC and the subordinated RC could be falsified. For both variables considered, antecedent and relative pronoun function, such a dependency was not significant. There was neither a tendency towards anti-parallelism between embedding RC and embedded RC, nor towards parallelism.

There were two unpredicted effects: an affinity of embedded (center embedded and extraposed) RCs to have PP antecedents, and a tendency of center embedded relative clauses to be subject RCs. Recall, however, that while all non-center embeddings were taken from Negra, center embeddings were heuristically extracted from the *Frankfurter Rundschau* corpus to enhance data. Since the recall was low, it may well be argued that the difference is due to properties of the extraction heuristics. Further research on increasing the recall from the Rundschau corpus may help to clarify this open issue.

6 Discussion

The data is compatible with models that explain the distribution of center embeddings and extrapositions as following from restrictions on short term memory. Two parameters, embedded clause length and extraposition distance that are predicted to be relevant by Hawkins's EIC principle were confirmed. However, extraposition distance seems to be the more prominent factor, whereas clause length is of secondary importance. The data may also be interpretable within related models, such as Gibson's SPLT theory, which assumes higher "integration cost" with increasing extraposition distance and/or clause length.

Predictions following from a phrase structure architecture assuming that the internal clause structure of the participating RCs matters could be falsified. Assumptions that parallelism or anti-parallelism between the RCs might simplify processing could not be confirmed with the present data. In order to test predictions of a more elaborated interference model like Lewis (1996), more data is needed.

The question may be raised whether the material investigated – newspaper texts – reflects production or perception preferences. Konieczny (2000), for instance, found significant acceptability assymetries between perception and production data.

7 Conclusion and Future Work

The paper demonstrated how a syntactically annotated corpus together with appropriate querying and processing tools can be used to semi-automatically prove or disprove hypotheses derived from a model of performance. This technique allows for an easy repetition of the study on different data or with additional parameters (e.g., restrictive vs. non-restrictive RC).

The data can be interpreted as following from a tradeoff between different principles of minimizing memory effort during parsing (or generation, respectively). Such principles are the principle of Early Immediate Constituents and constraints on embedding depth and phrase length (see e.g. Köhler (1999) for a wider framework). The main results are the quantification of the tendency to extrapose over short distances only, and to center embed especially shorter phrases whose potential extraposition distance is short. We did not find any evidence for the hypothesis that structural differences between superordinated and subordinated RC might have an impact.

The study confirms and supplements a prior study by Uszkoreit et al. (1998) with more data (20,000 instead of 12,000 sentences) and, especially, a shift towards self-embeddings. For the future, an automatic calculation of EIC scores is planned. A cross-check with functional equivalents of relative clauses (Köhler 1999) and psycholinguistic acceptability tests are two possible ways to supplement the study.

References

Brants, T. et al. (2000). TIGER Annotationsschema. Saarbrücken: Universität des Saarlandes, www.coli.uni-sb.de/cl/projects/tiger.

Chomsky, N. (1959). On Certain Formal Properties of Grammars. *Information and Control* (2), 137–167.

Eisenberg, P. (1986). *Grundriß der deutschen Grammatik*. Stuttgart: Metzler.

Gibson, E. (1998). Linguistic Complexity: Locality of Syntactic Dependencies. *Cognition* (68), 1–76.

Hawkins, J. A. (1994). *A Performance Theory of Order and Constituency*. Cambridge: Cambridge UP.

Köhler, R. (1999). Syntactic Structures: Properties and Interrelations. *Journal of Quantitative Linguistics 6*(1), 46–57.

Konieczny, L. (2000). Locality and Parsing Complexity. *Journal of Psycholinguistic Research 29*(6), 627–645.

Lewis, R. L. (1996). Interference in Short-term Memory: The Magical Number Two (or Three) in Sentence Processing. *Journal of Psycholinguistic Research 25*, 93–115.

Schiller, A., S. Teufel, C. Stöckert, and C. Thielen (1999). Guidelines für das Tagging: Kleines und großes Tagset. Stuttgart: Institut für maschinelle Sprachverarbeitung.

Uszkoreit, H., T. Brants, et al. (1998). Studien zur performanzorientierten Linguistik. *Kognitionswissenschaft 7*(3), 129–133.

Automating Proof in Non-standard Analysis

EWEN MACLEAN
Division of Informatics, University of Edinburgh
ewenm@dai.ed.ac.uk

ABSTRACT. Non-standard analysis provides a framework to carry out formally calculus proofs in a much more intuitive manner than in the $\epsilon - \delta$ formulation of Weierstraß. This paper introduces the notions of proof-planning and rippling, and gives an example showing how they can be applied to non-standard analysis to produce readable proofs. We present the related work in the area and give an outline of the proposed work.

1 Introduction

The work proposed in this project is a combination of proof-planning (Bundy 1988) and an area of mathematics known as non-standard analysis (Robinson 1966). Proof-planning is a technique which helps to automate proof by working at an abstracted level. Non-standard analysis allows intuitive calculus proofs to be performed with the assurance of correctness thanks to a solid logical construction of the number system that underlies it– namely the hyperreals. The aim of this work is to develop proof-plans which capture the common patterns of reasoning that take place when proving theorems in non-standard analysis, and thereby to automate their proof. The work presented here has been completed in the $\lambda Clam$ higher order proof-planner (Richardson et al. 2000)

We start by giving a brief overview of some concepts in both non-standard analysis and proof-planning; we then present an example proof and discuss some of the related work; finally we highlight possible future directions for our development.

2 Non-standard analysis

Non-standard analysis is a tool which provides an intuitive yet rigorous alternative to Weierstraß's tricky $\epsilon - \delta$ proofs. Abraham Robinson (Robinson

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

1966) brought together a number of ideas from mathematical logic to come up with a theory of analysis in the hyperreal domain. By formally introducing an "infinitely close" relation, definitions and proofs become simpler and more intuitive.

As an example of how a definition in non-standard analysis can be simpler and more intuitive than its standard counterpart, consider the standard and the non-standard definitions of a limit.

Definition 1. *The limit l, of a function f at a point a is defined as:*

$$\lim_{x \to a} f(x) = l \;\equiv\; (\forall \epsilon > 0)(\exists \delta > 0)\forall x.\, 0 < |x - a| < \delta \to |f(x) - l| < \epsilon$$

Proofs which make use of this definition are usually very difficult to automate, on account of the alternating quantifiers, which mean that instantiations for δ have to guessed early on in the proof. Theorem 1 gives us an alternative characterisation of a limit making use of the infinitely close relation $\approx$.

Theorem 1. *The limit l, of a function f at a point a can equivalently be defined in non-standard analysis by:*

$$\lim_{x \to a} f(x) = l \quad \Longleftrightarrow \quad \forall x.\; x \approx {}^*a \land x \neq {}^*a \to f^*(x) \approx {}^*l$$

As this theorem expresses the notion of limit over the hyperreals, denoted ${}^*\mathbb{R}$, a few points are worth mentioning: the function f^* corresponds to the original function f extended to accept hyperreal arguments whereas ${}^*a$ corresponds to the embedding of real value a in the hyperreals. These notions are introduced for correctness and will not be examined further here. Only an intuitive understanding of their behaviour is needed for the rest of this paper although the interest reader may find out more by examining (Robinson 1966).

This theorem gives us a simple non-standard characterisation for a limit which formalises the intuition that underlies the $\epsilon - \delta$ formulation of a limit. For a proof of theorem 1 the reader is urged to consult (Fleuriot and Paulson 2000), for example. We give a demonstration of the intuitive nature of proofs involving limits in section 4.

3 Proof-planning

Proof planning (Bundy 1988) is a technique for devising an overall plan for a proof, which can then be used to guide the proof search itself. A proof-plan consists of methods and critics. The methods embody common patterns within proofs, such as the use of induction, and the associated tactics carry out these methods explicitly in the object level prover. For each conjecture a precise proof-plan is built, but a general form of a proof-plan can be developed for certain classes of conjecture. As described by (Ireland 1992), a method in proof planning has a number of "slots" assigned to it which correspond to a name by which it is recognised, an input which is matched against the current goal, and a set of preconditions which determine whether the method is applicable to the current goal. The method also has a set of effects which define what happens to the goal after it has been applied, and a tactic which controls the object level prover, in which the plan can be executed. Finally, it has an output which is the result of applying the effects to the current goal.

3.1 Critics

The use of inductive theorem provers such as the Boyer-Moore Theorem Prover (Boyer and Moore 1979) shows that failed proofs can provide useful information about failure, and hence can help to suggest ways to render proofs successful. One piece of useful information that could come out of a failed proof attempt is a suggestion of how to pre-process the conjecture in some way to help the proof succeed. In the context of proof-planning, a critic is a component which analyses this failure. Ireland and others (Ireland 1992) have done considerable work in this area, which the interested reader is urged to consult.

3.2 Rippling

Rippling is a heuristic used in proof-planning for guiding the proof search. It was initially motivated by Aubin's observation (Aubin 1976) on how terms introduced by induction are affected by rewriting. Bundy formalised this idea into a theory of annotated rewriting (Bundy et al. 1993), and a formal calculus has been developed from which one can prove termination (Basin and Walsh 1996). Crucially, it has been shown that rippling can be extended to non-inductive settings by Yoshida (Yoshida et al. 1994), Walsh (Walsh et al. 1992) and Hutter (Hutter 1997). This is of importance to the current work.

Rippling annotates the conclusion to indicate the difference between the hypotheses and conclusion. The idea is that the extra structure which exists in the conclusion can be annotated, and one can use this annotation to reason about how the proof is proceeding. The basic properties of an annotation are directed wave fronts, wave holes and sinks. Wave fronts, wave holes and sinks are the meta annotation which encapsulate terms. Wave fronts must include wave holes. Sinks correspond to universally quantified variables in the hypotheses. We shall now briefly explain the important properties and definitions of rippling. For a more formal definition of rippling see (Basin and Walsh 1996).

Well annotated terms All wave fronts must contain at least one wave hole, and the term within the wave hole is well annotated. All the terms within the wave front, but outside the wave hole are unannotated. Unannotated terms are well annotated.

Skeletons The skeleton of an expression is the set of all expressions that are built up from the unannotated term and the possible wave holes within each wave front.

Erasure The erasure of a well annotated term is the term without any annotation.

For example consider the expression $f(x+y)$. Examples of possible annotations which render it a well annotated term are:

$$f(x+y),\ \boxed{f(\boxed{x})}^{\uparrow}+y,\ \boxed{f(\boxed{x}+\boxed{y})}^{\uparrow},\ \boxed{f(\boxed{x+y})}^{\uparrow},\ \boxed{f(\boxed{x})}^{\downarrow}+y,\ \boxed{f(\boxed{x}+\boxed{y})}^{\downarrow}$$

The erasure of all of these terms is simply the original expression $f(x+y)$. As an example, the skeleton of the term $\boxed{f(\boxed{x})}^{\uparrow}+y$ is $x+y$, because this is the expression built up from the term in the wave hole and the unannotated part. Similarly the skeleton of $\boxed{f(\boxed{x}+\boxed{y})}^{\uparrow}$ is $\{x, y\}$, because x and y are both wave holes of the same wave front, so the set of possibilities is constructed. The reason for these definitions will become clear when considering the properties of rippling.

Standard rewrite rules can be annotated to fit in with the rippling scheme. Annotated rewrite rules, called wave rules, have to be both skeleton preserving and measure decreasing. We will not describe here in detail the arguments for termination of rippling. Skeleton preservation means that for a rewrite rule $L \Rightarrow R$, the skeleton of L must be a superset of the skeleton of R. For example

$$\begin{aligned}\boxed{f(\boxed{x})}^{\uparrow}+y &\Rightarrow \boxed{f(\boxed{x+y})}^{\uparrow}\\ \boxed{f(\boxed{x+y})}^{\downarrow} &\Rightarrow \boxed{f(\boxed{x})}^{\downarrow}+y\end{aligned}$$

are wave rules because the skeleton, $x+y$ is preserved under the rewrite. It is possible to see that when rewriting an expression the rewrites move the wavefronts around the expression, encapsulating more or less of the term structure according to their direction. The arrows on the wave fronts can be thought of as describing whether the wave front surrounds more or less of the term structure under rewriting. Wave rules that maintain the direction of the arrow are called longitudinal, and those that change the direction of the arrow are called transverse. Transverse rippling is only allowed if the direction of the wave rules moves from "out" to "in" (up to down). When initially annotating a term the wave fronts are all directed "out". This condition helps one to think about termination; intuitively one can think of the measure monotonically decreasing because a wave front can move out as far as it can go and then move back in if it is desirable, but it cannot move back out once it has started to "ripple in". Wave rules match expressions if the annotation matches with a subterm within the expression.

The properties of rippling are

Well-formedness If a well annotated term is rewritten by a wave-rule then the result is a well annotated term.

Skeleton preserving The result of applying wave rule to a term has a skeleton which is a subset of the original term.

Correctness When a term is rewritten by a wave rule the erasure of the rewrite corresponds to a rewrite in the unannotated theory.

Termination Rippling terminates.

This has been a brief overview of the calculus of rippling. For a fuller description see (Basin and Walsh 1996), and for a description of the existing work on colouring terms for equational rewriting, which we do not use, see (Hutter 1997).

4 A brief case study

This section presents a plan which has been generated automatically by $\lambda Clam$. Other proofs which have been performed include $LIM+$, the chain rule for differentiation and the product rule.

A useful example of a limit theorem to consider in the context of non-standard analysis and proof-planning is that of $LIM\times$. This theorem states that the product of the limits of two real valued functions is equal to the limit of the product of the functions. This conjecture is difficult to automate in standard analysis (Melis et al. 2000), but we show here that the structure of its non-standard counterpart is very simple. This example proof demonstrates the advantage of using non-standard analysis over standard analysis, and shows how existing proof-planning machinery can be used to model the reasoning patterns in non-standard proofs. The standard version of this proof is challenging on paper, and is extremely hard to automate. The $\epsilon - \delta$ formulation involves quantifiers which are arranged in such a way that a guess has to made about the instantiation for quantified variables before enough information is available. The conjecture in standard analysis is stated as

$$\lim_{x \to a} f(x) = l_f \wedge \lim_{x \to a} g(x) = l_g \vdash \lim_{x \to a} f(x) \times g(x) = l_f \times l_g$$

The hypotheses to the conjecture become:

$$c, l_f, l_g : \mathbb{R} \tag{17.1}$$

$$f, g : \mathbb{R} \to \mathbb{R} \tag{17.2}$$

$$\forall x.\ x \approx {}^*c \wedge x \neq {}^*c \to f^*(x) \approx {}^*l_f \tag{17.3}$$

$$\forall x.\ x \approx {}^*c \wedge x \neq {}^*c \to g^*(x) \approx {}^*l_g \tag{17.4}$$

using the non-standard characterisation for a limit. Hypotheses (17.1) and (17.2) describe the typing information of the constants.

The hypotheses (17.3) and (17.4) can now be embedded in the conclusion. This yields the annotated conclusion

$$\boxed{x} \approx {}^*c \wedge \boxed{x} \neq {}^*c \quad \to \quad \boxed{\underline{f^*(\boxed{x})}_{red} \times \underline{g^*(\boxed{x})}_{blue}}^{\uparrow} \approx \boxed{\underline{{}^*l_f}_{red} \times \underline{{}^*l_g}_{blue}}^{\uparrow}$$

The terms belonging to different hypotheses inside the wave fronts are given different colours. The shading around the variable x in the conclusion refers to positions which correspond to universally quantified variables in the hypotheses. In rippling terminology these are referred to as *sinks*. The wave rules that are available to the system are the following:

$$[finite(Y) \wedge finite(B)] \quad \boxed{\underline{X}_{red} \times \underline{A}_{blue}}^{\uparrow} \approx \boxed{\underline{Y}_{red} \times \underline{B}_{blue}}^{\uparrow} \quad \Rightarrow \quad \boxed{\underline{X \approx Y}_{red} \wedge \underline{A \approx B}_{blue}}^{\uparrow} \tag{17.5}$$

$$A \to \boxed{\underline{B}_{red} \wedge \underline{C}_{blue}}^{\uparrow} \quad \Rightarrow \quad \boxed{\underline{A \to B}_{red} \wedge \underline{A \to C}_{blue}}^{\uparrow} \tag{17.6}$$

Here when we write $A \Rightarrow B$ we mean that A rewrites to B, and hence it is true that $B \to A$. The side conditions involving the predicate $finite$ refer to all hyperreal numbers which are not infinite— for example real numbers.

Wave rule (17.5) represents the continuity of $\times$, which is itself a complicated theorem to prove in standard analysis, and renders this proof very simple indeed. In non-standard analysis, it is a challenging problem involving mainly equational reasoning, but is much simpler than its standard counterpart as we discuss in section 4.1.

By inspecting the annotation on the conclusion and that on the rewrite rules, the general idea behind rippling can be seen. The wave fronts should increase until the terms in the wave holes can match with the hypotheses. After application of each of these rules, the conclusion becomes

$$\boxed{\underline{\boxed{x} \approx {}^*c \wedge \boxed{x} \neq {}^*c \to f^*(\boxed{x}) \approx {}^*l_f}_{red} \wedge \underline{\boxed{x} \approx {}^*c \wedge \boxed{x} \neq {}^*c \to g^*(\boxed{x}) \approx {}^*l_g}_{blue}}^{\uparrow}$$

at which point the holes are instances of the hypotheses as required. Now the conclusion contains instances of the hypotheses, and the plan of the proof is complete.

Part of the output from the plan given by $\lambda Clam$ showing wave rules (17.1) and (17.2) is given below. The formatting has been changed slightly by hand so as not to take up too much space.

```
Attempting...
cond_wave_method _841981 cont_times
cond_wave_method outward cont_times
succeeded

X:hyperreal, c:real, f:real → real, g:real → real, l_f:real, l_g:real
∀  Y:hyperreal. ((Y ≈ emb  c) ∧ ¬ (Y = emb c)) →  ext f Y ≈ emb l_f
∀  Y:hyperreal. ((Y ≈ emb  c) ∧ ¬ (Y = emb c)) →  ext g Y ≈ emb l_g
⊢  ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext f X ≈ emb l_f) ∧ (ext g X ≈ emb l_g)

1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: nil                   %%current plan node

Attempting...
patch_meth (wave_method outward _818076) wave_critic_strat
patch_meth (wave_method outward prop_wr1) wave_critic_strat
succeeded

X:hyperreal, c:real, f:real → real, g:real → real, l_f:real, l_g:real
∀ Y:hyperreal. ((Y ≈ emb c) ∧ ¬ (Y = emb c)) → ext f Y ≈ emb l_f
∀ Y:hyperreal. ((Y ≈ emb c) ∧ ¬ (Y = emb c)) → ext g Y ≈ emb l_g
⊢  ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext f X ≈ emb l_f) ∧
   ((X ≈ emb c) ∧ ¬ (X = emb c)) → (ext g X ≈ emb l_g)

1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: 1 :: nil
```

Here `emb` denotes the embedding of a real number in the hyperreals, and `ext` the non-standard extension of a real function.

The non-standard proof follows a simple pattern of reasoning which proof-planning is able to model. In this example, we have introduced wave rule (17.5) to the system so that the simplicity of the overall structure is apparent. The annotation gives the proof extra information which can be used to speculate lemmas, and proof transformations. For example, consider what would happen if the wave rule (17.5) had not been added to the system. The rewriting mechanism would then fail, prompting a critic to respond to this failure, and suggest a new wave rule according to the hypotheses. See section 4.1 for a discussion of this wave rule. This would enable the proof to automatically recover and go through even in the absence of the rewrite rule. The advantage of the annotation in this case, is that it can be made explicit which terms belong to which hypotheses, and hence which should be brought closer together.

4.1 The continuity of $\times$

The following conjecture represents the lemma which has to speculated and whose proof must be planned automatically in order to justify the claim that the construction of a plan for the proof $LIM\times$ has been automated. This work is currently in progress.

$$\vdash \forall a, b, x, y.\ \ M \wedge x \approx y \wedge a \approx b \ \ \rightarrow \ \ x \times a \approx y \times b$$

In this case the meta-variable M has to be instantiated to the conditions of the rule. After the usual $\forall$ and implication introduction, we proceed as follows:

$$\vdash x \times a \approx y \times a \wedge y \times a \approx y \times b$$

from transitivity. We then speculate the following lemma:

$$M_1 \wedge x \approx y \vdash x \times a \approx y \times a$$

where M_1 is a new meta-variable which must be instantiated to the conditions. Now we use an arithmetic rule to yield the conclusion:

$$\vdash x \times a - y \times a \approx {}^*0$$

and use the distributive law to write this as:

$$\vdash (x - y) \times a \approx {}^*0$$

and finally rewrite this using the following axioms:

$$X \approx {}^*0 \wedge finite(Y) \rightarrow X \times Y \approx {}^*0$$
$${}^*0 + x = x$$
$$x + (-x) = {}^*0$$

to yield:

$$\vdash finite(a) \wedge x \approx y$$

which completes the proof, and M_1 is instantiated to the condition $finite(a)$.

Using this lemma we can proceed with the original proof. The conclusion was at the state:

$$x \times a \approx y \times a \wedge y \times a \approx y \times b$$

which can now be rewritten as:

$$finite(a) \wedge x \approx y \wedge finite(y) \wedge a \approx b$$

which completes the proof, with the instantiation of the meta-variable M as $finite(a) \wedge finite(y)$. Including the commutativity of $\times$ and the symmetry of $\approx$ in the original equation, we can also derive new conditions, meaning that the most general condition for M is $(finite(x) \vee finite(y)) \wedge (finite(a) \vee finite(b))$. The use of transitivity in this example is analogous to the use of the triangle inequality in the standard proofs.

5 Related work

A reasonable amount of work has already been completed in both standard and non-standard analysis. (Bledsoe et al. 1972) is the first piece of work on automating parts of standard analysis, and presents a resolution based theorem prover which automates certain standard analysis proofs, and presents certain heuristics such as the *limit heuristic* which are specific to analysis proofs. (Bledsoe and Ballantyne 1977) presents the same theorem prover modified to automate proofs in non-standard analysis. The goal of this work was to demonstrate how non-standard analysis allows mathematicians to prove theorems in analysis more easily than by using the $\epsilon - \delta$ formulation. The paper presents the methods employed in the prover, and explains how it handles the various types that occur in non-standard analysis. Their work uses a generous axiomatisation and the prover is able to prove simple results about standard parts of numbers, as well as more significant results such as $LIM+$, and even the Bolzano-Weierstrass conjecture that a continuous function on a compact set is uniformly continuous. Their proofs, however, are not as readable as those generated by our proof-planning approach, as they are driven by resolution, and the output is linear. Through proof-planning the tree structure of method applications in the proof plan can be seen, and hence the structure of the proof itself is clearer.

In the ΩMEGA proof-planner and mathematical assistant (Benzmüller et al. 1997), there has been much work done in developing proof-plans for limit theorems using what is referred to as "multiple strategies". The ΩMEGA proof-planner builds on Bledsoe's work by introducing a "complex estimate", and a constraint solver which carefully manages the introduction of meta-variables in order to calculate the difficult instantiations of the variables in standard proofs. The system is able to prove a substantial number of theorems at present, and a proof presentation system has been included in the system to make the proofs readable. The work is documented in (Melis et al. 2000) and (Melis and Siekmann 1999) for example.

The Mathpert system developed by Beeson is a mathematical assistant which is also capable of automatically proving properties about functions in standard analysis such as continuity (Beeson 1998). Often the proofs involve evaluating limits, and Beeson found that this evaluation could be simplified using of some axioms from non-standard analysis (Beeson 1995).

The ACL2 theorem prover originally did not include a theory for reasoning about conjectures over the reals, but Gamboa (Gamboa 1999) used Nelson's Internal Set Theory (Nelson 1977) to extend the prover to automatically prove theorems about real valued functions.

The most significant work that has been done in this area has been done by Fleuriot (Fleuriot and Paulson 2000) in the interactive theorem prover Isabelle/HOL. This work is the first to mechanise the construction of the hyperreals using the ultrapower construction used by Robinson to develop non-standard analysis. A substantial amount of real analysis has also been proved in the hyperreal domain using this construction. It also highlights that there are proofs in non-standard analysis, though simpler than their standard counterparts, which cannot be done automatically. We believe that our proposed approach should enable these to be planned automatically.

Other work in progress that is related to the current work is that of Heneveld (Heneveld et al. 2000). This work models the cognitive processes of mathematicians solving integration problems. The $\lambda Clam$ proof-planner is also being used to do this, and many of the rules used, such as integration by parts, can potentially be proved using the work proposed here.

6 Proposed work

The work we propose in this project involves generating plans for proofs in non-standard analysis automatically, and thence executing them in an object level theorem prover. The methodology is one of genericity and automation. We have compiled a set of minimal axioms for non-standard analysis from which to derive all of the rules we require, and we hope to build a generic proof-plan whose initial state will be constant, but whose eventual shape will be transformed by the addition of intelligent critics. In addition to this, we have introduced a method for unfolding axioms by introducing such notions as *field homomorphism*. For example, the reals and the hyperreals form a field, so it is sensible to express field axioms in such a way, that the axioms specific to each type can be generated in a generic manner. This is similar to the work done in Isabelle on *Axiomatic Type Classes* (Wenzel 2000).

Part of the work still yet to be completed involves the collection of a set of conjectures with which to test the system. The initial area of interest lies within *limit* and *continuity* conjectures, which have already been attempted in standard analysis, for example in the $\Omega MEGA$ system. Some of these proofs have already been performed as a trial in an earlier version of $\lambda Clam$. Also of interest are fundamental proofs about calculus such as the chain rule, the product rule, and the fundamental theorem of calculus. The former two have also been attempted in the earlier version of $\lambda Clam$ with some success.

7 Conclusion

The main aim of this project is to gain an understanding of the structure of proofs in non-standard analysis. Proof-planning allows this to be done in a declarative way by formally specifying plans. We hope that this will show that non-standard analysis is a better way of going about proving theorems about analysis in general, and also, perhaps, would also be a more sensible alternative for teaching purposes than the infamous $\epsilon - \delta$ proofs of standard analysis.

References

Aubin, R. (1976). *Mechanizing Structural Induction.* Ph. D. thesis, University of Edinburgh.

Basin, D. and T. Walsh (1996). A calculus for and termination of rippling. *Journal of Automated Reasoning 16*(1–2), 147–180.

Beeson, M. (1995). Using nonstandard analysis to verify the correctness of computations. *International Journal of Foundations of Computer Science 6*(3), 299–338.

Beeson, M. (1998). Automatic generation of epsilon-delta proofs of continuity. *Artificial Intelligence and Symbolic Computation*, 67–83. Lecture Notes in Artificial Intelligence No. 1476.

Benzmüller, C., L. Cheikhrouhou, D. Fehrer, A. Fiedler, X. Huang, M. Kerber, K. Kohlhase, A. Meier, E. Melis, W. Schaarschmidt, J. Siekmann, and V. Sorge (1997). Ωmega: Towards a Mathematical Assistant. In W. McCune (Ed.), *14th International Conference on Automated Deduction*, pp. 252–255. Springer-Verlag.

Bledsoe, W. W. and A. M. Ballantyne (1977). Automatic Proofs of Theorems in Analysis Using Nonstandard Techniques. *Association for Computing Machinery 24*(3), 353–374.

Bledsoe, W. W., R. S. Boyer, and W. H. Henneman (1972). Computer Proofs of Limit Theorems. *Artificial Intelligence 3*, 27–60.

Boyer, R. S. and J. S. Moore (1979). *A Computational Logic.* Academic Press. ACM monograph series.

Bundy, A. (1988). The Use of Explicit Plans to Guide Inductive Proofs. In R. Lusk and R. Overbeek (Eds.), *9th International Conference on Automated Deduction*, pp. 111–120. Springer-Verlag. Longer version available from Edinburgh as DAI Research Paper No. 349.

Bundy, A., A. Stevens, F. van Harmelen, A. Ireland, and A. Smaill (1993). Rippling: A Heuristic for Guiding Inductive Proofs. *Artificial Intelligence 62*, 185–253. Also available from Edinburgh as DAI Research Paper No. 567.

Fleuriot, J. D. and L. C. Paulson (2000). Mechanizing Nonstandard Real Analysis. *LMS Journal of Computation and Mathematics 3*, 140–190.

Gamboa, R. (1999). *Mechanically Verifying Real-Valued Algorithms in ACL2.* Ph. D. thesis, The University of Texas at Austin.

Heneveld, A., E. Maclean, A. Bundy, J. Fleuriot, and A. Smaill (2000). Towards a Formalisation of College Calculus. In *Proceedings of the 2000 Calculemus Symposium: Systems for Integrated Computation and Deduction.*

Hutter, D. (1997). Colouring Terms to Control Equational Reasoning. *Journal of Automated Reasoning 18*, 399–442.

Ireland, A. (1992). The Use of Planning Critics in Mechanizing Inductive Proofs. In A. Voronkov (Ed.), *International Conference on Logic Programming and Automated Reasoning – LPAR 92, St. Petersburg*, Lecture Notes in Artificial Intelligence No. 624, pp. 178–189. Springer-Verlag. Also available from Edinburgh as DAI Research Paper 592.

Melis, E. and J. Siekmann (1999). Knowledge-Based Proof Planning. *Artificial Intelligence 115*(1), 65–105.

Melis, E., J. Zimmer, and T. Müller (2000). Extensions of Constraint Solving for Proof Planning. *European Conference on Artificial Intelligence*.

Nelson, E. (1977). Internal set theory: A new approach to nonstandard analysis. *Bulletin American Mathematical Society 83*. Available from `http://www.math.princeton.edu/~nelson/books.html`.

Richardson, J., L. Dennis, J. Gow, and M. Jackson (2000). User/Programmer Manual for the $\lambda Clam$ proof planner.

Robinson, A. (1966). *Non-standard Analysis.* North-Holland Publishing Company, Amsterdam. Studies in Logic and the Foundations of Mathematics.

Walsh, T., A. Nunes, and A. Bundy (1992). The Use of Proof Plans to Sum Series. In D. Kapur (Ed.), *11th International Conference on Automated Deduction*, pp. 325–339. Springer Verlag. Lecture Notes in Computer Science No. 607. Also available from Edinburgh as DAI Research Paper 563.

Wenzel, M. (2000). Using Axiomatic Type Classes in Isabelle. Available from `http://isabelle.in.tum.de/doc/`.

Yoshida, T., A. Bundy, I. Green, T. Walsh, and D. Basin (1994). Coloured rippling: An extension of a theorem proving heuristic. In A. G. Cohn (Ed.), *Proceedings of ECAI-94*, pp. 85–89. John Wiley.

On LPetri nets

Kundan Misra

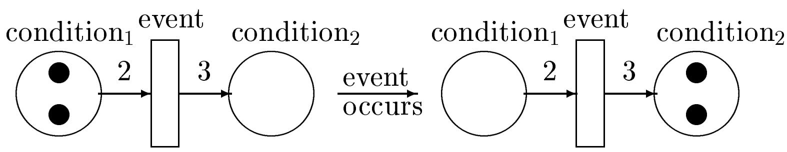
Department of Computer Science, University of Warwick, Coventry CV47AL, UK

kundan@dcs.warwick.ac.uk

Abstract. LPetri nets are coloured Petri nets where tokens are formulae from linear logic. LPetri nets are derived from linear logic Petri nets in Farwer (1996). LPetri nets are convenient for concisely modelling complex systems. A theory of simulations and intuitionistic linear logic specification language were developed for Petri nets in (Brown et al. 1991). We will extend these to LPetri nets.

1 Introduction

Petri nets are a system modelling schema that has been much used in many branches of engineering since the 1960s. A Petri net consists of conditions and events which are connected by directed, weighted arcs. Diagramatically, conditions are represented by circles and events by rectangles. Conditions contain tokens, and the "pre-condition" of an event is the number of tokens needed in each condition for the event to be enabled. The pre-condition of an event is indicated by the weights of the incoming arcs. An event can occur when its pre-condition is satisfied, causing its post-condition:



Coloured Petri nets (CPNs) allows tokens to be of different "colours". Each condition has a set of allowable colours called a *colour set*. Each condition contains a multisets of coloured tokens. (A *multiset* or "bag" is a set in which repetitions of elements is allowed.) Any event in a CPN may have a "guard function" associated. This allows the designer to specify complex criteria on the incoming arcs.

In (Farwer 1996), a type of CPN was devised in which the colour set of each condition is the set of linear logic (LL) formulae, so each LL formula is treated as a unique colour.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

The important characteristic of LL, which makes it well-suited to computer science, is that premises are treated as resources. This contrasts with classical logic, in that if we know that a premise holds true, then it is assumed to always be true. In a sense, classical logic provides an infinite "supply" of a true premise. For example, suppose $A \vdash B$ and $A \vdash C$. Then in LL, if we are given A we can have either B or C but not both.

LL is called a "substructural logic" because it lacks some of the rules of classical logic. In particular, LL does not have the rules of *contraction* and *weakening*. Omission of the contraction rule means that if two "copies" of the same premise are required, then one copy is not sufficient. For example, in LL if $A, A \vdash B$, then given only A, we cannot deduce B. On the other hand, in LL if we are given that $A, B \vdash D$, then we cannot say that $A, B, C \vdash D$. Rather, if we have $A, B \vdash D$, then we can only say $A, B, C \vdash D, C$. The paper (Girard 1987) is the detailed seminal work on LL. Also see (Troelstra 1993) for a more concise overview of linear logic.

Farwer calls the CPNs in which tokens are LL formulae *linear logic Petri nets* (LLPNs). We will consider a restricted form of LLPNs called *LPetri nets*, the difference being that LPetri nets do not have guard functions. The aim is to understand LPetri nets first, and advance to LLPNs in a later paper. Omitting the guard function gives a construct similar to standard Petri nets, making a categorical treatment more tractable.

Neither LPetri nets nor LLPNs fit the standard description of CPNs from (Jensen 1997), because they allow the tokens (LL formulae) to *interact* in the conditions under the proof rules of LL. So the marking of a net following the occurrence of an event is generally not the same as the tokens which determine whether the pre-condition of an event is satisfied. Suppose the LL formulae $J \multimap K$ and J enter a condition, b. Then J and $J \multimap K$ are "consumed" to produce K.

In (Brown et al. 1991), an intuitionistic linear logic (ILL) proof theory was developed for standard Petri nets, together with a categorical semantics. As in classical logic, ILL is LL without the law of the excluded middle. (Also see (Troelstra 1993) for an overview of ILL.) We will show that this proof theory can be extended from Petri nets to LPetri nets.

The diagram shows how earlier work relates to this paper:

$$
\begin{tikzcd}
 & \text{Petri nets} \arrow[r, "\text{Marti-Oliet and Meseguer}"] \arrow[d, "\text{this paper}"'] \arrow[dr, "\text{Brown, Gurr, de Paiva}"] & \text{categorical definition} \\
\text{LLPNs} \arrow[r, "\text{this paper}", "\supseteq"'] & \text{LPetri nets} \arrow[r, "\text{this paper}"] & \text{ILL proof theory} \\
 & \text{coloured Petri nets} \arrow[ul, "\text{Farwer}"] \arrow[u, "\text{this paper}"'] \arrow[ur, "\text{this paper}"'] &
\end{tikzcd}
$$

To motivate this work, we must show that LPetri nets can model real-world systems. We will do this with examples. To justify the categorical approach, (Brown et al. 1991) argue that: Structures derived from the category of Petri nets yields constructors on Petri nets that are required for a modular approach to Petri nets; if the category of Petri nets has an appropriate structure, we get a logic for reasoning about Petri nets; expressing models of concurrency as categories allows us to explore relationships between models by studying functors between the categories; and the generality often makes it easy to modify the structures being studied. Also, (Sassone 2000) says that the appeal of a categorical approach to Petri nets is that it "tends to devise neat algebraic structures that capture the *essential* nature of the class of systems considered".

For clarity of numbering, no distinction has been made between propositions, lemmas and theorems. All are called "proposition". Also, proofs have been omitted. Proofs will be included in a full journal version of this paper.

2 LPetri nets

An *LPetri net* is a coloured Petri net in which the colour sets are the formulae from the $(\otimes, +, \&, \oplus, \multimap, \perp, !)$-fragment of LL on finite alphabets. The set of all such formulae is denoted L. We base the definition of the category of LPetri nets, **LPetri**, on the definition of **Petri** given by (Marti-Oliet and Meseguer 1991).

Definition 2.1. *(a) An* LPetri net *LN is a 4-tuple $\langle E, (B^{\otimes})^L, pre, post\rangle$ where $(B^{\otimes})^L$ is the set of functions from L to the free commutative monoid on B. The set of conditions is B and the set of events is E. We define pre and post as functions $pre, post : E \to (B^{\otimes})^L$.*
(b) Let $LN' = \langle E', (B'^{\otimes})^L, pre', post'\rangle$. Then $\langle f, g\rangle : LN \to LN'$ is a morphism *of LPetri nets if*
(i) f is a function $f : E \to E'$ and
(ii) $g : B^{\otimes} \to B'^{\otimes}$ is a monoid homomorphism such that

$$\begin{tikzcd} E \arrow[r, "pre"] \arrow[d, "f"'] & (B^{\otimes})^L \arrow[d, "g"] \\ E' \arrow[r, "pre'"'] & (B'^{\otimes})^L \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} E \arrow[r, "post"] \arrow[d, "f"'] & (B^{\otimes})^L \arrow[d, "g"] \\ E' \arrow[r, "post'"'] & (B'^{\otimes})^L \end{tikzcd} \quad \text{commute.}$$

The category **LPetri** *has objects all LPetri nets and arrows all LPetri net morphisms.*

3 LPetri net simulations

We must first introduce an alternative definition of LPetri net. Unfortunately, we must redefine *pre* and *post*.

Definition 3.1. *An LPetri net is a 4-tuple $\langle E, B, pre, post\rangle$ where E, B are sets, and pre, post are functions or "multirelations" $E \times B \to \mathbb{N}^L$.*

We call $pre(e)$ the pre-condition set of e and $post(e)$ the post-condition set of e. We can extend the pre- and post-conditions to multisets of events in the obvious way, to enable multisets of events to occur simultaneously.

If M is a marking of LN, then the pair $\langle LN, M\rangle$ is the LPetri net with marking M. We say that the marking $\langle LN, M\rangle$ *enables* A, denoted $M{\downarrow}^{LN}A$, if for each $b \in B$ and each $l \in L$, the definition of "enable" for a standard Petri net is satisfied. Formally, for each $b \in B$ and each $l \in L$, we require $\Sigma_{e\in A} pre_l\langle e, b\rangle \le M(b)$. Here, we are using Σ in the sense of multisets.

If LN evolves under A from the marking M to M', we write $M{\leadsto_1}M'$. This happens if for each $l \in L$, $M' = (M - pre_l(A)) + post_l(A)$. We then say that the events of A occur *concurrently*.

3.1 The category GLNet

Definition 3.2. *The category* **GLNet** *has objects all LPetri nets and arrows $\langle E, B, pre, post\rangle \to \langle E', B', pre', post'\rangle$ all pairs of functions $\langle f, F\rangle$ where $f : E \to E'$ and $F : B \to B'$ such that the following squares weakly commute in the direction of the partial order signs and composition is function composition of each component in the pair.*

$$\begin{tikzcd} E \times B' \arrow[r, "1\times F"] \arrow[d, "f\times 1"'] \arrow[dr, phantom, "\le"] & E \times B \arrow[d, "pre"] \\ E' \times B' \arrow[r, "pre'"'] & \mathbb{N} \end{tikzcd} \qquad \begin{tikzcd} E \times B' \arrow[r, "1\times F"] \arrow[d, "f\times 1"'] \arrow[dr, phantom, "\ge"] & E \times B \arrow[d, "post"] \\ E' \times B' \arrow[r, "post'"'] & \mathbb{N} \end{tikzcd}$$

We define a partial ordering on $\mathbb{N}^L$ by: If $f, g \in \mathbb{N}^L$, then $f \preceq g$ if and only if $f(l) \le g(l)$ for all $l \in L$.

Definition 3.3. *For an LPetri net morphism $\langle f, F\rangle : LN \to LN'$, the pair of markings $\langle M, M'\rangle$ is F-*`ok` *if $MF \preceq M''$.*

Suppose all pairs $\langle M_i, {M'}_i\rangle$ are F-`ok` for $0 \le i \le n+1$. If an evolution LN steps through the M_is from M_0 to N_{n+1}, then the image of the evolution under f is its *simulation* in LN'. A simulation of an evolution of an LPetri net is essentially a copy of the evolution in another LPetri net.

The following proposition is adapted from Proposition 6.4 of (Brown et al. 1991).

Proposition 3.1. *Let $\langle LN, M\rangle$ and $\langle LN', M'\rangle$ be marked LPetri nets, and let $\langle f, F\rangle : \langle LN, M\rangle \to \langle LN', M'\rangle$ be a morphism in* **GLNet** *such that $\langle M, M'\rangle$ is F-*`ok`*. Then for all $e \in E$, $M{\downarrow}^{LN} e$ implies that $M'{\downarrow}^{LN'} fe$.*

This means that $\langle LN, M\rangle$ can simulate any one-step evolution of $\langle LN', M'\rangle$. The following variation of Theorem 6.6 of (Brown et al. 1991) tells us when $\langle LN, M\rangle$ can simulate any evolution of $\langle LN', M'\rangle$.

Proposition 3.2. *Let $\langle LN, M\rangle$ and $\langle LN', M'\rangle$ be marked nets, and let $\langle f, F\rangle : LN \to LN'$ be an arrow in* **GLNet**. *If $\langle M_0, {M'}_0\rangle$ is F-*`ok` *and $M_0 {\rightsquigarrow}^e M_1$, then ${M'}_0 {\rightsquigarrow}^{fe} {M'}_1$ and $\langle M_1, {M'}_1\rangle$ is F-*`ok`.

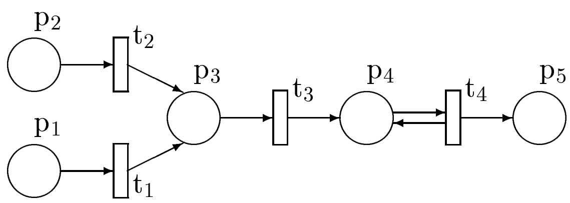
3.2 Example: Apple cake-baking

We now have the machinery to apply the simulation ideas from (Brown et al. 1991) to the apple cake-baking example of (Farwer 1996).

We have modified the LLPN which models an apple cake-baking process in (Farwer 1996) to an LPetri net. The process includes the making of dough, the mixing of apple pieces and baking at the correct temperature. LN' models this in detail, while LN models the process with less granularity, though LN only does things that LN' also does. In (Brown et al. 1991), simulations of Petri nets are given, and here we will give a simulation of LPetri nets $LN \to LN'$.

Let $LN' = \langle E', B', pre', post'\rangle$, where $E' = \{t_1, t_2, t_3, t_4\}$ and $B' = \{p_1, p_2, p_3, p_4\}$.

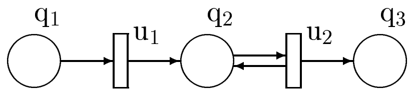
Symbols: I_j - some ingredient, A - apple, D - dough, H - heat in degrees Celsius, Q - apple quarter, R - ready to bake, C - cake.

$LN' =$ 

${M'}_0$ is defined by ${M'}_0(p_1) = \{\otimes_{j\in J} I_j\}$, ${M'}_0(p_2) = \{A^3\}$, ${M'}_0(p_4) = \{200H\}$ and ${M'}_0(p_3) = {M'}_0(p_5) = \emptyset$. (Note: we abbreviate n copies of A in $A \otimes \cdots \otimes A$ to A^n.) The events are defined by $pre'(t_1) = \{\otimes_{j\in J} I_j\}$, $post'(t_2) = \{A\}$, $post'(t_2) = \{Q^4\}$, $pre'(t_3) = \{D \otimes Q^{1}2\}$, $post'(t_3) = \{R\}$, $pre'(t_4) = \{R \otimes 200H\}$ and $post'(t_4) = \{\{200H\}_{p_4}, \{C\}_{p_5}\}$.

Suppose we let ${M'}_0(p_1) = \{\otimes_{j\in J} I_j, (I_{i_0} \otimes I_{i_1}) \multimap P\}$. The symbol P may stand for poison. Then the term $(I_{i_0} \otimes I_{i_1}) \multimap P$ models a contaminant which makes it impossible for dough to be produced. This could also be modelled in a CPN with guard functions, though not in so intuitively obvious a way.

Let $LN = \langle E, B, pre, post\rangle$, where $E = \{q_1, q_2, q_3\}$ and $B = \{u_1, u_2\}$.

$LN =$ 

M_0 is defined by $M_0(q_1) = \{D \otimes Q^{12}\}$, $M_0(q_2) = \{200H\}$ and $M_0(q_3) = \emptyset$. The events are defined by $pre(u_1) = \{D \otimes Q^{12}\}$, $post(u_1) = \{R\}$, $pre(u_2) = \{R \otimes 200H\}$, $post(u_2) = \{\{200H\}_{q_2}, \{C\}_{q_3}\}$.

Define $f : E \to E'$ by $f : u_1 \mapsto t_3$, $f : u_2 \mapsto t_4$, and define $F : B' \to B$ by $F : \{p_1, p_2, p_3\} \mapsto q_1$, $F : p_4 \mapsto q_2$, $F : p_5 \mapsto q_3$.

Then $\langle f, F\rangle : LN \to LN'$ is a morphism in **GLNet**. By Proposition 3.2, for any F-ok pair of markings $\langle M, M'\rangle$, the net LN' can simulate any behaviour of LN with marking M. Consider the pair of markings $\langle M, M'\rangle$ where $M = (\{D \otimes Q^{12}\}, \emptyset, \emptyset)$ and $M' = (\{\otimes_{j\in J} I_j, D \otimes Q^{12}\}, \{A^3, D \otimes Q^{12}\}, \{D \otimes Q^{12}\}, \emptyset, \emptyset)$. LN can evolve from M by $M {\rightsquigarrow_1}^{u_1} (\emptyset, R, \emptyset) {\rightsquigarrow_1}^{u_2} (\emptyset, \emptyset, C)$. The simulation of this evolution in LN' is:

$$M' {\rightsquigarrow_1}^{t_3} (\{\otimes_{j\in J} I_j, D \otimes Q^{12}\}, \{A^3, D \otimes Q^{12}\}, \emptyset, \{R\}, \emptyset) {\rightsquigarrow_1}^{t_4}$$
$$(\{\otimes_{j\in J} I_j, D \otimes Q^{12}\}, \{A^3, D \otimes Q^{12}\}, \emptyset, \emptyset, \{C\}).$$

4 Example: Message handler

We now consider an extension of the message handler and its more concrete version which was given in (Brown et al. 1991) and (Brown and Gurr 1992). Here, we add features beyond what (Brown et al. 1991) and (Brown and Gurr 1992) gave so as to illustrate the practical utility of LPetri nets. This example illustrates the type of simulation that is possible in **GLNet** as a result of Proposition 3.2. The approach is similar to that in the apple cake-baking example just given.

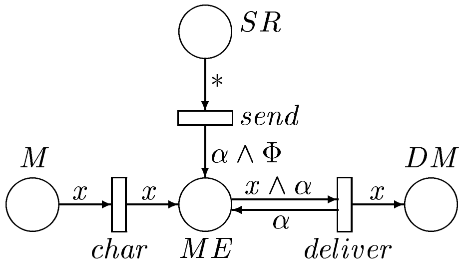
In the simple message handler N, when the sender is ready, characters are sent by M to ME which then ensures that the characters are delivered. The end of a message is indicated by $ENDMESS$, which disables ME. The message handler N' does everything that N does, but it also has a timeout feature. When a character is not sent within a stipulated time, the character is resent. Also, if a character is lost in N', then it is resent.

The *simulation morphism* $N \to N'$ simply maps

$B \mapsto B'$ for each condition B of N and $E \mapsto E'$ for each event E of N.

First consider the "unintelligent" message handler called N.

Symbols: SR - sender ready, M - message, ME - message sent into ether, DM - deliver message, $\Phi = E \multimap (N \multimap (D \multimap (M \multimap (E \multimap (S \multimap ((S \multimap \alpha^\perp) \parr (S^\perp \multimap \Phi)) \parr (S^\perp \multimap \Phi)) \parr (E^\perp \multimap \Phi)) \parr (M^\perp \multimap \Phi)) \parr (D^\perp \multimap \Phi)) \parr (N^\perp \multimap \Phi))$ where $\parr$ ("par") is the LL non-deterministic disjunction.

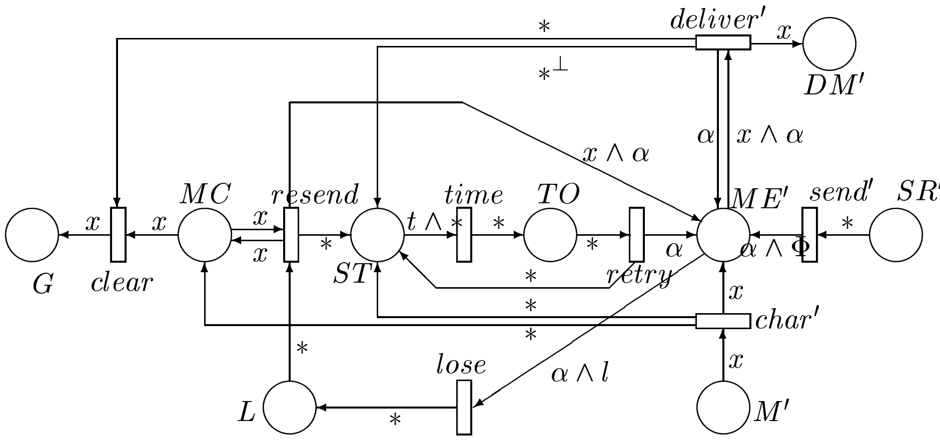


When the sender is ready, the *send* event occurs. This sends α and Φ to ME. It is implicit that the receipt of characters (each x is a character) via *char* is timed with SR. (At least, this synchrony is not accounted for by the model.) When a character arrives at ME and the α flag is there, then *deliver* occurs, which sends x to DM and returns the α flag to ME. This continues until the characters $E\ N\ D\ M\ E\ S\ S$ are received in sequence.

The formula Φ is "eroded" by each character in $ENDMESS$ until the flag α is removed when the last S is received. When the flag α is removed, the sending of characters is no longer enabled and thus ceases. Note that if one of the letters in "$ENDMESS$" is read but the next character received is not the subsequent letter of the word "$ENDMESS$", then the formula Φ is refreshed. When *send* occurs again placing the α flag in ME, then characters in the message can again be passed.

Now consider the more intelligent message handler, called N'.

Symbols: SR' - sender ready, M' - message, ME' - message sent into ether, DM' - deliver message, ST - start timer, TO - timeout, MC - copy message, L - lose message, Φ is the same linear logic expression as above, α is a flag used to enable message-sending.



The features of N' that are also in N work in the same way as in N. There are several additional features. When a character is sent by the event $char'$, the timer is started by a flag $*$ being sent to ST and a copy of the character

x is sent to MC. When the character is successfully sent, through the event *deliver*, the $*$ flag is deleted from ST by $*^{\perp}$ and the $*$ flag is sent to the *clear* event so that the character x is removed from MC.

If the timer "times out", then t occurs in ST so that the *time* event is enabled by $t \wedge *$ in ST. The event *time* then sends $*$ to TO. We represent the non-sending of the message in the allocated time by the loss of the α flag in the ME' condition. So the *retry* event sends the α flag to ME' again. The *retry* event then restarts the timer by sending the $*$ flag to ST.

If the character x is lost, then the *lose* event is enabled by the coming into being of the l flag in ME'. This passes the $*$ flag to condition L which enables the *resend* event. The *resend* event "passes" $x \wedge \alpha$ to ME' in another attempt to send the character. Note that the timer is not restarted.

5 A Proof System for LPetri Nets

Much of this section follows from (Brown et al. 1991), with variations for our extension from Petri nets to LPetri nets.

5.1 A category of multirelations

Definition 5.1. *Let* $\mathbf{C}$ *be a concrete category with finite products, and let* N *be an object of* $\mathbf{C}$ *equipped with a partial order* $\leq$. *The category* $\mathbf{M}_N\mathbf{C}$ *has* objects *the triples* $\langle E, B, \alpha\rangle$ *where* E *and* B *are objects of* $\mathbf{C}$ *and* $\alpha : E \times B \to N$ *is an arrow in* $\mathbf{C}$*;* arrows $\langle E, B, \alpha\rangle \to \langle E', B', \alpha'\rangle$ *are the pairs* $\langle f, F\rangle$ *of morphisms* $f : E \to E'$, $F : B' \to B$ *in* $\mathbf{C}$ *such that the following diagram commutes:*

$$\begin{tikzcd}
E \times B' \arrow[r, "1\times F"] \arrow[d, "f\times 1"'] \arrow[dr, phantom, "\geq"] & E \times B \arrow[d, "\alpha"] \\
E' \times B' \arrow[r, "\alpha'"'] & \mathbb{N}
\end{tikzcd}$$

where $\leq$ *is the partial order induced pointwise on* $\mathbf{C}(A, N)$ *by the partial order on* N, *and* composition *is composition in* $\mathbf{C}$ *for each component.*

The category $\mathbf{M}_N\mathbf{C}^*$ is defined in the same way as $\mathbf{M}_N\mathbf{C}$ except that the partial order sign in the definition of arrows is reversed.

Proposition 5.1. *The categories* $\mathbf{M}_N\mathbf{C}$ *and* $\mathbf{M}_N\mathbf{C}^*$ *are isomorphic.*

Evidently, an LPetri net can be regarded as an object of $\mathbf{M}_{\mathbb{N}\mathbb{L}}\,\mathbf{Set} \times \mathbf{M}_{\mathbb{N}\mathbb{L}}\,\mathbf{Set}$, where $\mathbf{M}_{\mathbb{N}\mathbb{L}}\,\mathbf{Set}$ is interchangeable with $\mathbf{M}_{\mathbb{N}\mathbb{L}}\,\mathbf{Set}^*$.

5.2 When $\mathbf{M}_N\mathbf{C}$ is a model of linear logic

Proposition 5.2. *(From (Brown et al. 1991)) If* $\mathbf{C}$ *is cartesian closed and has finite coproducts, and* $\langle N, \leq \rangle$ *is a partial order with a symmetric monoidal closed structure, then* $\mathbf{M}_N\mathbf{C}$ *has finite products, finite coproducts and a symmetric monoidal closed structure, then* $\mathbf{M}_N\mathbf{C}$ *is a sound model of linear logic. These are sufficient (not necessary) conditions.*

The linear connectives are interpreted as follows: $\wedge$ by product, $\oplus$ by coproduct, $\otimes$ by the symmetric monoidal product, $\multimap$ by the internal hom, $\invamp$ by a second symmetric monoidal product $\diamond$, and linear negation by the functor $(- \multimap \perp)$ where $\perp$ is the unit of $\diamond$. See (Brown 1991) for a proof.

Now, **Set** is cartesian closed. Also, $\langle \mathbb{N}^L, \leq \rangle$ is a closed ordered monoid, since truncated subtraction (where $f(l) - f(l') = 0$ if $f(l') > f(l)$ for $f \in \mathbb{N}^L$) is right adjoint to addition. Therefore, $\mathbf{M}_{\mathbb{N}^\mathbf{L}}\mathbf{Set}$ has all finite products and coproducts, and is symmetric monoidal closed. So $\mathbf{M}_{\mathbb{N}^\mathbf{L}}\mathbf{Set}$ is a sound interpretation of ILL, by Proposition 5.2.

5.3 ILL is a proof system for LPetri nets

The following four results are variations of results 7.1, 7.2, 7.3 and 9.1 from (Brown et al. 1991). They give the result that ILL is a proof system for LPetri nets.

Proposition 5.3. *The category* **GLNet** *is the pullback in* **Cat** *of the forgetful functor* $\mathcal{U} : \mathbf{M}_{\mathbb{N}^\mathbf{L}}\mathbf{Set} \to \mathbf{Set} \times \mathbf{Set}^{op}$ *along itself. That is,* **GLNet** *is the kernel pair of* $\mathcal{U}$ *in* **Cat**.

Proposition 5.4. *The functor* $\mathcal{U}$ *strictly preserves the product, coproduct and symmetric monoidal closed structure of* $\mathbf{M}_{\mathbb{N}^\mathbf{L}}\mathbf{Set}$.

Proposition 5.5. *The category* **GLNet** *has the products, coproducts and symmetric monoidal closed structure induced by those in* $\mathbf{M}_{\mathbb{N}^\mathbf{L}}\mathbf{Set}$.

Proposition 5.6. *Let* $[[\mathcal{N}]]$ *be an interpretation of the LPetri net* $\mathcal{N}$ *in* **GLNet**. *If* $\Gamma \vdash N$ *then there is an arrow* $[[\Gamma]] \to [[N]]$.

Proposition 5.6 allows us to perform calculations between LPetri nets, such as combining them by taking products and coproducts, just as though the LPetri nets were symbols in ILL formulae. The result is that we can model complex systems, using LPetri nets, by building up a large LPetri from simpler "component" LPetri nets. This is what (Brown et al. 1991) is referring to in their title "A linear specification for Petri nets", and here this specification language has been extended to LPetri nets.

6 Conclusion

We have considered a "pared down" version of LLPNs, and given some examples of how they might be useful. The aim has been to illustrate the convenience of using linear logic formulae as tokens, which highlights the usefulness of LLPNs as well as LPetri nets. We have also shown that the theory of simulations for Petri nets (from (Brown et al. 1991)) extends to LPetri nets. We did this by showing that certain properties of **GNet** also hold for **GLNet**. Finally, we have shown that the linear specification language for Petri nets, given in (Brown et al. 1991), is also a specification language for LPetri nets.

7 Further work

We would like to extend these results on LPetri nets to LLPNs as much as possible. Second, we would like to explore the connection between Petri nets and other models of concurrency, such as that of (Winskel and Nielsen 1995). Third, we would like to consider other possibilities for CPN colour sets - in particular, other substructural logics. Fourth, more study should be done on how to make this theory (more) useful to practitioners who work with Petri nets.

Finally, the author is presently doing work on semantics of data refinement for concurrency. If we consider Petri nets as a model for concurrency, and LPetri nets as a more powerful model for concurrency, then we would like to make our study of the semantics of data refinement encompass the linear specification language of Petri nets and LPetri nets.

References

Brown, C. (1991). *Linear Logic and Petri Nets: Categories, Algebra and Proof.* Ph. D. thesis, AI Laboratory, University of Edinburgh.

Brown, C. and D. Gurr (1992). Refinement and simulation of nets - a categorical characterization. *Lecture Notes in Computer Science 616.*

Brown, C., D. Gurr, and V. de Paiva (1991). A linear specification language for Petri nets. Technical Report 363, Computer Science Department, Aarus University.

Farwer, B. (1996). Towards linear logic Petri nets. Technical report, Faculty of Informatics, University of Hamburg.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science 50*, 1–102.

Jensen, K. (1997). *Coloured Petri Nets, Vol. 1.* New York, USA: Springer.

Marti-Oliet, N. and J. Meseguer (1991). From Petri nets to linear logic. *Mathematical Structures in Computer Science 1*, 69–101.

Sassone, V. (2000). On the Algebraic Structure of Petri nets. *Bulletin of the EATCS 72*, 133–148.

Troelstra, A. N. (1993). *Substructural Logics*, Chapter Tutorial on linear logic. Clarendon Press.

Winskel, G. and M. Nielsen (1995). *Semantic Modelling*, Volume 4 of *Handbook of Logic in Computer Science*, Chapter Models for concurrency. Oxford Science Publications.

A weakening of chromatic number

Tara Nicholson
Laboratory for Logic and Experimental Philosophy
Simon Fraser University
taraleen@sfu.ca

Abstract. Hadwiger's conjecture (1943), the claim that every graph with chromatic number $> n$ contains a K_{n+1} minor subgraph (for $n \in Z^+$), remains an open problem in graph theory. It has been proven for all $n \leq 5$, and for $n = 4$, it is equivalent to the Four Color Theorem. In this article, the conjecture is generalized to hypergraphs, and the model theory for a class of weakly aggregative modal logics is used to produce a construction with potential applications to a solution of this generalized formulation. More specifically, theorems of the K_n modal systems are used to derive a weakening and logically dual generalization of the "n-uncolourability" of a hypergraph. A contracting operation for hypergraphs is then defined and shown to preserve the weakened n-uncolourability property, relative to a specified input.

1 Introduction

In graph theory, scheduling problems constitute a common application of "chromatic number". For example, given a set Σ of committees, what is the least of amount of time required in order to schedule meetings in such a way that no conflicts arise? For a solution, it suffices to (a) form the system G consisting of the pair subsets of Σ whose memberships intersect, and (b) determine the cardinal j of the least partition π of Σ which biparts each element of G. The integer j, known as the **chromatic number** of G, is then the least number of time periods necessary for consistent scheduling. But, there are often occasions in which the consideration of such varying "human" factors as truancy, holidays, etc. makes for greater efficiency. E.g., there might be a smaller partition of Σ on which only elements of G whose common members are ill are a subset of some cell of the partition. In this way, putatively optimal solutions may be improved by determining which elements of the set system formed are (ir)relevant to a given application.

In logical, and particularly non-dialetheic, paraconsistent contexts, similar applications of "chromatic number" arise. Let Σ be a set of sentences of some language, and let H be the system of all minimally inconsistent subsets E of Σ. Then the chromatic number of H is the cardinal k of the least

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

partition of Σ into consistent cells. Preservationist logicians refer to this integer as the (in)coherence level of Σ, $\boldsymbol{l(\Sigma)}$(Jennings and Schotch 1989). As Jennings and Schotch have shown (1989), to avoid trivializing the closure of Σ under classical inference, it suffices to restrict inferences to those which follow classically from every element E' of a system H' of subsets of Σ, where the chromatic number of $\mathsf{H}' > k$. In this way, if a sentence α is inferred from Σ, then α follows classically from at least one cell in every partition of Σ into classically consistent subsets—i.e., $l(\Sigma \cup \alpha) = l(\Sigma)$.

Unfortunately however, the practical utility of this inferential strategy is questionable because the underlying problem—that of forming a set system H' with chromatic number $> k$—is probably not even in NP. Otherwise, we'd have CO–NP = NP, by the NP-completeness of COLOURABILITY. Still, in some contexts, it may be possible to form H' efficiently by weakening the restriction on its chromatic number. Here, rather than determining which elements of the set system formed are (ir)relevant to application at hand, we are, obversely, considering which partitions of the underlying set Σ are (ir)relevant to the preservation of the desired property, e.g., partitions of Σ which contain (in)consistent cells.

In what follows, a combinatoric formalization of this latter kind of weakening of "chromatic number" is derived from the model theory for a class of weakly aggregative modal logics, viz., the K_n modal logics, which represent the Jennings–Schotch paraconsistent inference relation. To begin, the relationship between $(n+1)$-ary aggregation and n-uncolourability is explored for the purposes of generalizing a logically dual characterization of chromatic number. This result is then exploited in a construction with potential applications to a solution of a generalized Hadwiger's conjecture for hypergraphs[1].

2 A modal representation of weak uncolourability

The K_n modal system is an $n+1$ aggregative theory which adjoins

- $[RM]$ $\frac{\vdash \alpha_i \rightarrow \alpha_j}{\vdash \Box\alpha_i \rightarrow \Box\alpha_j}$, and
- $[RN]$ $\frac{\vdash \alpha_i}{\vdash \Box\alpha_i}$ with
- $[K_n]$ $\vdash \Box p_1 \wedge \Box p_2 \wedge ... \wedge \Box p_{n+1} \ \rightarrow \ \Box 2/(n+1)(p_1, p_2, ..., p_{n+1})$,

where $2/(n+1)\{p_i\} = \vee[p_i \wedge p_j]$ $(1 \leq i \neq j \leq n+1)$.
Thus, K_1 is just the standard Kripke K.

Definition 2.1. *Let U be a non-empty set, let $n \in Z^+$, and let $\mathcal{R} : U \longrightarrow \wp(U^n)$. Then $\langle U, \mathcal{R}\rangle$ is an* $\boldsymbol{(n+1)}$***-ary relational frame*** $\mathfrak{F}$.

[1]The author wishes to thank the Natural Sciences and Engineering Research Council of Canada for supporting this research.

Definition 2.2. *Let Φ be a denumerable language with the operators $\wedge$, $\vee$, $\neg$, $\Box$, and $\Diamond$, where $At = \{p_i\}$ is the denumerable set of atoms of Φ. Let $\mathcal{F} = \langle U, \mathcal{R}\rangle$ be an $(n+1)$-ary relational frame. Let $\mathcal{V} : \Phi \longrightarrow \wp(U)$ where $\mathcal{V}$ is defined in the usual way for truth-functional operators, $\forall\alpha \in \Phi$, $V(\Box\alpha) = \{x \in U \mid \forall \boldsymbol{a} \in R(x), \exists y \in \boldsymbol{a} : y \in V(\alpha)\}$, and $\Diamond \stackrel{\text{def}}{=} \neg\Box\neg$. Then $\boldsymbol{\mathcal{M}}$ $= \langle \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{V}}\rangle$ is an **$(n+1)$-ary relational model** on the frame $\mathcal{F}$.*

Completeness for K_n with respect to the class of all $(n+1)$-ary relational models was first proven by Apostoli and Brown (1995), and later simplified by Nicholson, Jennings, and Sarenac (2000) in a proof which exploits a logically dual characterization of "n-uncolourability" for hypergraphs.

Definition 2.3. *A **hypergraph** $\mathsf{H} = \{E_i\}$ is any finite non-empty set of finite non-empty sets E_i, where $\cup\mathsf{H}$ is the **vertex** set of H, and $\{E_i\}$ is the set of its **edges**.*

If H is a hypergraph and $\forall E \in \mathsf{H}, |E| = 2$, then H is a simple graph without *isolated vertices*, or, vertices which fail to appear in edges of H. These are the only kinds of graphs we consider. In general, "G" denotes a graph, and "H" denotes a hypergraph. Moreover, none of the hypergraphs considered have isolated vertices; whence that graphs and hypergraphs alike are identified by their edge sets.

Definition 2.4. *Let Σ be a set. Then the **set of all k–decompositions δ of Σ**, $\boldsymbol{\Delta_k(\Sigma)} = \{\delta = \{d_i\} \mid \cup_{i=1}^{k} d_i = \Sigma\}$.*

Definition 2.5. *Let Σ be a set. Then the **set of all k–partitions π of Σ**, $\boldsymbol{\Pi_k(\Sigma)} = \{\pi = \{c_i\} \mid \cup_{i=1}^{k} c_i = \Sigma$, and $\forall i, j\ (1 \leq i \neq j \leq k)$, $c_i \cap c_j = \emptyset\}$.*

Definition 2.6. *Let H be a hypergraph. Then the **chromatic number** of H, $\boldsymbol{\chi(\mathsf{H})} = min\ \{j \in Z^+ \mid \exists\pi \in \Pi_j(\cup\mathsf{H}) : \forall c_i \in \pi, \forall E \in \mathsf{H},\ E \not\subseteq c_i\}$.*

Definition 2.7. *Let H be a hypergraph, and let $n \in Z^+$. Then H is **n–uncolourable** if $\chi(\mathsf{H}) > n$. H is **(properly) n–colourable**, else.*

If H is a hypergraph with a singleton edge, then since $\forall n \in Z^+$, no n-partition of $\cup\mathsf{H}$ biparts every H-edge, set $\chi(\mathsf{H}) = \infty$. In general, if $E \in \mathsf{H}$ is not biparted by a partition π, then we say that E is monochromatic on π. Because we are interested in (a) properties of hypergraphs with determinate chromatic number, and (b) non-trivial aggregation among formulae of a language, typically hypergraphs with singleton edges are excluded from consideration.

Definition 2.8. *Let $\mathsf{H} \subseteq \wp(\Phi)$. Then the **formulation** of H, $\boldsymbol{f(\mathsf{H})} = \vee[\wedge[E \in \mathsf{H}]]$. The **dual formulation** of H, $\boldsymbol{f^{\triangle}(\mathsf{H})} = \wedge[\vee[E \in \mathsf{H}]]$.*

Definition 2.9. *Let H be a hypergraph with a set $S \subseteq \cup\mathsf{H}$. Then S is a **transversal** for H if $\forall E \in \mathsf{H}, S \cap E \neq \emptyset$.*

Definition 2.10. *$\forall H$, the **transversal hypergraph** for H, $\boldsymbol{TrH}$ is the set of all inclusion–minimal transversals for H.*

Lemma 2.11. *$\forall H, TrTrH \subseteq H$. (Berge 1989) (Nicholson et al. 2000)*

Theorem 2.12. *$\forall H, \models f^{\triangle}(H) \leftrightarrow f(TrH)$, and $\models f^{\triangle}(TrH) \leftrightarrow f(H)$. (Nicholson et al. 2000)*

The preceding result suggests that $\forall \mathsf{H}, \mathsf{H}$ and $Tr\mathsf{H}$ are *duals* in the logical sense in which conjunctive and disjunctive normal forms are duals. To illustrate, let $\alpha \in \Phi$, with β the DNF of α. Let H be the underlying hypergraph whose edges are the disjuncts of β, and whose vertices are the conjunct literals. Then $\beta = f(\mathsf{H}) \equiv f^{\triangle}(Tr\mathsf{H}) =$ the CNF of α.

Definition 2.13. *Let $H \subseteq \wp(\Phi)$. Then $[J_n] = \wedge[\Box[\cup H]] \rightarrow \Box f(H)$, and $[J_n^{\triangle}] = \Diamond f^{\triangle}(H) \rightarrow \vee[\Diamond[\cup H]]$.*

Theorem 2.14. *Let $H \subseteq \wp(At)$, and let $\mathcal{C}$ be the class of all $(n+1)$-ary relational models. Then $\forall n \in Z^{+}$, $\chi(H) > n \;\Leftrightarrow\; \mathcal{C} \models [J_n]$.*

Lemma 2.15. *Let $H \subseteq \wp(\Phi)$. Then where $\mathcal{C}$ is the class of all $(n+1)$-ary relational models, $\mathcal{C} \models [J_n] \;\Leftrightarrow\; \mathcal{C} \models [J_n^{\triangle}]$.*

Corollary 2.16. *Let $H \subseteq \wp(At)$, and let $\mathcal{C}$ be the class of all $(n+1)$-ary relational models. Then $\chi(H) > n \;\Leftrightarrow\; \mathcal{C} \models \Diamond f(TrH) \rightarrow \vee[\Diamond[\cup H]]$.*

Expressed combinatorically (modullo hypergraphs with singleton edges), this corollary provides the following logically dual characterization of "n-uncolourability" for hypergraphs:

Theorem 2.17. *$\forall H, \forall n \in Z^{+}$, H is n-uncolourable $\Leftrightarrow \forall B \in \binom{TrH}{n}, \cap B \neq \emptyset$, and $|TrH| > n$. (Nicholson et al. 2000)*

To generalize this characterization, consider that $\forall B \in \binom{Tr\mathsf{H}}{n}, \cap B \neq \emptyset \Leftrightarrow \forall B \in \binom{Tr\mathsf{H}}{n}, \exists x \in \cup\mathsf{H} : \{x\}$ is a transversal for $B \Leftrightarrow$ every n-tuple of $Tr\mathsf{H}$-edges has a *singleton* transversal. Thus, if the size of minimum transversals for n-tuples of $Tr\mathsf{H}$-edges is allowed to vary, then H is n-colourable. In that event however, by weakening the $[J_n^{\triangle}]$ principle it is possible to define a subclass of n-partitions of $\cup\mathsf{H}$ which do not properly colour H. I.e., let $\mathsf{H} \subseteq \wp(At)$, let $\delta = \{d_i\}$ be a decomposition of $\cup\mathsf{H}$, and let $\mathcal{C}$ be the class of $(n+1)$-ary relational models. Then the following rule preserves validity with respect to $\mathcal{C}$:

$$\frac{\Diamond f^{\triangle}(\mathsf{H}) \rightarrow \Diamond\vee[d_1] \vee \Diamond\vee[d_2] \vee \ldots \vee \Diamond\vee[d_{|\delta|}]}{\Box\wedge[d_1] \wedge \Box\wedge[d_2] \wedge \ldots \wedge \Box\wedge[d_{|\delta|}] \rightarrow \Box f(\mathsf{H})}.$$

I.e., where $S = \{\pi \in \Pi_n(\cup\mathsf{H}) \mid \forall d \in \delta, \exists c \in \pi : d \subseteq c\}$, we have that $\forall \pi \in S$, some H-edge is monochromatic on π.

Definition 2.18. *Let S be a finite set with $k \in Z^+$, and $\delta \in \Delta_k(S)$. Then the set of* $\boldsymbol{\delta}$***-n-partitions of*** $\boldsymbol{S, \Pi_n^\delta(S)} =$ $\{\pi \in \Pi_n(S) \mid \forall d_i \in \delta, \exists c \in \pi : d_i \subseteq c\}$.

In words, $\Pi_n^\delta(S)$ is just the set of n-partitions of S on which each $d_i \in \delta$ is monochromatic.

Definition 2.19. *Let S be a finite set with $k \in Z^+$, and $\delta \in \Delta_k(S)$. Then the set of* $\boldsymbol{\delta}$***-n-decompositions of*** $\boldsymbol{S, \Delta_n^\delta(S)} =$ $\{\delta' \in \Delta_n(S) \mid \forall d_i \in \delta, \exists d' \in \delta' : d_i \subseteq d'\}$.

Definition 2.20. *Let H be a hypergraph, let $k, n \in Z^+$, and let $\delta \in \Delta_k(\cup \mathsf{H})$. Then H is* ***uncolourable on*** Π_n^δ, *if $\forall \pi \in \Pi_n^\delta(\cup \mathsf{H})$, $\exists E \in \mathsf{H} : E$ is monochromatic on π. H is (properly)* ***colourable on*** Π_n^δ, *else.*

In general, if H is a hypergraph and $\exists k, n \in Z^+$ with $\delta \in \Delta_k(\cup \mathsf{H})$, and H uncolourable on Π_n^δ, then H is said to be **weakly $\boldsymbol{n}$-uncolourable**.

Theorem 2.21. *Let H be a hypergraph, with $k, n \in Z^+$, and $\delta \in \Delta_k(\cup \mathsf{H})$. Then H is uncolourable on $\Pi_n^\delta \Leftrightarrow \forall B \in \binom{Tr H}{n}, \exists d_i \in \delta : d_i$ is a transversal for B.*

Proof. [$\Rightarrow$] Assume the antecedent. Suppose the consequent is false. Let B be an n-tuple of transversals for H with no d_i transversing B. Then $\cup \mathsf{H} - [B]$ induces an n-partition $\pi \in \Pi_n^\delta$ which biparts every H-edge, absurd.

[$\Leftarrow$] Assume the antecedent. Suppose that the consequent is false. In particular, let $\pi \in \Pi_n^\delta$ properly colour H. Then $\cup \mathsf{H} - [\pi]$ is a j-tuple ($j \leq n$) of transversals for H for which no $d_i \in \delta$ is a transversal, absurd.

Notice that the generalization of n-uncolourability provided by the preceding theorem suggests a parameter according to which the extent to which a hypergraph is (not) n-uncolourable may be measured. In the next section, a formulation of Hadwiger's conjecture is given which highlights this parameter, and its potential utility, with respect to the solution of the conjecture.

3 Hadwiger's conjecture for hypergraphs

Hadwiger's conjecture is the claim that $\forall n \in Z^+, \forall \mathsf{G}$, if $\chi(\mathsf{G}) > n$, then the complete graph on $n+1$ vertices, K_{n+1}, is obtainable from G in a finite amount of time by consecutively either deleting vertices or edges, or contracting edges. For $n = 4$, this conjecture is equivalent to the four-colour theorem. Moreover, Robertson, Seymour, and Thomas (1993) have shown that the four-colour theorem implies Hadwiger's conjecture for $n = 5$; for $n = 3$ it was proven by Dirac (1952); for $n = 2$, by Hadwiger (1943); and for $n = 1$ it is utterly trivial. So far however, for arbitrary $n \in Z^+$, there is no known proof of the conjecture. In this section, the conjecture is generalized to hypergraphs, and equivalently formulated in the language of weak-uncolourability.

Definition 3.1. *Let* H *and* H' *be hypergraphs. Then* H' *is a* ***contraction*** *of* H *if for some* $E = \{x, y_1, y_2, ...y_k\} \in \mathsf{H}$, *and some* $y_i \in E, \mathsf{H}'$ *is the result of substituting* x *for all occurrences of* y_i *in* H*-edges, and deleting any singleton edges which thereby arise.*

If H is a graph, then a contraction of H corresponds to the usual notion of an edge–contraction for a graph. Similarly, the deletion of a vertex x in a hypergraph H results in the hypergraph $\mathsf{H}' = \{E \in \mathsf{H} \mid E \subseteq \cup\mathsf{H} - x\}$, and the deletion of an H-edge E gives $\mathsf{H} - \{E\}$.

Definition 3.2. *Let* H *and* J *be hypergraphs. Then* H' *is a* ***minor*** *of* H *if* H' *is obtainable from* H *by a finite sequence of contractions, or deletions of vertices or edges.*

(Generalized) Hadwiger's conjecture:
$\forall \mathsf{H}, \forall n \in Z^+, \chi(\mathsf{H}) > n \;\Rightarrow \mathsf{H}$ contains a K_{n+1} minor.

Definition 3.3. *Let* H *be a hypergraph. Then* H *is* ***connected*** *if* $\forall x, y \in \cup\mathsf{H}$, *there is a sequence* $E_1, E_2, ..., E_k$ *of* H*-edges, with:*

- $x \in E_1$,
- $y \in E_k$, *and*
- $E_j \cap E_{j+1} \neq \emptyset$ $(1 \leq j < k)$.

Definition 3.4. *Let* H *be a hypergraph with* $S \subseteq \cup\mathsf{H}$. *Then the* ***restriction*** *of* H *to* S, $\mathsf{H}[\boldsymbol{S}] = \{E \in \mathsf{H} \mid E \subseteq S\}$.

Proposition 3.5. *Let* H *and* H' *be connected hypergraphs, with* $\cup\mathsf{H}' = [j]$. *Then* H *contains an* H' *minor iff* $\exists \pi \in \Pi_j(\cup\mathsf{H})$ *with non-empty cells such that* $\forall c \in \pi, \mathsf{H}[c]$ *is connected, and* $\forall E = \{j_i\} \in \mathsf{H}'$, $\exists E' \in \mathsf{H}$ *with the vertices in* E' *having indices which agree with those in* E.

Definition 3.6. *Let* H *and* H' *be connected hypergraphs where* H' *is a minor of* H, *and* $\cup\mathsf{H}' = [j]$. *Let* $\pi \in \Pi_j(\cup\mathsf{H})$ *be such that* $\forall c \in \pi, c \neq \emptyset$, $\mathsf{H}[c]$ *is connected, and* $\forall E = \{j_i\} \in \mathsf{H}'$, $\exists E' \in \mathsf{H}$ *with the indices of* E' *agreeing with those of* E. *Then* $\{\mathsf{H}[c_i]\}$ $(1 \leq i \leq j)$ *is a set of* ***exterior subgraphs*** *for* H', *and* $\{E \in \mathsf{H} \mid \forall c \in \pi, E \notin \mathsf{H}[c]\}$ *is an* $\boldsymbol{\mathsf{H}'}$***-interior of*** H, H/H' *(see Figure 1).*

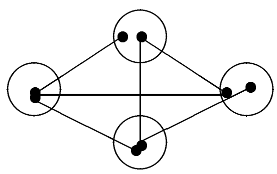


Figure 1: a K_4-interior G/K_4 of a graph G

Theorem 3.7. *Let H be a connected hypergraph. Then H contains a K_{n+1} minor iff there is an $(n+1)$–vertex minor G of H with exterior subgraphs $H_1, H_2, ..., H_{n+1}$ satisfying $\forall B \in \binom{Tr(H/G)}{n}$, $\exists H_i : \cup H_i$ is a transversal for B.*

Proof. $[\Rightarrow]$ Assume that H contains a K_{n+1} minor G with exterior subgraphs $\mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H}_{n+1}$. Then where $\delta = \{\cup H_1, \cup H_2, ..., \cup H_{n+1}\}, \mathsf{H}/\mathsf{G}$ is uncolourable on Π_n^δ. $\therefore \models \Box \wedge [\cup \mathsf{H}_1] \wedge \Box \wedge [\cup \mathsf{H}_2] \wedge ... \wedge \Box \wedge [\cup \mathsf{H}_{n+1}] \rightarrow \Box f(\mathsf{H}/\mathsf{G})$. $\therefore \models \Diamond f^{\triangle}(\mathsf{H}/\mathsf{G}) \rightarrow \vee[\Diamond \vee [\cup \mathsf{H}_i]](1 \leq i \leq n+1)$. Whence, $\forall B \in \binom{Tr(\mathsf{H}/\mathsf{G})}{n}, \exists \mathsf{H}_i : \cup \mathsf{H}_i$ is a transversal for B.

$[\Leftarrow]$ Assume the antecedent. Then $\models \Diamond f^{\triangle}(\mathsf{H}/\mathsf{G}) \rightarrow \vee[\Diamond \vee [\cup \mathsf{H}_i]]$ $(1 \leq i \leq n+1)$. So $\models \Box \wedge [\cup \mathsf{H}_1] \wedge \Box \wedge [\cup \mathsf{H}_2] \wedge ... \wedge \Box \wedge [\cup \mathsf{H}_{n+1}] \rightarrow \Box f(\mathsf{H}/\mathsf{G})$. $\therefore$ H contains a K_{n+1} minor subgraph.

Corollary 3.8. *Hadwiger's conjecture is true $\Leftrightarrow .\forall H, \chi(H) > n \Rightarrow \exists G, H' \subseteq H : G$ is a minor of H' with exterior subgraphs $H'_1, H'_2, ...H'_{n+1}$ such that $\forall B \in \binom{Tr(H'/G)}{n}$, $\exists H'_i : \cup H'_i$ is a transversal for B.*

4 weakly n-uncolourable minor subgraphs

Since "minor-hood" is transitive for hypergraphs, to prove Hadwiger's conjecture it suffices to show that every hypergraph admits of a chromatic number–preserving contraction. In Nicholson (2001) it is shown how to define such an operation for every hypergraph H with a vertex x satisfying: $\exists S \subseteq N_{\mathsf{H}}(x)(= \{E \in \mathsf{H} \mid x \in E\}) : |S| \geq |N_{\mathsf{H}}(x)| - (\chi(\mathsf{H}) - 3)$, and $\forall E \in \mathsf{H}, E \not\subseteq \cup S - \{x\}$. As a corollary, it follows that for every graph G, there is a chromatic number–preserving contraction of G if $\exists x \in \cup \mathsf{G}$ such that either x does not appear in any K_3 subgraph of G, or $|N_{\mathsf{G}}(x)| = \chi(\mathsf{G}) - 1$. The proof exploits the downward invariance of the dual characterization of "n-uncolourability" upon the deletion of a specified vertex from a given $Tr\mathsf{G}$-edge. Here, a similar strategy is invoked, but this time using the dual characterization of weak uncolourability.

Definition 4.1. *Let H be a hypergraph. Then the **covering hypergraph** for H, $\boldsymbol{Tr'}H$, is the set of all transversals for H.*

Definition 4.2. *Let H be a hypergraph, with $E \in Tr'H$, and $x \in E$. Let $n \in Z^+$. Then **x is IN_n^E** if $\exists B \in \binom{Tr'H}{n} : E \in B$ and $\cap B = \{x\}$; **x is OUT_n^E**, else (see Figure 2) .*

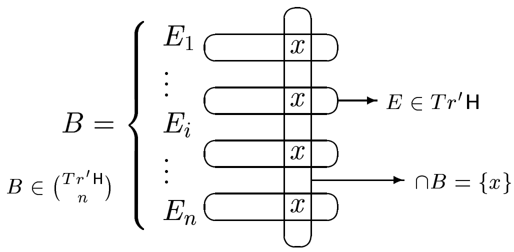


Figure 2: x is IN_n^E

Definition 4.3. *Let H be a hypergraph with $E \in Tr\mathsf{H}$, and $x \in E$. Then the* ***contractor set for x on E***, $\boldsymbol{C_E^x} = (\cup\{E' \in \mathsf{H} \mid E - \{x\} \cap E' = \emptyset\}) - \{x\}$ *(see Figure 3).*

Definition 4.4. *Let H be a hypergraph, and let $E \in Tr\mathsf{H}$, with $x \in E$. Then the* ***(block) contraction of the contractor set for x on E***, $\boldsymbol{C_E^x \mapsto x}$, *is the minor hypergraph obtained from H by identifying each vertex in the contractor set for x on E with x (and deleting any singleton edges which arise).*

Definition 4.5. *Let H be a hypergraph, and let $B \in \binom{Tr\mathsf{H}}{n+1}$ satisfy:*

- $\cap B = \emptyset$,
- $E \in B$, *and*
- $x \in E$.

Then B is a ***context*** *for $C_E^x \mapsto x$, E is a* ***base***, *and x is the* ***pivot*** *(see Figure 3).*

Definition 4.6. *Let H be a hypergraph with $E \in Tr\mathsf{H}$ and $x \in E$ such that for some $n \in Z^+$, x is OUT_n^E. Then a hypergraph H' is a* $\boldsymbol{\Psi_n}$***–contraction*** *of H, if $\mathsf{H}' \cong C_E^x \mapsto x$, and there is a context B for $C_E^x \mapsto x$.*

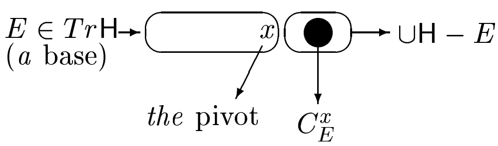


Figure 3: the contractor set C_E^x for x on E

Definition 4.7. *Let H be a hypergraph with a Ψ_n-contraction H'. Let $B = \{E_i\}$ be a context for the contraction, E_j a base, and x the pivot. Then* $\boldsymbol{\mu}$: $B \longrightarrow \wp(\cup\mathsf{H}')$, *where:*

- $\forall i \neq j,\ \mu(E_i) = (E_i - C^x_{E_j}) \cup \{x\}$, *and*
- $\mu(E_j) = E_j - \{x\}$.

Proposition 4.8. *Let H be a hypergraph with a Ψ_n-contraction H'. Let B be a context for H', E_j a base, and x the pivot. Then $\forall E \in B$, $\mu(E)$ is a transversal for H', and in particular, $\mu(E_j) \in Tr(\mathsf{H}')$.*

Lemma 4.9. *Let H be a hypergraph with a Ψ_n-contraction H'. Let $B = \{E_i\}$ be a context for H', E_j a base, and x the pivot. Let $C = \{c_i\} = \cup\mathsf{H} - [B]$, where $\forall i\ (1 \leq i \leq n+1)$, $c_i = \cup\mathsf{H} - E_i$. Lastly, let $C' = \{c'_i\} = \cup\mathsf{H}' - [\mu[B]]$, where $\forall i\ (1 \leq i \leq n+1), c'_i = \cup\mathsf{H}' - \mu(E_i)$.*
Then H is uncolourable on $\Pi^C_n \Rightarrow \mathsf{H}'$ is uncolourable on $\Pi^{C'}_n$.

Proof. Assume that H' is colourable on $\Pi^{C'}_n$. Let $\pi \in \Pi^{C'}_n(\cup\mathsf{H}')$ bipart every H'-edge. Then since $\mu(E_j) \in Tr\mathsf{H}', c'_j \in \pi$. $\therefore$ replacing x with $C^x_{E_j}$ in $c'_j \in \pi$ produces an n-partition π' of $\cup\mathsf{H} - \{x\}$ which biparts every H-edge. $\therefore \cup\mathsf{H} - [\pi'] \in \binom{Tr'\mathsf{H}}{n}$. But by construction, $\cap(\cup\mathsf{H} - [\pi']) = \{x\}$, which is absurd since x is $OUT^{E_j}_n$. $\therefore \mathsf{H}'$ is uncolourable on $\Pi^{C'}_n$ (see Figure 4).

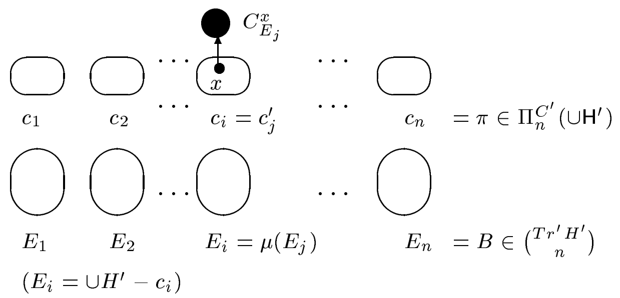


Figure 4: Ψ preserves weak-uncolourability

Definition 4.10. *Let H be a hypergraph with a Ψ_n-contraction H'. Let B be a context for H', E_j a base, and x the pivot. Then a **transformation $t[B]$** of the context B is the result of replacing each $E \in \mu[B]$ with any least subset $t(E)$ of E which is a transversal for H'.*

Proposition 4.11. *Let H be a hypergraph with a Ψ_n-contraction H'. Let $B = \{E_i\}$ be a context for H', E_j a base, and x the pivot. Further, let $C = \{c_i\} = \cup\mathsf{H}' - [\mu[B]]$, where $\forall i\ (1 \leq i \leq n+1), c_i = \cup\mathsf{H}' - \mu(E_i)$, and let $C' = \{c'\} = \cup\mathsf{H}' - t[B]$, for any transformation t of B, where $\forall i\ (1 \leq i \leq n+1), c'_i = \cup\mathsf{H}' - t(E_i)$.*
Then H' is uncolourable on $\Pi^C_n \Rightarrow \mathsf{H}'$ is uncolourable on $\Pi^{C'}_n$.

Corollary 4.12. *Let H be a hypergraph with a Ψ_n-contraction H'. Let $B = \{E_i\}$ be a context for H', E_j a base, and x the pivot. Let $C = \{c_i\} = \cup H - [B]$, where $\forall i\ (1 \leq i \leq n+1), c_i = \cup H - E_i$. Lastly, let $C\,' = \{c\,'_i\} = \cup H\,' - [t[B]]$, for any transformation t of B, where $\forall i\ (1 \leq i \leq n+1), c\,'_i = \cup \mathsf{H}\,' - t(b_i)$.*

Then H is n-uncolourable on $\Pi_n^C \Rightarrow \mathsf{H}\,'$ is uncolourable on $\Pi_n^{C\,'}$.

The preceding result may (more suggestively, perhaps) be represented as a validity–preserving rule for $(n+1)$-ary relational models:

$$\frac{\Box\wedge[c_1]\wedge\Box\wedge[c_2]\wedge\ldots\wedge\Box\wedge[c_{n+1}]\rightarrow\Box f(\mathsf{H})}{\Box\wedge[c'_1]\wedge\Box\wedge[c'_2]\wedge\ldots\wedge\Box\wedge[c'_{n+1}]\rightarrow\Box f(\mathsf{H}\,')}$$

Proposition 4.13. *Let H be a hypergraph with a Ψ_n-contraction $\mathsf{H}\,'$. Let B be a context for $\mathsf{H}\,', E_j$ a base, and x the pivot. Then for any transformation t of B, $\cap t[B] = \emptyset$.*

Definition 4.14. *Let $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H}_k$ and $B_0, B_1, B_2, ..., B_{k-1}$ be finite sequences of hypergraphs H_i and B_{i-1}, respectively. Let $E_0, E_1, E_2, ..., E_{k-1}$ be a finite sequence of sets E_{i-1}, and let $x_0, x_1, x_2, ..., x_{k-1}$ be a finite sequence of vertices x_{i-1}. Then $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H}_k$ is a* $\boldsymbol{\Psi_n}$***–sequence*** *(for $n \in Z^+$) if $\langle H_i\rangle, \langle B_{i-1}\rangle, \langle E_{i-1}\rangle$, and $\langle x_{i-1}\rangle$ satisfy:*

1. *$\forall i\ (0 \leq i \leq k-1)$,*
   - *$B_i \in \binom{Tr\mathsf{H}_i}{n+1}$,*
   - *$E_i \in B_i$,*
   - *$x_i \in E_i$,*
2. *$\forall i\ (1 \leq i \leq k), \mathsf{H}_i$ is a Ψ_n-contraction of H_{i-1} which uses B_{i-1} as a context, E_{i-1} as a base, and x_{i-1} as the pivot, and*
3. *$\forall i\ (1 \leq i \leq k-1)$, there is a transformation t: $B_i = t[B_{i-1}]$.*

Corollary 4.15. *If $\forall \mathsf{H}, \forall n \in Z^+$, $\chi(\mathsf{H}) > n \Rightarrow$ there is a Ψ_n-sequence $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2, ..., \mathsf{H}_k$ with $\mathsf{H} = \mathsf{H}_0$ and $|\cup \mathsf{H}_k| = n+1$, then Hadwiger's conjecture is true.*

References

Apostoli, P. and B. Brown (1995). A solution to the competeness problem for weakly aggregative modal logic. *Journal of Symbolic Logic 60*, 832–842.

Berge, C. (1989). *Hypergraphs: combinatorics of finite sets.* North-Holland.

Dirac, G. (1952). A property of 4-chromatic graphs and some remarks on critical graphs. *J. London Math. Soc. 27*, 85–92.

Hadwiger, H. (1943). Über eine Klassifikation der Streckenkomplexe. *Vierteljahrsch. Naturforsch. Ges. 88*, 133–42.

Jennings, R. and P. Schotch (1989). On Detonating. *Philosophia, Paraconsistent Logic*, 306–327.

Nicholson, T. (2001). Chromatic number preserving contractions of hypergraphs. (unpublished) presented at the Society for Exact Philosophy Conference, Université de Montréal, May, 2001.

Nicholson, T., R. Jennings, and D. Sarenac (2000). Revisiting completeness for the K_n modal logics: a new proof. *the logic journal of the IGPL 8*(1). on–line.

Roberston, N., P. Seymour, and R. Thomas (1993). Hadwiger's conjecture for K_6-free graphs. *Combinatorica 13*, 279–361.

A plural resolution logic

Rick Nouwen
UiL-OTS Universiteit Utrecht
rick.nouwen@let.uu.nl

Abstract. This paper presents an integration of centering theory into incremental dynamic logic. It furthermore discusses how such a combined system could deal with plural pronoun resolution.

1 Introduction

It is common in dynamic semantics to assume that anaphora are related to their antecedent by indexing. This way, dynamic semantics can analyze and describe the distribution of such relations. At the same time, it ignores one of the key aspects of language use: reference resolution. On the pragmatic side of the study of language we see the exact opposite situation. Theories like centering theory ((Grosz et al. 1995)) describe the way anaphoric elements choose their antecedents, but they say little to nothing about structural limitations of anaphoric relations. A combination of these two families of theories is thus a natural step in the study of the pragmasemantics of natural language.

Recently, a number of attempts have been made at combining dynamic semantics with centering theory ((Roberts 1998), (Beaver 1999), but see also (Hardt 1999)), as well as attempts to improve the formalism involved in dynamic semantics in order for it to better deal with discourse anaphora ((van Eijck 2000a),(van Eijck 2001)). All such studies have thus-far concentrated on reference resolution of singular pronouns. This is a dangerous simplification since dealing with plurals proves to be a far more complicated challenge. Plural pronoun reference resolution is a relatively poorly studied mechanism. Since plural anaphora can be linked to multiple antecedents, the basic insights from centering and other pragmatic theories of pronominal reference focusing on singulars lose their value.

In this paper I take a formalist standpoint towards pronoun resolution, by integrating a formalization of *centering theory* in a dynamic logic. The base system will be *incremental dynamics* as presented in (van Eijck 2001).

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

In Section 2 the formal details of the logic are presented. In the next section I consider its relation to centering and propose some translations of centering notions. In Section 4, I discuss the resolution of plurals, propose some changes to the system and I give some examples. This is followed by conclusions and some thoughts on future developments.

2 Presentation of the logic

2.1 Incremental Dynamics

Incremental dynamics (ID, (van Eijck 2001)) is a system which purports to do away with the destructive assignment problem of dynamic predicate logic (DPL, (Groenendijk and Stokhof 1991)) as well as provide a logic fit for a reformulation of discourse representation theory in Montagovian (i.e. bottom up) style. In DPL, the merging of two contexts sharing a certain variable will destroy the value of one of its occurrences. ID is variable free. Quantifier actions simply bind the next available index.

For the purpose of this paper we focus on one particular interesting aspect introduced in (van Eijck 2000a). Normally, dynamic meaning is seen as a relation between sets of individuals. In ID these sets are replaced by sequences reflecting a salience hierarchy: the most prominent individual in the discourse is the head of the current sequence. Salience of an individual is changed whenever it is involved in a predication. Predicates have the capability of permuting contexts in such a way that their arguments are assigned prominence according to the obliqueness hierarchy. Here is an example.

(20.1) i) A farmer who owns a donkey likes a woman.

ii) $\exists x; \text{farmer}(x); \exists y; \text{donkey}(y); \text{own}(x,y); \exists z; \text{woman}(z); \text{like}(x,z)$

Normally we would give the semantics for (20.1i) as in (20.1ii), producing a context where all assignments are such that x, y and z refer to a farmer, a donkey and a woman respectively. But notice that the salience hierarchy of (i) in no way reflects the order of introduction of variables in (ii). Things change if we assume predicates to be permuters. The *own* predicate will first push the farmer to the head of the sequence and the donkey to the second position. Processing *likes* later on will again push the farmer to the head, but now the woman to the second position. The output is thus a sequence where the referent for a farmer is most salient and the referent for the donkey is the least salient one, which is in accordance with the usual relation between salience and syntactic rôle.

Incremental dynamics is thus an ideal starting point for our integration of reference resolution and dynamic semantics. We will now proceed with the formal details of the logic and gradually extend the system to fit our purpose better.

2.2 The logic

In the current variant on the theme of ID, an *assignment* will be a triple $\langle l, \mathrm{i}, d\rangle$, where l is a natural number representing syntactic level, i is a natural number representing the index or name and d is an individual. Of each introduced individual we thus keep track of where it was introduced and by which label.

A *context* is a sequence of assignments. If c is a context and α an assignment, then we write $c : \alpha$ for the sequence resulting from pushing α to the top of c. We write $c[p]$ for the pth assignment in c, e.g. $c[0]$ returns the head of context c. If i is an index we write $c(\mathrm{i})$ for the entity paired with i in some assignment in c. $c(\mathrm{i}) =\uparrow$ whenever i is not part of any assignment in c.

Since our collections of assignments are represented as sequences we now have the opportunity to permute them without losing track of our level-index-entity labelling. We define permutation of contexts as follows (given a trivial definition of subtraction): $(p)c := (c - c[p]) : c[p]$. Given an assignment $a = \langle \alpha, \beta, \gamma\rangle$ we write $a^\bullet$ for γ. Similarly $c^\bullet$ is $\{a^\bullet | a \in c\}$.

The model and interpretation in the model are standardly defined:

Definition 2.1. (ontology/model) Let $\langle X, \Pi\rangle$ be an upper semilattice, then if $Y \subseteq X$, we define $\oplus Y$ as the unique z such that $\forall x \in Y : x\Pi z$ and $\forall x \in X : (\forall y \in Y : y\Pi x \rightarrow z\Pi x)$. We write $x \oplus y$ for $\oplus\{x, y\}$. Given a set X, we write X^* for the closure of X under $\oplus$. Let E be a set of entities. Let $\textsc{Pred}$ be a set of predicate symbols, with $\textsc{Ar}$ an arity function for $\textsc{Pred}$. A *model* is a structure $\langle I, E\rangle$, where I is a function from $\textsc{Pred}$ to tuples of length $\textsc{Ar}(\textsc{Pred})$ containing individuals from the E^* domain.

(interpretation) $M \models Pa_1 \ldots a_n$ if and only if $\langle a_1 \ldots a_n\rangle \in I_M(P)$ and $M =\!| Pa_1 \ldots a_n$ if and only if $\langle a_1 \ldots a_n\rangle \notin I_M(P)$

A *state* is a set of contexts of equal length. We often write S^i, where i specifies the cardinality of each of the contexts in S. Updates are performed on tuples $\langle n, S\rangle$, where S is a state and n is a counter specifying the current level.

Definition 2.2. (semantics) For $k \geq 1$:

$$
\begin{array}{rcl}
\langle n, S^i\rangle [\![\exists]\!] & := & \langle n, S'^{i+1}\rangle \text{ where} \\
& & S' = \{c : \langle n, i+1, d\rangle | d \in E^* \ \& \ c \in S\} \\
\langle n, S^i\rangle [\![\bar{\exists}]\!] & := & \langle n, S'^{i+1}\rangle \text{ where} \\
& & S' = \{c : \langle n, i+1, d\rangle | d \in (c^\bullet)^* \ \& \ c \in S\} \\
\langle n, S^i\rangle [\![P\mathrm{i}_1 \ldots \mathrm{i}_k]\!] & := & \langle n, S'^i\rangle \text{ where} \\
& & S' = \{(\mathrm{i}_1)\ldots(\mathrm{i}_k)c | c \in S \ \& \ M \models Pc(\mathrm{i}_1)\ldots c(\mathrm{i}_k)\} \\
\langle n, S^i\rangle [\![\mathrm{i} = \mathrm{j}_1 \oplus \ldots \oplus \mathrm{j}_k]\!] & := & \langle n, S'^i\rangle \text{ where} \\
& & S' = \{c | c \in S \ \& \ c(\mathrm{i}) = c(\mathrm{j}_1) \oplus \ldots \oplus c(\mathrm{j}_k)\} \\
\langle n, S^i\rangle [\![\varphi; \psi]\!] & := & (\langle n, S^i\rangle [\![\varphi]\!]) [\![\psi]\!]
\end{array}
$$

There are two quantifier actions. The ordinary existential quantifier $\exists$ expands the state such that each context now has a new index pointing to an individual in the domain. The action $\bar{\exists}$ differs from $\exists$ in that it assigns only contextually available individuals to the newly introduced index. In this way the two quantifiers constitute the indefiniteness/definiteness distinction.

Predicates are tests modulo permutation. If the condition of the predication succeeds, then a context is permuted in such a way that the arguments of the predicate form the head of the context in the order mirroring their syntactic rôles.

The distribution of indices to predicates is handled by the compositional semantics. I refer to (van Eijck 2000b) for a detailed exposition of the compositionality of incremental dynamics. In this paper I will take the compositional analysis for granted. For clarity I will write $\exists_{\mathrm{i}}$ for the quantifier action introducing index i into the context, but the reader should bear in mind that the reality of the formalism does not differentiate between quantifier actions with different indices. Another clarification method I will use is the graphic depiction of contexts. Here is an example.

(20.2) John saw Harry.
$\bar{\exists}_{21}; \mathrm{john}(21); \bar{\exists}_{22}; \mathrm{harry}(22); \mathrm{saw}\ 21\ 22$

Following the introduction of John and Harry we derive the context c', below (given a unique contextual *John* and *Harry*). Rows are assignments. The leftmost column represents the syntactic level of the referent represented by each row. The second column is the index or name of the referent. The final column specifies the value. Prominence is represented by the order of the rows. The top row has prominence zero, the second one, etc. So in c' the individual "h" is assigned to index 22 on level 0 and is the most prominent individual in the discourse. Following the predication "saw 1 2" the prominence hierarchy is permuted into c'', making John the more prominent of the two.

$$c' = \begin{array}{|c|c|c|}\hline 0 & 22 & \mathrm{h} \\ \hline 0 & 21 & \mathrm{j} \\ \hline \end{array}\ [\![\mathrm{saw}\ 21\ 22]\!]\ \begin{array}{|c|c|c|}\hline 0 & 21 & \mathrm{j} \\ \hline 0 & 22 & \mathrm{h} \\ \hline \end{array} = c''$$

2.3 Extending the system with syntactic levels

Since resolution mechanisms not only make use of a prominence hierarchy but also of a notion of *utterance, discourse unit* or any other segmentation of discourse, we need to express syntactic units. For this we have the first position in assignment tuples, but also the index keeping track of the current level during updating. We change levels[1] with "[" and "]". Here is the

[1] The use of *levels* is closely related to Visser and Vermeulen's work on dynamic bracketing, cf. (Visser and Vermeulen 1996).

semantics.[2]

$$\begin{array}{rcl}\langle n, S^i\rangle \,[\![\ [\ ]\!] & := & \langle n+1, S^i\rangle \\ \langle n+1, S^i\rangle \,[\![\ ]\ ]\!] & := & \langle n, S^i\rangle\end{array}$$

Next, I will consider *centering theory* as our candidate for integration with the formal system.

3 Centering Theory

Centering Theory (Grosz et al. 1995) is a theory of coherence in local discourse. In this paper we will embrace the basic ideas behind centering but nothing prevents us from altering some of its assumptions. In fact, the point is that we can make formal reference to the notions introduced by centering. We adopt here a specific version of centering known as the BFP-algorithm, (Brennan et al. 1987).

Central to centering is the assumption that each utterance produces a set of *Forward-looking centers* notated as (C_f). Furthermore, if an utterance is not discourse initial, then it is also assigned a backward-looking center or C_b. The elements in C_f are partially ordered reflecting the relative prominence in the utterance. Usually this order is thought to mirror the obliqueness hierarchy. The highest ordered element in C_f is most likely to be the future backward looking center. Centering theory calls this element C_p or *preferred center.*

Resolution occurs with the aid of *transition types.* Each type is defined in terms of C_b and C_f, i.e. a transition between utterances k and l is called a *shift* if $C_b^k \neq C_b^l$. Such a transition is called *retain* if $C_b^k = C_b^l$, but $C_b^l \neq C_p^l$. A *continue* transition occurs when the current backward-looking center equals the previous one and it is also the current preferred center. The transition types are ordered. This is what drives resolution: pronoun reference should be resolved to produce a transition which is least costly. The transition order is *continue* < *retain* < *smooth-shift* < *rough-shift*, where a shift is smooth if $C_b^l = C_p^l$ and rough otherwise. The complete centering strategy towards resolution can now be captured in an algorithmic fashion. First of all, all the possible resolutions are produced. Next the following possibilities are filtered out: (i) those which do not obey syntactic binding constraints, (ii) those which do not have the current backward-looking center mapped to element in the current C_f which is highest in the C_f of the preceding utterance and (iii) those where all pronouns, if there are any,

[2]The semantics of the closing bracket as given here is no doubt problematic, since it allows multiple utterances to be represented on the same level. In this paper however, the "] "-action plays no rôle. Still, to give a hint of an alternative definition we might propose that the closing bracket throws away all the referents introduced at the level to be closed. This means we want the output state to be $S'^j = \{c|c' = c\lceil\{0,\ldots,n\} \,\&\, j = |c'|\}$. Similar definitions could open the door to formalisms not only representing discourse units, but even *discourse structure* (cf. (Roberts 1998), (Polanyi 1988)).

fail to refer to the current C_b. We will refer to this final filter as Rule 1. Following these constraints we are left with a set of possible resolutions, each of which corresponds to a certain transition type. The least costly possibility is considered the optimal resolution.

I give a brief example. Consider: *"1. John has been having a lot of trouble arranging his vacation. 2. He cannot find anyone to take over his responsibilities. 3. He called up Mike yesterday to work out a plan. 4. Mike has annoyed him a lot recently. 5. He called him at 5 am on Friday last week."* (cf.(Grosz et al. 1995)(20)).

Say that we already processed (and resolved) everything up to the one but final sentence. Here are the relevant center assignments.

	1.	2.	3.	4.
C_b	$\emptyset$	John	John	John
C_f	⟨John⟩	⟨John⟩	⟨John,Mike⟩	⟨Mike,John⟩

For the resolution of the pronoun in (4.) we had a choice between resolving *him* to John or to Mike. Rule 1 as well as binding constraint B tell us to refer to John. Since the preferred center is no longer the backward-looking center, we may call (4.) a retain transition. Now we consider the final sentence. We have two possible resolutions (after application of binding constraints). The current C_b is Mike. We have a choice between $C_f = \langle \text{Mike}, \text{John} \rangle$ or $C_f = \langle \text{John}, \text{Mike} \rangle$. The first one produces a smooth shift, the latter a rough one. This means we will have to resolve *he* to Mike and *him* to John.

A different way of looking at centering is the view that it is simply a formalism on tuples. Given the logic we introduced in Section 2 we can view states as sets of forward looking center lists. Transition types are thus a notion directly related to the extended incremental dynamics system.

We write c^l for the assignments in c which share level l, i.e. c^l gives us the forward looking center list of discourse unit l. This also means that $c^l[0]$ is the preferred center for that level.

Continuing this line we can define the backward looking center of a discourse segment l as follows:[3]

$$\begin{aligned} c^l_{\leftarrow} &:= \begin{cases} 0 & c^{l-1} = \emptyset \\ \iota p.\exists q(c^l[p]^\bullet = c^{l-1}[q]^\bullet \ \&\ \forall p', q'(c^l[p']^\bullet = c^{l-1}[q']^\bullet \to q < q')) & \text{otherwise} \end{cases} \\ c^l_b &:= c^l[c^l_{\leftarrow}] \end{aligned}$$

We can now define the two transition building constraints as predicates over

[3] The original definition of backward looking center has it that no C_b is assigned to the discourse initial sentence. I follow Beaver (Beaver 2000) in selecting the subject in that case.

contexts and levels.[4]

$$\begin{array}{rcl} \text{Cohere}(c,l) & :\Leftrightarrow & c_b^l = c_b^{l-1} \\ \text{Align}(c,l) & :\Leftrightarrow & c_b^l = c^l[0] \end{array}$$

The transitions types are:

$$\begin{array}{rcrcr} \text{continue}(c,l) & :\Leftrightarrow & \text{Cohere}(c,l) & \wedge & \text{Align}(c,l) \\ \text{retain}(c,l) & :\Leftrightarrow & \text{Cohere}(c,l) & \wedge & \neg\text{Align}(c,l) \\ \text{smoothshift}(c,l) & :\Leftrightarrow & \neg\text{Cohere}(c,l) & \wedge & \text{Align}(c,l) \\ \text{roughshift}(c,l) & :\Leftrightarrow & \neg\text{Cohere}(c,l) & \wedge & \neg\text{Align}(c,l) \end{array}$$

Up to now we have simply shown that our extended incremental dynamics is capable of fully incorporating centering theory. Now it is time that we start thinking about plurality.

4 Resolving plurals

One of the strategies used by language to introduce plural referents is coordination. Conjunctive NPs unite two (singular or plural) individuals into a new plural one. In the current system, this is modeled well by making use of the summation operator $\oplus$ and of the =-predicate. In fact, this is similar to the way coordinated NPs are handled in e.g. DRT (see (Kamp and Reyle 1993):434). For instance, the NP *a man and a woman* is represented as $\exists_1; \text{man } 1; \exists_2; \text{woman } 2;\ \exists_3; 3 = 1 \oplus 2$. Plural pronouns can subsequently link to such plural referents. But an interesting feature of these constructions is that the subparts of the newly formed plural do not inherit salience from the position of the plural NP. Consider for example the contrast in (20.3).

(20.3) i) Tom talked with Dick. He spoke very softly.

ii) Tom and Mary talked with Dick. He spoke very softly.

In (20.3ii) *he* is readily resolved to refer to *Dick* even though *Tom* is in subject position. Another example is the contrast in (20.4).

(20.4) i) Tom, Susan and Mary went to a party the other day.
?They wore beautiful dresses.

ii) Tom took Susan and Mary to a party the other day.
They wore beautiful dresses.

Our system so far does a good job accounting for this fact. Given that it is predication which drives the salience game, the plural *Tom, Susan and Mary* is permuted to the front of the context. The singular parts of this

[4]The names of these predicates were inspired by David Beaver's optimality theoretic reformulation of the BFP-algorithm in (Beaver 2000).

referent play no rôle in any predication, so they are even less salient than the referents contributed by *a party* and *the other day*.

We thus easily predict inferior salience for parts of conjunctive NPs. Here is a detailed example. The figure in (20.7) shows a possible context update of (20.5)/(20.6), in a model where f is a farmer, m a man and w a woman and with m and w hitting f.[5]

(20.5) A man and a woman hit a farmer.

(20.6) $\exists_1; \text{man } 1; \exists_2; \text{woman } 2; \exists_3; 3 = 1 \oplus 2; \exists_4; \text{farmer } 4; \text{hit}^* \ 3\ 4$

(20.7)

0	4	f
0	3	$m \oplus w$
0	2	w
0	1	m

$\text{hit}^* \ 3\ 4$

0	3	$m \oplus w$
0	4	f
0	2	w
0	1	m

It is clear that the farmer is now more salient than the man. Subsequent anaphoric reference will thus prefer picking up the farmer-referent over the sub-individual of the plural subject.

I propose translating singular pronouns as $\bar{\exists}_\text{i}; sg \ \text{i}$ and plural ones as $\bar{\exists}_\text{i}; pl \ \text{i}$. Here, sg and pl are predicates such that in our model $I(sg) = E$ and $I(pl) = E^* - E$. Continuing (20.5) with e.g. *He was angry* asks for a subsequent update of the final context in (20.7) with $[; \bar{\exists}_5; sg \ 5$. The resulting state is:

0	3	$m \oplus w$
0	4	f
0	2	w
0	1	m

$[; \bar{\exists}_5; sg \ 5$

1	5	f
0	3	$m \oplus w$
0	4	f
0	2	w
0	1	m

1	5	w
0	3	$m \oplus w$
0	4	f
0	2	w
0	1	m

1	5	m
0	3	$m \oplus w$
0	4	f
0	2	w
0	1	m

There are a few complications here, though. First of all, centering's Rule 1, rules each of the resulting contexts out. Because of agreement, none of them has the pronoun referring to the backward looking center $m \oplus w$. Furthermore, would we disregard Rule 1, then all resulting contexts are simply (smooth) shifts. This is not against our intuition – we are changing the topic from the man and the woman to the farmer. However, we still want some way of expressing the fact that the farmer is the most salient *singular* referent and the fact that the farmer is accessible with pronominal reference. The key to the solution of these problems seems to lie in agreement. I propose here that the tension between transition types of resolutions is dependent on agreement. This means that as far as plural reference is concerned $m \oplus w$

[5] I write P^* for the pluralized version of predicate P. This also means that in this paper the distributive/collective distinction will be ignored.

is the backward-looking center in (20.7). For singular reference, though, it is the farmer who is the C_b.

Another problematic issue for the centering component is that of split antecedents.

(20.8) John took Mary to the cinema. They enjoyed the movie.

How will we be able to relate the plural construction in the second sentence to the two singulars in the first? It is easy to see that the setup thus far predicts the c_b of the second sentence to be absent (a violation of Rule 1). The example in (20.8), however, is fine. The reason for this is that although the plural individual consisting of John and Mary was not explicitly introduced in the first sentence, this group still inherits some of the salience of its parts. This is also illustrated by (20.9).

(20.9) John took Susan to see the Beatles. They...

Although the NP *the Beatles* is the only plural in the first sentence, the construed plural John and Susan is at least as salient as are the fab four (we are for instance not forced to repeat the names of John and Susan when we want to comment about the fact that they had a good time). This problem can be solved when we again assume agreement to be active during resolution. That is, processing the first sentence also creates the possibility to assign prominence to implicit plurals. In (20.9) I therefore take John to be the singular C_b while John and Susan form the plural C_b.

Implementing the ideas presented in the previous sections will happen through two operations on contexts. One, the *singular closure* of a context is the prominence hierarchy of all the singular referents. The other, the *plural closure*, restricts the context to plural referents, but also includes construed plurals and calculates their salience.

First we define the singular closure of a context.

$$sg(c) = \lambda a.a \in c \ \& \ a^\bullet \in E$$

Viewing sequences of assignments as four-place tuples, the plural closure creates a partially ordered C_f by allowing the position slot to receive the same index more than once. Construed plurals inherit the combined salience of their parts. Since the name of these referents is irrelevant, I assign them the dummy label $\star$. I write c^+ for the sum of all the positions taken in c. For a linearly ordered c of say length 2 this means that $c^+ = 1$, given that the first position is called zero. I write $maxl(c)$ for the maximum level in c.

$$pl(c) := \{\langle c'^+, maxl(c'), \star, \oplus(c'^\bullet)\rangle | c' \subseteq c \ \& \ \oplus(c'^\bullet) \in E^* - E\}$$

Thus, $pl(c)$ is the set of plural constructions possibly attainable from c where prominence is calculated by summating the prominence of the parts.

Given these definitions we can define the backward-looking centers.

$$\begin{array}{rcl} c^l_{b,sg} & := & (sg(c))^l[(sg(c))^l_{\leftarrow}] \\ c^l_{b,pl} & := & (pl(c))^l[(pl(c))^l_{\leftarrow}] \end{array}$$

Correspondingly we can weaken the two transition building constraints.

$$\begin{array}{rcl} \text{Cohere}(c, l) & :\Leftrightarrow & \exists v : c^l_{b,v} = c^{l-1}_{b,v} \\ \text{Align}(c, l) & :\Leftrightarrow & \exists v : c^l_{b,v} = c^l[0] \end{array}$$

Finally I define a pronoun resolution operator. Let S^i be a state and let n be a level defined for S. We define:

$$\lfloor S^i \rfloor_n \ := \ \lambda c.\neg\exists c' \in S : c' \prec_n c$$

Here the ordering is as follows. The relation $\prec_n$ is the minimal relation such that:

$$\begin{array}{c} \text{Cohere}(c, n) < \text{Cohere}(c', n) \rightarrow c \prec_n c' \\ \& \\ \text{Cohere}(c, n) = \text{Cohere}(c', n) \rightarrow (\text{Align}(c, n) < \text{Align}(c', n) \rightarrow c \prec_n c') \end{array}$$

This means that $\lfloor S \rfloor_n$ filters out all contexts but the ones representing the least costly transition at level n.

5 Final remarks

First of all, let me remark that there are a few reasons why this paper can only be considered to be a preliminary study. There is so little empirical data concerning plural resolution[6] that the generalizations I have made here are very likely to be too general. Also, it is clear that the centering framework is far from optimal; the system proposed in this paper inherits most of the problems inherent in centering.

There is also room for improvement by changing some design choices. For instance, the permutable contexts have the disadvantage that salience can only be expressed relative to other referents. This means that if we have a sentence with a conjunctive subject as the only NP, the subparts of this NP end up relatively high in the prominence hierarchy since there are no other referents to intervene. If we compare a sentence *A and B Pred*, with *A Pred B*, then in both cases B is second in the C_f. We might remedy this by letting predicates assign rôles, i.e. they will assign position 0 to their subject, 1 to their object, etcetera. In a system like that, existential introduction should assign some notion of *non-salience* to a new index and we should use "[" and "]" for a richer notion of constituency in the logic.

[6] But see (Koh 2001) for some recent interesting psycholinguistic results.

Here is what was achieved: Enriching contexts with labels expressing syntactic levels opens the possibility of equating contexts with sequences of forward-looking center lists. Applying the formalized notion of backward-looking center to a singular or plural closure of such contexts enables us to deal with some complicating issues in plural pronoun resolution.

References

Beaver, D. (1999). The logic of anaphora resolution. In P. Dekker (Ed.), *Proceedings of the 12th Amsterdam Colloquium*, pp. 61–66. ILLC Universiteit van Amsterdam.

Beaver, D. (2000, July). The optimization of discourse. Unpublished manuscript.

Brennan, S., M. W. Friedman, and C. Pollard (1987). A centering approach to pronouns. In *Proceedings of the 25th annual meeting of the ACL*, Stanford, pp. 155–162.

Groenendijk, J. and M. Stokhof (1991). Dynamic Predicate Logic. *Linguistics and Philosophy 14*, 39–100.

Grosz, B., A. Joshi, and S. Weinstein (1995). Centering: a framework for modelling the local coherence of discourse. *Computational Linguistics 21*(2), 203–226.

Hardt, D. (1999, April). Dynamic interpretation of verb phrase ellipsis. *Linguistics and Philosophy 22*(2), 187–221.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic.* Dordrecht: D. Reidel.

Koh, S. (2001, february). *Resolution of the antecedent of a plural pronoun.* Ph. D. thesis, Graduate School of the University of Massachusetts Amherst.

Polanyi, L. (1988). A formal model of the structure of discourse. *Journal of Pragmatics 12*, 601–638.

Roberts, C. (1998). The place of centering in a general theory of anaphora resolution. In M. Walker, A. Joshi, and E. Prince (Eds.), *Centering Theory in Discourse*, pp. 359–400. Oxford: Clarendon Press.

van Eijck, J. (2000a). Context semantics for NL. Unpublished manuscript.

van Eijck, J. (2000b). On the proper treatment of Context. In *Proceedings of CLIN99*, Utrecht.

van Eijck, J. (2001, Summer). Incremental Dynamics. *Journal of Logic, Language and Information 10*(3), 319–351.

Visser, A. and C. Vermeulen (1996). Dynamic bracketing and discourse representation. *Notre Dame Journal of Formal Logic 37*, 321–365.

Relational Concept Analysis for Structural Disambiguation

Jerome Piat

RACE Laboratory, University Of Tokyo, Komaba 5-4-1, Meguro-ku, Tokyo 153-8904, Japan

piat@race.u-tokyo.ac.jp

Abstract. Many methods currently used to disambiguate some grammatical structures are statistical or probabilistic ones and the knowledge learned is usually at the level of abstraction of the example itself, thus difficult to reuse for examples newly encountered. This paper aims at proposing a framework to solve some grammatical ambiguities using a Relational Concept Analysis. Such framework is used here to disambiguate a Japanese structure referred to as the Genitive Case Japanese *No* particle - Ga/No conversion by the use of two main methods that participate in correctly solving the syntactic ambiguity : a computational logical analysis, establishing logical semantic paths between concepts, and an empirical preference rule system, using conceptual rules learned from examples to help tackle cases difficult to solve by the first method. Practical results show that the proposed framework is computationally manageable, and can provide some adaptive conceptual generalization as well as some knowledge enrichment toward the understanding of the meaning of language structures.

1 Introduction

The Japanese grammatical structure *A no Adj B*, with A and B two nouns and Adj a regular adjective is grammatically ambiguous, when no oral or written marker is present. Indeed, Adj can refer to A or B, or both. This grammatical ambiguity will be later referenced as GCN ((Shinichi 1971), (Miyagawa 1992)). A system could rely on some statistical analysis, but the knowledge learned may be directly dependent on the number of occurrences encountered at the level of examples and thus sensitive to irregularities and lacks in the coverage of representative examples as described in (et al. 98). The present work aims at proposing a concept based framework to solve such ambiguity as well as some similar ones by using a network of concepts, and the construction of semantic paths between them in order to find dynamically the correct grammatical attachment afterward. The approach introduced here is class based in order to avoid as much as possible problems induced by the sparseness of data in training sets. Compared to some previously known class based approaches ((et al. 98), (Kurohashi and Sakai

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

1999)), the present one generalizes from examples, then constructs conceptual paths at learning time, or searches for some specific types of semantic paths at decision time to find appropriate conceptual bindings. This approach requires a rich database of concepts, that can link them together by different semantic relations. Wordnet ((Miller et al 1993), (Miller 1993)) has been used for the present study.

The first section gives a brief linguistic analysis of the Genitive case Japanese No ambiguity, a description of the system developed to tackle similar ambiguities is then described in two steps, then results obtained from the disambiguation of GCN are presented and discussed.

1.1 Linguistic analysis

Semantically non-ambiguous cases

1. Hanako no ookii me (Hanako's big eyes).
2. me no ookii Hanako (Big-eyed Hanako).

The syntactic structure is exactly the same in both cases. The adjective *ookii (big, large)* refers to the noun *me (eye)* in both sentences. Though inherently semantically ambiguous, it is not ambiguous for a human interpretation.

3. sono hito no kireina me (the beautiful eyes of this person).

The Japanese language uses two types of adjectival words: one type directly referred to as adjectives that end with the hiragana suffix -i, and another type followed by the hiragana suffix -na and referred here as nominal adjectives. The nominal adjective in the example 3. for instance attaches to the noun on the right of the structure. This means that the structure is syntactically not ambiguous in such case. A more detailed study can be found in (Kanzaki and Isahara 1997).

Semantically ambiguous cases

4. ? badominton-kooto no hiroi sentaa.

This sentence is syntactically and semantically ambiguous. In case the context is omitted, two alternative interpretations could reasonably be drawn by a Japanese native speaker:

- a. the center that contains a large badminton court.
- b. the large center that contains a badminton court.

As a primary conclusion, it appears that solving the GCN structure requires both a syntactic and a semantic approach. At the semantic level, a conceptual approach is needed to better analyse the semantic relations the two nouns of the structure have between each other. A system that is supposed to find the correct attachment is then expected to find whether the sentence is ambiguous for humans or not, and if not, determine the appropriate attachment.

1.2 The Relational Concept Approach

The Principle of Minimality

The approach to solve the GCN ambiguity is partly based on a maxim of quantity described by Grice as one part of a basic principle of communication in pragmatics: the principle of minimality indicating that something not adding information is generally omitted. This principle does not strictly belong to the Concept Relational Approach but participates in solving the GCN from the logical inclusion point of view and is one of several principles that can draw a semantic search within that framework.

The Conceptual Knowledge database

Though the present work deals with a Japanese structure, the English version of the Wordnet database has been used for its richness of information, and accessibility, providing also a valuable library of functions to process examples.

Formalization

Let C_i be a concept defined here as a non-intentional internal representation of information that can serves as the meaning of a linguistic entity, represented as a synset in the Wordnet database ((Miller et al 1993)). The relation $\sigma_i(C_j)$ is true if and only if there exists a semantic arc from C_i to C_j following a type of relation u. Some Wordnet relation types are symmetrical but haven't been used here, only the two asymmetrical relation types: hyponymy and meronymy are used. For instance, if u is the hyponymy relation, $\sigma_u(C_i)$=C_j is true if and only if C_j is the superordinate or hypernym of C_i, which can be written also C_i *"is a"* C_j.
The semantic arc from C_i to C_j following the relation u is noted $s_u(C_i, C_j)$.The semantic oriented homogeneous path from C_i to C_j following a unique relation type u is then so that: $p_u(C_i, C_j)= \bigcup_{k=i}^{j} s_u(C_k, C_{k+1})$ with i, j, and k $\in \mathbb{N}$.
A path is said to be homogeneous if and only if it is composed of semantic arcs having all the same type of relation, it is called heterogeneous otherwise. An heterogeneous path from C_i to C_j is composed of homogeneous paths from C_i to C_j possibly following different types of relations and noted $P_n^a(C_i, C_j)= \bigcup_{k=0}^{n} p_k$ with n defining the number of different relation types from C_i to C_j, a a unique identifier, and p_k an homogeneous path between two concepts.

Unilateral heterogeneous logical inclusion paths

As described just before, a first part of the study based on Grice minimality principle is so that if one of the two nouns of the GCN structure is conceptually an intrinsic part of the second noun, then the adjective is likely to determine the noun that contains the other, otherwise, the included noun would be merely useless. As a simple example, in *me no ookii hito (big-eyed person)*, *me (eye)* being an intrinsic part of a person, adding it in the structure would not add useful information if the adjective *ookii (big)* was qualifying *hito (person)*. Therefore, the adjective implicitly attaches to the noun that is a constituting part, or property of the other in that case.
The goal of this first part is to find some paths of logical inclusion based on the general definition of a path given in section 1.2, using a logical decision criteria $\mathcal{A}_l(C_i,C_j)$ for two concepts C_i and C_j, with Card($\{P_k^m(C_i,C_j)$, $m \in \mathbb{N}\}$) also noted in short Card($\{P_k,i,j)\}$) the total number of heterogeneous logical inclusion path occurrences linking C_i to C_j, a reliability criteria $r(i,j) = \frac{Card^2(\{P_k,i,j\})-Card^2(\{P_k,j,i\})}{Card(\{P_k,i,j\})+Card(\{P_k,j,i\})}$, and $t_0 \in \mathbb{R}_+^*$ a threshold parameter.

$$\mathcal{A}_l(C_i,C_j) =$$
$$\begin{cases} \text{undecidable} & \text{if Card}(\{P_k(C_i,C_j\}) + \text{Card}(\{P_k(C_j,C_i\}) = 0 \\ \text{ambiguous} & \text{if } |r(i,j)| < t_0 \\ \text{left attachment} & \text{if } r(i,j) \geq t_0 \\ \text{right attachment} & \text{if } r(i,j) \leq -t_0 \end{cases}$$

Practically, three types of logical inclusion paths have been instantiated:

Unary logical paths where the full logical inclusion path is $P_u^k(C_i,C_j)$ $= p_u(C_i,C_j)$ with u an hypernymy or meronymy semantic relation. These paths are direct semantic paths for which one element is directly found to be a conceptual part member or instance of the other. This is the only type where $P_u^k(C_i,C_j)$ is homogeneous, that is composed of only one type of relation.

Binary logical paths where the path $P_u^k(C_i,C_j) = p_u(C_i,C_m) \bigcup p_v(C_m,C_j)$ with the pair (u,v) being for instance (is part, is a), (is a, has member), or (is a, has part). In this case, the full path is heterogeneous as being composed of two semantically distinct homogeneous paths. m represents here the index of the concept C_m so that $\{C_m\} = p_u(C_i,C_m) \bigcap p_v(C_m,C_j)$.

Ternary logical paths where $P_u^k(C_i,C_j) = p_u(C_i,C_m) \bigcup p_v(C_m,C_n) \bigcup p_w(C_n,C_j)$ with (u,v,w) is a combination of three distinct hyponymic or meronymic homogeneous sub-paths with m and n so that: $\{C_m\} = p_u(C_i,C_m) \bigcap p_v(C_m,C_n)$ and $\{C_n\} = p_v(C_m,C_n) \bigcap p_w(C_n,C_j)$.

Searching for the existence and redundancy of these paths has proved to provide some valid information to disambiguate the GCN structure. Results are given in section 2.2. Figure 21.1 illustrates one binary and ternary paths of logical inclusion. Within each one, there is at least one common concept C_m bridging two different but logically complementary semantic sub-paths $p_u(C_i, C_m)$ and $p_v(C_m, C_j)$ so that at least one binary or ternary logical path of inclusion can be created to connect the first concept C_i with the second C_j.

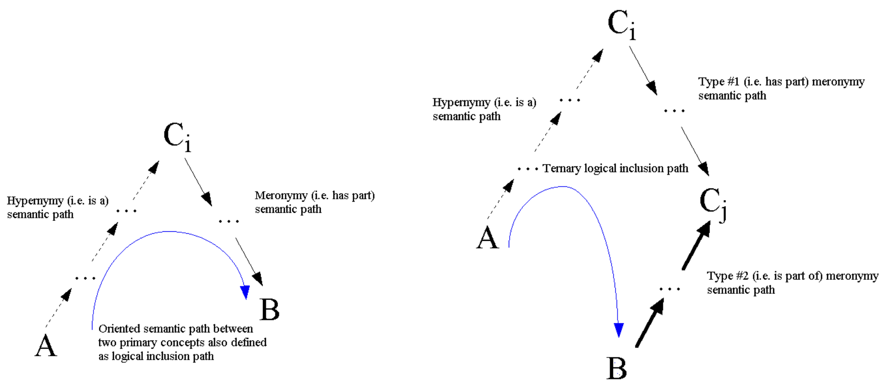


Figure 21.1: Binary and ternary logical inclusion paths.

Whether the logical inclusion paths found are unary, binary, or ternary, the reliability parameter *r(i,j)* is calculated afterward by including them all to determine the appropriate attachment. Some more complex paths could be searched, but the current implementation has been limited to ternary paths. It should be noted that not all binary or ternary combinations *(u,v,w)* are logically valid. For instance if a concept C_i is part of a concept C_m which has a part that is C_j, it cannot obviously be deduced that C_i is an intrinsic part of C_j. Thus only a restricted number of combinations have been used to search valid paths. Moreover, the search can be constrained in depth to limit the combinatorial explosion of possibilities especially in case of ternary paths like *(*is part, has part, is a) where a concept C_j is a C_m that has a part which is a C_i. It should be noted however that average computational time was short, generally few seconds on a PC Pentium III 750 Mhz with 128 Mb of RAM memory.

Learning Empirical semantic paths

The creation and learning of empirical preference rules based on examples was needed in order to:

- Extend Wordnet semantic relations for nouns and relate concepts like *sword* with *warrior* or *rent* with *apartment* by creating and adapting general rules at some low, medium or high levels of abstraction and create new semantic relations like *used by* for instance.

- Solve indecision cases for logically independent concepts.

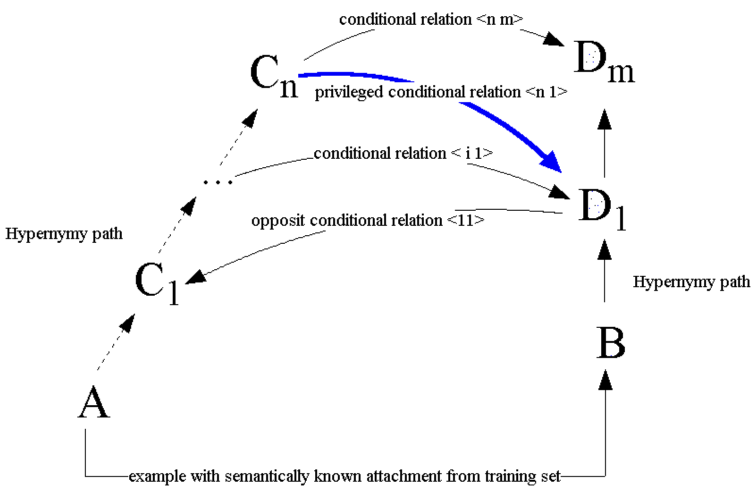


Figure 21.2: Learning preference rules at the right level of abstraction.

Formally, this second step intends to construct empirical paths from two logically disconnected concepts C_i and C_j in order to recreate, from examples, preference rules that will help to solve similar types of combinations by a generalization process. An empirical path is then composed of:

Two homogeneous hyponymic sub-paths $p_h(C_i, C_m)$ and $p_h(C_j, C_n)$ (h being the hyponymic relation) that are independent in the sense that they do not share common concepts.

A semantic arc $s_{u(C_r,C_s)}(C_m, C_n)$ artificially added between one concept C_r of the sub-path $p_h(C_i, C_m)$ and a concept C_s of the other sub-path $p_h(C_j, C_n)$. Here u designate one new type of relation between C_r and C_s that is none of those described in Wordnet database ((Miller et al 1993), (Miller 1993)).

An empirical path k linking two concepts C_i and C_j via an hyponymic sub-path $p_h(C_i, C_m)$, an artificial semantic arc $s_{u(C_r,C_s)}(C_m, C_n)$, and another hyponimic sub-path $p_h(C_j, C_n)$ is written:
$\mathcal{E}^k_{u(C_m,C_n)}(C_i, C_j) = p_h(C_i, C_m) \bigcup s_{u(C_r,C_s)}(C_m, C_n) \bigcup p_h(C_j, C_n)$.
The cardinality of $s_{u(C_r,C_s)}$ noted $Card(s_{u(C_r,C_s)})$ represents the number of learned examples (C_i, C_j) so that there exists a path $\mathcal{E}^k_{u(C_m,C_n)}(C_i, C_j)$ with m, n, i, and j distinct indexes $\in \mathbb{N}$. A reliability criteria for a preference rule is $r(s) = \frac{Card^2(s_{u(C_r,C_s)}) - Card^2(s_{u(C_s,C_r)})}{Card(s_{u(C_r,C_s)}) + Card(s_{u(C_s,C_r)})}$ with $Card(s_{u(C_r,C_s)}) +$

$Card(s_{u(C_s,C_r)}) \neq 0$. Let t_1 be a positive threshold parameter, the preference rule decision criteria A_e is obtained by calculating the average value of the n_0 first most reliable rules for two C_i and C_j as follows:

$$A_e(C_i, C_j) = \begin{cases} \text{ambiguous} & \text{if } \frac{1}{n_0+1} * |\sum_{k=0}^{n_0} r_{s_k}(i,j)| < t_1 \\ \text{left attachment} & \text{if } \frac{1}{n_0+1} * \sum_{k=0}^{n_0} r_{s_k}(i,j) \geq t_1 \\ \text{right attachment} & \text{otherwise} \end{cases}$$

2 Results

2.1 Results from examples of the Japanese literature

Nine hundred examples of the GCN structure have been extracted from the Japanese literature. For each example, three native speakers have been asked to add two types of information:

- The semantic attachment of the adjective to the left or right noun.
- The existence of an intrinsic logical inclusion between the concepts referenced by the two nouns. Practically, this logical inclusion uses only the meronymy type of relation of the Wordnet database. In the GCN structure *A no Adj B*, A is said to be logically included in B if and only if A is thought of being an intrinsic part of B by native speakers for that only experience, this is noted in the table below by the relation $A \sqsubset B$ and $A \sqsupset B$ otherwise.

Tests have been run independently and were blind in the sense that testers were not indicated the purpose of the tests. No notion of ambiguous attachment have been mentioned to the speakers thus no information have been collected on that issue here. Each percentage value given here is the average of percentage values from the three testers and represent the relative proportion of examples of one type of logical inclusion among all examples for the same semantic attachment (left or right).

attachment $\diagdown$ inclusion	$\sqsupset$	$\sqsubset$	$\emptyset$
right attachment	52.07%	2.25%	45.68%
left attachment	0.0	39.28%	60.72%

Table 21.1: Semantic attachments results by a human interpretation.

Two conclusions can be drawn from such results though the data set is relatively small and not representative of all possible combinations of the Japanese language for the GCN structure. The first one is that the total number of examples involving the search for some potential logical inclusion paths is relatively high. Secondly, in case native speakers think that there is a logical inclusion between the two concepts, that is results of the two first columns, there is a strong correlation between the direction of the attachment and the direction of the logical inclusion. This hinted for a conceptual relational approach to tackle a part of the problem. However, it should be noted that the existence of such correlation does not prove that logical inclusions paths are the underlying cognitive dynamic structures used to partially solve the ambiguity, though reconstructing them is used afterward as a method to solve the GCN structure. Some deeper cognitive study is thus needed at this point. Following results tend to confirm that correlation.

2.2 The search for logical inclusion paths

Out of these nine hundred examples, one hundred have been extracted so that side effects of translations from Japanese to English were minimized. All these examples have been considered as non-ambiguous by Japanese native speakers and thus having the adjective attaching either to the left noun or to the right noun only. Moreover, very abstract Wordnet synsets like *entity* or *something* have been ignored on purpose in the search of logical paths in order to avoid too abstract and meaningless matching. The following table shows, for the columns of left and right attachment, the percentage of correct semantic attachments found by the system on the sole search for inclusion paths.

Left attachment	Right Attachment
72.7%	75.5%

Table 21.2: Disambiguation results from the search of logical inclusion paths.

Results confirm that by finding some logical inclusion paths between concepts, following the decision criteria given in section 1.2, some promising results are found when there is originally some logical inclusion between the two nouns (or concepts in Wordnet). This can be generalized to arbitrary semantic paths depending on the grammar structure analyzed.

2.3 Learning semantic preference rules

As discussed before, searching for logical semantic paths cannot solve entirely the ambiguity as a significantly high number of real examples are not involving any logical inclusion relation, as confirmed by Wordnet. Consequently a system has been implemented to create and learn preference rules at the appropriate levels of abstraction to link unrelated hyponymic paths, and create empirical semantic paths as described in 1.2 for appropriate semantic bindings. The two most significant results are:

- Semantic rules, weighted by some adaptive reliability value, reveal step-by-step some useful abstract information to disambiguate the GCN. this could be for instance that a *rent* applies to artifacts and therefore that the adjective attaches to the left in an example like *yachin no takai heya (a room with an expensive rent)*. This type of information could be hardly extracted by a non-conceptual approach, participate directly in the generalization process, and could be stored in an efficient way in memory.

- Ambiguous examples for native speakers are also found ambiguous by the system, and at the opposite examples that are not considered ambiguous by humans have been generalized using a rather few number of examples by the system as counter-examples are rare by definition.

As one example, here is listed in order of reliability some of the most reliable rules calculated after 20 examples have been learned with closely related words like *payment*, *rent*, *cost* as the right noun of the GCN, and as the left noun some names of everyday life goods like *television*, *table* or *book*:

cost ⌈ *by rule #523* ⌋ applies to physical object
outgo ⌈ *by rule #608* ⌋ applies to physical object
expenditure ⌈ *by rule #693* ⌋ applies to physical object
cost ⌈ *by rule #521* ⌋ applies to artifact
outgo ⌈ *by rule #606* ⌋ applies to artifact
expenditure ⌈ *by rule #691* ⌋ applies to artifact

Here, rules found are rather abstract because the search depth has not been limited to low values. This means that if there are some contradictory examples afterward, these rules are very likely to have their reliability decreased because of their high level of abstraction, and most reliable rules will have a degree of abstraction lower but more appropriate to what have been learned. At decision time, inputting a completely new example like *nedan no sugokutakai fune (a very expensive boat)*, the system can correctly answer that the adjective attaches to the left noun *nedan (price)* even though words may be encountered for the first time.

2.4 Benefits and current limits

The combinatorial explosion of possibilities is not so high. The search for relation between concepts is limited by the fact that abstract concepts are much more rare than less abstract ones, and thus the processing is generally very quick at learning time and at decision time. In addition, the knowledge induced by each concept in relation to others can be kept local so that the creation of potential paths is done using a limited set of possible connections, and changes made only locally.

Moreover, when learning preference rules, each example influences considerably low, middle and high level rules, improving considerably the attachment decision for unlearned semantically similar examples. One example can modify several hundreds of abstract preference rules, and rules being generated at all levels of abstraction, the training phase does not need a high number of examples. This approach is also valuable for some deep understanding of the meaning of linguistic structures as understanding the appropriate semantic attachment give the possibility to draw new inferences from acquired and related knowledge.

Concerning limits and possible improvements of the approach, results depend highly on the richness of information found in Wordnet. It has been noticed that the information is not homogeneously distributed, this being valid for common concepts as well. Some noise and side effects have been inevitably induced by the necessary translation though examples have been chosen so that these problems are as few as possible. The fact that a good proportion of words are polysemous induce changes at all levels of abstraction when learning preference rules, and thus may modify inconsistently many relations at all levels. Pondering the reliability of preference rules by criteria like their degree of abstraction is being investigated.

Moreover, exact matching is sometimes not possible but some coordinate concepts may participate significantly in creating analogous semantic bindings. Therefore, introducing some semantic distance between concepts as in (Resnik 1992) may improve sensibly the efficiency of the system, at the expense of its speed. In addition, current preference rules are rather simple, and constructed to create appropriate local conceptual bindings. Learning more complex common sense based rules is currently studied.

3 Conclusion

A Relational Concept Approach can significantly participate in solving structural ambiguities, and single out the possible existence of some cognitive processes underlying the understanding of some linguistic structures. Moreover, such method can add implicit knowledge to the understanding of the structure by the analysis itself, providing a preference rules system based on learnable connection rules, and semantic paths that can generalize appropriate semantic attachments and draw further inferences toward a generalized understanding of the structure. Such framework can also be applied to grammatical ambiguities similar to the GCN structure like the Japanese grammatical structure *Adj Noun no Noun* or the English structure *Adj Noun Noun*. It can be supplemented by some collocation statistical or probabilistic methods and be used to semantically enrich the construction of the meaning of linguistic expressions.

References

et al., H. W. (98). Structural Disambiguation Based on Reliable Estimation of Strength of Association. *COLING-ACL*.

Kanzaki, K. and H. Isahara (1997). Lexical Semantics for Adnominal Constituents in Japanese. *NLPRS*.

Kurohashi, S. and Y. Sakai (1999). Semantic Analysis of Japanese Noun Phrases: A New Approach to Dictionnary-Based Understanding. *ACL*.

Miller, G. (1993). Nouns in Wordnet : A lexical Inheritance System, Revised Edition.

Miller et al, G. (1993). Introduction to Wordnet : An On-Line Lexical Database. Revised Edition.

Miyagawa, S. (1992). Case, Agreement, and GA/No Conversion. *Third Annual Southern California Japanese/Korean Linguistics Conference, San Diego in Japanese Korean Linguistics, Volume 3, Soonja Choi, SLA Stanford Editor*.

Resnik, P. (1992). Wordnet and Distributional Analysis: A Class-Based Approach to Lexical Discovery. *AAAI Workshop on Statistically-based Natural Language Processing Techniques*.

Shinichi, H. (1971). Ga-no conversion and idiolectal variations in Japanese. *Gengo kenkyuu 60*.

A Model-Based Semantics of the Mood Morphemes in Hungarian

György Rákosi
University of Debrecen, Hungary
rakosigy@hotmail.com

Abstract.

This paper outlines a novel model-based approach to the analysis of the basic semantic properties of Hungarian mood morphemes.[1] I assume that these morphemes are assigned a basic semantic feature already in the lexicon and they contribute this basic semantic impact to the interpretation of the sentence in which they occur. Though the present proposal does not aim to achieve anything more than providing an analysis where these features are established, it is believed at the same time to have a potential to support the claim made by Farkas (1992) that mood distribution is *at least partially* semantically motivated.

Traditionally, three different moods are distinguished in Hungarian: the indicative, the subjunctive and the conditional. First, I will briefly introduce some of their most typical uses, both in main and in subordinate clauses (*section 1*). Then I will propose an analysis that assumes that the indicative is in basic contrast with the non-indicative, this latter being an abstract superordinate category for the subjunctive and the conditional. The indicative is associated with the *compatibility* of the relevant proposition with a given contextually identified model of interpretation *M*, whereas the non-indicative is characterised by *incompatibility* with *M* (*section 2*). Finally, I will apply Quer's (1998) account of mood variation in Romance languages to an analysis of instances of mood variation in Hungarian, and partially modify that account for the purposes of the present proposal (*section 3*).

1 The distribution of mood morphemes in Hungarian clauses

I do not make an attempt here to catalogue all the different contexts in which mood morphemes occur: their most frequent uses are summarised here to facilitate the subsequent theoretical discussion in *section 2*.

[1]This paper has been preceded by a number of illuminating discussions on the topic and here I wish to express my gratitude to the four people that helped me the most with their valuable suggestions and criticism: *Tibor Laczkó*, *Tamás Mihálydeák*, *Péter Pelyvás* and *Enikõ Tóth*. I am also obliged to the anonymous ESSLLI reviewers for their useful comments. Of course, any errors remaining still are entirely mine. Special thanks to *György Balassa* for the LaTeX conversion.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

1.1 The indicative

The unmarked category of the indicative is denoted by a zero morph in Hungarian. It is typically used to express full epistemic commitment in main clauses (1) and in the complement clauses of factive predicates (2-3):

(1) *Péter otthon van.*
Peter at home is(IND)
'Peter is at home.'

(2) *Örülök, hogy Péter otthon van.*
glad.be.I(IND) that Peter at home is(IND)
'I am glad that Peter is at home.'

(3) *Érdekes, hogy Péter otthon van.*
Interesting that Peter at home is(IND)
'It is interesting that Peter is at home.'

The indicative can also be used however to express various degrees of epistemic commitment:

(4) *Talán otthon van.*
perhaps at home is(IND)
'Perhaps he is at home.'

(5) *Biztos, hogy otthon van.*
certain that at home is(IND)
'He must be at home.'

Moreover, the indicative is also used in the complement clauses of weak intensional predicates such as propositional attitude verbs (6), declaratives (7) and fiction verbs (8):

(6) *Péter azt hiszi, hogy Edit otthon van.*
Peter that.acc believes(IND) that Edith at home is(IND)
'Peter believes that Edith is at home.'

(7) *Péter azt mondja, hogy Edit otthon van.*
Peter that.acc says(IND) that Edith at home is(IND)
'Peter says that Edith is at home.'

(8) *Péter azt képzeli, hogy Edit otthon van.*
Peter that.acc fancies(IND) that Edith at home is(IND)
'Peter fancies that Edith is at home.'

Notice that the interpretation of the embedded clause may be attached to the referent of the matrix subject instead of the speaker, as in the last three sentences.

1.2 The subjunctive

The subjunctive affix is *-j*.[2] It is the mood primarily associated with imperative sentences:

(9)

Ülj	*le!*
sit.you(SUB)	down

'Sit down.'

We also find the subjunctive in the complement clauses of directive (10), desiderative (11) and non-epistemic modal (12) predicates:

(10)

Megparancsolom,	*hogy*	*ülj*	*le!*
order.I(IND)	that	sit.you(SUB)	down

'I order you to sit down.'

(11)

Arra	*vágyom,*	*hogy*	*leülj.*
that.for	long.I(IND)	that	down.sit.you(SUB)

'I am longing for you to sit down.'

(12)

Nem	*szabad,*	*hogy*	*leülj.*
not	allowed.to	that	down.sit.you(SUB)

'You are not allowed to sit down.'

Nevertheless, the subjunctive is not simply the mood that we could associate with imperative speech acts. It is obligatory, for example, in purpose adjunct clauses (13) and it appears in *too* and *enough* (14) contexts:

(13)

Azért	*jöttem,*	*hogy*	*beszéljek*	*veled.*
that.for	came.I(IND)	that	talk.I(SUB)	you.with

'I have come to talk to you.'

(14)

Túl	*fiatal*	*vagy*	*ahhoz,*	*hogy*	*ezt*
too	young	are.you(IND)	that.to	that	this.acc
megértsd.					
understand.you(SUB)					

'You are too young to understand this.'

[2] A syntactically motivated distinction is often made between an imperative and a subjunctive mood in Hungarian as the former reverses the order of the preverb and the verb both in main and in subordinate clauses, while the latter does not (cf., for example, Kenesei 1994). This is a purely syntactic distinction though and for this reason I am going to talk about the single morphological category of the subjunctive in what follows.

1.3 The conditional

The conditional affix is *-n*. It is the obligatory mood in both the main and the subordinate clauses of counterfactuals:

(15)

Ha	*ő*	*itt*	*lenne,*	*akkor*	*boldog*	*lennék.*
if	she	here	be.she(SUB)	then	happy	be.I(CON)

'If she were here, then I would be happy.'

It also has a volitional/boulomaic use, expressing the speaker's -or, as in (16), of the individual referent of the clausal subject- wishes:

(16)

Péter	*szívesen*	*inna*	*egy*	*sört.*
Peter	keenly	drink.he(CON)	a	beer.acc

'Peter would keenly drink a beer.'

The conditional can be used to express a remote future possibility, i.e. when we want to indicate that there is only a slight possibility that something will happen in the future:

(17)

Ne	*engedjétek*	*be,*	*ha*	*erre*	*jönne.*
not	let.you(SUB)	in	if	here.to	come.he(CON)

'Do not let him in if he happens to come this way.'

And finally, the conditional is obligatory in two other subordinate clause types: complement clauses governed by the matrix pronominal *anélkül* (18) and clauses with the complementizer *mintha* (19):

(18)

Elment	*anélkül,*	*hogy*	*köszönt*	*volna.*
left.he(IND)	that.without	that	said.goodbye.he	CON

'He left without saying goodbye.'

(19)

Úgy	*tett,*	*mintha*	*semmi*	*sem*	*történt*	*volna.*
so	did.he(IND)	as.if	nothing	not	happened.it	CON

'He behaved as if nothing had happened.

2 The semantics of the mood morphemes

The following discussion is based on the assumption that in the above examples the respective mood morphemes contribute a single meaning to the interpretation of the sentence in which they occur. This semantic feature is already assigned to them in the lexicon and it encodes the relation that exists between the propositional core of the relevant clause and a given contextually identified model of interpretation M. Two types of such relations are going to be established: *compatibility* and *incompatibility*. The present proposal is intended to be compatible with dynamic frameworks, but it itself is not dynamic as the sole purpose is the analysis of the meaning of mood morphemes and not that of the sentences in which they occur.

This proposal rests upon the following understanding of the notion of model. A model M is taken to be a single set containing propositions that the individual anchor of the model can assign a definite truth-value to. Every model is anchored to a given individual x and a given time t: in the default case the speaker and the time of utterance.[3] This notion of model corresponds in many ways to what is currently known in dynamic semantics as information state, as it is intended to represent the epistemic status of its individual anchor. The reason why I still talk about models and not information states is threefold. *First*, the present proposal itself is not dynamic as it is emphasised above. *Second*, propositions in modal semantics are interpreted with respect to models that assign them to (sets of) possible worlds, one of which is regarded to have a distinguished status in linguistic analyses: the actual world. As this paper is not concerned with an analysis of sentential meaning, the introduction of the whole inventory of a possible world semantics is not necessary. Models here are taken to represent *only* the relevant actual world, as it is known to a given individual. This terminology facilitates a transition to semantic analyses that aim at providing fully-fledged interpretations for sentences. *Third*, in some non-classical logical approaches models are indeed defined *through language*, without explicitly assuming a separate ontological domain of entities which expressions in the language can be assigned to (cf. Carnap's work). This notion of models is close to the present one.[4]

M is then a representation of the epistemic status of its individual anchor, which need not be the speaker as in (8), repeated here as (20):

(20)	*Péter*	*azt*	*képzeli,*	*hogy*	*Edit*	*otthon*	*van.*
	Peter	that.acc	fancies(IND)	that	Edit	at home	is(IND)
	'Peter fancies that Edit is at home.'						

Here the interpretation of the matrix clause is in a model anchored to the speaker as the default case, but the subordinate clause is to be interpreted in a model anchored to *Péter*, the main clause subject. Model shifts can be introduced by expressions such as *according to Peter* or by intensional predicates such as *képzeli* in (20). Nevertheless, even a short characterisation of model-shifting would exceed the limitations of this paper.[5] I investigate only one particular type of shifting models of interpretation in *section 3*: one which is the result of mood variation.

M contains propositions that the individual anchor either knows to be true or knows to be not true. As the proposal centres on the notions of

[3]In what follows the temporal parameter will be ignored and I will concentrate upon the default value: the present.

[4]My thanks are due to Tamás Mihálydeák for pointing this out to me.

[5]cf. Farkas 1992 for an introductory discussion of this issue.

compatibility and *incompatibility* (to be explained below), the issue of entailments needs to be mentioned.[6] From the present perspective, the problem arises when we want to represent inferences that speakers draw from the information that is available to them (the propositions in M). This is a very complex issue that I cannot tackle here properly, but what I assume is the following. Inferences are only represented in M if the individual anchor is "forced" to draw them in a given situation. Potential inferences -ones that are not relevant to be drawn in a given situation- are not to be represented in *M.* The reason why I do not assume theoretic closure is that M would cease to be a representation of any degree of psychological plausibility if we assumed that speakers were able to draw all the logically possible conclusions from the premises that are available to them. This, of course, renders a proper formal treatment very difficult, but this is not an idiosyncratic problem of the present proposal.

My basic claim is that the mood morphemes establish a relation between M and the propositional core of the clause in which they occur. We can receive this propositional core if we abstract away all attitudinal operators from the clause. Kiefer (1987) & (1988) claims that every sentence has the following schematic semantic structure:

(21) (Att, p')

where *Att* stands for attitudinal operators that determine sentence modality and *p'* represents the propositional content of the clause. If we consider for example the imperative sentence (9), repeated here as (22),

(22) *Ülj* *le.*
sit.you(SUB) down
'Sit down.'

then the relevant propositional content is *you are sitting down'* and the attitudinal operator is *Imp*(erative), which approximately means "*I want the future action described by the propositional content to become true*" (Kiefer: 1987, 79). The attitudinal operator of declarative sentences is *Decl*, with the approximate sense "*to consider something to be true*".

These preliminaries having been set, we can now turn to the discussion of the mood morphemes themselves. I claim that *the indicative* denotes the *compatibility* of the propositional core of the clause in which the indicative morpheme is present with the relevant model *M.* Compatibility is defined in (23):

(23) A proposition p is compatible with a given model M iff $\neg p \notin M$.

[6] As the reviewers of the abstract indicated, a proposition p can be incompatible with M even if $\neg p$ is not in *M*, but there is another proposition q in M which implies $\neg p$.

In the case of factive predicates in (1-3) p is in M as the speaker knows the relevant propositions to be true. In (6-8) the matrix predicate introduces another model in which the propositional content of the complement clause is to be interpreted, and within that model the embedded proposition is again taken to be true. It is considered to be an advantage of the present approach that it can also account for clauses which do not express full epistemic commitment, as (4-5): here neither p, nor $\neg p$ is in M and this is indicative inducing according to (23). That is why we also find the indicative in the complement clause of a matrix modal predicate of epistemic possibility, as in (24):

(24)	*Lehet,*	*hogy*	*otthon*	*van.*
	may(IND)	that	at home	is(IND)
	'He may be at home.'			

Here the speaker does not know whether the person talked about is at home or not, and this non-determinateness leads to no propositions in *M*. In this case again, the relevant propositional core *'he is at home'* is compatible with *M*. The above mentioned *Decl* attitudinal operator is present only in the matrix, but not in the embedded clause - what is asserted is the possibility that the embedded proposition is true.

As of the other two moods, let us consider *the subjunctive* first. The subjunctive morpheme is argued to denote the *incompatibility* of the relevant propositional core with *M*, where incompatibility is defined as:

(25) A proposition p is incompatible with a given model M iff $\neg p \in M$.

Considering our earlier example sentences (9-14), the intuitive insight about the function of the subjunctive is that it can be regarded as a kind of indicator that the relevant propositional core is incompatible with the relevant model M in the sense that p is known to be not true at the time of utterance. This complies with the description that Kiefer provides for the *Imp* operator. Let us consider the following sentence:

(26)	*Látogass*	*meg*	*holnap.*
	visit.you(SUB)	perf	tomorrow
	'Visit me tomorrow.'		

I assume that at the time of utterance M contains a proposition of the form *'(what I know now is that) you are not coming tomorrow'*, otherwise the use of an imperative sentence is not felicitous. It has to be emphasised again that the present proposal is restricted to a semantic analysis of the mood morphemes themselves and not to the sentences in which they occur. If we want to achieve the latter goal, then the present apparatus is insufficient.[7] What is claimed of the subjunctive morpheme is nothing more than what it contributes to the interpretation of the clause is that its propositional core is incompatible with M, as it has been defined. It will be shown in *section 3* that this approach helps to explain why the subjunctive occurs in contexts where futurity is not involved in any sense (cf. sentences 37 and 40 in *section 3*).

The conditional morpheme, as opposed to the subjunctive, is argued to have a lexically encoded function of introducing an obligatory model shift. In (15-19), the conditional morpheme introduces an alternative model M_C where the relevant propositional core is envisaged to be true, thus M_C contains p. However, the reference model is M here, too, just as well as it has been claimed for the subjunctive and it contains $\neg p$. In other words, the conditional morpheme also has the basic function of denoting the incompatibility of p with M, but as opposed to the subjunctive, it introduces at the same time an alternative model of interpretation.

The counterfactual reading we have in (15) and in (18-19) follows from this approach in an obvious way. I assume that the volitional use operates in the same way as it has been presented for the subjunctive: in the case of (16) Peter would like a situation to come about which is not true at the moment. We use (17), repeated here as (27), if what we know now is that he is not coming:

(27)	*Ne*	*engedjétek*	*be,*	*ha*	*erre*	*jönne.*
	not	let.you(SUB)	in	if	here.to	come.he(CON)
	'Do not let him in if he happens to come this way.'					

What happens here is that the speaker still maintains a slight possibility that a situation might happen, which he otherwise considers to be not very likely.

I have presented an analysis where both the subjunctive and the conditional morphemes are to be associated with *incompatibility*, as it is understood here. The conditional morpheme has the additional function that it always denotes an obligatory shift in models of interpretation - a shift into a model M_C which is incompatible with the reference model M. Thus we might as well regard the conditional as some kind of a special form of the subjunctive - in fact I claim that they both realize the same underlying non-indicative category of mood.[8] In this sense, the Hungarian mood sys-

[7] Cf., for example, the analysis of deontic modals in Enç 1997, where predicates of deontic modality are said to shift the evaluation time into the future. Then, as usual, the sentence is to be interpreted in a set of future possible worlds - but the introduction of the whole inventory of a possible world semantics is exactly what the present proposal intends to avoid.

[8] In *section 3* I will present data from Hungarian which are believed to support this claim.

tem is based upon a basic dichotomy: the indicative can be contrasted with the abstract superordinate category of the non-indicative. Figure 2 gives a summary of the system that has been proposed here.

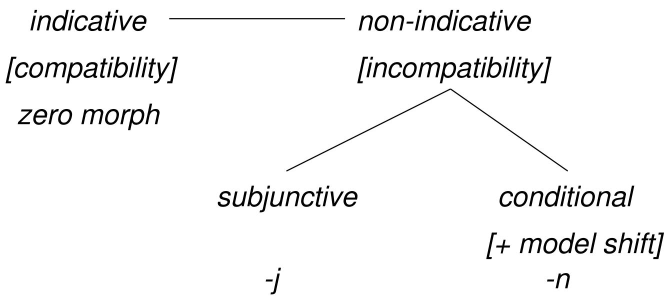


Figure 22.1: The Hungarian mood-system.

A final note of warning should be made here before closing this section. I agree with Farkas (1992) in that mood distribution is not completely arbitrary and there is ample evidence to suggest that it is in fact to some extent semantically motivated. This does not mean though that such motivation can always be legitimately established. In other words, it is not claimed here that there is an exact assignment from semantic to grammatical representations or vice versa.

3 Mood variation in Hungarian

3.1 Some instances of mood variation in Hungarian

In this brief section I apply the approach outlined above to mood variation in Hungarian. Some typical contexts where mood variation is grammatical are listed below. I provide only one English translation in each case, as there is no systematic way of rendering in English the differences brought about by the change of mood in these environments.

The grammaticality of mood variation is the most robust in polarity contexts. The presence of a conditional or a question operator (cf. 28-29) can be a licensing factor for an indicative-conditional variation to varying degrees:

(28)	*Ez*	*az*	*a*	*híres*	*palota?*
	This (IND-zero copula)	that	the	famous	palace

(29)	*Ez*	*lenne*	*az*	*a*	*híres*	*palota?*
	This	be.it(CON)	that	the	famous	palace

'Is this that famous palace?'[9]

[9] This example has been suggested to me by Ádám Nádasdy. I owe (32-33) to Tibor Laczkó.

(29) is typically uttered in a situation where the speaker finds the palace much different than he had expected: he in fact finds it unworthy of its fame.

The negative operator is, however, a much stronger licenser for mood variation. We find the indicative-conditional variation in the complement clause of negated factive verbs (30-31), negated propositional attitude verbs (*hisz* 'believe', *gondol* 'think'), predicates with a negative meaning component (32-33); and in clauses of reason when the matrix clause is negated (34-35):

(30) *Péter nem emlékszik arra, hogy János itt*
Peter not remember.he(IND) that.for that John here
volt.
was(IND)

(31) *Péter nem emlékszik arra, hogy János itt lett*
Peter not remember.he(IND) that.for that John here was
volna.
CON
'Peter does not remember that John was here.'

(32) *János tagadta, hogy õ ölte meg az*
John denied.he(IND) that he killed.he(IND) perf the
elnököt.
president.acc

(33) *János tagadta, hogy õ ölte volna meg az*
John denied.he(IND) that he killed.he CON perf the
elnököt.
president.acc
'John denied that he killed the president.'

(34) *Péter nem azért maradt otthon, mert fáradt*
Peter not that.for stayed.he(IND) at home because tired
volt.
was.he(IND)

(35) *Péter nem azért maradt otthon, mert fáradt*
Peter not that.for stayed.he(IND) at home because tired
lett volna.
was.he CON
'Peter did not stay at home because he was tired.'

Where the indicative is used (30, 32, 34), the speaker takes the proposition expressed by the complement clause to be true. In (31, 33) the speaker's epistemic commitment to the truth of the embedded preposition is left unspecified, in (35) the speaker takes the embedded proposition to be false. Negated epistemic predicates expressing likelihood or probability show an indicative-subjunctive-conditional alternation[10] (36-38), and an indicative-subjunctive alternation appears in the complement clauses of negated matrix predicates of epistemic possibility[11] (39-40):

(36)	*Nem*	*valószínû,*	*hogy*	*ez*	*a*	*történet*	*igaz.*	
	not	likely	that	this	the	story	true	

(IND-zero copula)

(37)	*Nem*	*valószínû,*	*hogy*	*ez*	*a*	*történet*	*igaz*	*legyen.*
	not	likely	that	this	the	story	true	be(SUB)

(38)	*Nem*	*valószínû,*	*hogy*	*ez*	*a*	*történet*	*igaz*	*lenne.*
	not	likely	that	this	the	story	true	be(CON)

'It is not likely that this story is true.'

(39)	*Az*	*nem*	*lehet,*	*hogy*	*ilyen*	*hülye*	*vagy.*
	that	not	may.it(IND)	that	so	stupid	are.you(IND)

(40)	*Az*	*nem*	*lehet,*	*hogy*	*ilyen*	*hülye*	*legyél.*
	that	not	may.it(IND)	that	so	stupid	are.you(CON)

'You cannot be that stupid.'

(36) and (39) mean something like '*Though it does not seem to be very likely, I still have to believe it as my sources are completely reliable*'. No such assumption is present in (37-38) and (40): the embedded proposition is not believed to be true by the speaker.[12]

In all the previous examples, the indicative was in variation with the subjunctive or the conditional, or both. But there is also a subjunctive-conditional variation in purpose clauses[13] :

[10]This threefold alternation also appears in the context of free choice pronouns and in concessive clauses. I leave these contexts out for lack of space, but the present proposal is intended to provide an account of these instances of mood variation, too.

[11]There is a restricted occurrence of the subjunctive in the complement clauses of these predicates, too. The question operator, for example, licenses this mood: *Lehetséges, hogy tényleg ilyen hülye lennél?* 'Is it really possible that you are that stupid?'

[12]Native speakers have differing intuitions about the interpretation of this set of sentences. Some of them find (36) -and to a lesser degree (39), too- acceptable even if the speaker does not believe that the embedded proposition is true. Even these native speakers believe though that the speaker is more likely to assume the truth of the embedded proposition if he uses the indicative than if he uses one of the non-indicative moods.

[13]The conditional is usually argued to be less acceptable in these contexts in descriptive grammars.

(41)	*Megkérte,*	*hogy*	*írjon*	*egy*	*levelet.*
	asked.he(IND)	that	write.he(SUB)	a	letter.acc

(42)	*Megkérte,*	*hogy*	*írna*	*egy*	*levelet.*
	asked.he(IND)	that	write.he(CON)	a	letter.acc

'He asked him to write a letter.'

There is no semantic difference between the two complement clauses here: both express the same purposive meaning.

3.2 Explaining mood variation

Quer (1998) claims that the grammatical function of mood variation in Romance languages is to mark shifts in models of interpretation. What the indicative denotes in contexts of mood variation is that the interpretation of the relevant proposition is in a model anchored to the speaker and not to the referent of the main clause subject (if he is different from the speaker). The subjunctive has the function of denoting that the interpretation is not in the speaker's, but in another, contextually available model (for example in that of the referent of the main clause subject). Then the speaker's epistemic commitment towards the truth of the relevant proposition is left unspecified. I argue that this proposal has the right predictions for the Hungarian data, too, with some modifications to be explained below.

Following the claim made by Quer, I assume that the indicative morpheme in all the above depicted mood variation contexts has the function of denoting the compatibility of the relevant proposition with the *speaker*'s model. Thus when the speaker opts for (32), then he can only go on saying '*... but I am sure that he did*'. Accepting the denial put forward by Peter is not a possibility in this case. These indicative clauses express the speaker's full epistemic commitment towards the truth of the relevant proposition, which is brought about by the presence of the *Decl* attitudinal operator in these environments. This is an important difference between (39) and (24): *Decl* is assumed to be present in the complement clause only in the former, but not in the latter case.

I claim that in mood variation contexts both the subjunctive and the conditional morphemes have the basic function of denoting that the relevant proposition is incompatible with the relevant model. Thus they are assumed to operate in the same way whether they are in mood variation environments or not, and it is only the indicative that has a special function in the former case. When the individual anchor of the model is not the speaker, then incompatibility arises with respect to *that* model, and the speaker's commitment towards the truth of the relevant proposition is left unspecified. In (31) for example, the matrix predicate introduces a model anchored to the matrix predicate where the embedded proposition is to be interpreted. Hence it is possible that the speaker thinks that John was there just as well that he believes (in another situation) that he was not. When no other model of interpretation is available in the context (as in 29, 35, 37, 38 and 40), then the two morphemes simply denote the incompatibility of p with the speaker's own model.

If they are both grammatical in these contexts, then the subjunctive and the indicative are almost completely interchangeable as both appear to contribute the same incompatibility feature to the interpretation of the clause. This is especially true for (41) and (42). I believe that these data support the fundamental claim made in this paper: that the subjunctive and the conditional are both subtypes of the same abstract non-indicative mood, which is in basic contrast with the indicative.

References

Enç, M. (1997). Tense and Modality. In S. Lappin (Ed.), *The Handbook of Contemporary Semantic Theory*, pp. 345–358. Oxford: Blackwell Publishers Ltd.

Farkas, D. (1992). Mood choice in complement clauses. In I. Kenesei and C. Pléh (Eds.), *Approaches to Hungarian*, Volume 4, pp. 207–25. Szeged: JATE.

Kenesei, I. (1994). Subordinate clauses. In F. Kiefer and K. É. Kiss (Eds.), *The Syntactic Structure of Hungarian*, Volume 27 of *Syntax and Semantics*, pp. 275–354. San Diego: Academic Press.

Kiefer, F. (1987). On defining modality. *Folia Linguistica XXX*(1), 67–94.

Kiefer, F. (1988). Mondatfajta – mondatmodalitás – mondathasználat. [Sentence type – sentence modality – sentence use]. *Tertium non datur 5*, 5–15.

Quer, J. F. (1998). *Mood at the Interface*. Ph. D. thesis, The Hague: Holland Academic Graphics.

Unprovability of Herbrand Consistency in Weak Arithmetics

Saeed Salehi
Institute of Mathematics, Polish Academy of Sciences, P.O.BOX 137, 00-950 Warsaw, and
Turku Center for Computer Sciences, Lemminkaisenkatu 14 A - 4th floor, FIN-20520 Turku
saeed@cs.utu.fi

Abstract. By introducing an appropriate definition of Herbrand Consistency in weak arithmetics, we show Gödel's Second Incompleteness Theorem for Herbrand consistency of theories containing $I\Delta_0$.

1 Introduction

Consider a formula ϕ in the prenex normal form

$$\forall x_1 \exists y_1 \cdots \forall x_m \exists y_m \overline{\theta}(x_1, y_1, \cdots, x_m, y_m)$$

with the Skolem functions $f_1^\theta, \cdots, f_m^\theta$; its Skolemized form by definition is

$$\forall x_1 \cdots \forall x_m \overline{\theta}(x_1, f_1^\theta(x_1), \cdots, x_m, f_m^\theta(x_1, \ldots, x_m)).$$

For a sequence of terms $\sigma = \langle t_1, \cdots, t_m \rangle$, the *Skolem instance* $Sk(\theta, \sigma)$ is

$$\overline{\theta}(t_1, f_1^\theta(t_1), \cdots, t_m, f_m^\theta(t_1, \ldots, t_m)).$$

Herbrands's Theorem states that a theory T is consistent if and only if every finite set of its Skolem instances is propositionally satisfiable (see (5).) Let Λ be a set of Skolem terms of T (i.e. constructed from the Skolem function symbols of T) *available Skolem instances* of θ in Λ are $Sk(\theta, \sigma)$ for all sequence of terms $\sigma = \langle t_1, \cdots, t_m \rangle \subseteq \Lambda$ such that $\{f_1^\theta(t_1), \cdots, f_m^\theta(t_1, \ldots, t_m)\}$ is a subset of Λ too.

Any function, p, whose domain is a set of atomic formulae and its range is $\{0, 1\}$ is called an *evaluation*, if it preserves the equality (for all a, b and atomic formulae φ, $p[a = b] = 1$ implies $p[\varphi(a)] = p[\varphi(b)]$) and satisfies the equality axioms ($p[a = a] = 1$ for all a.) For a set of terms Λ, an *evaluation on* Λ is an evaluation whose domain is the set of all atomic formulae with

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

constants from Λ (i.e. the variables are substituted by the terms from Λ.) An evaluation p *satisfies* an atomic formula φ if $p[\varphi]=1$. This definition can be extended to all open (quantifier-less) formulae in a unique way.
Evaluation p on Λ is an T-*evaluation* for a theory T, if it satisfies all the available Skolem instances of T in Λ.
When Λ is the set of all Skolem terms of T, any T-evaluation on Λ determines a Herbrand model of T (see (5).)

Toward formalizing the definition of Herbrand Consistency, we read the above Herbrand's Theorem as:

A theory T is consistent if and only if for every finite set of Skolem terms of T, say Λ, there is an T-evaluation on Λ.

So Herbrand Consistency of a theory T can be defined as:

"for every set of Skolem terms of T, there is an T-evaluation on it."

Herbrand's Theorem is provable in $I\Delta_0 + SupExp$, and it is known that Herbrand consistency is not equivalent to the standard, say Hilbert's, consistency in $I\Delta_0 + Exp$ (see (3), (7).) Unprovability or provability of Herbrand's Consistency for weak arithmetics (i.e. proper fragments of $I\Delta_0+ Exp$) had been an open problem (see (6),(7).) Herbrand Consistency of $I\Delta_0 + Exp$ is unprovable in itself ((3),(7).)

Adamowicz ((1)) has shown the unprovability of Herbrand Consistency of $I\Delta_0 + \Omega_2$ in itself (also in another unpublished paper for $I\Delta_0 + \Omega_1$.)

In this paper we modify the definition of Herbrand Consistency such that its negation gives a real Herbrand proof of contradiction even when Exp is not available, and show unprovability of formalized Herbrand Consistency of $I\Delta_0$ (by the new definition) in itself. So it turns out that $I\Delta_0$ does not prove its own Herbrand Consistency, since the new Herbrand Consistency predicate is implied by the old one.

2 Formalization of Herbrand Consistency in $I\Delta_0$

We take the language of arithmetic $\mathcal{L} = \{0, S, +, ., \leq\}$ in which the operations "S" (successor) "$+$" (addition) and "." (multiplication) are regarded as predicates. For example "$x+y=z$" is a 3-array predicate, and the traditional statements should be re-read in this language by using the predicates $S, +, .$; as an example $\forall x, y, z(x + (y + z) = (x + y) + z)$ can be read as $\forall x, y, z, u, v, w(``y + z = v" \wedge ``x + v = w" \wedge ``x + y = u" \rightarrow ``u + z = w")$.

So we may need some extra universal quantifiers (and variables) to represent the arithmetical formulae in this language, but for simplicity, and when there is no confusion, we will use the old notation.

All atomic formulae in our language are of the form $x_1 = x_2$, $x_2 = S(x_1)$, $x_1 + x_2 = x_3$, $x_1.x_2 = x_3$ and $x_1 \leq x_2$, where x_1, x_2, x_3 are variables or the constant 0.

Denote the cardinal of a set A by $|A|$; by *terms* we mean terms constructed from the Skolem function symbols of a theory T under consideration.

For a set of terms Λ, there are $2|\Lambda|^3+3|\Lambda|^2$ different atomic formulae with constants from Λ. So there are $2^{2|\Lambda|^3+3|\Lambda|^2}$ different evaluations on Λ. This shows that the above definition has a deficiency in weak arithmetics, from the viewpoint of incompleteness: unprovability of the consistency of T in T is equivalent to having a model of T which contains a proof of contradiction from T. By the above definition, a Herbrand proof of contradiction consists of a set of terms, say Λ, such that there is no T-evaluation on it. If Exp is not available in T, it may happen that all the (few) available evaluations in the model are T-evaluations. This doesn't give a real Herbrand proof in the model! because *not all the evaluations are accessible in the model* (their number $2^{2|\Lambda|^3+3|\Lambda|^2}$ might be too large to exist.) It would be more reasonable if we could find a model with a sufficiently small set of terms in it, such that none of the evaluations on this set (which can be counted in the model) is an T-evaluation. An upper bound for the codes of the evaluations on a set of terms is given below.

We use the Hajek-Pudlak's coding of sets-sequences and terms ((3)) the main properties of this coding are:

$*$ $\mathsf{code}(\langle x_1,\cdots,x_l\rangle),\mathsf{code}(\{x_1,\cdots,x_l\}) \leq (9(1+max\{x_1,\cdots,x_l\})^2)^l$

(i.e. the set $\{x_1,\cdots,x_l\}$ or the sequence $\langle x_1,\cdots,x_l\rangle$ can have a code which is less than or equal to $[9(1+max\{x_1,\cdots,x_l\})^2]^l$)

$*$ $\mathsf{code}(A\cup B),\mathsf{code}(A*B) \leq 64.\mathsf{code}(A).\mathsf{code}(B)$

Code the ordered pair $\langle a,b\rangle$ by $(a+b)^2+b+1$.

Fix the function symbols $f_k^{i,j}$ which is supposed to be the i-th, k-array Skolem function for the j-th axiom of a theory T (so if the j-th axiom is $\exists x\forall y\exists u\exists v A(x,y,u,v)$ then its Skolemized is $\forall y A(f_0^{1,j},y,f_1^{1,j}(y),f_1^{2,j}(y))$.)

Code $f_k^{i,j}$ by $\langle 1,\langle i,\langle j,k\rangle\rangle\rangle$, the symbol ")" by $\langle 2,0\rangle$, "(" by $\langle 2,1\rangle$ and the constant 0 by $\langle 2,2\rangle$.

And fix the function symbols f_l^i which is supposed to be the i-th, l-array function, these symbols are reserved to be Skolem function of a formula θ in the definition of $HCon_T(\theta)$, and code it by $\langle 0,\langle i,l\rangle\rangle$.

Terms are well-bracketing sequence constructed from $\{(,)\}\cup\{f_k^j\}_{j,k}\cup\{f_l\}_l$ (see (3).)

Example Let $i\geq 1$, and define $c_0=0$, $c_{k+1}=f_1^{1,1}(c_k)$ for $k\leq i$. There is a natural number $\mathbf{A}$ such that $\mathsf{code}(\{c_0,\cdots,c_i\})\leq \mathbf{A}^{i^2}$.

Since we have $\mathsf{code}(c_{k+1}) \leq 64^4\mathsf{code}(f_1^{1,1})\mathsf{code}(\text{“(”})\mathsf{code}(c_k)\mathsf{code}(\text{“)”}) \leq 64^4\langle 1,\langle 1,\langle 1,1\rangle\rangle\rangle\langle 2,0\rangle\langle 2,1\rangle\langle 2,1\rangle\mathsf{code}(c_k)$,

let $m=64^4\langle 1,\langle 1,\langle 1,1\rangle\rangle\rangle\langle 2,0\rangle\langle 2,1\rangle\langle 2,1\rangle\mathsf{code}(c_k)$, so we have $\mathsf{code}(c_k)\leq m^k.\mathsf{code}(c_0)$.

Hence $\mathsf{code}(\{c_0,\cdots c_i\}) \leq (9(1+\mathsf{code}(c_i))^2)^i \leq 9^i(2^2(m^i\mathsf{code}(c_0))^2)^i \leq 36^i\langle 2,2\rangle^{2i}m^{2i^2} \leq (36\langle 2,2\rangle m^2)^{i^2}$, we can take $\mathbf{A}=36\langle 2,2\rangle m^2$.

Let Λ be a set of terms with code y, we compute an upper bound for

evaluations on Λ: each evaluation is of the form
$\{\langle y_1 = y_2, p[y_1 = y_2]\rangle \mid y_1, y_2 \in \Lambda\} \cup \{\langle y_1 \leq y_2, p[y_1 \leq y_2]\rangle \mid y_1, y_2 \in \Lambda\} \cup \{\langle y_2 = S(y_1), p[y_2 = S(y_1)]\rangle \mid y_1, y_2 \in \Lambda\} \cup \{\langle y_1.y_2 = y_3, p[y_1.y_2 = y_3]\rangle \mid y_1, y_2, y_3 \in \Lambda\} \cup \{\langle y_1 + y_2 = y_3, p[y_1 + y_2 = y_3]\rangle \mid y_1, y_2, y_3 \in \Lambda\}$;
in which $p[\phi] \in \{0, 1\}$ for any atomic formula ϕ with constants from Λ.
Code " $=$ " by $\langle 3, 0\rangle$, " $\leq$ " by $\langle 3, 1\rangle$, "S" by $\langle 3, 2\rangle$, " $+$ " by $\langle 3, 3\rangle$, and "." by $\langle 3, 4\rangle$.
We code formulae by Polish notation, for example

$\mathsf{code}(x_1 + x_2 = x_3) = \mathsf{code}(+(x_1.x_2.x_3)) =$

$\mathsf{code}(\langle\langle 3, 3\rangle, \langle 2, 0\rangle, \mathsf{code}(x_1), \mathsf{code}(x_2), \mathsf{code}(x_3), \langle 2, 1\rangle\rangle)$.

There is a natural number $\mathbf{a}$ such that for any $k \in \{0, 1\}$

$\mathsf{code}(\langle y_1 = y_2, k\rangle) \leq 2 + (1 + \mathbf{a}y_1y_2)^2$,

$\mathsf{code}(\langle y_1 \leq y_2, k\rangle) \leq 2 + (1 + \mathbf{a}y_1y_2)^2$,

$\mathsf{code}(\langle y_2 = S(y_1), k\rangle) \leq 2 + (1 + \mathbf{a}y_1y_2)^2$,

$\mathsf{code}(\langle y_1 + y_2 = y_3, k\rangle) \leq 2 + (1 + \mathbf{a}y_1y_2y_3)^2$, and

$\mathsf{code}(\langle y_1.y_2 = y_3, k\rangle) \leq 2 + (1 + \mathbf{a}y_1y_2y_3)^2$.

So $\mathsf{code}(\langle\phi, k\rangle) \leq 2 + (1 + \mathbf{a}y^3)^2$ for all $k \in \{0, 1\}$ and atomic ϕ with constants from Λ, with $\mathsf{code}(\Lambda) = y$. Hence $\mathsf{code}(p) \leq (9.(3 + (1 + \mathbf{a}y^3)^2)^2)^{2|y|^3+3|y|^2} \leq (81(1 + \mathbf{a}y^3)^4)^{2|y|^3+3|y|^2}$, (we identify $|\Lambda|$ with $|y|$) for all evaluation p on Λ.
Call a set of terms Λ with $\mathsf{code}(\Lambda) = y$, *admissible* if $F(y) = (81(1 + \mathbf{a}y^3)^4)^{2|y|^3+3|y|^2}$ exists.
We modify the definition of Herbrand Consistency of a theory T as: " for every admissible set of Skolem terms of T, there is an T-evaluation on it". This is formalized below.

By "terms" we mean terms constructed from the Skolem function symbols $\{f_k^{i,j}\}_{i,j,k} \cup \{f_l^i\}_{i,l}$ introduced above, the bounded formula $\mathsf{Terms}(y)$ means "y is a set of terms constructed from those symbols".

There are bounded formulae $\mathsf{eva}(x)$ and $\mathsf{eval}(x, y)$ which represent "x is an evaluation" and "y is a set of terms and x is an evaluation on y".

For atomic formula ϕ, $p[\phi] = 1$ is a bounded formula, for more complex ϕ the statement $p[\phi] = 1$ can be written by a Π_1-formula:

let the bounded formula $\mathsf{Sat}(p, \phi, s)$ be

"$\mathsf{eva}(p)$& s is a sequence of pairs $\langle a_i, b_i\rangle$, such that:

1) each a_i is (the code of) a formula and each b_i is 0 or 1,

2) for $k = \mathsf{length}(s)$, $a_k = \phi$ and $b_k = 1$,

3) each a_i is either of the form

3.1) $a_i = a_j \wedge a_k$ for some $j, k < i$ and $b_i = b_j.b_k$,

or 3.2) $a_i = a_j \vee a_k$ for some $j, k < i$ and $b_i = b_j + b_k - b_j.b_k$,

or 3.3) $a_i = a_j \rightarrow a_k$ for some $i, j < k$ and $b_i = 1 + b_j.b_k - b_j$,

or 3.4) $a_i = \neg a_j$ for some $j < i$ and $b_i = 1 - b_j$,

or 3.5) a_i is atomic and $b_i = p[a_i]$. "

Let $S(\theta)$ be the number of subformulae of the formula θ. For the above sequence s we have $\mathsf{code}(s) \leq (9(1 + \mathsf{code}(\langle\phi, 1\rangle)^2)^{S(\phi)}$

$\leq (9(1 + 2 + (\phi + 1)^2)^2)^{S(\phi)} \leq (81(1 + \phi)^4)^{S(\phi)}$.

So we can write $p[\phi]=1$ as: $\forall z\Big(z\geq(81(1+\phi)^4)^{S(\phi)}\rightarrow\exists s\leq zSat(p,\phi,s)\Big)$.

Let $|\theta|$ be the number of existential quantifiers in the prenex normal form of θ (we can assume it has the form $\theta=\forall x_1\exists y_1\cdots\forall x_m\exists y_m\overline{\theta}(x_1,y_1,\cdots,x_m,y_m)$, so $|\theta|=m$ in this case.)
For a formula θ fix its Skolem functions as $f_1^\theta,\cdots,f_\alpha^\theta$ where $\alpha=|\theta|$. Write $\sigma=\langle t_1,\cdots,t_\alpha\rangle\subseteq\Lambda$ for a set of terms Λ such that $\{f_1^\theta(t_1),\cdots,f_\alpha^\theta(t_1,\ldots,t_\alpha)\}$ is a subset of Λ too.

We have $\mathsf{code}(Sk(\theta,\sigma))\leq\mathsf{code}(\theta*\sigma*(f_1^\theta(t_1),\cdots,f_\alpha^\theta(t_1,\ldots,t_\alpha)))$.

On the other hand
$\mathsf{code}((f_1^\theta(t_1),\cdots,f_\alpha^\theta(t_1,\ldots,t_\alpha))))\leq 18^\alpha\mathsf{code}(f_\alpha^\theta(t_1,\ldots,t_\alpha)))^{2\alpha}$, and also
$\mathsf{code}(f_\alpha^\theta(t_1,\ldots,t_\alpha)))\leq 64^{3+\alpha}\mathsf{code}(f_\alpha^\theta)\mathsf{code}(\text{“(”})\mathsf{code}(\text{“)”})\mathsf{code}(t_1)\ldots\mathsf{code}(t_\alpha)$.

So with $\mathsf{code}(\Lambda)=y$ we have

$\mathsf{code}(Sk(\theta,\sigma))\leq 64^3\theta(18.55^2)^\alpha 64^{2\alpha(3+\alpha)}\mathsf{code}(f_\alpha^\theta)^{2\alpha}.y^{2\alpha^2+\alpha}$.

Let $G(\theta,y)=[81(1+64^3.(18.55^2)^{|\theta|}.64^{2|\theta|(3+|\theta|)}.\theta.\mathsf{code}(f_{|\theta|}^\theta)^{2|\theta|}.y^{2|\theta|^2+|\theta|})^4]^{S(\theta)}$.

Noting that “$u=Sk(\theta,y)$” is a bounded formula, we can write “p is an θ-evaluation on y” as:

$\mathsf{Terms}(y)\wedge\mathsf{eval}(p,y)\wedge\forall z[z\geq G(\theta,y)\rightarrow\forall u\leq z\forall\sigma\leq y\{\sigma=(t_1,\cdots,t_{|\theta|})\subseteq y\wedge\{f_1^\theta(t_1),\cdots,f_{|\theta|}^\theta(t_1,\ldots,t_\alpha))\}\subseteq y\wedge\text{“}u=Sk(\theta,\sigma)\text{”}\rightarrow\exists s\leq z\mathsf{Sat}(p,u,s)\}]$.
Denote its bounded counterpart by $\mathsf{SatAvail}(p,y,\theta,z)$, that is:

$\mathsf{SatAvail}(p,y,\theta,z)=\mathsf{Terms}(y)\wedge\mathsf{eval}(p,y)\wedge\forall u\leq z\forall\sigma\leq y\{\sigma=(t_1,\cdots,t_{|\theta|})\subseteq y\wedge\{f_1^\theta(t_1),\cdots,f_{|\theta|}^\theta(t_1,\ldots,t_\alpha))\}\subseteq y\wedge\text{“}u=Sk(\theta,\sigma)\text{”}\rightarrow\exists s\leq z\mathsf{Sat}(p,u,s)\}]$.
For a finite theory $\{T_1,\cdots,T_n\}$, define the predicate $HCon_T(x)$, as:
$\forall z\Big(\forall y\leq z\ [\ \mathsf{Terms}(y)\ \wedge\ z\geq F(y)\ \wedge\ \bigwedge_{1\leq j\leq n}z\geq G(T_j,y)\ \wedge\ z\geq G(x,y)\rightarrow\exists p\leq z\exists s\leq z\{\mathsf{eval}(p,y)\wedge\bigwedge_{1\leq j\leq n}\mathsf{SatAvail}(p,T_j,y,s)\wedge\mathsf{SatAvail}(p,x,y,s)\}]\Big)$.

We note that the bounds $G(T_j,y)$ and for a standard x the bound $G(x,y)$ for z, are polynomial with respect to y, so for large enough, also for non-standard ys, they are less than the bound $F(y)$.

The cut log^2 is defined (informally) by: $x\in log^2\iff 2^{2^x}$ *exists*. A formal definition is given in the next section.

The predicate $HCon_T^*(x)$ is obtained from $HCon_T(x)$ by restricting the (only unbounded) universal quantifier to log^2:

$\forall z\in log^2\Big(\forall y\leq z\ [\ \mathsf{Terms}(y)\ \wedge\ z\geq F(y)\ \wedge\ \bigwedge_{1\leq j\leq n}z\geq G(T_j,y)\ \wedge\ z\geq G(x,y)\ \rightarrow\ \exists p\leq z\exists s\leq z\{\mathsf{eval}(p,y)\wedge\bigwedge_{1\leq j\leq n}\mathsf{SatAvail}(p,T_j,y,s)\wedge\mathsf{SatAvail}(p,x,y,s)\}]\Big)$.

Proposition 2.1. *The above formulae $HCon_T(\phi)$ and $HCon_T^*(\phi)$ binumerate “Herbrand Consistency of T with ϕ” in $\mathbf{N}$:*

$\mathbf{N}\models HCon_T(\theta)$ *iff* $\mathbf{N}\models HCon_T^*(\theta)$ *iff “$\{\phi\}\cup T$ is Herbrand consistent.”*

Herbrand Consistency of T, $HCon(T)$, is $HCon_T(\text{“}0=0\text{”})$.

For a moment assume we have proved the following proposition:

Proposition 2.2. *There is a finite set of $I\Delta_0$-derivable sentences, say B, such that for every bounded formula $\theta(x)$ with x as the only free variable, and for any finite theory α (in the language of arithmetic) whose axioms contain the set B,*

$$I\Delta_0 \vdash HCon(\alpha) \wedge \exists x \in log^2\theta(x) \to HCon^*_\alpha(\text{“}\exists x \in log^2\theta(x)\text{”})$$

Now we can prove our main theorem:

Theorem 2.3. *Take B as the previous proposition, and let D be the union of B and a finite fragment of $I\Delta_0$ containing PA^- such that the last proposition is provable in D, then for any finite consistent theory (in the language of arithmetic) whose axioms contain the set D, we have $\alpha \not\vdash HCon(\alpha)$.*

Proof. Let τ be the fixed point of $HCon^*_\alpha(\neg\tau) \equiv \tau$ (it is available in PA^-, i.e. $PA^- \vdash HCon^*_\alpha(\neg\tau) \equiv \tau$, see (4).)

The theory $\alpha + \neg\tau$ is consistent, since otherwise, by proposition 2.1, we would have $\mathbf{N} \models \neg HCon^*_\alpha(\neg\tau)$ and so by the fact that PA^- is Σ_1-complete ((4)) we would get $PA^- \vdash \neg HCon^*_\alpha(\neg\tau)$, hence $\alpha \vdash \neg\tau$, then α would be inconsistent.

Write $\neg\tau \equiv \exists x \in log^2\theta(x)$ for a bounded θ, then

$\alpha + \neg\tau + HCon(\alpha) \vdash HCon(\alpha) \wedge \exists x \in log^2\theta(x)$,

so by proposition 2.2 we get $\alpha + \neg\tau + HCon(\alpha) \vdash HCon^*_\alpha(\text{“}\exists x \in log^2\theta(x)\text{”})$,

and then $\alpha + \neg\tau + HCon(\alpha) \vdash HCon^*_\alpha(\neg\tau)$, or $\alpha + \neg\tau + HCon(\alpha) \vdash \tau$.

So $\alpha \vdash HCon(\alpha) \to \tau$, and this shows that $\alpha \not\vdash HCon(\alpha)$. $\triangle$

3 A Herbrand Σ_1-Completeness Theorem in $I\Delta_0$

This section is devoted to prove proposition 2.2.

Godel's original second incompleteness theorem states unprovability of (formalized) consistency of T in T, for strong enough theories T. Being strong enough means being able to code sets-sequences, terms and some other logical (syntaical) concepts, like provability and prove their properties.

Of those properties are:

1. $T \vdash Pr_T(\varphi) \wedge Pr_T(\varphi \to \psi) \to Pr_T(\psi)$
2. $T \vdash Pr_T(\varphi) \to Pr_T(Pr_T(\varphi))$

Usually the property 2 is proved by use of formalized Σ_1-completeness theorem:

$T \vdash \varphi \to Pr_T(\varphi)$ for any Σ_1-formula φ.

So how can one show Godel's second incompleteness theorem for weak arithmetics, which are not that strong to prove those properties?

One may have two options here:

1) try to find a model of T which does not satisfy $Con(T)$,

2) try to show some weak forms of Σ_1-completeness in T, which can prove $T \nvdash Con(T)$ (by a similar argument of our main theorem's proof.)

The first method is applied in (2) to show $Q \nvdash Con(Q)$ for Robinson's arithmetic Q.

There is no hope to use this way for more complex theories like $I\Delta_0$ (and its super-fragments) since there is no recursive non-standard model for them (see (3).)

So the difficulty rises when one seeks for a kind of formalized Σ_1-completeness theorem which can be proved in the (weak) theory and at the same time is powerful enough to show unprovabolity of the theory's consistency in itself.

A weak form of Σ_1-incompleteness theorem can be like:

$T \vdash Con(T) \wedge \exists x\theta(x) \rightarrow Con_T(\exists x\theta(x))$ for Δ_0-formulae $\theta(x)$ (cf (1).)
Our proposition 2.2 is a form of weak Σ_1-incompleteness theorem, in which the witness x for $\theta(x)$ is small (restricted to log^2) and the second consistency predicate is rather weak (that is $HCon^*_T$ instead of $HCon_T$.)

Take A be the axiom system:

$A1.\ \forall x \exists y$ "$y = S(x)$"

$A2.\ \forall x, y, z($"$y = S(x)$" $\wedge$ "$z = S(x)$" $\rightarrow y = z)$

$A3.\ \forall x\ (x \leq x)$

$A4.\ \forall x, y, z\ (x \leq y \wedge y \leq z \rightarrow x \leq z)$

$A5.\ \forall x\ (x \leq 0 \rightarrow x = 0)$

$A6.\ \forall x, y, z\ ($"$y = S(z)$" $\wedge\, x \leq y \rightarrow x \leq z \vee x = y)$

$A7.\ \forall x, y($"$y = S(x)$" $\rightarrow x \leq y)$

$A8.\ \forall x$ "$x + 0 = x$"

$A9.\ \forall x, y, z, u, v\ ($"$z = S(y)$" $\wedge$ "$x + y = u$" $\wedge$ "$v = S(u)$" $\rightarrow$ "$x + z = v$"$)$

$A10.\ \forall x$ "$x.0 = 0$"

$A11.\ \forall x, y, z, u, v\ ($"$z = S(y)$" $\wedge$ "$x.y = u$" $\wedge$ "$u + x = v$" $\rightarrow$ "$x.z = v$"$)$

$A12.\ \forall x, y\ ($"$y = S(x)$" $\rightarrow \neg y \leq x)$

Fix the terms $c_0 = 0$, $c_{j+1} = f_1^{1,1}(c_j)$.

The term c_i is represented as the i-th numeral in every A-evaluation p: $p[c_0 = 0] = 1$ and $p[c_{j+1} = S(c_j)] = 1$.

Lemma 3.1. *($I\Delta_0$) Suppose for an i, we have $\{c_0, \cdots, c_i\} \subseteq \Lambda$ for a set of terms Λ, and p is an A-evaluation on Λ, then*
1) If $p[a \leq c_i] = 1$ for an $a \in \Lambda$, then there is an $j \leq i$ such that $p[a = c_j] = 1$.
2) If γ is an open formula and $\gamma(x_1, \cdots, x_m)$ holds for $x_1 \cdots x_m \leq i$, then $p[\gamma(c_{x_1}, \cdots, c_{x_m})] = 1$.

Proof. 1) by induction on j, one can prove that if $p[a \leq c_j] = 1$ then $p[a = c_k] = 1$ for a $k \leq j$: for $j = 0$ use $A5$, and for $j + 1$ use $A6$.

2) The assertion can be proved for the atomic or negated atomic formulae. For $x_1 \leq x_2$ use induction on x_2, for $x_2 = 0$ by $A3$ and for $x_2 + 1$ by

$A3$, $A4$ and $A7$. Similarly for $x_1 + x_2 = x_3$ and $x_1.x_2 = x_3$ use induction on x_2 and $A8$, $A9$, $A10$ and $A11$. For $\neg x_1 = x_2$: if $\neg x_1 = x_2$ then either $x_1 + 1 \leq x_2$ or $x_2 + 1 \leq x_1$, e.g. for $x_1 + 1 \leq x_2$ we have $p[c_{x_1+1} \leq c_{x_2}] = 1$, now use $A12$. For $\neg S(x_1) = x_2$ use $A2$, and the cases $\neg x_1 + x_2 = x_3$ and $\neg x_1.x_2 = x_3$ can be derived from the previous cases. For $\neg x_1 \leq x_2$: if $\neg x_1 \leq x_2$ then $x_2 + 1 \leq x_1$ so $p[c_{x_2+1} \leq c_{x_1}] = 1$, now use $A4$ and $A12$.

The induction cases for $\wedge, \vee, \rightarrow$ are straightforward. (Note we have assumed that the formula θ is in normal form: the negation appears only in front of atomic formulas.) $\triangle$

Recall Godel's beta function:
$\beta(a, b, i) = r$ if $a = (q + 1)[(i + 1)b + 1] + r \ \wedge \ r \leq (i + 1)b$ for some q (cf (4).) Define the ordered pairs by $\langle a, b\rangle = a + \frac{1}{2}(a + b + 1)(a + b)$.

Let $\Psi(z, i) = \forall x \leq z \forall y \leq z \forall j < i\{\langle x, y\rangle = z \rightarrow x \geq (i + 1)y + 1 \wedge \beta(x, y, 0) = 2 \wedge \beta(x, y, j + 1) = (\beta(x, y, j))^2\}$.

The formula $\Psi(z, i)$ states that z is a (β)-code of a sequence whose length is at least $i + 1$, and its first term is 2 and every term is the square of its preceding term. So such a sequence looks like: $\langle 2, 2^2, 2^{2^2}, \cdots, 2^{2^i}, \ldots\rangle$.

We can define the cut log^2 as: $x \in log^2 \iff \exists z \Psi(z, x)$.

Denote the open part of Ψ by $\overline{\Psi}$, so $\Psi(z, x) = \forall \mathbf{u} \overline{\Psi}(z, x, \mathbf{u})$, in which $\mathbf{u} = (u_1, \cdots, u_k)$ for a natural k.

To get the B asserted in the proposition, we add the following axioms to A:

$A13$. $\Psi(33, 0)$

$A14$. $\forall x \forall i \exists y(\Psi(x, i) \rightarrow \Psi(y, i + 1))$

The axiom $A14$ is in fact the $I\Delta_0$-derivable statement $i \in log^2 \rightarrow i{+}1 \in log^2$. To be more precise we (*can*) write the axiom $A14$ in the prenex normal form:

$A14$. $\forall x \forall i \exists y \forall \mathbf{u} \forall \mathbf{v}(\overline{\Psi}(x, i, \mathbf{u}) \rightarrow \overline{\Psi}(y, i + 1, \mathbf{v}))$.

Fix the terms $z_0 = c_{33}$, $z_{j+1} = f_2^{1,14}(z_j, c_j)$.

The term z_i is represented as a (β)-code of the sequence $\langle 2, 2^2, \cdots, 2^{2^i}\rangle$ in any B-evaluation (note that $33 = \langle 5, 2\rangle =$ a β-code for $\langle 2\rangle$.)

Lemma 3.2. *($I\Delta_0$) Suppose for $i \geq 33$, $\{c_0, \cdots, c_i, z_0, \cdots, z_i\} \subseteq \Lambda$, then for any B-evaluation p on Λ, p satisfies all the available Skolem instances of $\Psi(z_i, c_i)$.*

Proof. By induction on $j \leq i$ one can show that any such p satisfies all the available Skolem instances of $\Psi(z_j, c_j)$. $\triangle$

Now we are close to the proof of the proposition, let α be a theory whose axioms contain the set B, and take a model $M \models I\Delta_0$ such that $M \models HCon(\alpha)$ and $M \models i \in log^2 \wedge \theta(i)$ for an $i \in M$. Take a set of terms Λ with $\mathsf{code}(\Lambda) = y$ such that $F(y)$ exists and is in $log^2(M)$ (we can assume i and y are non-standard) then we find an admissible set of terms Λ', so by the assumption $HCon(\alpha)$ there is an α-evaluation on Λ' which induces an $(\alpha \cup \{\exists x \in log^2\theta(x)\})$-evaluation on Λ. This shows $M \models HCon^*_\alpha(\exists x \in log^2\theta(x))$.

Write $\theta(x) = \forall x_1 \leq \alpha_1 \exists y_1 \leq \beta_1 \cdots \forall x_m \leq \alpha_m \exists y_m \leq \beta_m \overline{\theta}(x, x_1, y_1, \cdots, x_m, y_m)$. There are (partial) functions on M, $g_1, \cdots, g_m$ (we may assume, $g_j : [0,i]^j \to M$) such that for all $a_1, \cdots, a_m \in M$

$M \models a_1 \leq \alpha'_1 \to [g_1(a_1) \leq \beta'_1 \wedge \cdots [a_m \leq \alpha'_m \to [g_m(a_1, \ldots, a_m) \leq \beta'_m \wedge \overline{\theta}(i, a_1, g_1(a_1), \cdots, g_m(a_1, \ldots, a_m))]] \ldots]$,

in which $(\alpha'_j, \beta'_j;\ j \leq m)$ is the image of $(\alpha_j, \beta_j;\ j \leq m)$ under the substitution $\{x \mapsto i, x_j \mapsto a_j, y_j \mapsto g_j(a_1, \cdots a_j);\ j \leq m\}$.

Consider the formula

$\exists x \in log^2\theta(x) \ \equiv$

$\exists x \exists z \forall x_1 \leq \alpha_1 \exists y_1 \leq \beta_1 \cdots \forall x_m \leq \alpha_m \exists y_m \leq \beta_m \forall \mathbf{u}\{\overline{\Psi}(z, x, \mathbf{u}) \wedge$
$\wedge \overline{\theta}(x, x_1, y_1, \cdots, x_m, y_m)\}$. Its Skolemized form is:

$\forall x_1 \cdots \forall x_m \forall \mathbf{u}\{\overline{\Psi}(f^2_0, f^1_0, \mathbf{u}) \ \wedge \ x_{\leq}\alpha''_1 \to [f^1_1(x_1) \leq \beta''_1 \wedge \cdots [x_m \leq \alpha''_m \to [f^1_m(x_1, \ldots, x_m) \leq \beta''_m \wedge \overline{\theta}(f^1_0, x_1, f^1_1(x_1), \cdots, x_m, f^1_m(x_1, \ldots, x_m))]] \cdots]\}$,

in which $(\alpha''_j, \beta''_j;\ j \leq m)$ is the image of $(\alpha_j, \beta_j;\ j \leq m)$ under the substitution $\{x \mapsto f^1_0, y_j \mapsto f^1_j(x_1, \cdots x_j);\ j \leq m\}$.

Define the operation $\Re$ on terms by:

- $f^1_0 \mapsto c_i$
- $f^2_0 \mapsto z_i$
- $f^1_1(c_j) \mapsto c_{g_1(j)}$

$\vdots$

- $f^1_m(c_{j_1}, \cdots, c_{j_m}) \mapsto c_{g_m(j_1, \cdots, j_m)}$

That is the term f^1_0 is mapped (under $\Re$) to c_i, and f^2_0 is mapped to z_i and for any $1 \leq t \leq m$ the term $f^1_t(c_{j_1}, \cdots, c_{j_t})$ is mapped to $c_{g_t(j_1, \cdots, j_t)}$.

By an argument similar to the example in the previous section, it can be shown that there is a natural $\mathbf{K}$ such that $\mathsf{code}(c_j), \mathsf{code}(z_j) \leq \mathbf{K}^j$ for any $j \geq 1$, and $\mathsf{code}(\{c_0, \cdots, c_i, z_0, \cdots, z_i\}) \leq \mathbf{K}^{i^2}$ for any $i \geq 1$.

For any term t, $\mathsf{code}(\Re(t)) \leq \mathsf{code}(t * (z_i)^{|t|} * (c_i)^{|t|}) \leq$

$\leq 64^3.t.36^{3|t|}.\mathsf{code}(z_i)^{|t|}.\mathsf{code}(c_i)^{|t|} \leq 64^3.t.36^{3t}.\mathbf{K}^{2it}$, so $max\{\mathsf{code}(\Re(t)) | t \in \Lambda\} \leq 64^3.y.36^{3y}.\mathbf{K}^{2i^2y}$, hence $\mathsf{code}(\Re(y)) \leq 36^{|y|}.[64^3.t.36^{3t}.\mathbf{K}^{2it}]^{|y|}$.

Let $\Lambda' = \Re(\Lambda) \cup \{c_0, \cdots, c_i, z_0, \cdots, z_i\}$.

So $\mathsf{code}(\Lambda') = y' \leq 64.\mathbf{K}^{2i^2}.36^{|y|}.64^{3|y|}.y^{|y|}.36^{3y|y|}.\mathbf{K}^{2i|y|}$.

We show that $F(y')$ exists. Note that $y \in log^2$ because $y < F(y) \in log^2$.

Assuming that i, y are non-standard we can write: $F(y') \leq (y')^{4|y'|^4} = (y')^{4(2i+|y|)^4} \leq (y')^{4y^5}.(y')^{4(2i)^5}$, and this is less than $(2^{2^i})^{14}$ if $y \leq i$, and is less than $(2^{2^y})^{14}$ if $i \leq y$. So Λ' is admissible.

Hence by the assumption $HCon(\alpha)$ there is an α-evaluation q on Λ'. Define the evaluation p on Λ by

$p[\varphi(a_1, \cdots, a_l)] = q[\varphi(\Re(a_1), \cdots, \Re(a_l))]$ for any atomic φ.

It can be shown that the above equality holds for open formulae φ as well. We show that p satisfies all the available Skolem instances of $\{\exists x \in log^2\theta(x)\} \cup \alpha$ in Λ:

1) p is an α-evaluation, since q is so and the operation $\Re$ has nothing to do with the Skolem functions of α: $p[\phi(t_1, f_1^{1,j}(t_1), \cdots, t_k, f_k^{1,j}(t_1, \ldots, t_k))] =$

$q[\phi(\Re(t_1), \Re(f_1^{1,j}(t_1)), \cdots, \Re(t_k), \Re(f_k^{1,j}(t_1, \ldots, t_k)))] =$

$q[\phi(\Re(t_1), f_1^{1,j}(\Re(t_1)), \cdots, \Re(t_k), f_k^{1,j}(\Re(t_1, \ldots, t_k)))] = 1.$

2) p satisfies all the available Skoelm instances of $\exists x \in log^2\theta(x)$ in Λ:

2.1) $p[\overline{\Psi}(f_0^2, f_0^1, t_1, \cdots, t_{24})] = q[\overline{\Psi}(\Re(f_0^2), \Re(f_0^1), \Re(t_1), \cdots, \Re(t_{24}))] =$

$q[\overline{\Psi}(z_i, c_i, \Re(t_1), \cdots, \Re(t_{24}))] = 1$

since by lemma 3.2, q satisfies all the available Skolem instances of $\Psi(z_i, c_i)$ then the latter equality holds.

2.2) by lemma 3.1 for any term t and any $k \leq i$, if $p[t \leq c_k] = 1$ then $p[t = c_j] = 1$ for some $j \leq k$. So for evaluating $\theta(x)$ it is enough to consider Skolem instances like $\bar{\theta}(f_0^1, c_{j_1}, f_1^1(c_{j_1}), \cdots, c_{j_m}, f_m^1(c_{j_1}, \ldots, c_{j_m}))$:

$p[\bar{\theta}(f_0^1, c_{j_1}, f_1^1(c_{j_1}), \cdots, c_{j_m}, f_m^1(c_{j_1}, \ldots, c_{j_m}))] =$

$q[\bar{\theta}(\Re(f_0^1), \Re(c_{j_1}), \Re(f_1^1(c_{j_1})), \cdots, \Re(c_{j_m}), \Re(f_m^1(c_{j_1}, \ldots, c_{j_m})))] =$

$q[\bar{\theta}(c_i, c_{j_1}, c_{g_1(j_1)}, \cdots, c_{j_m}, c_{g_m(j_1, \ldots, j_m)})] = 1$

the latter equality holds by $M \models \bar{\theta}(i, j_1, g_1(j_1), \cdots, j_m, g_m(j_1, \ldots, j_m))$ and lemma 3.1.

This completes the proof of the proposition.

Acknowledgement I would like to thank professor Zofia Adamowicz for reading the draft of the paper and for her fruitful criticisms.

References

[1] Adamowicz Z. & Zbierski P. "On Herbrand Type Consistency in Weak Theories", to appear in *Archive for Mathematical Logic.*

[2] Bezboruah A. & Shepherdson J.C. "Gödel's Second Incompleteness Theorem For Q", in *The Journal of Symbolic Logic.* (**41**) 1976, pp. 503-512

[3] Hájek P. & Pudlák P. *Metamathematics of First Order Arithmetic*, Springer-Verlag 1991.

[4] Kay R. *Models of Peano arithmetic*, Oxford Logic Guides 15,. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1991

[5] Nerode A. & Shore R.A. *Logic for Applications*, Springer-Verlag 1993.

[6] Paris J. & Wilkie A. "Δ_0-sets and inducton" Proceedings of the Jadwisin Logic Conference, Open days in Model Theory and Set Theory, Poland, Leeds University Press, 1981, pp. 237-248.

[7] Pudlák P. "Cuts, Consistency Statements and interpretation" in *The Journal of Symbolic Logic*, (**50**) 1985, pp. 423-442.

Monads for natural language semantics

Chung-chieh Shan
Harvard University, 33 Oxford St, Cambridge, MA 02138, USA
ken@digitas.harvard.edu

Abstract. Accounts of semantic phenomena often involve extending types of meanings and revising composition rules at the same time. The concept of *monads* allows many such accounts—for intensionality, variable binding, quantification and focus—to be stated uniformly and compositionally.

1 Introduction

The Montague grammar tradition formulates formal semantics for natural languages in terms of the λ-calculus. Each utterance is considered a tree in which each leaf node is a lexical item whose meaning is a (usually typed) value in the λ-calculus. The leaf node meanings then determine meanings for subtrees, through recursive application of one or more *composition rules*. A composition rule specifies the meaning of a tree in terms of the meanings of its sub-trees. One simple composition rule is function application:

$$[\![x\,y]\!] = [\![x]\!]\big([\![y]\!]\big) : \beta \qquad \text{where } [\![x]\!] : \alpha \to \beta \text{ and } [\![y]\!] : \alpha. \tag{24.1}$$

Here α and β are type variables, and we denote function types by $\to$.

To handle phenomena such as intensionality, variable binding, quantification and focus, we often introduce new types in which to embed existing aspects of meaning and accommodate additional ones. Having introduced new types, we then need to revise our composition rules to reimplement existing functionality. In this way, we often augment semantic theories by simultaneously extending the types of meanings and stipulating new composition rules. When we augment a grammar, its original lexical meanings and composition rules become invalid and require global renovation (typically described as "generalizing to the worst case" (Partee 1996)). Each time we consider a new aspect of meaning, all lexical meanings and composition rules have to be revised.

Over the past decade, the category-theoretic concept of *monads* has gained popularity in computer science as a tool to structure denotational semantics (Moggi 1990; Moggi 1991) and functional programs (Wadler 1992a;

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

Wadler 1992b). When used to structure computer programs, monads allow the substance of a computation to be defined separately from the plumbing that supports its execution, increasing modularity. Many accounts of phenomena in natural language semantics can also be phrased in terms of monads, thus clarifying the account and simplifying the presentation.

In this paper, I will present the concept of monads and show how they can be applied to natural language semantics. To illustrate the approach, I will use four monads to state analyses of well-known phenomena *uniformly* and *compositionally*. By "uniformly" I mean that, even though the analyses make use of a variety of monads, they all invoke monad primitives in the same way. By "compositionally" I mean that the analyses define composition rules in the spirit of Montague grammar. After presenting the monadic analyses, I will discuss combining monads to account for interaction between semantic phenomena.

2 Monadic analyses

Intuitively, a monad is a transformation on types equipped with a composition method for transformed values. Formally, a monad is a triple $(\mathbb{M}, \eta, \star)$, where $\mathbb{M}$ is a type constructor (a map from each type α to a corresponding type $\mathbb{M}\alpha$), and η and $\star$ are functions (pronounced "unit" and "bind")

$$\eta : \alpha \to \mathbb{M}\alpha, \qquad \star : \mathbb{M}\alpha \to (\alpha \to \mathbb{M}\beta) \to \mathbb{M}\beta. \tag{24.2}$$

These two functions are polymorphic in the sense that they must be defined for all types α and β. Roughly speaking, η specifies how ordinary values can be injected into the monad, and $\star$ specifies how computations within the monad compose with each other.[1] Some concrete examples follow.

2.1 The powerset monad; interrogatives

As a first example, consider sets. Corresponding to each type α we have a type $\alpha \to t$, the type of subsets of α. We define[2]

$$\begin{aligned}
\mathbb{M}\alpha &= \alpha \to t && \forall \alpha, && \text{(24.4a)}\\
\eta(a) &= \{a\} : \mathbb{M}\alpha && \forall a : \alpha, && \text{(24.4b)}\\
m \star k &= \textstyle\bigcup_{a \in m} k(a) : \mathbb{M}\beta && \forall m : \alpha \to t,\ k : \alpha \to \beta \to t. && \text{(24.4c)}
\end{aligned}$$

[1] By definition, η and $\star$ must satisfy left identity, right identity, and associativity:

$$\begin{aligned}
\eta(a) \star k &= k(a) && \forall a : \alpha,\ k : \alpha \to \mathbb{M}\beta, && \text{(24.3a)}\\
m \star \eta &= m && \forall m : \mathbb{M}\alpha, && \text{(24.3b)}\\
(m \star k) \star l &= m \star (\lambda v.\, k(v) \star l) && \forall m : \mathbb{M}\alpha,\ k : \alpha \to \mathbb{M}\beta,\ l : \beta \to \mathbb{M}\gamma. && \text{(24.3c)}
\end{aligned}$$

[2] In this section and the next, we treat types as sets in order to define the powerset and pointed powerset monads. These two monads do not exist in every model of the λ-calculus.

The powerset monad is a crude model of non-determinism. For example, the set of individuals m_1 defined by

$$m_1 = \{\text{John}, \text{Mary}\} : \mathbb{M}e$$

can be thought of as a non-deterministic individual—it is ambiguous between John and Mary. Similarly, the function k_1 defined by

$$\begin{aligned} k_1 &: e \to \mathbb{M}(e \to t) \\ k_1(a) &= \{\lambda x.\, \text{like}(a, x), \lambda x.\, \text{hate}(a, x)\} : \mathbb{M}(e \to t) \end{aligned}$$

maps each individual to a non-deterministic property. To apply the function k_1 to the individual m_1, we compute

$$\begin{aligned} m_1 \star k_1 &= \textstyle\bigcup_{a \in \{\text{John},\text{Mary}\}} \{\lambda x.\, \text{like}(a, x), \lambda x.\, \text{hate}(a, x)\} \\ &= \{\lambda x.\, \text{like}(\text{John}, x), \lambda x.\, \text{hate}(\text{John}, x), \\ &\qquad \lambda x.\, \text{like}(\text{Mary}, x), \lambda x.\, \text{hate}(\text{Mary}, x)\} : \mathbb{M}(e \to t). \end{aligned}$$

We see that the non-determinism in both m_1 and k_1 is carried through to produce a 4-way-ambiguous result.

Most words in natural language are not ambiguous in the way m_1 and k_1 are. To upgrade an ordinary (deterministic) value of any type α to the corresponding non-deterministic type $\mathbb{M}\alpha$, we can apply η to the ordinary value, say John:

$$\begin{aligned} \eta(\text{John}) &= \{\text{John}\} : \mathbb{M}e, \\ \{\text{John}\} \star k_1 &= \{\lambda x.\, \text{like}(\text{John}, x), \lambda x.\, \text{hate}(\text{John}, x)\} : \mathbb{M}(e \to t). \end{aligned} \tag{24.5}$$

Similarly, to convert an ordinary function to a non-deterministic function, we can apply η to the output of the ordinary function, say k_2 below:

$$\begin{aligned} k_2 &= \lambda a.\, \lambda x.\, \text{like}(a, x) : e \to e \to t, \\ \eta \circ k_2 &= \lambda a. \{\lambda x.\, \text{like}(a, x)\} : e \to \mathbb{M}(e \to t), \\ m_1 \star (\eta \circ k_2) &= \{\lambda x.\, \text{like}(\text{John}, x), \lambda x.\, \text{like}(\text{Mary}, x)\} : \mathbb{M}(e \to t). \end{aligned} \tag{24.6}$$

In both (24.5) and (24.6), an ordinary value is made to work with a non-deterministic value by upgrading it to the non-deterministic type.

Consider now the function application rule (24.1). We can regard it as a two-argument function, denoted A and defined by

$$\begin{aligned} A &: (\alpha \to \beta) \to \alpha \to \beta, \\ A(f)(x) &= f(x) : \beta \qquad \forall f : \alpha \to \beta,\, x : \alpha. \end{aligned} \tag{24.7}$$

We can lift ordinary function application A to non-deterministic function application $A_{\mathbb{M}}$, defined by

$$\begin{aligned} A_{\mathbb{M}} &: \mathbb{M}(\alpha \to \beta) \to \mathbb{M}\alpha \to \mathbb{M}\beta, \\ A_{\mathbb{M}}(f)(x) &= f \star \big[\lambda a.\, x \star [\lambda b.\, \eta(a(b))]\big] : \mathbb{M}\beta \\ &\qquad \forall f : \mathbb{M}(\alpha \to \beta),\, x : \mathbb{M}\alpha. \end{aligned} \tag{24.8}$$

Substituting (24.4) into (24.8), we get

$$A_{\mathbb{M}}(f)(x) = \{\, a(b) \mid a \in f,\, b \in x \,\} \subseteq \beta \qquad \forall f \subseteq \alpha \to \beta,\, x \subseteq \alpha. \tag{24.9}$$

Just as the definition of A in (24.7) gives rise to the original composition rule (24.1), that is,

$$[\![x\,y]\!] = A([\![x]\!])([\![y]\!]), \tag{24.10}$$

the definition of $A_{\mathbb{M}}$ in (24.8) gives rise to the revised composition rule

$$[\![x\,y]\!] = A_{\mathbb{M}}([\![x]\!])([\![y]\!]). \tag{24.11}$$

For the powerset monad, this revised rule is the set-tolerant composition rule in the alternative semantics analysis of interrogatives first proposed by Hamblin (1973). In Hamblin's analysis, the meaning of each interrogative constituent is a set of alternatives available as answers to the question; this corresponds to the definition of $\mathbb{M}$ in (24.4). By contrast, the meaning of each non-interrogative constituent is a singleton set; this corresponds to the definition of η in (24.4).

To support question-taking verbs (such as *know* and *ask*), we (and Hamblin) need a secondary composition rule in which A is lifted with respect to the function f but not the argument x:

$$\begin{aligned} [\![x\,y]\!] &= A'_{\mathbb{M}}([\![x]\!])([\![y]\!]) \qquad \text{where} \\ A'_{\mathbb{M}} &: \mathbb{M}(\mathbb{M}\alpha \to \beta) \to \mathbb{M}\alpha \to \mathbb{M}\beta, \\ A'_{\mathbb{M}}(f)(x) &= f \star \big[\lambda a.\, \eta(a(x))\big] : \mathbb{M}\beta \qquad \forall f : \mathbb{M}(\mathbb{M}\alpha \to \beta),\, x : \mathbb{M}\alpha. \end{aligned} \tag{24.12}$$

Substituting (24.4) into (24.12), we get

$$A'_{\mathbb{M}}(f)(x) = \{\, a(x) \mid a \in f \,\} \subseteq \beta \qquad \forall f \subseteq (\alpha \to t) \to \beta,\, x \subseteq \alpha, \tag{24.13}$$

Note that, for any given pair of types of $[\![x]\!]$ and $[\![y]\!]$, at most one of $A_{\mathbb{M}}$ (24.8) and $A'_{\mathbb{M}}$ (24.12) can apply. Thus the primary composition rule (24.11) and the secondary composition rule (24.12) never conflict.

2.2 The pointed powerset monad; focus

A variation on the powerset monad (24.4) is the *pointed powerset monad*; it is implicitly involved in Rooth's (1996) account of focus in natural language. A *pointed set* is a nonempty set with a distinguished member. In other words, a pointed set x is a pair $x = (x_0, x_1)$, such that x_0 is a member of the set x_1. Define the pointed powerset monad by

$$\mathbb{M}\alpha = \big\{\, (x_0, x_1) \mid x_0 \in x_1 \subseteq \alpha \,\big\} \qquad \forall \alpha, \tag{24.14a}$$

$$\eta(a) = \big(a, \{a\}\big) : \mathbb{M}\alpha \qquad \forall a : \alpha, \tag{24.14b}$$

$$m \star k = \big([k(m_0)]_0, \textstyle\bigcup_{a \in m_1} [k(a)]_1\big) : \mathbb{M}\beta \qquad \forall m : \mathbb{M}\alpha,\ k : \alpha \to \mathbb{M}\beta. \tag{24.14c}$$

This definition captures the intuition that we want to keep track of both a set of non-deterministic alternatives and a particular alternative in the set. As with the powerset monad, the definition of $m\star k$ carries the non-determinism in both m and k through to the result.

Substituting our new monad definition (24.14) into the previously lifted application formula (24.8) gives

$$A_{\mathbb{M}}(f_0, f_1)(x_0, x_1) = \bigl(f_0(x_0), \{\, a(b) \mid a \in f_1,\, b \in x_1 \,\}\bigr) : \mathbb{M}\beta \\ \forall (f_0, f_1) : \mathbb{M}(\alpha \to \beta),\ (x_0, x_1) : \mathbb{M}\alpha. \tag{24.15}$$

This formula, in conjunction with the primary composition rule (24.11), is equivalent to Rooth's recursive definition of focus semantic values.

Crucially, even though the pointed powerset monad extends our meaning types to accommodate focus information, neither our definition of $A_{\mathbb{M}}$ in (24.8) nor our composition rule (24.11) needs to change from before. Moreover, the majority of our lexical meanings have nothing to do with focus and thus need not change either. For example, in the hypothetical lexicon entry $[\![\mathit{John}]\!] = \eta(\text{John})$, the upgrade from meaning type e to meaning type $\mathbb{M}e$ occurs automatically due to the redefinition of η.

2.3 The reader monad; intensionality and variable binding

Another monad often seen in computer science is the *reader monad*, also known as the *environment monad*. This monad encodes dependence of values on some given input. To define the reader monad, fix a type ρ—say the type s of possible worlds, or the type g of variable assignments—then let

$$\mathbb{M}\alpha = \rho \to \alpha \qquad \forall \alpha, \tag{24.16a}$$

$$\eta(a) = \lambda w.\, a : \mathbb{M}\alpha \qquad \forall a : \alpha, \tag{24.16b}$$

$$m \star k = \lambda w.\, k\bigl(m(w)\bigr)(w) : \mathbb{M}\beta \qquad \forall m : \mathbb{M}\alpha,\ k : \alpha \to \mathbb{M}\beta. \tag{24.16c}$$

Note how the definition of $m\star k$ threads the input w through both m and k to produce the result. To see this threading process in action, let us once again substitute our monad definition (24.16) into the definition of $A_{\mathbb{M}}$ in (24.8):

$$A_{\mathbb{M}}(f)(x) = \lambda w.\, f(w)\bigl(x(w)\bigr) : \mathbb{M}\beta \qquad \forall f : \mathbb{M}(\alpha \to \beta),\ x : \mathbb{M}\alpha. \tag{24.17}$$

For $\rho = s$, we can think of $\mathbb{M}$ as the intensionality monad, noting that (24.17) is exactly the usual extensional composition rule. While words such as *student* and *know* have meanings that depend on the possible world w, words such as *is* and *and* do not. We can upgrade the latter by applying η.

For $\rho = g$, we can think of $\mathbb{M}$ as the variable binding monad, noting that (24.17) is the usual assignment-preserving composition rule. Except for pronominals, most word meanings do not refer to the variable assignment. Thus we can upgrade the majority of word meanings by applying η.

If we substitute the same monad definition (24.16) into the secondary composition rule (24.12), the result is

$$A'_{\mathbb{M}}(f)(x) = \lambda w.\, f(w)(x) : \mathbb{M}\beta \qquad \forall f : \mathbb{M}(\mathbb{M}\alpha \to \beta),\, x : \mathbb{M}\alpha. \tag{24.18}$$

For $\rho = s$, this is the intensional composition rule; it handles sentence-taking verbs such as *know* and *believe* (of type $s \to (s \to t) \to e \to t$) by allowing them to take arguments of type $s \to t$ rather than type t. The monad laws, by the way, guarantee that $A'_{\mathbb{M}}(f)(x) = A_{\mathbb{M}}(f)\big(\eta(x)\big)$ for all f and x; the function η (in this case a map from $s \to t$ to $s \to s \to t$) is simply the intension (up) operator, usually written $^\wedge$.

For $\rho = g$, the same formula (24.18) is often involved in accounts of quantification that assume quantifier raising at LF, such as that in Heim and Kratzer (1998). It handles raised quantifiers (of type $g \to (g \to t) \to t$) by allowing them to take arguments of type $g \to t$ rather than type t. The function η (in this case a map from $g \to t$ to $g \to g \to t$) is simply the variable abstraction operator.

2.4 The continuation monad; quantification

Barker (2000) proposed an analysis of quantification in terms of *continuations.* The basic idea is to *continuize* a grammar by replacing each meaning type α with its corresponding continuized type $(\alpha \to t) \to t$ throughout. As a special case, the meaning type of NPs is changed from e to $(e \to t) \to t$, matching the original treatment of English quantification by Montague (1974).

In general, for any fixed type ω (say t), we can define a *continuation monad* with *answer type* ω:

$$\mathbb{M}\alpha = (\alpha \to \omega) \to \omega \qquad \forall \alpha, \tag{24.19a}$$

$$\eta(a) = \lambda c.\, c(a) : \mathbb{M}\alpha \qquad \forall a : \alpha, \tag{24.19b}$$

$$m \star k = \lambda c.\, m\big(\lambda a.\, k(a)(c)\big) : \mathbb{M}\beta \qquad \forall m : \mathbb{M}\alpha,\ k : \alpha \to \mathbb{M}\beta. \tag{24.19c}$$

The value c manipulated in these definitions is known as the *continuation.* Intuitively, "the continuation represents an entire (default) future for the computation" (Kelsey, R., W. Clinger, and J. Rees (Eds.) 1998). Each value of type $\mathbb{M}\alpha$ must turn a continuation (of type $\alpha \to \omega$) into an answer (of type ω). The most obvious way to do so, encoded in the definition of η above, is to feed the continuation a value of type α:

$$[\![\mathit{John}]\!] = \eta(\text{John}) = \lambda c.\, c(\text{John}) : \mathbb{M}e, \tag{24.20a}$$

$$[\![\mathit{smokes}]\!] = \eta(\text{smoke}) = \lambda c.\, c(\text{smoke}) : \mathbb{M}(e \to t). \tag{24.20b}$$

To compute the meaning of *John smokes*, we first substitute our monad definition (24.19) into the primary composition operation $A_{\mathbb{M}}$ (24.8):

$$\begin{gathered} A_{\mathbb{M}}(f)(x) = \lambda c.\, f\big(\lambda g.\, x\big(\lambda y.\, c(g(y))\big)\big) : \mathbb{M}\beta \\ \forall f : \mathbb{M}(\alpha \to \beta),\, x : \mathbb{M}\alpha. \end{gathered} \tag{24.21}$$

Letting $f = \eta(\text{smoke})$ and $x = \eta(\text{John})$ then gives

$$\begin{aligned}[\![\textit{John smokes}]\!] &= \lambda c.\, \eta(\text{smoke})\big(\lambda g.\, \eta(\text{John})\big(\lambda y.\, c(g(y))\big)\big)\\ &= \lambda c.\, \eta(\text{John})\big(\lambda y.\, c(\text{smoke}(y))\big) = \lambda c.\, c(\text{smoke}(\text{John})) : \mathbb{M}t.\end{aligned}$$

In the second step above, note how the term $\lambda y.\, c(\text{smoke}(y))$ represents the future for the computation of $[\![\textit{John}]\!]$, namely to check whether he smokes, then pass the result to the context c containing the clause. If *John smokes* is the main clause, then the context c is simply the identity function id_ω. We define an evaluation operator $\varepsilon : \mathbb{M}\omega \to \omega$ by $\varepsilon(m) = m(\mathrm{id}_\omega)$. Fixing $\omega = t$, we then have $\varepsilon\big([\![\textit{John smokes}]\!]\big) = \text{smoke}(\text{John})$, as desired.

Continuing to fix $\omega = t$, we can specify a meaning for *everyone*:

$$[\![\textit{everyone}]\!] = \lambda c.\, \forall x.\, c(x) : \mathbb{M}e. \tag{24.22}$$

This formula is not of the form $\lambda c.\, c(\dots)$. In other words, the meaning of *everyone* non-trivially manipulates the continuation, and so cannot be obtained from applying η to an ordinary value. Using the continuized composition rule (24.21), we now compute a denotation for *everyone smokes*:

$$\begin{aligned}[\![\textit{everyone smokes}]\!] &= \lambda c.\, \eta(\text{smoke})\big(\lambda g. [\![\textit{everyone}]\!]\big(\lambda y.\, c(g(y))\big)\big)\\ &= \lambda c. [\![\textit{everyone}]\!]\big(\lambda y.\, c(\text{smoke}(y))\big) = \lambda c.\, \forall x.\, c(\text{smoke}(x)) : \mathbb{M}t,\end{aligned}$$

giving $\varepsilon\big([\![\textit{everyone smokes}]\!]\big) = \forall x.\, \text{smoke}(x) : t$, as desired.

The main theoretical advantage of this analysis is that it is a compositional, in-situ analysis that does not invoke quantifier raising. Moreover, note that a grammar continuized is still a grammar—the continuized composition rule (24.21) is perfectly interpretable using the standard machinery of Montague grammar. In particular, we do not invoke any type ambiguity or flexibility as proposed by Partee and Rooth (1983) and Hendriks (1993); the interpretation mechanism performs no type-shifting at "run-time".

This desirable property also holds of the other monadic analyses I have presented. For instance, in a grammar with intensionality, meanings that use intensionality (for example $[\![\textit{student}]\!]$) are identical in type to meanings that do not (for example $[\![\textit{is}]\!]$). The interpretation mechanism does not dynamically shift the type of *is* to match that of *student*.

It is worth relating the present analysis to the computer science literature. Danvy and Filinski (1990) studied *composable continuations*, which they manipulated using two operators "shift" and "reset". We can define shift and reset for the continuation monad by (Wadler 1994)

$$\text{shift} = \lambda h.\, \lambda c.\, \varepsilon\big(h(\lambda a.\, \lambda c'.\, c'(c(a)))\big) : \big((\alpha \to \mathbb{M}\omega) \to \mathbb{M}\omega\big) \to \mathbb{M}\alpha, \tag{24.23}$$

$$\text{reset} = \lambda m.\, \lambda c.\, c\big(\varepsilon(m)\big) : \mathbb{M}\omega \to \mathbb{M}\omega. \tag{24.24}$$

Assuming that the $\forall$ operator is of type $(e \to t) \to t$, the meaning of *everyone* specified in (24.22) is simply $\text{shift}\big(\lambda c.(\eta \circ \forall)(\varepsilon \circ c)\big)$. To encode scope islands, Barker implicitly used reset. Filinski (1999) proved that, in a certain sense, composable continuations can simulate monads.

3 Combining monads

Having placed various semantic phenomena in a monadic framework, we now ask a natural question: Can we somehow combine monads in a modular fashion to characterize interaction between semantic phenomena, for example between intensionality and quantification?

Unfortunately, there exists no general construction for composing two arbitrary monads, say $(\mathbb{M}_1, \eta_1, \star_1)$ and $(\mathbb{M}_2, \eta_2, \star_2)$, into a new monad of the form $(\mathbb{M}_3 = \mathbb{M}_1 \circ \mathbb{M}_2, \eta_3, \star_3)$ (King and Wadler 1993; Jones and Duponcheel 1993). One might still hope to specialize and combine monads with additional structure, to generalize and combine monads as instances of a broader concept, or even to find that obstacles in combining monads are reflected in semantic constraints in natural language.

Researchers in denotational semantics of programming languages have made several proposals towards combining monadic functionality, none of which are completely satisfactory (Moggi 1990; Steele 1994; Liang et al. 1995; Espinosa 1995; Filinski 1999). In this section, I will relate one prominent approach to natural language semantics.

3.1 Monad morphisms

One approach to combining monads, taken by Moggi, Liang et al., and Filinski, is to compose *monad morphisms* instead of monads themselves. A monad morphism (also known as a *monad transformer* or a *monad layering*) is a map from monads to monads; it takes an arbitrary monad and transforms it into a new monad, presumably defined in terms of the old monad and supporting a new layer of functionality. For instance, given any monad $(\mathbb{M}_1, \eta_1, \star_1)$ and fixing a type ρ, the *reader monad morphism* constructs a new monad $(\mathbb{M}_2, \eta_2, \star_2)$, defined by

$$\mathbb{M}_2\,\alpha = \rho \to \mathbb{M}_1\,\alpha \qquad \forall \alpha, \tag{24.25a}$$

$$\eta_2(a) = \lambda w.\, \eta_1(a) : \mathbb{M}_2\,\alpha \qquad \forall a : \alpha, \tag{24.25b}$$

$$m \star_2 k = \lambda w. \big[m(w) \star_1 \lambda a.\, k(a)(w)\big] : \mathbb{M}_2\,\beta \qquad \forall m : \mathbb{M}_2\,\alpha,\; k : \alpha \to \mathbb{M}_2\,\beta. \tag{24.25c}$$

If we let the old monad $(\mathbb{M}_1, \eta_1, \star_1)$ be the *identity monad*, defined by $\mathbb{M}_1\,\alpha = \alpha$, $\eta_1(a) = a$, and $m \star_1 k = k(m)$, then the new monad (24.25) is just the reader monad (24.16). If we let the old monad be some other monad—even the reader monad itself—the new monad adds reader functionality.

By definition, each monad morphism must specify how to embed computations inside the old monad into the new monad. More precisely, each monad morphism must provide a function (pronounced "lift")

$$\ell : \mathbb{M}_1\,\alpha \to \mathbb{M}_2\,\alpha, \tag{24.26}$$

polymorphic in α.[3] For the reader monad morphism, ℓ is defined by

$$\ell(m) = \lambda w.\, m : \mathbb{M}_2\,\alpha \qquad \forall m : \mathbb{M}_1\,\alpha. \tag{24.28}$$

The continuation monad also generalizes to a monad morphism. Fixing an answer type ω, the *continuation monad morphism* takes any monad $(\mathbb{M}_1, \eta_1, \star_1)$ to the monad $(\mathbb{M}_2, \eta_2, \star_2)$ defined by

$$\mathbb{M}_2\,\alpha = (\alpha \to \mathbb{M}_1\,\omega) \to \mathbb{M}_1\,\omega \qquad \forall \alpha, \tag{24.29a}$$

$$\eta_2(a) = \lambda c.\, c(a) : \mathbb{M}_2\,\alpha \qquad \forall a : \alpha, \tag{24.29b}$$

$$m \star_2 k = \lambda c.\, m\big(\lambda a.\, k(a)(c)\big) : \mathbb{M}_2\,\beta \qquad \forall m : \mathbb{M}_2\,\alpha,\ k : \alpha \to \mathbb{M}_2\,\beta. \tag{24.29c}$$

The lifting function ℓ for the continuation monad morphism is defined by

$$\ell(m) = \lambda c.(m \star_1 c) : \mathbb{M}_2\,\alpha \qquad \forall m : \mathbb{M}_1\,\alpha. \tag{24.30}$$

Monad morphisms can be freely composed with each other, though the order of composition is significant. Applying the continuation monad morphism to the reader monad is equivalent to applying to the identity monad the composition of the continuation monad morphism and the reader monad morphism, and yields a monad with type constructor $\mathbb{M}\alpha = (\alpha \to \rho \to \omega) \to \rho \to \omega$. Applying the reader monad morphism to the continuation monad is equivalent to applying to the identity monad the composition of the reader monad morphism and the continuation monad morphism, and yields a different monad, with type constructor $\mathbb{M}\alpha = \rho \to (\alpha \to \omega) \to \omega$.

3.2 Translating monads to monad morphisms

The monad morphisms (24.25) and (24.29) may appear mysterious, but we can in fact obtain them from their monad counterparts (24.16) and (24.19) via a mechanical translation. The translation takes a monad $(\mathbb{M}_0, \eta_0, \star_0)$ whose η_0 and $\star_0$ operations are λ-terms, and produces a morphism mapping any old monad $(\mathbb{M}_1, \eta_1, \star_1)$ to a new monad $(\mathbb{M}_2, \eta_2, \star_2)$. The translation is defined recursively on the structure of λ-types and λ-terms, as follows.[4]

Every type τ is either a function type or a base type. A function type has the form $\tau_1 \to \tau_2$, where τ_1 and τ_2 are types. A base type ι is a type fixed in $\mathbb{M}_0$ (ρ and ω in our cases), a polymorphic type variable (α and β

[3] By definition, ℓ must satisfy naturality:

$$\ell\big(\eta_1(a)\big) = \eta_2(a) \qquad \forall a : \alpha, \tag{24.27a}$$

$$\ell\big(m \star_1 k\big) = \ell(m) \star_2 (\ell \circ k) \qquad \forall m : \mathbb{M}_1\,\alpha,\ k : \alpha \to \mathbb{M}_1\,\beta. \tag{24.27b}$$

[4] Among other things, the translation requires $\mathbb{M}_0$ to be defined as a λ-type, and η_0 and $\star_0$ to be defined as λ-terms. Thus the translation cannot apply to the powerset and pointed powerset monads (see footnote 2). Nevertheless, any monad morphism (including ones produced by the translation) can be applied to any monad (including these two monads).

as appearing in η and $\star$ (24.2) and ℓ (24.26)), or the terminal type ! (also known as the unit type or the void type). For every type τ, we recursively define its *computation translation* $\lceil\tau\rceil$ and its *value translation* $\lfloor\tau\rfloor$:

$$\lceil\iota\rceil = \mathbb{M}_1\,\iota, \qquad \lfloor\iota\rfloor = \iota, \qquad \lceil\tau_1 \to \tau_2\rceil = \lfloor\tau_1 \to \tau_2\rfloor = \lfloor\tau_1\rfloor \to \lceil\tau_2\rceil\,, \tag{24.31}$$

where ι is any base type.

Each term $e : \tau$ is an application term, an abstraction term, a variable term, or the terminal term. An application term has the form $(e_1 : \tau_1 \to \tau_2)$ $(e_2 : \tau_1) : \tau_2$, where e_1 and e_2 are terms. An abstraction term has the form $(\lambda x : \tau_1.\, e : \tau_2) : \tau_1 \to \tau_2$, where e is a term. A variable term has the form $x : \tau$, where x is the name of a variable of type τ. The terminal term is $! : !$ and represents the unique value of the terminal type !. For every term $e : \tau$, we recursively define its *term translation* $\lceil e\rceil : \lceil\tau\rceil$:

$$\begin{aligned}
&\big\lceil(e_1 : \iota \to \tau_1 \to \cdots \to \tau_n \to \iota')(e_2 : \iota)\big\rceil = \\
&\qquad \lambda y_1 : \lfloor\tau_1\rfloor.\ldots\lambda y_n : \lfloor\tau_n\rfloor.\big[\lceil e_2\rceil \star_1 \big(\lambda y_0 : \iota.\, \lceil e_1\rceil\,(y_0)\ldots(y_n)\big)\big\rceil, && (24.32a)\\
&\big\lceil(e_1 : (\tau_1 \to \tau_2) \to \tau_3)(e_2 : \tau_1 \to \tau_2)\big\rceil = \lceil e_1\rceil\,\big(\lceil e_2\rceil\big), && (24.32b)\\
&\lceil\lambda x : \tau.\, e\rceil = \lambda x : \lfloor\tau\rfloor.\,\lceil e\rceil\,, && (24.32c)\\
&\lceil x : \iota\rceil = \eta_1(x), && (24.32d)\\
&\lceil x : \tau_1 \to \tau_2\rceil = x, && (24.32e)\\
&\lceil !\rceil = \eta_1(!), && (24.32f)
\end{aligned}$$

where ι and ι' are any base types, and $y_0, \ldots, y_n$ are fresh variable names.

Finally, to construct the new monad $(\mathbb{M}_2, \eta_2, \star_2)$, we specify

$$\begin{aligned}
\mathbb{M}_2\,\alpha &= \lceil\mathbb{M}_0\,\alpha\rceil\,, \qquad \eta_2 = \lceil\eta_0\rceil\,, \qquad \star_2 = \lceil\star_0\rceil\,,\\
\ell(m) &= \big\lceil\lambda f : ! \to \alpha.\,\eta_0(f(!))\big\rceil\,(\lambda ! : !.\, m) \quad \forall m : \mathbb{M}_1\,\alpha.
\end{aligned} \tag{24.33}$$

To illustrate this translation, let us expand out $\mathbb{M}_2$ and $\star_2$ in the special case where $(\mathbb{M}_0, \eta_0, \star_0)$ is the reader monad (24.16). From the type translation rules (24.31) and the specification of $\mathbb{M}_2$ in (24.33), we have

$$\mathbb{M}_2\,\alpha = \lceil\rho \to \alpha\rceil = \lfloor\rho\rfloor \to \lceil\alpha\rceil = \rho \to \mathbb{M}_1\,\alpha,$$

matching (24.25a) as desired. From the term translation rules (24.32) and the specification of $\star_2$ in (24.33), we have

$$\begin{aligned}
\star_2 &= \big\lceil\lambda m : \rho \to \alpha.\, \lambda k : \alpha \to \rho \to \beta.\, \lambda w : \rho.\, k\big(m(w)\big)(w)\big\rceil && \text{by (24.16c)}\\
&= \lambda m : \rho \to \mathbb{M}_1\,\alpha.\, \lambda k : \alpha \to \rho \to \mathbb{M}_1\beta.\, \lambda w : \rho.\, \big\lceil k\big(m(w)\big)(w)\big\rceil && \text{by (24.32c)},
\end{aligned}$$

in which

$$\begin{aligned}
\big\lceil k\big(m(w)\big)(w)\big\rceil &= \eta_1(w) \star_1 \lambda y_0 : \rho.\, \big\lceil k\big(m(w)\big)\big\rceil\,(y_0) && \text{by (24.32a), (24.32d)}\\
&= \big\lceil k\big(m(w)\big)\big\rceil\,(w) && \text{by (24.3a)}\\
&= \big(\lceil w\rceil \star_1 \lambda y_0 : \rho.\, \lceil m\rceil\,(y_0)\big) \star_1 \lambda y_0 : \alpha.\, \lceil k\rceil\,(y_0)(w) && \text{by (24.32a)}\\
&= \big(\eta_1(w) \star_1 \lambda y_0 : \rho.\, m(y_0)\big) \star_1 \lambda y_0 : \alpha.\, k(y_0)(w) && \text{by (24.32d), (24.32e)}\\
&= m(w) \star_1 \lambda y_0 : \alpha.\, k(y_0)(w) && \text{by (24.3a)},
\end{aligned}$$

matching (24.25c) as desired.

The intuition behind our translation is to treat the λ-calculus with which $(\mathbb{M}_0, \eta_0, \star_0)$ is defined as a programming language whose terms may have computational side effects. Our translation specifies a semantics for this programming language in terms of $(\mathbb{M}_1, \eta_1, \star_1)$ that is call-by-value and that allows side effects only at base types. That the semantics is call-by-value rather than call-by-name is reflected in the type translation rules (24.31), where we define $\lfloor \tau_1 \to \tau_2 \rfloor$ to be $\lfloor \tau_1 \rfloor \to \lceil \tau_2 \rceil$ rather than $\lceil \tau_1 \rceil \to \lceil \tau_2 \rceil$. That side effects occur only at base types is also reflected in the rules, where we define $\lceil \tau_1 \to \tau_2 \rceil$ to be $\lfloor \tau_1 \to \tau_2 \rfloor$ rather than $\mathbb{M}_1 \lfloor \tau_1 \to \tau_2 \rfloor$. Overall, our translation is a hybrid between the call-by-name Algol translation (Benton et al. 2000, §3.1.2) and the standard call-by-value translation (Wadler 1992a, §8; Benton et al., §3.1.3).

3.3 A call-by-name translation of monads

Curiously, the semantic types generated by monad morphisms seem sometimes not powerful enough. As noted at the end of §3.1, the reader and continuation monad morphisms together give rise to two different monads, depending on the order in which we compose the monad morphisms. Fixing $\rho = s$ for the reader monad morphism and $\omega = t$ for the continuation monad morphism, the two combined monads have the type constructors

$$\mathbb{M}_{cr}\,\alpha = (\alpha \to s \to t) \to s \to t, \qquad \mathbb{M}_{rc}\,\alpha = s \to (\alpha \to t) \to t. \tag{24.34}$$

Consider now sentences such as

John wanted to date every professor (at the party). (24.35)

This sentence has a reading where every professor at the party is a person John wanted to date, but John may not be aware that they are professors. On this reading, note that the world where the property of professorship is evaluated is distinct from the world where the property of dating is evaluated. Therefore, assuming that *to date every professor* is a constituent, its semantic type should mention s in contravariant position at least twice. Unfortunately, the type constructors $\mathbb{M}_{cr}$ and $\mathbb{M}_{rc}$ (24.34) each have only one such occurrence, so neither $\mathbb{M}_{cr}\,t$ nor $\mathbb{M}_{rc}t$ can be the correct type.

Intuitively, the semantic type of *to date every professor* ought to be

$$\big((s \to t) \to s \to t\big) \to s \to t \tag{24.36}$$

or an even larger type. The type (24.36) is precisely equal to $\mathbb{M}_{cr}\,(s \to t)$, but simply assigning $\mathbb{M}_{cr}\,(s \to t)$ as the semantic type of *to date every professor* would be against our preference for the reader monad morphism to be the only component of the semantic system that knows about the type s. Instead, what we would like is to equip the transformation on types taking each α to $\big((s \to \alpha) \to s \to t\big) \to s \to t$ with a composition method for transformed values.

One tentative idea for synthesizing such a composition method is to replace the call-by-value translation described in §3.2 with a call-by-name translation, such as the Algol translation mentioned earlier (Benton et al., §3.1.2).[5] For every type τ, this translation recursively defines a type $[\![\tau]\!]$:

$$[\![\iota]\!] = \mathbb{M}_1\,\iota, \qquad [\![\tau_1 \to \tau_2]\!] = [\![\tau_1]\!] \to [\![\tau_2]\!], \tag{24.37}$$

where ι is any base type. For every term $e : \tau$, this translation recursively defines a term $[\![e]\!] : [\![\tau]\!]$:

$$\begin{aligned} [\![e_1(e_2)]\!] &= [\![e_1]\!]\big([\![e_2]\!]\big), & [\![x]\!] &= x, \\ [\![\lambda x : \tau.\, e]\!] &= \lambda x : [\![\tau]\!].[\![e]\!], & [\![!]\!] &= \eta_1(!). \end{aligned} \tag{24.38}$$

Applying this translation to a monad $(\mathbb{M}_0, \eta_0, \star_0)$ gives the types

$$[\![\eta_0]\!] : \mathbb{M}_1\,\alpha \to [\![\mathbb{M}_0\,\alpha]\!], \tag{24.39a}$$

$$[\![\star_0]\!] : [\![\mathbb{M}_0\,\alpha]\!] \to \big(\mathbb{M}_1\,\alpha \to [\![\mathbb{M}_0\,\beta]\!]\big) \to [\![\mathbb{M}_0\,\beta]\!]. \tag{24.39b}$$

If we let $(\mathbb{M}_0, \eta_0, \star_0)$ be the continuation monad (24.19) and $(\mathbb{M}_1, \eta_1, \star_1)$ be the reader monad (24.16), then the type t transformed is

$$[\![\mathbb{M}_0\,t]\!] = (\mathbb{M}_1\,t \to \mathbb{M}_1\,t) \to \mathbb{M}_1\,t = \big((s \to t) \to s \to t\big) \to s \to t,$$

as desired. However, unless $(\mathbb{M}_1, \eta_1, \star_1)$ is the identity monad, the types in (24.39) do not match the definition of monads in (24.2). In other words, though our call-by-name translation does give the type transform we want as well as a composition method in some sense, its output is not a monad morphism.

4 Conclusion

In this paper, I used monads to characterize the similarity between several semantic accounts—for interrogatives, focus, intensionality, variable binding, and quantification.[6] In each case, the same monadic composition rules and mostly the same lexicon were specialized to a different monad. The monad primitives η and $\star$ recur in semantics with striking frequency.

It remains to be seen whether monads would provide the appropriate conceptual encapsulation for a semantic theory with broader coverage. In particular, for both natural and programming language semantics, combining monads—or perhaps monad-like objects—remains an open issue that promises additional insight.

Acknowledgments Thanks to Stuart Shieber, Dylan Thurston, Chris Barker, and the anonymous referees for helpful discussions and comments.

[5]Another possible translation is the standard ("Haskell") call-by-name one (Wadler 1992a, §8; Benton et al., §3.1.1). It produces strictly larger types than the Algol call-by-name translation, for instance $s \to \big(s \to (s \to \alpha) \to s \to t\big) \to s \to t$.

[6]Other phenomena that may fall under the monadic umbrella include presuppositions (the error monad) and dynamic semantics (the state monad).

References

Barker, C. (2000). Continuations and the Nature of Quantification. Manuscript, University of California, San Diego, 4 November 2000, `http://semanticsarchive.net/Archive/902ad5f7/`.

Benton, N., J. Hughes, and E. Moggi (2000). Monads and Effects. Lecture notes, International Summer School on Applied Semantics, 5 September 2000, `http://www.disi.unige.it/person/MoggiE/APPSEM00/`.

Danvy, O. and A. Filinski (1990). Abstracting Control. In *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, pp. 151–160. New York: ACM Press.

Espinosa, D. A. (1995). *Semantic Lego*. Ph. D. thesis, Graduate School of Arts and Sciences, Columbia University.

Filinski, A. (1999). Representing Layered Monads. In *POPL '99: Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, San Antonio, TX, pp. 175–188. New York: ACM Press.

Hamblin, C. L. (1973). Questions in Montague English. *Foundations of Language 10*, 41–53.

Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar*. Oxford: Blackwell.

Hendriks, H. (1993). *Studied Flexibility: Categories and Types in Syntax and Semantics*. Ph. D. thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Jones, M. P. and L. Duponcheel (1993). Composing Monads. Technical Report YALEU/DCS/RR-1004, Department of Computer Science, Yale University, New Haven.

Kelsey, R., W. Clinger, and J. Rees (Eds.) (1998). Revised^5 Report on the Algorithmic Language Scheme. *Higher-Order and Symbolic Computation 11*(1), 7–105. Also in *ACM SIGPLAN Notices 33*(9), 26–76.

King, D. J. and P. Wadler (1993). Combining Monads. In J. Launchbury and P. M. Sansom (Eds.), *Functional Programming, Glasgow 1992: Proceedings of the 1992 Glasgow Workshop on Functional Programming*, Ayr, Scotland. Berlin: Springer-Verlag.

Lappin, S. (Ed.) (1996). *The Handbook of Contemporary Semantic Theory*. Oxford: Blackwell.

Liang, S., P. Hudak, and M. Jones (1995). Monad Transformers and Modular Interpreters. In *POPL '95: Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, San Francisco, CA, pp. 333–343. New York: ACM Press.

Moggi, E. (1990). An Abstract View of Programming Languages. Technical Report ECS-LFCS-90-113, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, Edinburgh.

Moggi, E. (1991). Notions of Computation and Monads. *Information and Computation 93*(1), 55–92.

Montague, R. (1974). The Proper Treatment of Quantification in Ordinary English. In R. Thomason (Ed.), *Formal Philosophy: Selected Papers of Richard Montague*, pp. 247–270. New Haven: Yale University Press.

Partee, B. (1996). The Development of Formal Semantics. See Lappin (1996), pp. 11–38.

Partee, B. and M. Rooth (1983). Generalized conjunction and type ambiguity. In R. Bauerle, C. Schwartze, and A. von Stechow (Eds.), *Meaning, Use and Interpretation of Language*, pp. 361–383. Berlin: De Gruyter.

Rooth, M. (1996). Focus. See Lappin (1996), pp. 271–297.

Steele, Jr., G. L. (1994). Building Interpreters by Composing Monads. In *POPL '94: Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, Portland, OR, pp. 472–492. New York: ACM Press.

Wadler, P. (1992a). Comprehending Monads. *Mathematical Structures in Computer Science 2*(4), 461–493.

Wadler, P. (1992b). The Essence of Functional Programming. In *POPL '92: Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, Albuquerque, NM, pp. 1–14. New York: ACM Press.

Wadler, P. (1994). Monads and Composable Continuations. *Lisp and Symbolic Computation 7*(1), 39–56.

Incomplete Definite Descriptions, Demonstrative Completion and Redundancy

Isidora Stojanovic
Ecole Polytechnique, CREA, 1 rue Descartes, F-75005 Paris, France
stojanov@poly.polytechnique.fr

Abstract.
"Incomplete" definite descriptions (i.e. descriptions that violate the uniqueness constraint) have been offered various accounts in semantics. Among them, the so-called ellipsis account, which analyzes "the F" as elliptical for "the F which is *that* F". I begin by arguing that the objections raised against this account have not been conclusive, and go on to supply a new argument against it, which consists in showing such demonstrative completions to be semantically redundant.

1 The Problem

A friend, whose dog I was playing with, said to me once:

(1) The dog likes you.

or perhaps:

(2) That dog likes you,

I don't quite remember. To know which was exactly the sentence he used does not seem to matter much. As against such layman's intuitions, most semantic theories see a huge difference between the way in which definite descriptions, like "the dog", and complex demonstratives, like "that dog" contribute to the truth conditions. It is widely agreed on that "that dog" contributes to the truth conditions of (2) by the dog itself, Fido, so that (2) is true iff Fido likes me (at the time of the utterance)[1]. But when it comes

[1]This view of demonstratives is common to the theories that embody David Kaplan's insights from *Demonstratives*, in Almog J., Perry J. and Wettstein H. (eds.), *Themes from Kaplan*, Oxford University Press, 1989. Such is, for instance, the theory of Larson R. and Segal G. in *Knowledge of Meaning*, MIT Press, 1995.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

to definite descriptions, semanticians are at pains to agree on how "the dog" contributes to the truth conditions of (1). If, following Bertrand Russell[2], "the such-and-such is so-and-so" is true iff exactly one thing is such-and-such, and it is also so-and-so, then (1) is bound to be false, it seems, since there are plenty of dogs in the universe.

"Incomplete" definite descriptions, which violate Russell's uniqueness condition, seem to pose a problem only to the accounts that do *not* view definite descriptions as referential expressions[3]. Why not say, then, that definite descriptions can at least be referentially *used*, so that (1) and (2) end up receiving the same truth conditions[4]? Yet another option is to say that (1) literally says something false, but conveys something true, namely, that Fido likes me[5]. Some might be tempted to say that the domain of discourse relevant to interpreting (1) contains only one dog, Fido, so that the uniqueness condition becomes fulfilled after all. The portion of the universe over which the quantifier ranges would be then specified by the context[6]. At last, some might suggest that not the whole sentence is phonetically realized in (1). (1) would then simply serve as a shorthand for something more complex, like:

(3) The dog that you are playing with right now likes you.

(1) might as well be a shorthand for (2). Then all one needs is recover a completion that will prevent the failure of the uniqueness condition[7].

In the last half of the century, many accounts of incomplete definite descriptions have seen light, but none has received a unanimous support, and, more importantly, none has been refuted either, *pace* claims to the contrary. In this paper, I focus on the last approach, the so-called *ellipsis* approach. I begin by arguing that the arguments offered against it did not prove conclusive. I provide two replies to the argument from *indeterminacy*, which

[2]See *On Denoting*, Mind 59, 1905. Russell's proposal has been incorporated in the framework of generalized quantifiers, as endorsed e.g. by Stephen Neale's in *Descriptions*, MIT Press Cambridge, 1990, or by Larson and Segal (op. cit.)

[3]"Referential" accounts of definite descriptions Cf. Strawson's criticism of Russell, in *On Referring*, Mind 59, 1950. A more recent account can be found e.g. in Heim I., *Articles and Definiteness*, in Stechow A. and Wunderlich D. (eds.) *Semantics. An International Handbook of Contemporary Research*, Berlin: De Gruyter, 1991. The idea is already neat in G. Wilson's *On Definite and Indefinite Descriptions*, The Philosophical Review 87, 1978.

[4]See Donnellan, K., *Reference and Definite Descriptions*, The Philosophical Review 75, 1966, reprinted in Garfield J. and Kiteley M., (eds.), *Meaning and Truth*, Paragon House NY, 1991.

[5]See S. Kripke, *Speaker's Reference and Semantic Reference*, Midwest Studies in Philosophy 2, 1977, and K. Bach in *Conversational Impliciture*, Mind and Language 9, 1994.

[6]The suggestion goes back to Barwise, J. and Cooper, R., *Generalized Quantifiers*, Linguistics and Philosophy 2, 1981. Stanley J. and Szabo S. contend that this is the only sound view of incomplete descriptions. See *On Quantifier Domain Restriction*, Mind and Language 15, 2000.

[7]Stephen Neale, *On Being Explicit*, Mind and Language 15, 2000.

objects to the ellipsis approach that it fails to recover a determinate completion. One reply is that the recovery devices may themselves depend upon the context in a way that allows them to yield a determinate completion. The other reply is that one can associate to every incomplete description a *canonical* completion, namely, a demonstrative completion that recovers "that" over "the". I then argue that the argument from knowledge, which points that one may be ignorant of demonstratives while competent at using incomplete definite descriptions, does not go through for the simple fact that, in general, one does not have to know the phonetic realization of some particular expression in order to make an ellipsis on that expression.

Even though it has got replies to those two arguments, the ellipsis account of incomplete descriptions seems to be deeply mistaken. The reason, I argue, is that demonstrative completions over definite descriptions, likely to be forced upon the account by the argument from indeterminacy, are actually redundant. The cases of incomplete definite descriptions not meant to refer to particular objects – that is, of descriptions used attributively and anaphorically – may help us to get to the redundancy result. It is clear that such incomplete descriptions may be completed into complex demonstratives salva congruitate. The question is how demonstratives used attributively or anaphorically are to be accounted for. If such demonstratives are seen as definite descriptions in disguise, the ellipsis account will be circular and lack a solution to the problem that it purports to solve. If, on the other hand, they are still seen as genuine referential expressions, the account will lack motivations for *not* considering definite descriptions as referential expressions at the outset[8].

2 The Argument from Indeterminacy

In its typical instances, bare ellipsis consists in leaving out some expression possible to recover from the discourse, as in:

(4) Bill wants pie for dessert and Al pudding.

(5) Bill has one child and Al four.

Thus the speaker of (4) fails to pronounce the whole sentence "Al wants pudding for dessert", while the speaker of (5) omits "has" and "children", recoverable from the previously used "child", given the plural constraint due to the numeral "four[9]."

[8]It is noteworthy that from the standpoint of etymology, the ties between demonstratives and the definite article are straightforward. The idea that "the" is a reduced form of "that" may be found e.g. in Jespersen O., *The Philosophy of Grammar*, Allen & Unwin, London, 1924.

[9](4) is an example from K. Bach, who wrote: "Utterances are elliptical, strictly speaking, only if the suppressed material is recoverable (..) by grammatical means alone." (ibid., p. 131)

Some cases of incomplete descriptions fit fine to this picture. Consider:

(6) I saw a neighbor of mine kissing a woman in the staircase. The woman was his wife.

It could be that the description "the woman" is a shorthand for e.g. "the woman that I saw being kissed by a neighbor of mine in the staircase", which is recoverable from the discourse alone. However, not many incomplete descriptions are amenable to this sort of analysis. Take this well-known example[10]: having come upon Smith's cruelly mutilated body, the inspector, without suspecting anyone in particular, says:

(7) The murdered must be insane.

There is nothing in the discourse itself to complete the description. To keep viewing "the murderer" as a shorthand for another description respectful of the uniqueness constraint, the recovery of the elided material had better depend not only upon the linguistic context, but upon the context in its most general sense. Let us call narrow the approach that constrains the recovery of the elided material down to linguistic inputs only, and *broad* the other one. Both philosophers and linguists seem to agree that no narrow approach can be worked out – as (7) already suggests –, and that no broad approach can be worked out either, since there is no algorithm that tells us how to recover the elided material. H. Wettstein wrote: "When one says, e.g., 'The table is covered with books', the table the speaker has in mind can be more fully described in any number of ways (..) Since these more complete descriptions are not synonymous, it follows that each time we replace (..) 'the table' with a different one of these 'Russellian' descriptions, it would seem that we obtain an expression for a different proposition[11]."

Some have thought, erroneously, that Wettstein's worry would not arise if "non- descriptive" completions were possible: completions with referential expressions which, albeit different, would pick out the same referent[12]. But as pointed out by Marga Reimer, "even if completions of incomplete descriptions are stipulated to be non-descriptive, the problem of adjudicating between non-equivalent, co-denoting descriptions remains[13]."

So let me put the argument from indeterminacy as follows: *The ellipsis approach to incomplete descriptions fails because it does not offer any algorithm that may tell us how to recover a determinate completion over an incomplete description.*

[10]See Donnellan, ibid., p. 147.

[11]*Demonstrative Reference and Definite Descriptions*, Philosophical Studies 40, 1981, p. 246.

[12]That was S. Neale's view in *Descriptions*, cf. p. 99 et passim.

[13]*Incomplete Descriptions*, Erkenntnis 37, 1992, p.353. F. Recanati similarly noted: "Even if we accept that the expressed content is singular, still it is totally indeterminate which particular sentence expressing that content the uttered sentence is elliptical for." (*Domains of Discourse*, Linguistics and Philosophy 19, 1996, p.449 fn.)

To rule out this argument, it will not do just to have non-descriptive completions, whose values depend on the context of utterance. The "algorithm" that yields those completions should also be able to depend on the context. Now, Wettstein has argued convincingly enough that there was no obvious algorithm leading from an incomplete description to its completion, and that even the speaker may be unable to point to some determinate completion. But this does not show yet that there are utterances of incomplete descriptions which lack a determinate completion. The point may be made clear by looking at the behavior of demonstratives. It has been agreed on that the linguistic meaning of a demonstrative such as "that" does not correspond to any expression à la "what I am pointing to" or "what I have in mind". There is no algorithm, in other words, that leads from a demonstrative to its referent. But this does not mean that there are utterances of "that" without a determinate referent. The meaning of "that" might exploit all the cues that the context makes available, take those cues as inputs, and give us the referent of "that" as an output. This is perfectly compatible with the fact that *we* did not manage to come up with such an algorithm. Similarly, one may suggest that the meaning of "the" exploits any contextual cue available so as to single out a completion[14].

There is a reply, then, to the argument from indeterminacy. True, it 'passes the buck' to the semantics of "the", but this simply shows that there is no special link between indeterminacy and ellipsis. Besides, it takes little to realize that other accounts of incomplete descriptions are not any better off in this regard. Indeterminacy is a very general phenomenon, and Stanley and Szabo were wrong to think that indeterminacy obliged the ellipsis approach to place "intolerable burdens on any possible solution" to the semantics of "the". They wrote: "context has to provide a specific predicate (..) And it is exceedingly hard to see what feature of context could [select the predicate F among other candidates][15]." At the same time, they did not see the same "intolerable burden" placed on the contextual restriction of the domain of discourse. But take (7). What feature of context could restrict the domain down to a set that contains no other murderers than whoever happens to have murdered Smith? Why should there belong some person rather than another? The only way to restrict the domain appropriately would use, it seems, some clause like "the murderer of the person whose mutilated body we are looking at." And it makes no difference whether this clause is supposed to restrict the domain of discourse, or to be

[14]It is somewhat ironical that Wettstein, who supports this view for demonstratives – see *How to Bridge the Gap Between Meaning and Reference*, Synthese 58, 1984 –, should thus provide a possible reply to his own argument against the ellipsis account of incomplete descriptions.

[15]Ibid., p. 238. The issue of indeterminacy is also known as the issue of underdetermination. For discussion, see Recanati, F., *Direct Reference*, Blackwell Oxford, 1993, p. 235 et passim.

recovered over some other expression.

3 The Argument from Knowledge

There is another possible reply to the argument from indeterminacy, which moreover makes a good case for the narrow ellipsis approach to incomplete descriptions. The reply consists in showing that for every incomplete description, it is possible to come up with a determinate, canonical completion. Thus every definite description "the F" that violates the uniqueness constraint while empirically yielding true utterances may be seen as elliptical for "the F which is *that* F", or simply for "*that* F". The user of an incomplete description may be then seen as having failed to pronounce /at/ after /th/. It looks as if he had used a definite article, whereas he has used a demonstrative.

First of all, it should be shown that canonical completion works. That is obvious for those cases in which there is something contextually salient that the description singles out, in the way in which "the dog" singles out Fido in (1). The cases that are worrisome are rather of the same ilk as (6), in which the description is anaphoric, or as (7), in which it is attributive. But, as it turns out, complex demonstratives also allow for anaphoric as well as for attributive uses[16]. One can utter: "*that* such-and- such is so-and-so", just as one can utter: "*the* such-and-such is so-and-so", in order to attribute the property of being so-and-so to *whatever* is such-and-such, however there is nothing *of* which the speaker wishes to say that *it* is such-and-such. Thus, instead of (7), the inspector might have said as well:

(8) That murdered must be insane.

If we turn to definite descriptions used anaphorically, we also see that they can always be replaced, in a more or less felicitous manner, by complex demonstratives. Thus, instead of (6), I might have said as well:

(9) I saw a neighbor of mine kissing a woman in the staircase. That woman was his wife.

Let us assume, for the sake of the argument, that substituting "that" for "the" generally works. The output of the substitution may be less felicitous than the input, but there shall be no difference in the truth conditions. Will there be anything wrong, then, with the narrow ellipsis approach? Stanley and Szabo thought they had a knockdown argument against it, which I shall call the argument from *knowledge*. They wrote: "Suppose that Max is not a

[16] In *Are Complex 'That' Phrases Devices of Direct Reference?*, Noûs 33, 1999, Jeffrey King uses precisely this sort of cases to motivate an account of complex demonstratives in terms of restricted quantifiers.

fully competent speaker of English. (..) The use of demonstrative pronouns is not discussed until unit 7 and Max is not there yet. (..) Since Max does not know the word 'that', he cannot identify the sentence uttered by the speaker of (1) which contains that word as an unarticulated constituent[17]." As Max, competent in the use of the definite article, has no difficulties to understand (1), Stanley and Szabo were led to conclude that (1) cannot be elliptical for (2).

If this argument were to be taken seriously, it would have been a knock-down argument against ellipsis in general. Thus, suppose that Max does not know what the plural of "child" is, but knows that it is irregular. As he does not want to reveal his ignorance, he says "Bill has one child and Al four", as in (5). Shall we say then that his utterance cannot be elliptical for "Bill has one child and Al has four *children*"? No. Ignorance of the phonetic realization of some grammatical form of an expression is not enough to prevent us from making utterances elliptical on that expression, still less from understanding them[18].

4 The Argument from Redundancy

Let me take stock. The narrow ellipsis approach, which over every incomplete description recovers a canonical, demonstrative completion, meets the argument from indeterminacy in that it does provide a determinate completion, whereas the argument from knowledge has simply proved flawed. Now, the question was how to ascribe correct truth conditions to an utterance that involves a definite description that does not single out anything. If the ellipsis approach is to provide a solution, it must be able to ascribe correct truth conditions to utterances that involve complex demonstratives in lieu of incomplete definite descriptions. The view on which "that" is a referential expression whose truth-conditional contribution consists of its referent has no problem with those cases in which there is some contextually salient referent, as in Fido's case. Rather, the question is how that view deals with other cases, in which demonstratives are used attributively or anaphorically. If demonstratives lose their referential character on such uses, and turn out to be quantifiers in disguise, then the ellipsis approach clearly lacks a clue to the problem of incomplete descriptions. Conversely, if demonstratives keep referential even on such uses, then it must be possible to provide definite descriptions themselves with an account that sees them as referential ex-

[17]Ibid., p. 238; "unarticulated" reads as "elided". I have taken the liberty of making necessary changes so that the quote fits the examples that I have been using here.

[18]Had Max have no *concept* of demonstratives, perhaps he would have given us reasons to be suspicious toward the ellipsis approach. But it is far from clear that anyone without any idea of how demonstratives work could ever grasp what (1) says. And it is not much more clear that the mastery of "that" requires anything more than does the mastery of "the".

pressions. In either case, the recovery of the demonstrative over the definite article must prove unattractive, since the material recovered turns out to be redundant. Albeit possible, the ellipsis approach is far from attractive, for either it lacks a solution to the problem that it purports to solve, or it lacks motivations for not analyzing definite descriptions as referential expressions, while being able to do so[19].

In the remainder of the paper, let me try to see whether there is a referential account of demonstratives that covers their anaphoric and attributive uses. (Keep in mind, though, that the answer does not affect the argument itself.)

The cases of anaphora that we have seen are not, so to say, difficult enough, since there is something singled out by the antecedent. Consider this harder case instead:

(10) Whenever you see a neighbor of yours kissing a woman in the staircase, the woman (/ that woman) (/ she) may easily happen to be his wife.

The quantifier "whenever", which, for our purposes, may be taken to range over situations, makes it clear that there is not a single woman relevant to the truth of (10). How can "that woman" then occur as a referential expression in (10)? It seems that if "that woman" were a genuine demonstrative, there would have to be some particular woman whom it would stand for.

In relation to the same issue as it arises with pronouns, Gareth Evans argued that, notwithstanding appearances, the pronoun "she" may receive a referential analysis even in (10) and the sentences of the same ilk. Evans wrote: "If we adopt a Fregean account of satisfaction, we have only to give an account of the pronoun-antecedent construction as it occurs in singular sentences – no further explanation need be given of pronouns with quantifier antecedents. (..) A natural explanation of the role of pronouns with singular antecedents is in terms of co-reference – the pronoun refers to whatever the antecedent refers to[20]."

Without tackling the details of Evans' account, let us adapt the idea to our needs. We can roughly say that (10) is true iff C is true for every

[19] The cases that pose problems for referential accounts of definite descriptions – cases which had originally motivated Russell's view –, are precisely the cases of descriptions attributively used and the cases of anaphora. If there is a referential account of such cases when they involve demonstratives, that same account should work with descriptions too. The choice, then, to reserve the referential account for demonstratives and to subject definite descriptions to a non-referential one can only be arbitrary and unmotivated.

[20] *Pronouns*, in Collected Papers, Oxford University Press, 1985, p. 227. Frege's account of satisfaction is like the substitutional account, except that, instead of assuming that the language has all the expressions needed for substitution instances, it simply assumes that it can always be *enriched* with any such expression. Cf. *Pronouns, Quantifiers and Relative Clauses*, in ibid.

pair of referential terms substituted for *(x)* and *(y)* respectively, where C is the conditional: 'when you see *(x)* kissing *(y)* in the staircase and *(x)* is a neighbor of yours and *(y)* is a woman, *(y)* may easily happen to be *(x)*'s wife.' The general idea is that every situation in which you can truly say: "I see *him*, a neighbor of mine, kissing *her* in the staircase", is also a situation in which you can truly say: "*she* may well happen to be *his* wife." In other words, there are no more quantifiers that properly range over individuals, but only quantifiers that range over situations[21].

There is one step left from a referential account of demonstratives anaphorically used to a referential account of demonstratives attributively used, hence of definite descriptions. Recall the situation. Watching Smith's body cruelly mutilated, the inspector says:

(11) That murderer must be insane.

How can "that" occur as a genuine demonstrative in (11)? – there is nobody salient in the context that the inspector was talking of, nobody he meant to speak of otherwise than as of *whoever* happened to be the murderer. And if "that murderer" is not a demonstrative, but a definite description in disguise, then it is plainly circular to suggest that "the murder" used under the same circumstances should be elliptical for "that murderer".

Once again, however, it seems possible to continue treating "that murderer" as a demonstrative. All we need to do is make our quantifiers range not over individuals, but over something else – situations, possible substitution instances, events, and so on. On Russellian approaches, (11) is true iff there is one and only one individual who is a murderer, and who, moreover, must be insane. Swaying the range of our quantifiers from individuals to situations, we would say that (11) is true iff there is one and only one situation in which one would speak truly if one said: "That murder must be insane", using "that" in reference to someone demonstrated in that situation. Such an account would clearly preserve the referential character of "that murderer". So the idea is that the demonstrative always acts as a referential expression, i.e. as an expression contributing to the truth conditions by its referent. Yet, in order to fix the referent, one must have settled on some particular situation. Most often, the situation settled on is the current one, that is, the situation in which the utterance is taking place. But that need not be the case. One may also settle on other situations, or even not settle on any particular situation at all. Thus one may simply state the existence of a situation, so that the situation relevant to fixing the reference of "that"

[21] Alternatively, all quantifiers range over (possible) substitution instances. Presumably, this option is more faithful to Evans' original proposal. There are other frameworks favorable to the idea. E.g. one can make quantifiers range over events, or over (possible) contexts. It goes without saying that within the boundaries of the present paper I can at best *hint* at a referential account of unbound anaphora.

would merely be a hypothetical situation – which is what probably happens in (11).

To be sure, the proposal that I have just canvassed does not amount to providing a solution to attributive uses of expressions normally seen as referential, like complex demonstratives, or even pronouns[22]. But what it does show is that the possibility of using an expression attributively is compatible with the possibility of giving that expression a referential account[23]. The referential account that can be given for "that", including its anaphoric and attributive uses, straightforwardly extends into a referential account of the definite article itself. In consequence, an account of definite descriptions in terms of quantifiers that posits a demonstrative completion when it comes to incomplete descriptions, is simply unmotivated – or *dis*motivated, as might be a more proper way of putting it –, given that the initial motivations for a non- referential account of definite descriptions vanish in the presence of a referential account of demonstratives that behave in the same way as definite descriptions do.

[22]Indeed, "that murderer" may be successfully replaced by "he" in (11). Another well-known example is that of a person who, spotting giantic footprints, says: "*He* must be a giant."

[23]In this way, the referential account of "the", built out from the referential account of "that" attributively used, will not blur the epistemological distinction between two manners of individuating things – namely, by direct reference vs. by description –, brought to light by Russell in *Knowledge by Acquaintance and Knowledge by Description*, Proceedings of the Aristotelian Society 11, 1910.

Belief Fragments and Ontological Categories

Jan Westerhoff

Trinity College, Cambridge (UK)

jcw26@cam.ac.uk

Abstract.

There have been attempts to get some logic out of belief dynamics, i.e. attempts to define the constants of propositional logic in terms of functions from sets of beliefs to sets of beliefs.[1] It would be interesting to see whether something similar could be done for ontological categories, i.e. ontological constants. The theory presented here will be a (modest) expansion of belief dynamics: it will not only incorporate beliefs, but also parts of beliefs, so called *belief fragments*. On the basis of this we will give a belief-dynamical account of the ontological categories of states of affairs, individuals, properties of arbitrary adicities and properties of arbitrary orders.

1 The Background: Belief Fragments

The fundamental idea of this paper is that two beliefs can have something in common, namely, a part of a belief, which we will also call a belief fragments. Consider the following two beliefs:

1. *The Duke of Edinburgh is over fifty.*

2. *The American President is over fifty.*

What do these two beliefs have in common? They have something in common which isn't itself a belief, but rather a part of a belief, a belief fragment: this is IS OVER FIFTY. It is important to note that IS OVER FIFTY isn't the same as 'is over fifty' — we are not talking about syntax here. To see this consider a third belief:

3. *The Chancellor of Cambridge University is male.*

The syntactic elements which the *sentences* 1 and 3 have in common ('the', 'of', 'is') are not the belief fragments which the two *beliefs* 1 and 3 have in common. What the beliefs have in common is that they are both

[1] See (3), chapter 6.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

about a certain person, namely about Prince Phillip. The belief fragment PRINCE PHILLIP which they have in common isn't identical to any syntactical element which the English formulations of the two beliefs given above have in common.

Note that we are not assuming that beliefs have any *particular* structure. They will have *some* structure, but for our purposes it is irrelevant what this is. The sentences given above which express beliefs 1–3 happen to share some syntactic elements. This is due to the way we have chosen to express them. But of course we could have expressed them in all sorts of ways (by formulating them in different languages, by using pictures or diagrams), including ways in which they did not have any syntactic features in common. This, however, would not change the fact that the *beliefs expressed* share a common belief fragment.

We are not specifying what kind of entity a fragment is, whether it is linguistic, psychological, material (a brain state, say), functional or whatever. All we claim here is that if beliefs are supposed to be entities of kind X, belief fragments should be considered to be parts of entities of kind X.[2]

There are beliefs which have more than one thing in common. Take for example the following two beliefs:

4. *The apple and the banana are on the table.*

5. *The apple and the banana are both to the left of the vase.*

4 and 5 have two belief fragments in common, namely THE APPLE and THE BANANA. One might be tempted to treat them just as a single fragment THE APPLE AND THE BANANA. But such a procedure cannot be generalized as can be seen from the following example:

6. *Peter washes his white shirt.*

7. *Charles washes his white car.*

Here WASHES and WHITE are fragments common to both 6 and 7 but we would not want to say that there is *one* fragment which somehow contains these two as parts which 6 and 7 have in common. For the sake of simplicity we are therefore going to disregard such cases in our inquiry and confine ourselves to beliefs which have at most a single fragment in common.

For our account we need two primitive notions. The first is the meet operation $\cap_B$. This takes two beliefs and returns the largest fragment which they have in common, else it returns $\emptyset$. Meeting can be iterated: $\cap_B$ can also return a ('smaller') fragment which two fragments have in common. And finally the meet operation can take a belief and a belief fragment and return the smaller fragment which the belief and the belief fragment have

[2] Cf. (4), 7–9.

in common. We will write $B_i, B_j, \ldots$ for beliefs and $B_i^\sharp, B_j^\sharp, \ldots$ for belief fragments.

The second primitive notion is the two-place $\boxplus$ operation which allows us to expand fragments and thus to build complexes out of fragments. The expansion operation takes two belief fragments $B_i^\sharp, B_j^\sharp$ and returns either

1. a belief B_n or
2. another belief fragment $B_m^\sharp$ or
3. the set $\{B_i^\sharp, B_j^\sharp\}$ of the two belief fragments.

Let us consider an example for each possibility in turn. For the first case, let $B_i^\sharp$ be PIERO and $B_j^\sharp$ MARRIED LUCREZIA, then $B_i^\sharp \boxplus B_j^\sharp = B_n =$ PIERO MARRIED LUCREZIA.

For the second possibility, let $B_i^\sharp$ be IS MARRIED TO and $B_j^\sharp$ LORENZO, then $B_i^\sharp \boxplus B_j^\sharp = B_m =$ LORENZO IS MARRIED TO (or IS MARRIED TO LORENZO — see below).

For the final option let $B_i^\sharp$ again be IS MARRIED TO but $B_j^\sharp$ IS THE BROTHER OF. Clearly in this case there is no belief or belief fragment which can be constructed out of $B_i^\sharp$ and $B_j^\sharp$ so $\boxplus$ will return the set of the two as a default value.[3]

We suppose that expansion is commutative and associative, i.e.

(COMM) $B_i^\sharp \boxplus B_j^\sharp = B_j^\sharp \boxplus B_i^\sharp$

(ASS) $(B_i^\sharp \boxplus B_j^\sharp) \boxplus B_k^\sharp = B_i^\sharp \boxplus (B_j^\sharp \boxplus B_k^\sharp)$

As reader will have already noted from the example of $(B \boxplus B_1^\sharp)$, $\boxplus$ behaves quite differently from the expansion operator in standard belief dynamics.

Firstly, $(B_i^\sharp \boxplus B_j^\sharp) \boxplus B_j^\sharp$ will not be the same as $(B_i^\sharp \boxplus B_j^\sharp)$, i.e. expanding a fragment twice by another fragment will not give the same result as doing it only once (as is true for standard expansion where $(A + \alpha) + \alpha = A + \alpha$). For example, expanding the above fragment IS MARRIED TO *twice* by the fragment LORENZO gives LORENZO IS MARRIED TO LORENZO.

Secondly, in the standard case the expansion of a set of beliefs will always deliver a unique object of the same kind, namely a (now expanded) set of beliefs. But with our expansion operator it will not always be the case that an expansion of some fragment will yield some unique new belief fragment (or belief). Suppose some fragment corresponds to the dyadic property 'adores'. If we want to expand this by the belief fragment 'Dante', it is clear

[3] We could plausibly extend $\boxplus$ by allowing it to take a belief fragment and a belief and return a belief fragment. For example, it could take the belief PIERO IS MARRIED TO LUCREZIA and expand it by the fragment ENTAILS THAT. This would then return the belief fragment ENTAILS THAT PIERO IS MARRIED TO LUCREZIA (or PIERO IS MARRIED TO LUCREZIA ENTAILS THAT).

that we get another belief fragment, namely a monadic property. But it is not clear what this property is; it may be either 'Dante adores' or 'adores Dante'. This point generalizes: for every fragment corresponding to an n-adic property (where $n \geq 2$) which is expanded by a belief fragment there are n different possible results.

In order to give a systematic account of our notion of expansion, there are now two possibilities. We could either say that expanding a fragment corresponding to an n-adic property yields a set Γ containing the different possibilities, so that in the above case the result of expanding 'adores' by 'Dante' would be {adores Dante, Dante adores}. Alternatively we could define the expansion in terms of a function $\gamma(\Gamma)$ which selects exactly one member of Γ.

The problem with the first construction is that it brings in an unwelcome asymmetry between e.g. fragments corresponding to 'naturally' monadic properties and those corresponding to monadic properties which were produced from n-adic properties by 'filling in' $n-1$ places. Obviously they will behave differently under expansion: the expansion of the naturally monadic predicates by the appropriate fragments will yield a belief, while the expansion of the formerly n-adic properties will yield a *set* of beliefs. The expansion of IS DEAD by BEATRICE will just give the belief BEATRICE IS DEAD, while expanding the result of expanding ADORES by DANTE by BEATRICE gives the set {DANTE ADORES BEATRICE, BEATRICE ADORES DANTE}.

We will therefore use the second construction. The expansion operator is then defined as a function γ which selects exactly one member from the set Γ of the various possibilities.

2 The Program: Which Fragment is Which?

As we saw above, $\cap_B$ will supply us with various kinds of belief fragments. What we want to try to do is to group the fragments according to the ontological categories they correspond to. There is a fairly natural way in which beliefs correspond to states of affairs so that in a similar way fragments of beliefs should correspond to fragments of states of affairs (i.e. properties and individuals). Is there a way of telling which groups of fragments correspond to which categories?

We will look at an attempt of grouping fragments by their behaviour under expansion. Clearly not every fragment can be used to expand other every fragment: if $B_i^\sharp$ is IS YELLOW and $B_j^\sharp$ is IS BIGGER THAN expanding the one by the other will not be a belief, i.e. there is no belief B such that $B_i^\sharp \boxplus B_j^\sharp = B$.

2.1 Adicity

One way in which this can be exploited is in defining different adicities of properties. Suppose we have to properties $B_i^\sharp$ and $B_j^\sharp$ such that $B_i^\sharp \boxplus B_j^\sharp = B_k^\sharp$ (i.e. expanding the one by the other gives another belief fragment) while $B_k^\sharp \boxplus B_i^\sharp = B$ (i.e. expanding the result by the first fragment gives a belief). In this case the belief fragment $B_j^\sharp$ must correspond to a dyadic property. (Note that this doesn't entail that $B_i^\sharp$ must correspond to an individual — $B_j^\sharp$ could correspond to a property of order n and $B_i^\sharp$ to one of order $n-1$.) For example, $B_i^\sharp$ could be PETER and $B_j^\sharp$ IS MARRIED TO so that expanding $B_j^\sharp$ by $B_i^\sharp$ gives the belief fragment PETER IS MARRIED TO (or IS MARRIED TO PETER) (thus a fragment corresponding to a monadic property), while another expansion by $B_i^\sharp$ gives the belief (corresponding to a state of affairs) PETER IS MARRIED TO PETER). In another possible interpretation $B_i^\sharp$ could be BIGGER THAN and $B_j^\sharp$ IS THE INVERSE OF, so that $B_i^\sharp$ would not correspond to an individual.

We can generalize this by introducing the notion of *n-copy-saturability.* We say that some belief fragment $B_i^\sharp$ is n copy-saturable ($n$cs) by some belief fragment $B_j^\sharp$ iff there is some belief B_k such that $(B_j^\sharp \ldots \boxplus (B_j^\sharp \boxplus (B_j^\sharp \boxplus B_i^\sharp))) = B_k$, where $\boxplus$ occurs n times on the left-hand-side of the equation.

Thus the notion of n copy-saturability, which says that n expansions by some other belief fragment make the expanded belief fragment into a belief allows us to group belief fragments into classes. The class of dyadic properties will correspond to the class of fragments which are 2cs by some $B_m^\sharp$, the class of triadic properties to the class of fragments which is 3cs by some $B_k^\sharp$ and in general the class of i-adic properties will correspond to those fragments which are $i$cs by some $B_l^\sharp$.

While the notion of copy-saturability does the trick for all adicities greater that two, it fails for the monadic case. In this case the asymmetry between the expanded and the expander will break down since $B_i^\sharp$ will be 1cs by $B_j^\sharp$ and also $B_j^\sharp$ will be 1cs by $B_i^\sharp$. So the class of fragments which are 1cs by some $B_i^\sharp$ will *not* correspond to the class of monadic properties since it will e.g. also contain individuals.

The above procedure also fails to give us a way of distinguishing between different orders of properties of one adicity. If some $B_i^\sharp$ is 2cs by some $B_j^\sharp$ we can be sure that $B_i^\sharp$ is dyadic, but we cannot know what the order of either $B_i^\sharp$ or of $B_j^\sharp$ is, since it could be the case that $B_i^\sharp$ corresponds to a first-order dyadic property (which we will in the future abbreviate as P_2^1) and $B_j^\sharp$ an individual (abbreviated as i), or that $B_i^\sharp = P_2^2$ and $B_j^\sharp = P_n^1$ or $B_i^\sharp = P_2^3$ and $B_j^\sharp = P_m^2$ and so on.

So we need some way of distinguishing first-order properties from individuals and also a method for distinguishing orders of properties. Once we have a notion of an individual, the definition of orders of properties is easy.

Before proceeding to this discussion, however, let us discuss a couple of properties of belief fragments which can be characterized in this framework. The above introduction of the selection function γ gives us a quite natural way of characterizing symmetry.

We can say that a belief fragment $B_i^\sharp$ corresponding to a dyadic property corresponds to a *symmetric* property iff for any two distinct fragments $B_j^\sharp$, $B_k^\sharp$ such that repeatedly expanding $B_i^\sharp$ by them gives some belief B, what B is does not in any way depend on γ.

This corresponds to the idea that what we mean by saying that some property (such as 'married') is symmetric is that 'Romeo married Juliet', 'Juliet married Romeo', 'Juliet and Romeo are married', 'Romeo and Juliet are married', 'A marriage took place between Romeo and Juliet' ... are all the same belief. So iff some fragment corresponds to a symmetric property it makes no difference which of the different possibilities (i.e. members of Γ) γ picks, since these possibilities really just all amount to the same.

We will say that a fragment $B_i^\sharp$ corresponding to a dyadic property corresponds to a *reflexive* property iff for some fragments $B_j^\sharp$, $B_k^\sharp$ and some set of beliefs $\mathfrak{B}$ $((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_k^\sharp) \cup \mathfrak{B} = ((((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_k^\sharp) \boxplus ((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_j^\sharp)) \boxplus ((B_i^\sharp \boxplus B_k^\sharp) \boxplus B_k^\sharp)) \cup \mathfrak{B}$. This characterization depends on the fact that $\mathfrak{B}$ is logically closed so that given we expand it by some belief aPb, when P is reflexive we get the very same result if we afterwards *also* expand by aPa and bPb.

Transitivity is a bit more complicated to characterize. Here we have to make our characterization relative to the structure of γ. Suppose γ selects the expansion which fills in the first place (i.e. iff the dyadic P is expanded by some a, γ selects aP rather than Pa from the set Γ of possibilities.). We can then say that a fragment $B_i^\sharp$ corresponding to a dyadic property corresponds to a transitive property iff for some fragments $B_j^\sharp$, $B_k^\sharp$ and $B_l^\sharp$ and some set of beliefs $\mathfrak{B}$ $(((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_k^\sharp) \boxplus ((B_i^\sharp \boxplus B_k^\sharp) \boxplus B_l^\sharp)) \boxplus \mathfrak{B} = (((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_k^\sharp) \boxplus ((B_i^\sharp \boxplus B_k^\sharp) \boxplus B_l^\sharp) \boxplus ((B_i^\sharp \boxplus B_j^\sharp) \boxplus B_l^\sharp)) \boxplus \mathfrak{B}$. Again this characterization depends on the fact that $\mathfrak{B}$ is logically closed so that given we expand it by some beliefs aPb and bPc when P is transitive we get the very same result if we afterwards *also* expand by aPc.

2.2 Order

To give an account of the different orders of properties different belief fragments correspond to we start from the class of fragments which are 1cs by some $B_i^\sharp$, i.e. the class $\mathcal{G}$ containing fragments corresponding to i, P_1^1, P_1^2 ....

We then split $\mathcal{G}$ into partitions which fulfill the following properties:

— For no two members from any G_i from $\mathcal{G}$ will the expansion of the one by the other yield a belief.

— If for some $B^\sharp_i$ in some G_j from $\mathcal{G}$ and some some $B^\sharp_k$ in some distinct G_l from $\mathcal{G}$ it is the case that $B^\sharp_i \boxplus B^\sharp_k$ yields a belief this is also true for all other members of G_j and G_l. (Call G_j and G_l *corresponding classes* in this case). For every G_i from $\mathcal{G}$ there is exactly one corresponding class.

Such a partitioning ensures that every order of properties is contained wholly and purely in some partition of $\mathcal{G}$. We just don't know which partition contains which.

We will now proceed by considering the following construction: Take some $B^\sharp_i \in \mathcal{G}$ and select some distinct $B^\sharp_j \in \mathcal{G}$ such that $B^\sharp_i \boxplus B^\sharp_j$ yields a belief. Now take $B^\sharp_j$ and select another $B^\sharp_k$ (if possible from a partition different from the one of which $B^\sharp_i$ is a member) such that $B^\sharp_j \boxplus B^\sharp_k$ yields a belief. Then consider $B^\sharp_k$ and so on. We call such a sequence an *expansion sequence* which can be written in the following way:

$$
\begin{array}{llllllll}
B^\sharp_i & \overset{\boxplus}{\hookleftarrow} & B^\sharp_j & = B_m & & & & \\
 & & | & & & & & \\
 & & B^\sharp_j & \overset{\boxplus}{\hookleftarrow} & B^\sharp_k & = B_n & & \\
 & & & & | & & & \\
 & & & & B^\sharp_k & \overset{\boxplus}{\hookleftarrow} & B^\sharp_l & = B_o \\
 & & & & & \vdots & &
\end{array}
$$

(Here the arrow indicates that the fragment on the left of the arrow is expanded by the fragment on the right in the respective step and yields the belief to the right of the equation sign.)

If we take a concrete example and let $B^\sharp_i$ be P^2_1 the expansion sequence looks like this:

$$
\begin{array}{llllllll}
P^2_1 & \overset{\boxplus}{\hookleftarrow} & P^1_1 & = B_m & & & & \\
 & & | & & & & & \\
 & & P^1_1 & \overset{\boxplus}{\hookleftarrow} & i & = B_n & & \\
 & & & & | & & & \\
 & & & & i & \overset{\boxplus}{\hookleftarrow} & P^1_1 & = B_o \\
 & & & & & \vdots & &
\end{array}
$$

Remember that what we want to do is finding the partition containing those and only those beliefs corresponding to individuals. Since the expansion sequences do not end, we cannot in any way identify i as the expansion in the last step of such a sequence. But if we look at the above example we see that the expansion which corresponds to i is *the only expansion both immediately preceded and immediately followed by expansions which come from the same partition.* In the third step in this sequence it wasn't possible to choose the fragment which expands from a *new* partition (as we demanded above, where $B_i^\sharp$ and $B_k^\sharp$ were supposed to come from different partitions). In fact we had to select the fragment which expands in step three (namely P_1^1) from *the very same partition* from which we selected the fragment which expanded in step one.

We can thus define the partition of belief fragments corresponding to individuals in the following way:

> it is the class of all those fragments $B_n^\sharp$ which are used in expansion sequences such that $B_{n-1}^\sharp$ (the expansion immediately preceding it) and $B_{n+1}^\sharp$ (the expansion immediately following it) are elements of the same $G_k \in \mathcal{G}$ if there is such a $B_n^\sharp$
>
> or, if there is none in some expansion sequence it is the first member of that sequence (in this case the belief to be expanded corresponds to a first-order property).

Thus given that we know which partition contains the belief fragments corresponding to individuals, it is easy to give a recursive way of telling which partitions contain fragments corresponding to properties of which order. Suppose $B_n^\sharp$ corresponds to an individual. Then some $G_k \in \mathcal{G}$ contains all and only the first-order properties iff for every $B_m^\sharp \in G_k$ there is some belief B_1 such that $B_n^\sharp \boxplus B_m^\sharp = B_1$. Some $G_l \in \mathcal{G}$ contains all and only the second-order properties iff for every $B_o^\sharp \in G_l$ there is some belief B_2 such that $B_m^\sharp \boxplus B_o^\sharp = B_2$ and so on. It is then obvious how to define fragments corresponding to higher-order properties of adicities > 1.

3 Philosophical Reflections

It should be noted that although the above characterization relies on differences in the possibilities of putting together fragments to form beliefs this does not mean that we have to assume that some class of fragments is somehow 'complete' while the other is incomplete and cannot stand on its own. This line is taken by Strawson who argues that

> A subject-expression is one which, in a sense, presents a fact in its own right and is to that extent complete. A predicate expression is one which in no sense presents a fact in its own right

> and is to that extent incomplete. [...] The predicate-expression, on the new criterion is one that can be completed only by explicit coupling with another. [...] We find an additional depth in Frege's metaphor of saturated and the unsaturated constituents. (5), 187–188

On our account none of the fragments is able to represent a state of affairs (not even in 'a certain sense') — this is something which only beliefs can do. And although we consider belief fragments to be incomplete (to the extent to which they are incomplete *beliefs*) this does not mean that they cannot represent as they are; as we have seen above we can make sense of some of them representing individuals while others represent properties of different kinds.

The ontological categories we considered above — states of affairs, individuals, properties of arbitrary orders and adicities — are certainly not all the ontological categories there are. Just think of categories as abstract object, event, mathematical structure, trope or material object. All of these are categories which are as legitimate objects of ontological discussion as the ones we considered. Nevertheless, they seem to be distinguished by their *universality* as well as by their *coherence*.

They are universal because relative to every other ontological category there seem to be states of affairs, individuals and properties. The set containing the number seven (an abstract object) is an *individual*, amongst others it has the *property* of having just one member and it is a *state of affairs* that it has this property. The Seven Years' War (an event) is an individual, it has the property of having lasted for seven years and it is a state of affairs that it has lasted for seven years, and so on.

These ontological categories have a particular coherence because they can all be constructed from a rather confined basis. It is possible to account for all of them just in terms of states of affairs and individuals. This is done by employing a procedure due to Ajduckiewizc (1) (which was later developed in (2)) for giving a theory of grammatical categories. We use a primitive functor (intuitively interpreted as 'takes __ and returns ...') and write the complex ontological categories as fractions. Monadic first order properties are defined as $\frac{i}{s}$ (because they take an individual and return a state of affairs), while dyadic first order properties are $\frac{i,i}{s}$ and so forth. Monadic second order properties are taken to be of the form $\frac{\frac{i}{s}}{s}$; and in a similar way all the ontological categories of the different adicities and orders can be constructed. This constructional coherence does not seem to be possessed by other sets of ontological categories. The ensemble of states of affairs, individuals and properties of different adicities and orders which can be defined in a belief-dynamical framework therefore seems to occupy a special place in the system of ontological categories.*

References

[1] Kasiemirz Ajduckiewicz. *'The Scientific World-Perspective and other Essays. 1931–1963.* Reidel, PWN Polish Publishers, 1978.

[2] Yehoshua Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language,* 29:47–8, 1953. MIT Press, Cambridge MA, 1988.

[3] Peter Gärdenfors. *Knowledge in Flux.* MIT Press, Cambridge MA, 1988.

[4] Sven Ove Hansson. *A Textbook of Belief Dynamics. Theory Change and Database Updating.* Kluwer, Dordrecht, Boston, London, 1999.

[5] P. F. Strawson. *Individuals. An Essay in Descriptive Metaphysics.* Methuen & Co., London, 1964.

*Thanks are due to Matthias Hild for providing the initial motivation for this paper.

Grammar conversion from LTAG to HPSG

NAOKI YOSHINAGA
Department of Information Science, Graduate school of Science, University of Tokyo
yoshinag@is.s.u-tokyo.ac.jp

YUSUKE MIYAO
Department of Information Science, Graduate school of Science, University of Tokyo
yusuke@is.s.u-tokyo.ac.jp

ABSTRACT. We propose a grammar conversion algorithm from an arbitrary Feature-Based Lexicalized Tree Adjoining Grammar (FB-LTAG) grammar into a strongly equivalent Head-Driven Phrase Structure Grammar (HPSG)-style grammar. Our algorithm converts LTAG elementary trees into HPSG feature structures by encoding a tree structure in a list. A set of pre-determined rules manipulate the list to emulate substitution and adjunction. By using our algorithm, we can obtain HPSG-style grammars from existing LTAG grammars. We apply this algorithm to the XTAG English grammar and report some findings.

1 Introduction

Recent intelligent NLP applications (Kay et al. 1994) take advantages of computationally and linguistically motivated grammar formalisms, such as Feature-Based Lexicalized Tree Adjoining Grammar (FB-LTAG[1]) (Vijay-Shanker 1987; Vijay-Shanker and Joshi 1988) and Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1994). Although both formalisms have the same motivation to model natural languages, the correspondence between the formalisms is not well discussed. We believe that collaboration between the communities will be beneficial for their further improvement.

This research aims at providing a basis for the collaboration, and the proposal is a grammar conversion from LTAG into HPSG. Our conversion guarantees the strong equivalence, that is, parsing results (derivation trees) of an LTAG grammar can be derived from parsing results of the obtained HPSG-style grammar. Strongly equivalent grammars based on the different

[1] If not confusing, we use the term LTAG to refer to FB-LTAG in this paper.

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

formalisms are valuable for both communities in practical, computational, and theoretical aspects (Becker and Lopez 2000):

Resource sharing HPSG-based applications can make use of LTAG resources (lexicons and grammars) such as large-scale English (Doran et al. 2000) and French grammars (Abeillé and Candito 2000). Our algorithm can reduce the considerable workload to develop huge resources from scratch.

Parsing efficiency We can apply HPSG parsing techniques to LTAG parsing by applying them to the obtained HPSG-style grammar. Recent development of the HPSG parsing techniques allows the use of HPSG-based processing in practical application contexts (Flickinger et al. 2000). Experiments using state-of-the-art parsers show the efficiency of an HPSG parser compared to an LTAG parser.

Linguistic correspondence We can explore the linguistic correspondence between the formalisms. Since the obtained grammar has computational features of HPSG and linguistic features of LTAG, their difference will be apparent by comparing the obtained grammar with existing HPSG grammars.

We implemented the conversion algorithm, and successfully converted the latest version of the XTAG English grammar (The XTAG Research Group 2001), which is a large-scale FB-LTAG grammar. The strong equivalence was empirically attested by the fact that parsing with the original and the obtained grammars generated equivalent parse results. Strongly equivalent grammars enable the fair comparison of LTAG and HPSG parsers, and another work reports that an efficient HPSG parser achieved a parsing speed that was significantly higher than an existing LTAG parser (Yoshinaga et al. 2001). In this paper, we investigated the types of linguistic phenomena covered by the XTAG English grammar, and correspondence to their analysis in the HPSG formalism.

Tateisi et al. also translated LTAG into HPSG (Tateisi et al. 1998). However, their method depended on translator's intuitive analysis of the original grammar, and thus the translation was manual and grammar dependent. The manual translation demanded considerable efforts from the translator, and obscures the strong equivalence between the original and obtained grammars. Other works converted HPSG into LTAG (Kasper et al. 1995; Becker and Lopez 2000). Given the greater generative power of HPSG, the conversion required some restrictions on HPSG to suppress its generative capacity. As a result, the conversion loses the strong equivalence of the grammars. The existing works cannot gain the above advantages, which are attainable only when the strong equivalence is preserved.

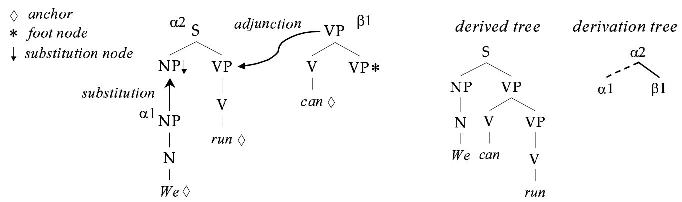


Figure 27.1: Tree Adjoining Grammar: basic structures and their composing operations

2 Grammar formalisms

2.1 Feature-Based Lexicalized Tree Adjoining Grammar (FB-LTAG)

LTAG (Schabes et al. 1988), an input of our algorithm, is a grammar formalism that provides syntactic analyses for a sentence by composing *elementary trees* with two operations called *substitution* and *adjunction*. Elementary trees are classified into two types, *initial trees* ($\alpha 1$ and $\alpha 2$ in Figure 27.1) and *auxiliary trees* ($\beta 1$ in Figure 27.1). An elementary tree has at least one leaf node labeled with a terminal symbol called an *anchor* (marked with $\diamond$). In an auxiliary tree, one leaf node is labeled with the same symbol as the root node and is specially marked as a *foot node* (marked with $*$). In an elementary tree, leaf nodes with the exception of anchors and a foot node are called *substitution nodes* (marked with $\downarrow$).

The left-hand side of Figure 27.1 illustrates the two operations. *Substitution* replaces a leaf node (substitution node) with an initial tree and *adjunction* grafts an auxiliary tree with the root node and a foot node labeled x onto a node with the same symbol x. Results of analysis are described not only by *derived trees* (i.e., parse trees) but also by *derivation trees* (Figure 27.1). A derivation tree represents the history of combinations of trees and is a structural description in LTAG.

FB-LTAG (Vijay-Shanker 1987; Vijay-Shanker and Joshi 1988) is an extension of the LTAG formalism. In FB-LTAG, each node in elementary trees has a feature structure, containing grammatical constraints on the node.

2.2 Head-Driven Phrase Structure Grammar (HPSG)

We define HPSG, an output of the algorithm, following the computational specifications in (Pollard and Sag 1994). It consists of *lexical entries* and

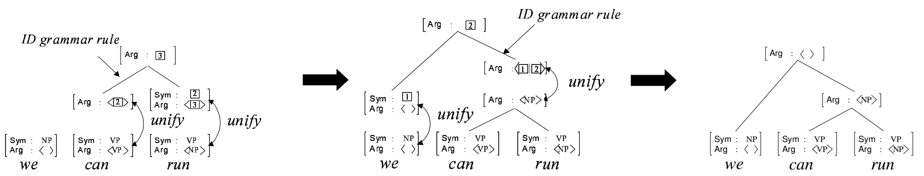


Figure 27.2: Parsing with an HPSG grammar

ID grammar rules, each of which is described with typed feature structures (Carpenter 1992). A lexical entry for each word expresses characteristics of the word, such as subcategorization frame and grammatical category. An ID grammar rule represents a grammatical relation between a mother and its daughters, and is independent of lexical characteristics.

Figure 27.2 illustrates an example of bottom-up parsing with an HPSG grammar. As we can see in this example, in HPSG a parse tree is generated by incrementally applying ID grammar rules to lexical entries and constructing each branching one by one, while in LTAG it is done by composing elementary trees with the two operations. Thus, the key points of the conversion are 1) how to encode a tree structure of an elementary tree as an HPSG feature structure, and 2) how to emulate substitution and adjunction. We should notice that there is not one-to-one correspondence between an elementary tree and an HPSG lexical entry because one elementary tree can have multiple anchors to represent compound expressions.

3 Algorithm

This section describes an algorithm for grammar conversion in detail. As described in Section 2, tree structures of LTAG elementary trees should be encoded in HPSG lexical entries, and the tree composing operations should be emulated by ID grammar rules. Thus, we propose a conversion algorithm which consists of: 1) conversion of elementary trees into HPSG lexical entries and 2) emulation of substitution and adjunction by pre-determined ID grammar rules.

In the following, we first state the notion of *canonical*, and show *canonical elementary trees*, which have a one-to-one correspondence to an HPSG lexical entry[2]. Canonical elementary trees are elementary trees which satisfy the conditions below:

[2]In this paper, we discuss a conversion of elementary trees which consist of binary branchings. A conversion of trees with unary branchings is straightforward and trees with n-ary ($n \geq 3$) branchings can be converted into trees with binary branchings.

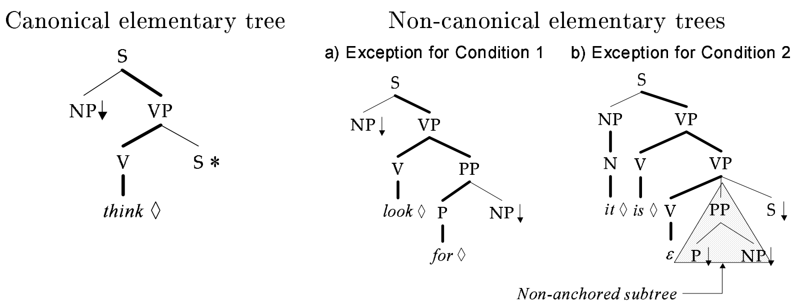


Figure 27.3: A canonical elementary tree and exceptions

Condition 1 A tree must have only one anchor.

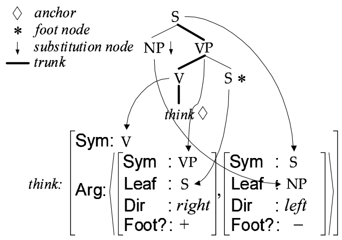
Condition 2 All branchings in a tree must contain trunk nodes.

Trunk nodes are nodes on a *trunk*, which is a path from an anchor to the root node (the thick line in Figure 27.3) (Kasper et al. 1995), with the exception of the anchors. Condition 1 guarantees that a canonical tree has only one trunk, and Condition 2 guarantees that each branching consists of a trunk node, a leaf node, and their mother (also a trunk node) (the left-hand side of Figure 27.3). The right-hand side of Figure 27.3 shows non-canonical ones. We call a subtree which is a depth 1 or above with no anchor a *non-anchored subtree*. Non-canonical elementary trees are first converted to canonical ones, and then applied the algorithm for canonical ones.

3.1 Conversion of canonical elementary trees

This section describes a conversion algorithm of an LTAG canonical elementary tree into an HPSG lexical entry. Canonical elementary trees can be directly converted to HPSG lexical entries as shown in the left-hand side of Figure 27.4.

The procedure `convert_tree_into_lexical_entry` in Figure 27.4 depicts an algorithm for converting an elementary tree T into an HPSG lexical entry L. In the algorithm, arg is a list of branchings b_i described with a quadruplet $\langle n_{i-1}, l_i, d_i, t_i \rangle$ along the trunk. The parameter n_{i-1} represents the mother node of the trunk node n_i. The parameters l_i, d_i and t_i represent a leaf node with depth i; they represent the non-terminal symbol, the direction (on which side of the trunk node n_i the leaf node is), and the type (whether a foot node or a substitution node) respectively. We call this list the *arguments* of the word. Finally the converted lexical entry L is described



Sym: symbol of a trunk node
Leaf: symbol of a leaf node
Dir: the direction of a leaf node against the trunk
Foot?: the type of a leaf node

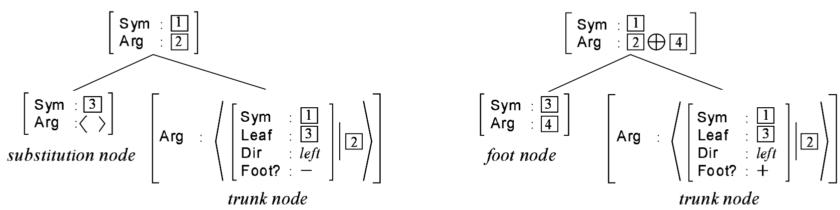
```
procedure convert_tree_into_lexical_entry(T)
begin
  arg := []
  for i := 1 to depth(T)−1
    n_{i−1} := trunk(T, i − 1)
    l_i := leaf(T, i)
    d_i := direct(T, i)
    t_i := type(T, i)
    b_i := ⟨n_{i−1}, l_i, d_i, t_i⟩
    arg := [b_i] ⊕ arg
  end for
  L := (n_{depth(T)−1} , arg)
  return L;
end

depth:   returns the depth of the anchor.
trunk:   returns the symbol of the trunk node.
leaf:    returns the symbol of the leaf node in depth i
direct:  returns which side of the trunk node
         the leaf node is in depth i
type:    returns the type of the leaf node in depth i
```

Figure 27.4: A conversion algorithm from a canonical elementary tree T into an HPSG lexical entry L

with the arguments arg and the mother of the anchor, namely, $n_{\mathtt{depth(T)}-1}$ where `depth(T)` is the depth of the tree T.

In the left-hand side of Figure 27.4, the value of the Sym feature is the symbol $n_{depth(T)-1}$, the value of the Arg feature contains the arguments in arg, as a list of feature structures with four features Sym, Leaf, Dir and Foot? corresponding to n_{i-1}, l_i, d_i and t_i respectively.



Left substitution rule *Left adjunction rule*

Figure 27.5: Grammar rules: substitution rule and adjuction rule

3.2 Definition of grammar rules

In this section, we give the definition of grammar rules to emulate substitution and adjunction (Figure 27.5)[3]. We should note that these grammar

[3]In the figure, we give rules whose right daughter is a trunk node. Of course, there are symmetric rules whose left daughter is a trunk node.

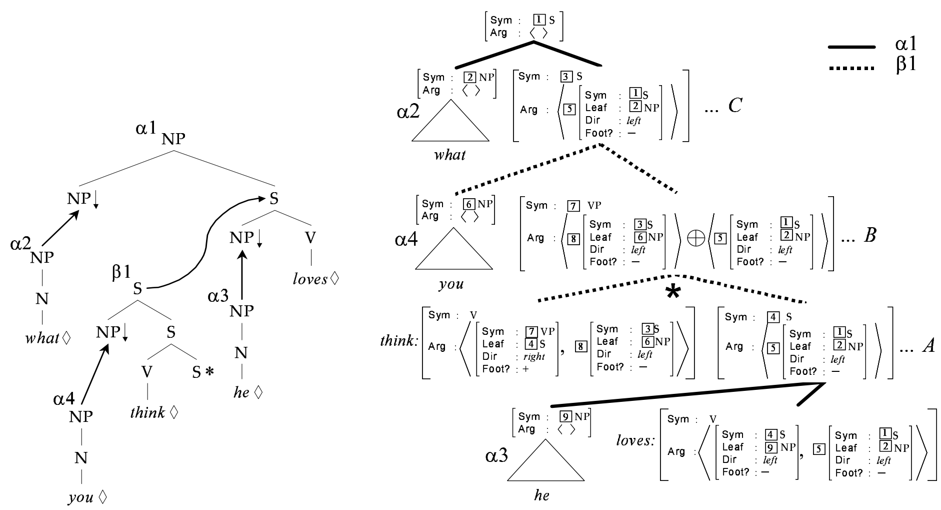


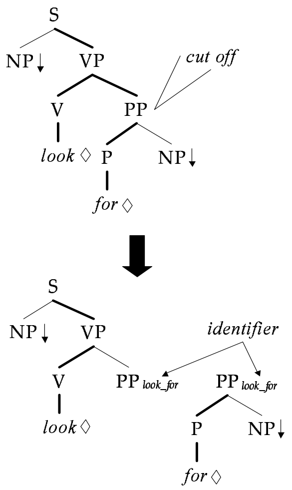
Figure 27.6: LTAG derivation and HPSG rule applications for a phrase "*what you think he loves*"

rules are independent of the original grammar because they do not specify any characteristics given in the original grammar.

Substitution rule: The Sym feature of the substitution node must have the value of the Leaf feature $\boxed{3}$ of the trunk node. The Arg feature of the substitution node must be a null list, because the substitution node must be unified only with the node corresponding to the root node of the initial tree. The substitution rule percolates the tail elements $\boxed{2}$ of the Arg feature of the trunk node to the mother in order to continue constructing the tree.

Adjunction rule: The Sym feature of a foot node must have the same value as the Leaf feature $\boxed{3}$. The value of the Arg feature of the mother node is a concatenation list of both Arg features $\boxed{2}$ and $\boxed{4}$ of its daughters because we first construct the tree corresponding to the adjoining tree and next continue constructing the tree corresponding to the adjoined tree. The value "+" or "−" of the Foot? feature explicitly determines whether the next rule application is the adjunction rule or the substitution rule.

Figure 27.6 shows an example of rule applications. The thick line indicates the adjoined tree ($\alpha 1$) and the dashed line indicates the adjoining tree ($\beta 1$). The adjunction rule is applied to construct the branching marked with $\star$, where "think" takes as an argument a node whose Sym feature's value is




```
procedure divide_tree_into_subtrees(MT)
begin
  A := select(MT)
  ⟨ST, 𝒯⟩ := divtree(MT, A)                          ··· (1)
  foreach T (𝒯)
    𝒮𝒮𝒯 := divide_tree_into_subtrees(𝒯)
    𝒮𝒯 := 𝒮𝒮𝒯 ∪ 𝒮𝒯
  end foreach
  𝒮𝒯 := 𝒮𝒯 ∪ {ST}
  return 𝒮𝒯;
end

procedure divtree(MT, A)
begin
  𝒯 := φ
  for i := 1 to depth(MT, A)-1
    if (nonleaf(arg(trunk(i))))
      ⟨MT′, T⟩ := cut(MT, arg(trunk(i)))          ··· (2)
      address(trunk(i), Address)
      mark(Address, MT′, T)                         ··· (3)
      𝒯 = 𝒯 ∪ T
      MT := MT′
    end if
  end for
  ST := MT
  return ⟨ST, 𝒯⟩
end

select:  returns one of anchors.
depth:   returns the depth of the anchor.
trunk:   returns the trunk node in depth i.
arg:     returns the sister of the trunk node.
cut:     cuts off the tree at the sister of the trunk node and
         returns a subtree whose root node is the sister node.
nonleaf:returns true if not a leaf node.
address:returns address in the elementary tree.
mark:    marks the address to each cut-off node.
```

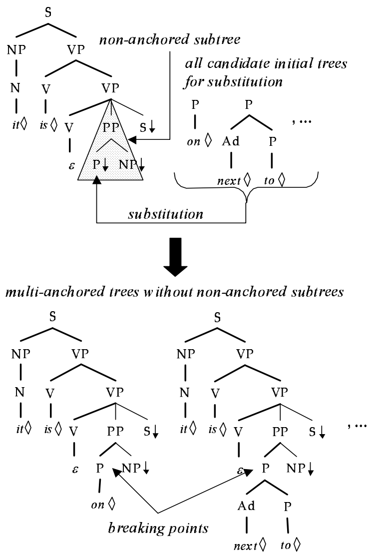
Figure 27.7: An algorithm for dividing a multi-anchored elementary tree MT into single-anchored trees $\mathcal{ST}$

S. By applying the adjunction rule, the Arg feature of the mother node (B) becomes a concatenation list of both Arg features of $\beta 1$ ($\boxed{8}$) and $\alpha 1$ ($\boxed{5}$). Note that when the construction of $\beta 1$ is completed, the Arg feature of the trunk node (C) will be its former state (A). We can continue constructing $\alpha 1$ as if nothing had happened.

3.3 Division of multi-anchored trees

Multi-anchored elementary trees, which violate Condition 1, are divided into multiple canonical elementary trees. We call the cutting nodes in the divided trees *cut-off nodes*. Note that a cut-off node is marked by an *identifier* to preserve a co-occurrence relation among the multiple anchors (Figure 27.7).

The procedure `divide_tree_into_subtrees` in Figure 27.7 depicts an algorithm for dividing a multi-anchored elementary tree MT into a set of single-anchored subtrees $\mathcal{ST}$. One anchor A is selected, and a single-anchored tree ST which can be constructed from the anchor A is picked up (`(1)` in Figure 27.7). We check the path from the root node to the anchor A, cut off




```
procedure expand_tree_into_anchored_tree(NT)
begin
  BR := na_br(NT)
  MT := {NT}
  foreach BR (BR)
    S := select(BR)                          ··· (1)
    IT := initial(S)                         ··· (2)
    MMT := φ
    foreach MT (MT)
      TMT := substitute(MT, S, IT)           ··· (3)
      mark(S)
      MMT := TMT ∩ MMT
    end foreach
    MT := MMT
  end foreach
  return MT
end

na_br:       returns the deepest branchings, whose
             daughters do not consist of an anchor.
initial:     returns all initial tree whose root node is the
             same as the leaf node.
select:      returns one of leaf nodes.
substitute:  causes substitution to the leaf node and
             returns a set of resulting multi-anchored trees.
mark:        marks the substituted node.
```

Figure 27.8: An algorithm for converting a non-anchored subtree NT into anchored trees $\mathcal{MT}$

the sister node[4] `arg(trunk(`i`))` if it is not a leaf node, and store the *address* of an elementary tree into the cut-off node as an identifier (`(2)` and `(3)` in Figure 27.7).

3.4 Substitution to non-anchored subtrees

Non-canonical elementary trees violating Condition 2 have non-anchored subtrees. A non-anchored subtree is converted into multi-anchored trees by substituting the deepest node with initial trees (Figure 27.8). Substituted nodes are marked as *breaking points* to remember that the nodes originate from the substitution nodes. In the resulting trees, all subtrees are anchored so that we can apply the above conversion algorithms.

The procedure `expand_tree_into_anchored_tree` in Figure 27.8 depicts an algorithm for combining a non-anchored subtree NT with initial trees into anchored trees $\mathcal{MT}$. For each branching consisting of substitution nodes or foot nodes, one substitution node S is selected (`(1)` in Figure 27.8). The function `substitute` causes a substitution to the node S with all initial trees which can substitute to S (`(3)` in Figure 27.8).

[4]We should mention that the path from the foot node to the root node (*spine*) in an auxiliary tree must not be cut because the spine represents the chain of head signs between the root node and the foot node, which are unified with the same internal node in the other elementary trees.

We should consider the termination of this algorithm. Since the function `initial` returns all initial trees whose root node is the same as the leaf node (`(2)` in Figure 27.8), the initial trees might include a tree violating Condition 2, to which this algorithm should also be applied. This seems to cause an infinite recursion of application of this algorithm, but it can be avoided by removing non-anchored subtrees from such an initial tree before applying this algorithm.

3.5 Strong equivalence

Our algorithm guarantees the strong equivalence of the original and obtained grammars. In the obtained grammar, the above grammar rules are applied only to feature structures corresponding to nodes which can substitute/be adjoined in the LTAG canonical elementary trees because encoded branchings in Arg specify the nodes to be subcategorized next. The strong equivalence holds also for conversion of non-canonical elementary trees. For trees violating Condition 1, we can distinguish the cut-off nodes from the substitution nodes owing to identifiers, which recover the co-occurrence relation in the original elementary trees between the divided trees. For trees violating Condition 2, we can identify substitution nodes in a combined tree because they are marked as breaking points, and we can consider the combined tree as two trees in the LTAG derivation.

By following a history of rule applications and mapping each of them to substitution or adjunction, we can recover an LTAG derivation tree from an HPSG parse tree. First, we find the syntactic head of a parse tree by following a trunk node when the substitution rule was applied, or a foot node when the adjunction rule was applied. We then follow the path from the syntactic head to the root node. If we find an application of the substitution rule, we can readily map it to LTAG substitution. If we find an application of the adjunction rule, the node takes adjunction. We remember the length of the Arg feature of the node to identify the adjoining tree. If the length of the Arg feature of a trunk node is longer than the one we remembered, we are in the adjoining tree. When the length becomes equal to the one we remembered, the construction of the adjoining tree reaches to its root node and the construction of the adjoined tree restarts. Thus we can bring down the adjoining tree. Let us consider the case in Figure 27.6. First we find that "*love*" is a syntactic head of the parse tree. We then follow the path from the anchor to the root node of $\alpha 1$. Since we first find an application of the substitution rule, we can map it to the substitution of $\alpha 3$ to $\alpha 1$. Then, the next rule is the adjunction rule (marked with $\star$), and we can find the node A takes adjunction. We thus remember the length of the Arg feature of the node A, and follow the trunk until the length of the Arg feature is equal to that of the node A. At the node C, the length of the Arg feature is equal to the node A. This implies that the construction of the adjoining tree $\beta 1$ is completed at the node C.

3.6 Extension to FB-LTAG

The above algorithm gives the conversion of LTAG, and it can be easily extended to handle an FB-LTAG grammar by merely storing a feature structure of each node into the Sym feature and Leaf feature together with the non-terminal symbol. ID grammar rules execute feature structure unification done in LTAG substitution and adjunction.

3.7 Remaining issue

The above algorithm gives a formal link between HPSG and LTAG, but the linguistic aspects are not well discussed. In this section, we discuss the linguistic aspects according to the notion of the syntactic head, which is a central notion in HPSG.

Distinction between predicative/modifying auxiliary trees We should mention that there are two kinds of auxiliary trees in LTAG (Kroch 1989; Schabes and Shieber 1994). One is a predicative auxiliary tree, and the other is a modifier auxiliary tree. The former introduces a predicate that subcategorizes for a phrase of the category of its foot node and assigns a thematic role to the phrase, while the latter introduces a modifying, complement or dislocated phrase. This distinction between the two kinds of auxiliary trees is given roughly by determining which daughter is head, a foot node or a trunk node. Tateisi et al. (Tateisi et al. 1998) distinguished these trees by manually analyzing feature percolation in auxiliary trees and by assigning HPSG rule schemata separately to them. In this paper, we consider that all trunk nodes are head, but it can be manually or semi-automatically determined by giving some linguistic cues or by analyzing feature percolation.

Head selection It is not entirely clear how we should `select` the anchor A in Figure 27.7. Since most multi-anchored trees represent compound expressions or idioms, such as "*in front of*" and "*kick the bucket*", this problem can be replaced with another problem that which word of a phrase should be syntactic/semantic head. In the current implementation, we simply select an anchor from the left most one, though we can select one by using the idea of projection or some other linguistic approaches. There is room for argument on how we should `select` the leaf node S to be substituted in Figure 27.8. Since elementary trees with non-anchored subtrees represent constructions requiring a specification beyond immediate dominance, such as *it-clefts* and *equative be*, this problem can be rephrased as the problem

Table 27.1: The classification of elementary trees in the XTAG English grammar (LTAG) and converted lexical entries corresponding to them (HPSG): $\mathcal{A}$: canonical elementary trees, $\mathcal{B}$: elementary trees violating only Condition 1, $\mathcal{C}$: elementary trees violating only Condition 2, $\mathcal{D}$: elementary trees violating both conditions

Grammar	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	**Total**
LTAG	326	764	54	50	1,194
HPSG	326	1,992	1,083	2,474	5,875

that which leaf node takes the most important syntactic role in the constructions and should be expanded for HPSG analysis. We can solve this problem by using the same idea as we have mentioned above, but for now we simply select a leaf node from the left most one.

4 Experiments

The algorithm was applied to the latest version of the XTAG English grammar (The XTAG Research Group 2001)[5], which is a large-scale FB-LTAG grammar for English. We successfully converted all the elementary trees[6] in the XTAG English grammar to HPSG lexical entries. Table 27.1 shows the classifications of elementary trees of the XTAG English grammar, according to the conditions we introduced, and also shows the number of corresponding HPSG lexical entries.

We acquired exactly the same number of the derivation trees by using the original and the obtained grammar in the parsing experiment with 457 sentences from the ATIS corpus (Marcus et al. 1994)[7] (the average length is 6.32 words). This result empirically attested the strong equivalence of our algorithm. The experimental result also shows that an efficient HPSG parser (Torisawa et al. 2000) achieves a drastic speed-up against the LTAG parser (Sarkar 2000). The speed-up is because of not only the careful implementation but also the difference of the parsing scheme in the HPSG and LTAG parsing (Yoshinaga et al. 2001).

[5]We used the grammar attached to the latest distribution of an LTAG parser which we used for the parsing experiment. The parser is available at: ftp://ftp.cis.upenn.edu/pub/xtag/lem/lem-0.13.0.i686.tgz

[6]*Elementary trees* should be in fact denoted as *elementary tree templates*. That is, elementary trees are abstracted from lexicalized trees, and one elementary tree template is defined for one syntactic construction, which is assigned to a number of words.

[7]We eliminated 59 sentences because of a time-out of the parsers, and 61 sentences because the LTAG parser does not produce correct derivation trees because of bugs in its preprocessor.

Table 27.2: The classification of the elementary trees violating the conditions in Table 27.1: multi-anchored ones (corresponding to $\mathcal{B}$) (left), and ones with non-anchored subtrees (corresponding to $\mathcal{C} \cup \mathcal{D}$) (right)

Construction	**# of trees**
Compound expressions	414
Verb with PP	194
Idioms	140
Others	16
Total	**764**

Construction	**# of trees**
Verb with PP	85
It-cleft	12
Others	7
Total	**104**

The left-hand side of Table 27.2 shows how multi-anchored elementary trees are used in the XTAG English grammar. The table shows they are mainly used for compound expressions or idioms. Although such expressions seem to be difficult to explain in the HPSG formalism, the obtained grammar can handle them with a proper size of lexical entries by dividing them into multiple lexical entries. Another case for multi-anchored trees is *verb with PP*. The obtained grammar expresses this construction by cut-off nodes to require specified subtrees. On the other hand in linguistic specifications of HPSG, such a constraint is expressed by having the FORM feature, which requires the type of a phrase. This analysis seems to be consistent with the obtained grammar, i.e., the LTAG analysis.

The right-hand side of Table 27.2 shows the usage of elementary trees with non-anchored subtrees. These elementary trees express constructions requiring specifications beyond immediate dominance. As we can see in Table 27.1, these trees are expanded to quite large number of lexical entries. This result expects that these constructions might be difficult to handle in the HPSG formalism. The major case of such constructions is *verb with PP*, which causes a Wh-extraction of NP from PP. Since this construction is explained by using SLASH feature in HPSG, the difference between the formalisms appears here. Another case is *it-cleft*, which is not obvious to handle in HPSG, and it might be worth discussing.

5 Conclusion

We proposed a grammar conversion from an FB-LTAG grammar into an HPSG-style grammar. The grammar conversion guarantees the strong equivalence, and hence we can obtain parsing results of an LTAG grammar from those of the obtained HPSG-style grammar. Thus, our grammar conversion enables to share LTAG resources in the HPSG community, to apply HPSG parsing techniques to LTAG grammars, and to clarify the difference of linguistic analysis in both grammar formalisms. We implemented this algorithm, and successfully converted the latest version of the XTAG English grammar. Although we did not give a formal proof of strong equivalence, it will be possible owing to the formulation of our conversion algorithm.

Acknowledgment The authors wish to thank Mr. Anoop Sarkar for his help in using his parser in our experiment. The authors are also indebted to two anonymous reviewers for their valuable comments on this paper.

References

Abeillé, A. and M.-H. Candito (2000). Evolution of the XTAG system. In A. Abeillé and O. Rambow (Eds.), *Tree Adjoining Grammars: Formal, Computational and Linguistic Aspects*. CSLI publications.

Becker, T. and P. Lopez (2000). Adapting HPSG-to-TAG compilation to wide-coverage grammars. In *Proceedings of the 5th of International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+5)*, pp. 47–54.

Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press.

Doran, C., B. A. Hockey, A. Sarkar, B. Srinivas, and F. Xia (2000). FTAG: A Lexicalized Tree Adjoining Grammar for French. In A. Abeillé and O. Rambow (Eds.), *Tree Adjoining Grammars: Formal, Computational and Linguistic Aspects*. CSLI publications.

Flickinger, D., S. Oepen, and J. Tsujii (Eds.) (2000). *Natural Language Engineering – Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation*. Cambridge University Press.

Kasper, R., B. Kiefer, K. Netter, and K. Vijay-Shanker (1995). Compilation of HPSG to TAG. In *Proceedings of 33rd Meeting of the Association for Computational Linguistics (ACL '94)*, pp. 92–99.

Kay, M., J. Gawron, and P. Norvig (1994). *Verbmobil: A Translation System for Face-to-Face Dialog*. CSLI Publications.

Kroch, A. (1989). Asymmetries in long-distance extraction in a Tree-Adjoining Grammar. In M. Baltin and A. Kroch (Eds.), *Alternative Conceptions of Phrase Structure*. University of Chicago Press.

Marcus, M., B. Santorini, and M. A. Marcinkiewicz (1994). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics 19*(2), 313–330.

Pollard, C. and I. A. Sag (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications.

Sarkar, A. (2000). Practical experiments in parsing using Tree Adjoining Grammars. In *Proceedings of the 5th of International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+5)*, pp. 193–198.

Schabes, Y., A. Abeille, and A. K. Joshi (1988). Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. In *Proceedings of 12th International Conference on Computational Linguistics (COLING '92)*, pp. 578–583.

Schabes, Y. and S. M. Shieber (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics 20*(1), 91–124.

Tateisi, Y., K. Torisawa, Y. Miyao, and J. Tsujii (1998). Translating the XTAG English grammar to HPSG. In *Proceedings of the 4th of International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, pp. 172–175.

The XTAG Research Group (2001). A Lexicalized Tree Adjoining Grammar for English. http://www.cis.upenn.edu/~xtag/.

Torisawa, K., K. Nishida, Y. Miyao, and J. Tsujii (2000). An HPSG parser with CFG filtering. *Natural Language Engineering – Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation 6*(1), 63–80.

Vijay-Shanker, K. (1987). *A Study of Tree Adjoining Grammars*. Ph. D. thesis, Department of Computer & Information Science, University of Pennsylvania.

Vijay-Shanker, K. and A. K. Joshi (1988). Feature structures based Tree Adjoining Grammars. In *Proceedings of 12th International Conference on Computational Linguistics (COLING '92)*, pp. 714–719.

Yoshinaga, N., Y. Miyao, K. Torisawa, and J. Tsujii (2001). Efficient LTAG parsing using HPSG parsers. In *Proceedings of Pacific Association for Computational Linguistics (PACLING 2001)*. To appear.

Infinitary Expressibility of Necessity in Terms of Contingency

Evgeni Zolin
Department of Mathematical Logic
Faculty of Mathematics and Mechanics
Moscow State University, 119899 Moscow, Russia
zolin@lpcs.math.msu.ru

Abstract. This paper consists of two parts. In the first part we present an axiomatization of the "epistemic" modal logic **KD45** in the language with non-contingency operator as the sole modal primitive symbol. The second part, having a certain philosophical flavour, is devoted to the old question about the possibility of defining the necessity operator in terms of the contingency operator. Here we give a new, positive answer to this question by constructing an infinitary operator defined in terms of contingency which behaves like some necessity.

1 Introduction

The non-contingency operator $\rhd$ is defined in terms of the necessity operator $\Box$ by putting $\rhd A := \Box A \vee \Box\neg A$. A natural question arises here: is necessity expressible in terms of non-contingency? The answer depends upon the understanding of the notion of expressibility.

In general, a notion α is said to be *definable* (or *expressible*) in terms of a notion β if there exists an expression $A = A(\beta)$ containing β such that A is equal (or equivalent) to α. In our case this means that $\Box$ would be definable in terms of $\rhd$ if there exists a formula $\varphi(p)$ such that all occurrences of $\Box$ in φ are in contexts of the form $\rhd\psi$ and the equivalence $\Box p \leftrightarrow \varphi(p)$ is valid. From this point of view, $\Box$ is *not* definable in terms of $\rhd$ (cf. (3; 4)).

However, this understanding is rather confined, by the author's opinion, and more appropriate is to say that A *behaves* like α, or A subjects the same laws as α. This approach proves to be successful to give a new, positive answer to the above question. In this paper we construct an operator $\boxtimes$ (by giving its infinitary definition in terms of $\rhd$) which behaves like some necessity. To be more exact, we show that, for any normal modal logic L (describing the behaviour of $\Box$) the corresponding logic describing the behaviour of $\boxtimes$ is normal and, for some L, it even contains L (up to

Proceedings of the Sixth ESSLLI Student Session
Kristina Striegnitz (editor)

replacement of $\Box$ by $\boxtimes$). The operator $\boxtimes$ plays an important rôle in the proof of completeness theorem for non-contingency logic of **KD45**.

2 Preliminaries

The propositional modal language consists of a denumerable set of variables $\mathrm{Var} = \{p_0, p_1, \ldots\}$, symbols for falsehood $\bot$, implication $\to$, and a unary modal operator $\Box$. Other connectives are taken as standard abbreviations. The set of formulas of this language, $\mathbf{Fm}^{\Box}$, is defined as usual, in particular, if A is a formula then so is $\Box A$. This language will be referred to as a $\Box$-language and its formulas as $\Box$-formulas. A $\rhd$-language and the set $\mathbf{Fm}^{\rhd}$ of $\rhd$-formulas are defined similarly, just by replacing the symbol $\Box$ by $\rhd$. Fix a translation $\mathsf{tr}\colon \mathbf{Fm}^{\rhd} \to \mathbf{Fm}^{\Box}$ which respects boolean connectives and sets $\mathsf{tr}(\rhd A) := \Box\mathsf{tr}(A) \lor \Box\neg\mathsf{tr}(A)$.

A (Kripke) *frame* is a structure $\langle W, \uparrow\rangle$, where W is a nonempty set of "worlds" and $\uparrow$ is a binary "accessibility" relation on W. By $\downarrow$ we denote the converse relation of $\uparrow$. Quantification over worlds accessible from a given world $w \in W$ will be written as $\forall x{\downarrow}w$ and $\exists x{\downarrow}w$. A *model* $M = \langle F, \models\rangle$ consists of a frame F and a valuation $\models\, \subseteq (W \times \mathrm{Var})$. The notion "$A$ is true in M at w" (written $M{,}w \models A$ and M usually omitted) is defined for both $\Box$- and $\rhd$-formulas in the standard way; the modal clauses are as follows:

$$\begin{array}{lcl} w \models \Box A & \leftrightharpoons & \forall x{\downarrow}w \quad x \models A; \\ w \models \rhd A & \leftrightharpoons & \big(\forall x{\downarrow}w \quad x \models A\big) \text{ or } \big(\forall x{\downarrow}w \quad x \not\models A\big). \end{array}$$

Obviously, $w \models A \Leftrightarrow w \models \mathsf{tr}(A)$, for any $\rhd$-formula A. A formula A is *valid* in a frame F ($F \models A$, in symbols) if A is true at every world in every model based on F. If Γ is a set of formulas then a Γ-*frame* is a frame validating Γ. A logic L is called *complete* w.r.t. a class of frames $\mathcal{F}$ if, for any formula A, $L \vdash A \Leftrightarrow \mathcal{F} \models A$.

The *minimal normal* modal logic **K** has the following axioms and the inference rules of modus ponens, substitution, and necessitation:

$$\begin{array}{ll} (\mathsf{A}^{\Box}_{\top}) & \text{All classical tautologies in the } \Box\text{-language} \\ (\mathsf{A}^{\Box}_{\mathbf{K}}) & \Box(p \to q) \to (\Box p \to \Box q) \qquad \text{(distributivity)} \end{array}$$

$$(\mathsf{MP})\ \frac{A \quad A \to B}{B} \qquad (\mathsf{Sub})\ \frac{A}{A[B/p]} \qquad (\mathsf{Nec})\ \frac{A}{\Box A}$$

The systems $\mathbf{K\Sigma}$, $\mathbf{\Sigma} \subseteq \{\mathbf{D}, \mathbf{4}, \mathbf{5}\}$, are obtained by adding to **K** the axioms $(\mathsf{A}^{\Box}_{\mathfrak{S}})$, $\mathfrak{S} \in \mathbf{\Sigma}$, listed below (the class of frames characterized by an axiom $(\mathsf{A}^{\Box}_{\mathfrak{S}})$ is first-order definable by a formula also shown below).

$$\begin{array}{ll|lr} (\mathsf{A}^{\Box}_{\mathbf{D}}) & \Box p \to \Diamond p & \forall w\ \exists x \quad w{\uparrow}x & \text{(seriality)} \\ (\mathsf{A}^{\Box}_{\mathbf{4}}) & \Box p \to \Box\Box p & \forall w\ \forall x{\downarrow}w\ \forall y{\downarrow}x \quad w{\uparrow}y & \text{(transitivity)} \\ (\mathsf{A}^{\Box}_{\mathbf{5}}) & \Diamond p \to \Box\Diamond p & \forall w\ \forall x{\downarrow}w\ \forall y{\downarrow}w \quad x{\uparrow}y & \text{(euclideanness)} \end{array}$$

The logic **KD45** is known to capture the principles of reasoning involving epistemic judgments: the postulates of this logic are valid under the (informal) interpretation of a sentence of the form $\Box A$ as "A is known (to some idealized person)." In this context, the non-contingency assertion $\rhd A$ means "the truth value of A is known."

We shall consider only *normal* modal $\Box$-logics, i.e., sets of $\Box$-formulas containing the axioms of **K** and closed under the rules of **K**. Given a logic L, a *non-contingency logic of* L (a $\rhd$*-logic of* L, for short), denoted by $L^{\rhd}$, is the set of all $\rhd$-formulas whose translations are theorems of L:

$$L^{\rhd} = \{A \in \mathbf{Fm}^{\rhd} \mid \mathsf{tr}(A) \in L\} = \mathsf{tr}^{-1}(L).$$

Montgomery and Routley (5; 6; 7) axiomatized the non-contingency logics of **T**, **S4**, and **S5**. It is worth noting that if L contains **T**, or more specifically, the reflexivity scheme $\Box A \to A$, necessity is definable in terms of non-contingency ($\rhd$-definable, for short) by $\Box A = A \,\&\, \rhd A$. In the logic **Ver**, the same effect is observed: it proves, for any A, a formula $\Box A \leftrightarrow \top$, which can be viewed as a $\rhd$-definition of $\Box$. Cresswell (2) provides an example of logic **H** such that $\mathbf{H} \not\supseteq \mathbf{T}$, $\mathbf{H} \neq \mathbf{Ver}$, but $\Box$ is $\rhd$-definable in **H**.

A systematic study of non-contingency logic (in particular, the cases when $\Box$ is counted to be $\rhd$-*un*definable), was initiated by Humberstone. In his paper (3), a (rather complicated) system axiomatizing the non-contingency logic of **K** was presented. Kuhn (4) succeeded in simplifying this system and proposed a finite axiomatization of the non-contingency logic of **K4**.

3 Axiomatizations of non-contingency logics

Now we formulate our systems for $\rhd$-logics of the logics L described above. For notation simplicity, we denote the systems by $L^{\rhd}$; Theorem 3.1 below justifies the notation. The logic $\mathbf{K}^{\rhd}$ has the rules (MP) and (Sub) as well as the following axioms and the "noncontingentization" rule (cf. (3)):

$(\mathsf{A}^{\rhd}_{\top})$	All classical tautologies in the $\rhd$-language		(NCR) $\dfrac{A}{\rhd A}$
$(\mathsf{A}^{\rhd}_{\mathbf{K}})$	$\rhd(p \leftrightarrow q) \to (\rhd p \leftrightarrow \rhd q)$	(equivalence)	
$(\mathsf{A}^{\rhd}_{\neg})$	$\rhd p \leftrightarrow \rhd\neg p$	(mirror axiom)	
$(\mathsf{A}^{\rhd}_{\vee})$	$\rhd p \to \big[\rhd(q \to p) \vee \rhd(p \to r)\big]$	(dichotomy)	

To obtain the system $\mathbf{K\Sigma}^{\rhd}$, $\mathbf{\Sigma} \subseteq \{\mathbf{D}, \mathbf{4}, \mathbf{5}\}$, we add to $\mathbf{K}^{\rhd}$ the corresponding axioms (note that no axiom corresponds to seriality):

$(\mathsf{A}^{\rhd}_{\mathbf{4}})$	$\rhd p \to \rhd(q \to \rhd p)$	(weak transitivity)
$(\mathsf{A}^{\rhd}_{\mathbf{5}})$	$\neg\rhd p \to \rhd(q \to \neg\rhd p)$	(weak euclideanness)

The classes of frames characterized by these two axioms strictly contain the classes of transitive (resp. euclidean) frames as well as the class of functional frames (where each world "sees" at most one world); hence their names.

The systems for $\mathbf{K}^{\rhd}$ and $\mathbf{K4}^{\rhd}$ proposed by Kuhn (4) differ from ours: instead of the axiom $(\mathsf{A}^{\rhd}_{\mathbf{K}})$, his systems contain the rule of equivalent replacement $\frac{A \leftrightarrow B}{\rhd A \leftrightarrow \rhd B}$ as well as an additional axiom $\rhd p \,\&\, \rhd q \to \rhd(p \,\&\, q)$. Our systems are similar to the standard axiomatizations of normal logics.

The main result of this section is formulated in Theorem 3.1, stating that the systems $\mathbf{K\Sigma}^{\rhd}$ axiomatize the $\rhd$-logics of $\mathbf{K\Sigma}$, $\mathbf{\Sigma} \subseteq \{\mathbf{4}, \mathbf{5}\}$ (logics containing the seriality axiom will be considered at the end of the section).

Theorem 3.1 (Completeness). *For any $\mathbf{\Sigma} \subseteq \{\mathbf{4}, \mathbf{5}\}$ and a $\rhd$-formula A, the following statements are equivalent:*

(1) $\mathbf{K\Sigma}^{\rhd} \vdash A$;

(2) $\mathbf{K\Sigma} \vdash \mathsf{tr}(A)$;

(3) *A is valid in all $\mathbf{K\Sigma}$-frames.*

Proof. We follow the scheme (1) $\Rightarrow$ (2) $\Leftrightarrow$ (3) $\Rightarrow$ (1). The equivalence (2) $\Leftrightarrow$ (3) is well-known (cf. (1)) completeness of $\mathbf{K\Sigma}$ w.r.t. $\mathbf{K\Sigma}$-frames. In the rest of the proof we refer to $\rhd$-formulas as just formulas.

$\underline{(1) \Rightarrow (2)}$ The axioms of $\mathbf{K}^{\rhd}$ are valid in any frame, so the translations thereof are provable in $\mathbf{K}$. For the axiom $(\mathsf{A}^{\rhd}_{\mathbf{4}})$, the proof is in (4). We give a sketch of a derivation of (the translation of) the axiom $(\mathsf{A}^{\rhd}_{\mathbf{5}})$ in $\mathbf{K5}$:

$$\begin{tikzcd}[row sep=tiny] & \Diamond p \to \Box\Diamond p \arrow[dr] & \\ \mathbf{K5} \vdash \neg\rhd p \arrow[ur] \arrow[dr] & & \Box\neg\rhd p \to \Box(q \to \neg\rhd p) \to \rhd(q \to \neg\rhd p). \\ & \Diamond\neg p \to \Box\Diamond\neg p \arrow[ur] & \end{tikzcd}$$

$\underline{(3) \Rightarrow (1)}$ We construct the canonical model $M_{\mathfrak{L}} = \langle W_{\mathfrak{L}}, \uparrow, \models\rangle$ for the logic $\mathfrak{L} = \mathbf{K\Sigma}^{\rhd}$. Its worlds are maximal $\mathfrak{L}$-consistent sets of formulas. A valuation is defined in the usual way: $w \models p \Leftrightarrow p \in w$, for any world w and a variable p. Before defining the relation $\uparrow$, we introduce some notation.

For a formula A, denote $\boxtimes A := \{\rhd(B \to A) \mid B \in \mathbf{Fm}^{\rhd}\}$. In the subsequent proof, the symbol $\boxtimes$ plays the rôle similar to that of $\Box$ in the standard canonical model argument for $\Box$-logics. The difference is in their "types": the operator $\Box$ maps a formula to a formula, whereas $\boxtimes$ maps a formula to a set of formulas. Note that semantically $\boxtimes$ is *not* equivalent to $\Box$, i.e., the truth at a world w of the formula $\Box A$ is not equivalent to the truth at w of all formulas in the set $\boxtimes A$. The next section is devoted to investigation of interconnection between the operators $\Box$ and $\boxtimes$.

Now denote $\sharp w := \{A \in \mathbf{Fm}^{\rhd} \mid \boxtimes A \subseteq w\}$. Finally, put $w{\uparrow}x$ iff $\sharp w \subseteq x$.

Lemma 3.2. *For any world $w \in W_{\mathfrak{L}}$, the following properties are satisfied:*

1° **(Dichotomy)** *If $\rhd A \in w$ then either $A \in \sharp w$ or $\neg A \in \sharp w$.*

2° *The set $\sharp w$ is closed under (even empty) conjunction (hence $\sharp w \neq \varnothing$).*

3° *The set $\sharp w$ is closed under derivability in $\mathfrak{L}$: if $A \in \sharp w$ and $\mathfrak{L} \vdash A \to B$, then $B \in \sharp w$.*

4° *The dichotomy property is reversible: if $A \in \sharp w$ then $\rhd A \in w$.*

▶ **1°.** Suppose $A, \neg A \notin \sharp w$, then by definition of $\sharp w$, for some formulas B, C we have: $\neg\rhd(B \to A) \in w$, $\neg\rhd(C \to \neg A) \in w$. However, using the dichotomy axiom, we derive: $\mathbf{K}^{\rhd} \vdash \rhd A \to \big[\rhd(B \to A) \vee \rhd(C \to \neg A)\big]$, hence w is even $\mathbf{K}^{\rhd}$-inconsistent, which contradicts our assumptions.

2°. By definition, the empty conjunction is $\top$. Since $\mathbf{K}^{\rhd} \vdash \rhd(B \to \top)$, for any formula B, we have $\boxtimes\top \subseteq \mathbf{K}^{\rhd} \subseteq \mathcal{L} \subseteq w$ and so $\top \in \sharp w$.

Now let $A, B \in \sharp w$ and prove that $(A\&B) \in \sharp w$, i.e., $\rhd[C\to(A \,\&\, B)] \in w$, for any formula C. From $\boxtimes A \subseteq w$ and $\boxtimes B \subseteq w$ it follows that $\rhd(C\to A) \in w$ and $\rhd(C\to B) \in w$. Then we derive:

$$\begin{aligned}\mathbf{K}^{\rhd} \vdash \rhd(C \to A) \,\&\, \rhd(C \to B) &\longrightarrow \rhd[(C \to A) \,\&\, (C \to B)] \longleftrightarrow \\ &\longrightarrow \rhd[C \to (A \,\&\, B)].\end{aligned}$$

Since w is closed under conjunction and derivability in $\mathbf{K}^{\rhd}$ (and even in $\mathcal{L}$), we conclude: $\rhd[C \to (A \,\&\, B)] \in w$.

3°. To prove that $B \in \sharp w$, we take an arbitrary formula C and show that $\rhd(C \to B) \in w$. Since $\boxtimes A \subseteq w$, we have $\rhd[\neg(C \to B) \to A] \in w$. The assumption $\mathcal{L} \vdash A \to B$ truth-functionally implies $\mathcal{L} \vdash [\neg(C \to B) \to A] \leftrightarrow [C \to B]$, and this finally yields $\rhd(C \to B) \in w$.

4°. $\boxtimes A \subseteq w$ implies $\rhd(\top\to A) \in w$, which is equivalent to $\rhd A \in w$. ◀

Lemma 3.3 **($\models\; = \;\ni$).** *For any formula A and a world w, $w \models A \Leftrightarrow A \in w$.*

▶ By induction on A. Consider the only interesting case $A = \rhd B$.

$$\begin{array}{lll}(\Leftarrow) & \rhd B \in w & \Rightarrow \text{(by dichotomy } \mathbf{1}^\circ) \\ & B \in \sharp w \ \text{ or } \ \neg B \in \sharp w & \Rightarrow \text{(by definition of } \uparrow) \\ & (\forall x{\downarrow}w \ B \in x) \ \text{ or } \ (\forall x{\downarrow}w \ \neg B \in x) & \Rightarrow \text{(by consistency of } x) \\ & (\forall x{\downarrow}w \ B \in x) \ \text{ or } \ (\forall x{\downarrow}w \ B \notin x) & \Rightarrow \text{(by induction hypothesis)} \\ & (\forall x{\downarrow}w \ x \models B) \ \text{ or } \ (\forall x{\downarrow}w \ x \not\models B) & \Rightarrow w \models \rhd B.\end{array}$$

($\Rightarrow$) Suppose $\rhd B \notin w$. Then the sets $X = \sharp w \cup \{B\}$ and $Y = \sharp w \cup \{\neg B\}$ are $\mathcal{L}$-consistent. For, if Y is not then $\mathcal{L} \vdash (A_1 \,\&\, \ldots \,\&\, A_n) \to B$ for some formulas $A_1, \ldots, A_n \in \sharp w$ and $n \geqslant 0$. By **2°**, $(A_1 \,\&\, \ldots \,\&\, A_n) \in \sharp w$, then $B \in \sharp w$ by **3°** and $\rhd B \in w$ by **4°**, which is not the case. The argument for X is similar except for additional use of the mirror axiom.

Therefore, X and Y are contained in some worlds x and y. Since $\sharp w \subseteq x$ and $\sharp w \subseteq y$, we have $w{\uparrow}x$ and $w{\uparrow}y$; by induction hypothesis, $B \in x$ and $B \notin y$ imply $x \models B$ and $y \not\models B$, thus $w \not\models \rhd B$. ◀

By this lemma, the canonical model falsifies all the nontheorems of L. To conclude the proof, it remains to check that the canonical frame is a $\mathbf{K\Sigma}$-frame. The case $\mathbf{\Sigma} = \varnothing$ is trivial.

Suppose $\mathbf{4} \in \mathbf{\Sigma}$ and prove that $\uparrow$ is transitive. Let $w{\uparrow}x{\uparrow}y$ and show that $w{\uparrow}y$, i.e., $\sharp w \subseteq y$. Take any $A \in \sharp w$, then $\rhd(B \to A) \in w$, for every B. By the axiom $(\mathsf{A}^{\rhd}_{\mathbf{4}})$, $\mathbf{K4}^{\rhd} \vdash \rhd(B \to A) \to \rhd[C \to \rhd(B \to A)]$, for any C. Since w is closed under $\mathbf{K4}^{\rhd}$-derivability, $\rhd[C \to \rhd(B \to A)] \in w$.

Hence $\boxtimes\rhd(B \to A) \subseteq w$ and $\rhd(B \to A) \in \sharp w \subseteq x$, whence $\boxtimes A \subseteq x$ and $A \in \sharp x \subseteq y$, as desired.

Suppose $\mathbf{5} \in \mathbf{\Sigma}$ and prove that $\uparrow$ is euclidean. Let $w{\uparrow}x$, $w{\uparrow}y$ and show that $x{\uparrow}y$, i.e., $\sharp x \subseteq y$. Take any $A \notin y$, then $A \notin \sharp w$ by $\sharp w \subseteq y$, hence $\neg\rhd(B{\to}A) \in w$, for some B. Since w is closed under $\mathbf{K5}^{\rhd}$-derivability, we apply $(\mathsf{A}^{\rhd}_{\mathbf{5}})$ to obtain $\rhd[C \to \neg\rhd(B \to A)] \in w$, for all C, therefore $\boxtimes\neg\rhd(B \to A) \subseteq w$. By $w{\uparrow}x$, we conclude: $\neg\rhd(B \to A) \in x$, thus $\boxtimes A \not\subseteq x$ and $A \notin \sharp x$, hence the claim. $\dashv$

Now we show, following (3), that adding the axiom $(\mathsf{A}^{\Box}_{\mathbf{D}})$ to some $\Box$-logics does not change $\rhd$-logic thereof. Let $F = \langle W, \uparrow\rangle$ be a frame. We denote the set of worlds accessible from $w \in W$ by $w{\uparrow} := \{x \in W \mid w{\uparrow}x\}$. Turning "blind" worlds into worlds "seeing" only itself yields a frame $\widehat{F} := \langle W, \Uparrow\rangle$, where $\Uparrow := \uparrow \cup \{\langle w, w\rangle \mid w{\uparrow} = \varnothing\}$. For a class of frames $\mathfrak{F}$, put $\widehat{\mathfrak{F}} := \{\widehat{F} \mid F \in \mathfrak{F}\}$. In (3) it is noted that F and $\widehat{F}$ validate the same $\rhd$-formulas.

Theorem 3.4. *Suppose a $\Box$-logic L is complete w.r.t. a class $\mathfrak{F}$ and $L\mathbf{D}$ is the smallest logic containing L and $(\mathsf{A}^{\Box}_{\mathbf{D}})$. If $\widehat{\mathfrak{F}} \subseteq \mathfrak{F}$ then $L\mathbf{D}^{\rhd} = L^{\rhd}$.*

Proof. The inclusion '$\supseteq$' is trivial. Now take any $A \in L\mathbf{D}^{\rhd}$; clearly, $A \in L^{\rhd} \Leftrightarrow \mathsf{tr}(A) \in L \Leftrightarrow \mathfrak{F} \models \mathsf{tr}(A) \Leftrightarrow \mathfrak{F} \models A$, so it remains to show that $F \models A$, for any frame $F \in \mathfrak{F}$. Since $\widehat{\mathfrak{F}} \subseteq \mathfrak{F}$, we have $\widehat{F} \in \mathfrak{F}$ and so $\widehat{F} \models L$; besides, $\widehat{F}$ is serial, hence $\widehat{F} \models (\mathsf{A}^{\Box}_{\mathbf{D}})$. Thus $\widehat{F} \models L\mathbf{D}$, whence $\widehat{F} \models L\mathbf{D}^{\rhd}$, in particular, $\widehat{F} \models A$. By the above, this is equivalent to $F \models A$. $\dashv$

As a consequence, $\mathbf{KD\Sigma}^{\rhd} = \mathbf{K\Sigma}^{\rhd}$, for any $\mathbf{\Sigma} \subseteq \{\mathbf{4}, \mathbf{5}\}$, since the transitivity and euclideanness properties are preserved as we pass from F to $\widehat{F}$. For the case $\mathbf{\Sigma} = \varnothing$ the result was obtained in (3).

4 Infinitary operator

Roughly speaking, the following "infinitary operator" occurs in the proof of Theorem 3.1 (we replace a set of formulas by a conjunction thereof):

$$\boxtimes A = \bigwedge_{B \in \mathbf{Fm}^{\rhd}} \rhd(B \to A).$$

¿From this equality one can read off a natural Kripke semantics of the operator $\boxtimes$. The question arises immediately: What modal principles are valid for this operator? Surprisingly enough, the operator $\boxtimes$ subjects the laws of some normal modal logic (which, of course, depends on the normal logic describing the behaviour of the initial necessity operator $\Box$).

To put it in a more precise form, consider the infinitary $\rhd$-language containing the set of variables Var as above, negation $\neg$, infinitary conjunction $\bigwedge$ and a unary modal operator $\rhd$. The set of formulas, $\mathbf{Fm}^{\rhd}_{\infty}$, is defined by induction: every variable p_i is a formula; if A is a formula then so are $\neg A$ and

$\triangleright A$; if Φ is a finite or countable set of formulas then $\bigwedge\Phi$ is a formula. Other connectives can be introduced as usual, e.g., $(A \to B) \leftrightharpoons \neg\bigwedge\{A, \neg B\}$; therefore we can assume that $\mathbf{Fm}^{\triangleright} \subset \mathbf{Fm}^{\triangleright}_{\infty}$. Kripke semantics for this language is defined in an obvious way.

Further, we introduce a $\boxtimes$-language obtained from the $\Box$-language by replacing the symbol $\Box$ by $\boxtimes$. Finally, we define a translation $\mathsf{Tr}\colon \mathbf{Fm}^{\boxtimes} \to \mathbf{Fm}^{\triangleright}_{\infty}$ which respects boolean connectives and has the following inductive item:

$$\mathsf{Tr}(\boxtimes A) = \bigwedge_{B \in \mathbf{Fm}^{\triangleright}} \triangleright(B \to \mathsf{Tr}(A)).$$

This translation induces semantics for the $\boxtimes$-language: $F \models A \leftrightharpoons F \models \mathsf{Tr}(A)$, for any $\boxtimes$-formula A. One can even define semantics for "mixed" formulas containing $\Box$, $\triangleright$, and $\boxtimes$. Note that the implication $\Box A \to \boxtimes A$ is valid in any frame, whereas the converse one is not.

Now, given a $\Box$-logic L, we define a *$\boxtimes$-logic of L* as the set of all $\boxtimes$-formulas valid in any L-frame:

$$L^{\boxtimes} \leftrightharpoons \{A \in \mathbf{Fm}^{\boxtimes} \mid \text{ for any frame } F \ (F \models L \Rightarrow F \models A)\}.$$

It is easily seen, for example, that $\mathbf{Ver}^{\boxtimes} = \mathbf{Ver}$ (from now on, we understand such equalities as well as inclusions up to replacement of $\Box$ by $\boxtimes$).

Theorem 4.1. *If L is a normal $\Box$-logic then $L^{\boxtimes}$ is a normal $\boxtimes$-logic.*

Proof. Since $L^{\boxtimes}$ is clearly closed under the rules of $\mathbf{K}$, we only need to verify that $\boxtimes(p \to q) \to (\boxtimes p \to \boxtimes q)$ is valid. Assume the contrary, i.e., there exists a model M and its world w such that

$$w \models \boxtimes(p \to q), \qquad w \models \boxtimes p, \qquad w \not\models \boxtimes q.$$

The latter implies that $w \not\models \triangleright(A \to q)$ for some $\triangleright$-formula A, and so

$$\begin{array}{ll} \exists x{\downarrow}w & x \models A \to q, \\ \exists y{\downarrow}w & y \models A, \neg q. \end{array}$$

By our assumptions, $w \models \triangleright(p \to q), \triangleright p$. Hence $w \models \triangleright[(p \to q) \,\&\, p]$, or equivalently, $w \models \triangleright(p \,\&\, q)$. But the case $w \models \Box(p \,\&\, q)$ is impossible, for $y \not\models p \,\&\, q$, therefore we have $w \models \Box(\neg p \vee \neg q)$. Now consider two cases:

1) $x \models q$. Then $x \not\models p$, since $x \models (\neg p \vee \neg q)$ by the above. Using $w \models \triangleright p$, we conclude $y \not\models p$. This yields a contradiction: on the one hand, $w \models \triangleright(q \to p)$, since $w \models \boxtimes p$; on the other, $x \not\models q \to p$ and $y \models q \to p$.

2) $x \not\models q$. Then $x \not\models A$, for $x \models A \to q$. Since $w \models \triangleright p$, there are two subcases:

2a) $x \models p$ and $y \models p$. Then from $w \models \triangleright[A \to (p \to q)]$ it follows that:

- either $w \models \Box[A \to (p \to q)]$, which is not the case, for $y \models A, p, \neg q$;
- or $w \models \Box\neg[A \to (p \to q)]$, so $w \models \Box A$, in contradiction with $x \not\models A$.

2b) $x \not\models p$ and $y \not\models p$. Then from $w \models \rhd[A \to p]$ it follows that:
- either $w \models \Box[A \to p]$, which is not the case, for $y \models A, \neg p$;
- or $w \models \Box\neg[A \to p]$, hence $w \models \Box A$, in contradiction with $x \not\models A$. $\dashv$

This result implies that the infinitary operator $\boxtimes$ defined in terms of non-contingency behaves like *some*, possibly different from the initial, necessity.

Theorem 4.2. *For any* $\mathbf{\Sigma} \subseteq \{\mathbf{4}, \mathbf{5}\}$*, we have* $\mathbf{K\Sigma}^{\boxtimes} \supseteq \mathbf{K\Sigma}$*.*

PROOF. For $\mathbf{\Sigma} = \varnothing$ the statement follows from Theorem 4.1.

$\underline{\mathbf{4} \in \mathbf{\Sigma}.}$ We shall prove that $\boxtimes p \to \boxtimes\boxtimes p$ is valid on any transitive frame. Assume that for a world w of some transitive model we have $w \models \boxtimes p$ and $w \not\models \boxtimes\boxtimes p$. This means that $w \not\models \rhd(A \to \boxtimes p)$ for some $\rhd$-formula A, i.e.,

$$\begin{array}{ll} \exists x{\downarrow}w & x \models A \to \boxtimes p, \\ \exists y{\downarrow}w & y \models A, \neg\boxtimes p. \end{array}$$

The latter, in turn, implies the existence of a $\rhd$-formula B such that

$$\begin{array}{ll} \exists s{\downarrow}y & s \models B \to p, \\ \exists t{\downarrow}y & t \not\models B \to p. \end{array}$$

By transitivity, $w{\uparrow}s$, $w{\uparrow}t$, so $w \not\models \rhd(B \to p)$, in contradiction with $w \models \boxtimes p$.

$\underline{\mathbf{4} \in \mathbf{\Sigma}.}$ We show that $\neg\boxtimes p \to \boxtimes\neg\boxtimes p$ is true at any world w of any euclidean model. Let $w \models \neg\boxtimes p$, i.e., $w \models \neg\rhd(A \to p)$ for some $\rhd$-formula A. By the axiom $(\mathsf{A}^{\rhd}_{\mathbf{5}})$, we conclude $w \models \rhd[B \to \neg\rhd(A \to p)]$ for any $\rhd$-formula B, i.e., $w \models \boxtimes\neg\rhd(A \to p)$. Since the implication $\neg\rhd(A \to p) \to \neg\boxtimes p$ is valid in any frame, the formula $\boxtimes\neg\rhd(A \to p) \to \boxtimes\neg\boxtimes p$ is valid too (we can use the monotonicity principle "from $\varphi \to \psi$ it follows that $\boxtimes\varphi \to \boxtimes\psi$", for $\boxtimes$ is a normal modal operator). Thus we have $w \models \boxtimes\neg\boxtimes p$. $\dashv$

This theorem cannot be generalized to all logics. A counterexample is $\mathbf{KB}^{\boxtimes} \not\supseteq \mathbf{KB}$, where $\mathbf{KB} = \mathbf{K} + (\mathsf{A}^{\Box}_{\mathbf{B}})$ and $(\mathsf{A}^{\Box}_{\mathbf{B}})$ is the symmetricity axiom $p \to \Box\Diamond p$. One can easily construct a finite symmetric frame falsifying the formula $p \to \boxtimes\neg\boxtimes\neg p$.

It is worth noting that all the previous reasoning is valid if, in the definition of $\boxtimes$, the infinitary conjunction is taken only over the set of *literals* $\mathrm{L} := \{p, \neg p \mid p \in \mathrm{Var}\}$. So, in what follows, we assume that $\boxtimes$ is defined as

$$\boxtimes A := \bigwedge_{\ell \in \mathrm{L}} \rhd(\ell \to A).$$

(In fact, this new operator $\boxtimes$ is *not* semantically equivalent to the previous one, as can be easily shown; however, the results obtained above remain true under new definition of $\boxtimes$ as well). Recall that, starting from $\Box$, we have defined the operator $\rhd$ and then the operator $\boxtimes$. What if we iterate the procedure? Schematically, the next iteration looks like:

$$\blacktriangleright A := \boxtimes A \vee \boxtimes\neg A; \qquad \boxplus A := \bigwedge_{\ell \in \mathrm{L}} \blacktriangleright(\ell \to A).$$

Fortunately, this iteration of the construction is redundant.

Theorem 4.3. *The operators $\boxtimes$ and $\boxplus$ are semantically equivalent, i.e., the formula $\boxtimes p \leftrightarrow \boxplus p$ is valid in any frame. Moreover, we have $\models \triangleright p \leftrightarrow \blacktriangleright p$.*

Proof. Validity of the implication $\triangleright p \to \blacktriangleright p$ follows from $\models \Box p \to \boxtimes p$. Now, using $\models \boxtimes p \to \triangleright p$, we obtain the converse implication:
$\models \blacktriangleright p \longleftrightarrow (\boxtimes p \vee \boxtimes \neg p) \longrightarrow (\triangleright p \vee \triangleright \neg p) \longleftrightarrow \triangleright p.$ $\dashv$

Let us observe the following distinctive feature of the operator $\boxtimes$. We have established that $\boxtimes$ possesses the following two properties: (a) $\boxtimes$ is a normal modal operator; (b) the operator $\triangleright$ is $\boxtimes$-definable by the equality $\triangleright A = \boxtimes A \vee \boxtimes \neg A$ (since $\triangleright$ and $\blacktriangleright$ are equivalent). It turns out that $\boxtimes$ is the weakest modality possessing (a) and (b) simultaneously (under modality here we mean any unary operator supplied by Kripke semantics; of course, this is not a formal definition; for example, any modal formula of one variable suits for ours purposes). Indeed, assume that $\boxminus$ is a modality satisfying (a) and (b). To prove that $\models \boxminus p \to \boxtimes p$, take any literal ℓ and put $A := (\ell \to p)$. From (b) it follows that $\models (\boxminus A \vee \boxminus \neg A) \to \triangleright A$, hence $\models \boxminus A \to \triangleright A$. By normality of $\boxminus$, we have $\models \boxminus p \to \boxminus(\ell \to p)$. Therefore $\models \boxminus p \to \triangleright(\ell \to p)$, for any ℓ, hence the claim: $\models \boxminus p \to \boxtimes p$.

Theorem 4.3 immediately implies that the infinitary $\Box$- and $\boxtimes$-logics of **K** are distinct. Formally, denote by L_∞ the set of all infinitary $\Box$-formulas (defined similarly to infinitary $\triangleright$-formulas) that are valid in any L-frame:

$$L_\infty = \{A \in \mathbf{Fm}_\infty^\Box \mid \text{ for any frame } F \ (F \models L \Rightarrow F \models A)\}.$$

One can define $L_\infty^\boxtimes$ in the same manner. Now observe that $\mathbf{K}_\infty \neq \mathbf{K}_\infty^\boxtimes$, for the logic $\mathbf{K}_\infty^\boxtimes$ contains the formula $\boxtimes p \leftrightarrow \boxplus p$, or explicitly,

$$\boxtimes p \leftrightarrow \bigwedge_{\ell \in \mathrm{L}} \Big(\boxtimes(\ell \to p) \vee \boxtimes \neg(\ell \to p)\Big),$$

whereas the logic $\mathbf{K}_\infty$ does not contain the corresponding infinitary $\Box$-formula (since $\Box$ and $\boxtimes$ are not equivalent).

5 Conclusion

The aim of this paper was to introduce a new modal operator $\boxtimes$ defined in terms of the non-contingency operator. As we have observed, $\boxtimes$ is a necessity operator (Theorem 4.1), which is similar to the original necessity $\Box$ is some aspects (Theorem 4.2) and different from it is some others ($\mathbf{KB}^\boxtimes \not\supseteq \mathbf{KB}$, $\mathbf{K}_\infty^\boxtimes \neq \mathbf{K}_\infty$). The new necessity has several distinctive features (idempotency of the construction $\Box \mapsto \boxtimes$, the fact that $\boxtimes$ is the weakest necessity such that $\triangleright$ is $\boxtimes$-definable in a natural manner).

Our main conjecture is that $\mathbf{K}^\boxtimes = \mathbf{K}$. If this is the case then the construction of $\boxtimes$ may be regarded as a solution of the problem concerning definability of necessity in terms of contingency. If not then the logic $\mathbf{K}^\boxtimes$ is a new modal logic of particular interest, like **K**, **K4** etc. Another interesting issue is axiomatization of infinitary $\boxtimes$-logics $L^{\boxtimes}_{\infty}$ over various modal logics L. This is a rather natural question, for the very definition of $\boxtimes$ is infinitary. Our candidate for $\mathbf{K}^{\boxtimes}_{\infty}$ is $\mathbf{K}_{\infty}[\boxtimes/\Box] + \{\boxtimes p \leftrightarrow \boxplus p\}$. These questions seem to be of both technical and philosophical interest, and the answers may shed a new light on the interconnection between necessity and contingency.

References

[1] A. Chagrov, M. Zakharyaschev, *Modal Logic*, Oxford Science Publications, 1997.

[2] M. J. Cresswell, Necessity and contingency, *Studia Logica*, vol. 47 (1988), pp. 145–149.

[3] I. L. Humberstone, The logic of non-contingency, *Notre Dame Journal of Formal Logic*, 1995, 36(2):214–229.

[4] S. T. Kuhn, Minimal non-contingency logic, *Notre Dame Journal of Formal Logic*, 1995, 36(2):230–234.

[5] H. Montgomery, R. Routley, Contingency and non-contingency bases for normal modal logics, *Logique et analyse*, 9 (1966), 318–328.

[6] H. Montgomery, R. Routley, Non-contingency axioms for $S4$ and $S5$, *Logique et analyse*, 11 (1968), 422–424.

[7] H. Montgomery, R. Routley, Modalities is a sequence of normal non-contingency modal systems, *Logique et analyse*, 12 (1969), 225–227.