CLAIRE GARDENT AND KRISTINA STRIEGNITZ

GENERATING BRIDGING DEFINITE DESCRIPTIONS

1. Introduction

It has long been known that knowledge based reasoning is a crucial component of natural language processing (NLP). Yet the complexity involved in representing and using knowledge efficiently has led most NLP work to focus on more tractable aspects of language such as syntax, prosody or semantic construction.

The generation of definite descriptions (that is, noun phrases with a definite article such as "the rabbit") is a case in point. The goal of this sub-task of natural language generation (which is the production of a text satisfying a given communicative goal) is to construct a noun phrase that allows the hearer to uniquely identify its referent in the context of utterance.

The standard algorithm for this task (on which most other proposals are based) is presented in (Dale and Reiter, 1995). But neither this algorithm nor the extensions proposed in (Horacek, 1997) and (Krahmer and Theune, 2001) take world knowledge into account when considering the context of utterance. For all these algorithms, the context is a *fixed* set of positive literals specifying entities and relations between them, which is intended to represent the current (linguistic and situational) context of utterance.

Yet many definite descriptions either refer to inferable entities (entities not explicitly mentioned or present in the context of utterance but inferable from it) or refer to contextually salient entities using inferred rather than explicitly mentioned relations (Poesio and Vieira, 1998). For instance, the use of the definite article to refer to the patrons, the waitress, the busboys, etc. in (1) can only be explained by taking into account that world knowledge supports the assumption that these are somehow related to the restaurant.

(1) The young woman scans **the restaurant** with this new information. She sees all *the patrons* eating, lost in conversations. *The tired waitress*, taking orders. *The busboys* going through the motions,

collecting dishes. *The manager* complaining to the cook about something.

One proposal which does integrate knowledge based reasoning into the generation of definite descriptions is that presented in Stone, 1998. The philosophy underlying Stone's proposal is that knowledge based reasoning should be integrated with sentence planning to reason about what the context (including a representation of the previous discourse as well as world and situational knowledge) entails. In this paper, we follow up on Stone's proposal and show how to integrate surface realization and inference into Dale and Reiter's algorithm to support the generation of such definite descriptions as illustrated in (1).

We start (Section 2) by presenting Dale and Reiter's base algorithm. Section 3 then summarizes the range of definite descriptions found in corpora while Section 4 focuses on cases involving knowledge based reasoning. In Section 5, we consider the two defining characteristics of definite descriptions, uniqueness and familiarity, and show how these can be defined to encompass not only directly coreferential, but also inference-based definite descriptions. Section 6 presents the extended algorithm and an implementation of it based on description logic. Section 7 concludes and points to further research.

2. The Standard Algorithm for Generating Definite Descriptions

Most algorithms for generating definite descriptions that are described in the current literature have a common core based on Dale's greedy algorithm (Dale, 1989; Dale, 1992) and on the incremental algorithm proposed in (Dale and Reiter, 1995). We now describe this standard base algorithm.

2.1. *The Task*

The task is to find a description of an object (the *target entity*) that allows the hearer to uniquely identify that object in a given situation, that is, to find a description that does not fit any other object in that situation. In Dale and Reiter's algorithm, this situation is represented by a set C of positive literals such as shown in Figure 1. This representation is meant to capture the knowledge that speaker and hearer share.

$$\{rabbit(r_1), rabbit(r_2), rabbit(r_3), hat(h_1), hat(h_2), bathtub(b_1),$$
$$white(r_1), black(r_2), white(r_3), in(r_1, h_1), in(r_2, h_2), in(r_3, b_1)\}.$$

Figure 1. Representation of the discourse context in Dale and Reiter's algorithm.

The output of Dale and Reiter's algorithm, that is, the description of the target entity, is a subset L of C which uniquely identifies the target entity. The target is uniquely identified by L if there are no *distractors* for the target in C given L. Distractors are defined as follows:

DEFINITION 1. **(Distractors)** *Given a description L ($L \subseteq C$) and an entity a, the set of distractors of a in C are all those entities $b \neq a$ for which there exists a substitution σ (substituting entity symbols for entity symbols) such that $\sigma(a) = b$ and $\sigma(L) \subseteq C$.*

Let's assume that the goal is to describe the object r_1. The set $\{rabbit(r_1)\}$ would not be a uniquely identifying description in this case, as r_2 and r_3 are *distractors* of r_1 with respect to this description. In contrast, the description $\{rabbit(r_1), white(r_1), in(r_1, h_1), hat(h_1)\}$ would rule out all distractors and therefore be uniquely identifying.

2.2. *The Base Algorithm*

The standard algorithm takes as input a target entity t and a set C of positive literals as shown in Figure 1. It either returns a subset of C that uniquely identifies t (the description) as output or fails in case such a description cannot be built.

The algorithm starts with an empty set and then incrementally adds literals until the description uniquely identifies all entities mentioned in it. The search problem the algorithm has to solve is given in Figure 2. States are determined by the description that has been built so far and the list of those entities that the description mentions but does not identify uniquely. In the initial state, the description is empty and the target list contains only t, which is the initial target provided as input. Goal states are all those states where the target list is empty, which means that all entities mentioned in the description are uniquely identified. A state $s(i + 1)$ can be derived from state $s(i)$ by applying the search operator to it. This operator first adds a new literal to the description. This literal is chosen from the given context and has to fulfill the following two constraints. It has to extend the description, which means that it has to mention an entity also mentioned in $\mathsf{Description}(i)$. Furthermore, it has to rule out at least one distractor of at least one entity mentioned in $\mathsf{Description}(i+1)$. This means that $\mathsf{Description}(i+1)$ mentions at least one entity for which the set of distractors given $\mathsf{Description}(i)$ is smaller than the set of distractors given $\mathsf{Description}(i+1)$. Finally, the list of target entities is updated. This involves first adding all those entities to the list that are mentioned in the newly added literal, but were not mentioned in the description before, and then deleting all those entities that are uniquely identified by the new description.[1]

There are different ways in which the search can be performed. Dale and Reiter discuss three, which they call *Full Brevity*, *Greedy Heuristics*, and *Incremental Algorithm*.[2] The *Full Brevity* strategy is a breadth first search which stops as soon as a goal state is found. It is guaranteed to always find the shortest descriptions possible. However, as Dale and Reiter point out, it has worst-case runtimes which are exponential in the size of the final description. They therefore propose to use heuristics which allow to approximate the optimal solution with a greedy search mechanism. The search strategy that Dale and Reiter call *Greedy Heuristic* always chooses the search step that rules out the most distractors. The *Incremental Algorithm* uses the following heuristics. It assumes that properties are ordered according to a domain dependent preference order. Going back to the example context in Figure 1, we could, for example, assume that sortal information (*rabbit, hat, bathtub*) ranks higher than color information, which ranks higher than location information. This order can then be used to order all literals concerning a particular entity. For entity r_1, for example, we would get the ordering $\langle rabbit(r_1),\ white(r_1),\ in(r_1,h_1)\rangle$ and for entity h_1 the ordering $\langle hat(h_1),\ in(r_1,h_1)\rangle$. This then determines in which order literals are added to the description. Of all the literals that are applicable in a given state, the one that is ranked highest in the order has to be chosen. Hence, the Incremental Algorithm just has to

[1] In the 1995 paper, Dale and Reiter actually do not deal with relations between objects (binary predicates). So, what we are presenting here is a simple extension along the lines of (Dale and Haddock, 1991).

[2] Note that the names are somewhat misleading: all three algorithms build the description incrementally and both the *Greedy Heuristics* as well as the *Incremental Algorithm* perform a greedy search.

input: a set of literals C (the context)
a target entity t

state $s(i)$**:** $\mathsf{Description}(i)$, a set of literals
$\mathsf{Targets}(i)$, a list of target objects

initial state: $\mathsf{Description}(0) = \emptyset$
(state $s(0)$) $\mathsf{Targets}(0) = \{t\}$

goal state: State $s(i)$ is a goal state if
$\mathsf{Targets}(i) = \emptyset$

operator:
1) Select a literal P from C, where P extends the description and rules out at least one distractor.
2) $\mathsf{Description}(i+1) = \mathsf{Description}(i) \cup \{P\}$.
3) $\mathsf{Targets}(i+1)$ is derived from $\mathsf{Targets}(i)$ by updating it with P: first, those entities mentioned in P that are not mentioned in $\mathsf{Description}(i)$ are added and then, all those entities which given $\mathsf{Description}(i+1)$ do not have any distractors (Definition 1) are deleted.

Figure 2. Searching a uniquely identifying description.

go stepwise through the ordered lists of literals for the entities that are part of the description and either include or discard each literal.

The output of the base algorithm is a list of properties that uniquely identify the target entity. So, this base algorithm determines the semantic content a referring expression has to express in order to be successful, it does not, however, produce the surface form of such a referring expression. It is therefore possible that the base algorithm produces an output that cannot be verbalized.

The general idea for avoiding this problem is to interleave property selection with surface realization (Horacek, 1997). This allows us to immediately check that every selected property can be incorporated in the syntactic tree, and to make sure that the final description has no "holes" which have to be filled due to syntactic reasons. For example, "the red" is not a good noun phrase in English (or at least it requires a special context). To turn it into one we would have to include some

State	Description	Targets	Distractors
0	$\emptyset$	$\{r_1\}$	r_1: all entities other than r_1
1	$\{rabbit(r_1)\}$	$\{r_1\}$	r_1: $\{r_2, r_3\}$
2	$\{rabbit(r_1),\ white(r_1)\}$	$\{r_1\}$	r_1: $\{r_3\}$ h_1: all entities other than h_1
3	$\{rabbit(r_1),\ white(r_1),\ in(r_1,h_1)\}$	$\{r_1, h_1\}$	r_1: $\{r_3\}$ h_1: $\{b_1\}$
4	$\{rabbit(r_1),\ white(r_1),\ in(r_1,h_1),\ hat(h_1)\}$	$\emptyset$	r_1: $\emptyset$ h_1: $\emptyset$

Figure 3. Running Dale and Reiter's algorithm with target r_1 on the context in Figure 1.

property that can be realized as a head noun, even if it doesn't rule out any distractors.

2.3. *An Example*

To illustrate how the base algorithm works, we will now go through the example of Section 2.1. Let us assume that the input context is as given in Figure 1 and that the target entity is r_1. Figure 3 shows state by state how the greedy search advances.

In the beginning, state 0, the description is empty. Hence, it does not distinguish the target r_1 from any other entity. The distractor set contains all entities mentioned in the context. Adding literals to the description cuts down the distractor sets of the targets more and more. In state 3, for example, r_3 is a distractor of r_1, because substituting r_3 for r_1 and b_1 for h_1 would yield the description $\{rabbit(r_3), white(r_3), in(r_3, b_1)\}$, which is a subset of the context.

The example is following Dale and Reiter's *Incremental Algorithm* and literals are added in the order described above: first sortal information, then color information, and then location information.

3. Definite Descriptions in Real Texts

We now survey the types of definite descriptions that can be found in corpora thereby giving a list of the different cases that an algorithm for generating definite descriptions should be able to deal with.

Two properties are generally taken to characterize definite descriptions namely, *uniqueness* and *familiarity*. Roughly, uniqueness says that the referent denoted by the definite description must be the only referent satisfying the given description – this property is most prominently exposed in (Russell, 1906). Familiarity on the other hand, requires that this referent be known to the hearer – this is perhaps most strongly demonstrated in (Heim, 1982).

Indeed these two properties are the properties taken into account by the base algorithm: the set of relations it outputs must allow the hearer to uniquely identify the intended referent (uniqueness) and it must do so on the basis of shared knowledge about that referent (familiarity).

The familiarity/uniqueness explanation is a fairly high-level one however, and as shown by, for example, Hawkins (1978) or Prince (1981), a finer grained examination of the phenomenon reveals a much more complex typology of possible uses. For a start, uniqueness is always relative to some restricted context. Here are some examples.

(2) b. If a rabbit sees a carrot, the rabbit eats the carrot.
c. There once was a doctor in London. The doctor was Welsh.

In (2a), uniqueness is relative to the quantification domain: for each rabbit and for each carrot that this rabbit sees, the rabbit eats the carrot that it sees. Similarly in (2b), uniqueness is relative to the domain of discourse: (2b) does not imply that there is a unique Welsh doctor in London but that there is a unique Welsh doctor in London *that the speaker is talking about.*

Although the base algorithm simply assumes an already restricted context, Krahmer and Theune (2001) show how it can be extended to deal with discourse domain restrictions. Interaction with quantification remains an open question and will probably remain so for a while as quantifiers have received little attention in the generation literature.

Moreover – and this is the main point of this paper – familiarity can be of different types, and only some of them are covered by the base algorithm. Following Poesio and Vieira (1998), for instance, we can identify four main familiarity classes: *coreferential* (direct or indirect), *bridging*, *larger situation*, and *unfamiliar* uses. In *coreferential uses*, the referent of the definite description is familiar in virtue of having been mentioned in the previous discourse (the referent is *discourse old* in Prince's (1981) terminology). In such cases, the hearer will know the intended referent either because the speaker uses the same description as was used in the previous discourse (*direct coreference*, as in (3a)) or because she uses a description which on the basis of world or lexical knowledge, the hearer can infer to be true of the previously mentioned entity (*indirect coreference*, as in (3b))

(3) a. *A woman* came in. *The woman* was wearing a beautiful hat.
b. *An actress* entered the stage. *The woman* was wearing a beautiful hat.

In a *bridging use*, the referent of the definite description is discourse new but related by world knowledge to some previously mentioned, that is, discourse old, entity. In (4) for instance, the referent of *the ceiling* is related by a *part-of* relation to the discourse old entity denoted by the NP *the room.* The ceiling is not just any ceiling but the ceiling of the room that was mentioned in the previous sentence.

(4) A woman came into *the room. The ceiling* was very high.

Larger situation uses (see (5)) are cases where the definite description denotes a discourse new but hearer old object: the described entity has not been mentioned in the previous discourse but is assumed by the speaker to be part of the hearer's general knowledge about the world (5a) and situation of utterance (5b).

(5) a. *The sun* is rising.
b. Pass *the salt* please !

The *unfamiliar* class covers all remaining uses of definite descriptions; that is, uses where the referent of the description is neither discourse/hearer old nor related by lexical knowledge to some discourse old entity. It encompasses definite descriptions with sentential complements (6a) and with modifiers relating the referent of the definite description to some either discourse or hearer old object (6b-c).

(6) a. Bill is amazed by *the fact that his father is black.*
b. *The man John met yesterday* is interesting.
c. *The Iran/Iraq war* is over.

In sum, a definite description can be familiar either because it refers to some known (that is, either discourse or hearer old) entity (coreferential and situational use); or because it is related, either explicitly by the description (unfamiliar use) or implicitly by lexical knowledge (bridging), to some known entity.

The base algorithm, because it only resorts to information that is explicit in the context of utterance (either through the previous discourse or through the situation), can only account for directly coreferring or larger situation uses. Indirect coreferences and bridging uses cannot be dealt with as they require an interaction between generation and inference. In the next section, we look at these *inference based definite descriptions* in more detail to see what is needed to extend the base algorithm so that it can deal with them.

4. Definite Descriptions and Inference

Inference based definite descriptions, that is, bridging and indirect coreferential uses, represent a non negligible proportion of uses in real text. An empirical study of the Wall Street Journal by Poesio and Vieira (1998) shows that out of 1412 definite descriptions being studied, 24% were inference based uses, with 9% bridging cases and 15% indirect coreference.

In both cases, processing requires reasoning based on world knowledge and the discourse context. With bridging uses, the hearer must be able to infer the implicit relation holding between the referent of the definite description and a discourse or hearer old entity. And in cases of indirect coreferential uses, the hearer must be able to infer that the properties used in the speaker's definite description, although not part of the common ground between speaker and hearer, hold of some discourse or hearer old entity.

We now consider these two cases in more detail using (here and in what follows) the following terminology. We call the referent of the definite description the *target* and the (discourse or hearer old) entity with which it is either coreferential or related via world knowledge the *anchor*.

4.1. *Bridging*

In a bridging use, a definite description relates the target entity to its anchor via some inferable relation, which we will call *bridging relation.*

The term *bridging* was first introduced by Clark (1977), who identified several types of different bridging relations, such as the *part-of* relation, semantic roles of verbs, reasons, consequences. In this paper, we will concentrate on the *part-of* relation. Clark distinguishes three subcases of bridges involving the *part-of* relation, which are illustrated by the following examples.

(7) a. John entered *the room. The ceiling* was very high.
b. John entered *the room. The windows* looked out to the bay.
c. John entered *the room. The chandelier* was sparkling brightly.

In (7a), *the ceiling* is a part of the room mentioned in the previous sentence and it is a *necessary* part: to be a room, a room must have a ceiling. In contrast, *the windows* and *the chandelier* in (7b-c) are what Clark called *probable* and *inducible* parts. Rooms don't necessarily have windows, nor do they necessarily have a chandelier. But while most rooms have windows, rooms with chandeliers are actually rare. Nevertheless, it is plausible to link the chandelier in (7c) to the room as rooms typically have lamps and a chandelier is a type of lamp.

There is, of course, a wide variety of different kinds of *part-of* relations. Kleiber (1997) and Gardent et al. (2003) study how different part of relations can be involved in bridging anaphora. In this paper, we gloss over the differences and use *bridge* to stand for a generic *part-of* relation that subsumes the different varieties of relevant relations.

4.2. *Indirect coreferential uses*

Indirect coreferential uses are cases where the definite description refers to a discourse old entity using a hearer new description. So, in this case target and anchor are the same entity. Although the description used is not part of the common ground, the hearer can nevertheless identify the intended object because the description used by the speaker is compatible with the common ground and with world knowledge and allows the hearer to establish the link to the anchor. Here are some examples.

(8) a. *An actress* entered the stage. *The woman* was wearing a beautiful hat.
b. I met *a man* yesterday. *The bastard* stole my money.
c. John bought *a new car. The Volvo* delights him.

In the first case, the property used in the definite description is entailed by world knowledge, as *woman* is a hypernym of *actress.* But as the other two examples show, world knowledge does not necessarily entail that the property used in the description holds of the target. Thus, a man can, but need not, be a bastard – *x is a bastard* is a proposition that is compatible but not entailed by the proposition *x is a man*. Similarly, a car can but need not be a Volvo – in this case, however, the two properties are not merely compatible, but stand in a hyponymic relation (Volvos are cars).

So, for both kinds of inference based definite descriptions, bridging uses and indirect coreferential uses, there are cases where the existence of an entity fitting the description is not logically entailed by the discourse context (Examples (7b,c) and (8b,c)). We focus here on cases where the existence of an entity fitting the description is entailed and cases where the description is related through hyponymy to a property of target that is entailed.

5. Familiarity and Uniqueness of Bridging Anaphora

As we saw in section 3, familiarity and uniqueness are two defining characteristics of definite descriptions. In this section, we show how to formulate these properties so as to encompass not only directly coreferential definite descriptions (as is done in the base algorithm) but also indirectly coreferential and bridging uses. We start by presenting the structured context the extended algorithm is working with. We then go on to give an intuitive explanation of how uniqueness and familiarity differ in our algorithm from the way these are defined in the base algorithm. Finally, we present our definitions.

5.1. *The Discourse Context*

As seen in Section 2, a context in the base algorithm is an unstructured set of facts assumed to be shared by hearer and speaker. To deal with bridging anaphora, we need a slightly more sophisticated notion of context.

First, we need to distinguish between knowledge that is shared between hearer and speaker and the private knowledge of the speaker. The speaker should, for example, only use the bridging description *the ceiling* in Example (7a) if he assumes that the hearer knows the rule that rooms have ceilings. The target entity, on the other hand, is, in the case of bridging descriptions, an entity which is inferable but new for the hearer. All explicit knowledge about the target entity should

$$\begin{array}{|ll|}\hline C_{shared}: & \mathit{book}(b), \\ & \forall y[\mathit{book}(y) \rightarrow \exists x[\mathit{author}(x) \wedge \mathit{of}(x,y)]], \\ & \forall xy[\mathit{of}(x,y) \rightarrow \mathit{bridge}(x,y)] \\ C_{private}: & \mathit{author}(a),\ \mathit{of}(a,b), \\ & \mathit{cockroach}(c),\ \mathit{of}(c,b) \\ \hline \end{array}$$

Figure 4. Example context to illustrate the definitions of anchors.

therefore be private knowledge of the speaker. Hence, we will use a structured context representation consisting of the speaker's/system's private knowledge $C_{private}$ and the knowledge that speaker and hearer share C_{shared}.

Secondly and as we alluded to above, we need rule based knowledge to capture the way concepts such as *rooms* and *ceiling* are linked through bridging relations. So, C_{shared} consists not only of positive literals but also of rules of the form *all entities with propery A are related to an entity with property B through a bridging relation.* Figure 4 shows an example context.

5.2. *Familiarity and uniqueness in the extended algorithm*

The uniqueness condition for coreferential definite descriptions described in Section 2 requires that the target entity be the only entity in the discourse context fitting the description. Furthermore, no familiarity condition was mentioned in the specification of the base algorithm. This is because it only generates coreferential definite descriptions, and without making it explicit, it assumes a very simple familiarity condition: The use of a definite description is only licensed if the target entity and the property used to describe that entity are discourse old. We now describe how these notions of familiarity and uniqueness can be extended to also capture bridging description. Our proposal will also be able to handle indirectly coreferential definite descriptions.

When looking at bridging descriptions, familiarity is important because we have to distinguish those discourse new entities which can be anaphorically linked to the previous discourse (the familiar ones) from those discourse new entities which do not license an anaphoric link to the previous discourse (the unfamiliar ones).[3] The use of *the cockroach* in (9), for example, is odd since familiarity is not given. Even if the

[3] Note that in our terminology, not only discourse old entities are familiar. Inferable discourse new entities are also familiar.

speaker had some information that links the cockroach to the book and intended the book to be the anchor, it is not part of general world knowledge that cockroaches are parts of books, and so the speaker cannot expect the hearer to make this link. In other words, to ensure familiarity *the definite descriptions that the speaker uses should be such that they let the hearer find at least one of the anchors intended by the speaker.*

(9) I picked up a book and *the cockroach* fell out.

Now, consider Example (10). Assume that the speaker intended the Italian restaurant to be the anchor for the cook. In this case, using the definite description *the cook* is not appropriate because from the hearers point of view either of the restaurants could be the anchor. That is, the hearer considers an entity to be a possible anchor which the speaker does not intend to be an anchor. So, this is one part of the uniqueness condition that definite descriptions have to satisfy: *definite descriptions should be such that they do not allow the hearer to consider entities to be anchors which the speaker does not intend to be anchors.*

(10) There are an Italian and a Chinese restaurant. *The cook* is excellent.

But there is another way in which uniqueness needs to ensured: the description should make the target unique with respect to the anchor. (11) shows an example where this condition is violated.

(11) I picked up a book and *the page* fell out.

The definite description *the page* lets the hearer identify the book as a possible anchor (because world knowledge tells him that books have pages), and there are no other entities which could be an anchor. So the familiarity condition and the first part of the uniqueness condition are satisfied. However, world knowledge says that a book usually has more than one page. So, the second part of the uniqueness condition requires that *the definite description should be such that it is plausible to assume that there are no entities besides the target which fit the description and are related to the anchor via a bridging relation.*

These informal definitions of familiarity and uniqueness crucially depend on two sets: the set of entities that the speaker intends to be anchors, which we call the *speaker anchors*, and the set of entities that the hearer considers to be possible anchors, or the *hearer anchors.* In

$$\begin{array}{|ll|}\hline C_{\mathit{shared}}: & \mathit{book}(b), \\ & \forall y[\mathit{book}(y) \rightarrow \exists x[\mathit{author}(x) \wedge \mathit{of}(x,y)]], \\ & \forall xy[\mathit{of}(x,y) \rightarrow \mathit{bridge}(x,y)] \\ C_{\mathit{private}}: & \mathit{author}(a),\ \mathit{of}(a,b), \\ & \mathit{cockroach}(c),\ \mathit{of}(c,b) \\ \hline \end{array}$$

Figure 5. Example context to illustrate the definitions of anchors.

the following section, we will make the notions of speaker and hearer anchor more precise, and in Section 5.4, we will give formal definitions of the familiarity and uniqueness conditions based on these notions.

5.3. *Defining Hearer and Speaker Anchors*

We will now define the sets of *speaker anchors* and *hearer anchors*. The intuition behind these sets is as follows: the speaker anchors are the entities intended by the speaker to be possible anchors and the hearer anchors are those entities which the hearer considers as anchors (or rather which the speaker thinks the hearer considers as anchors). For an anaphoric expression to be successful, the two sets have to coincide in a way that will be made more explicit in this section and the next.

The set of speaker anchors contains all entities a such that the speaker knows that a is identical to or related to target entity t. In other words, the set of speaker anchors contains all entities which could act as anchors for the target. The set of speaker anchors is defined as follows.

DEFINITION 2. **(Speaker Anchors)** *The set of* direct speaker anchors *(*dSA*) for a given target entity* t *is simply* $\mathsf{dSA}(t) = \{t\}$. *The set of* indirect speaker anchors *(*iSA*) for a given target entity* t *in a given context* C *is defined as follows:*

$$\mathsf{iSA}(t, C) = \{a \mid C_{\mathit{private}} \cup C_{\mathit{shared}} \models \mathit{bridge}(t, a)\}.$$

The set of speaker anchors *(*SA*) for a given target entity* t *in a given context* C *is*

$$\mathsf{SA}(t, C) = \mathsf{dSA}(t) \cup \mathsf{iSA}(t, C).$$

For instance, given the context in Figure 5 the set of speaker anchors for entity b is $\{b\}$, the set of speaker anchors for a is $\{a, b\}$ and the set of speaker anchors for c is $\{c, b\}$.

The intuition behind the set of hearer anchors is that the speaker tries to model how the hearer will interpret a given description. The set of hearer anchors, hence, contains all those entities (known to both speaker and hearer) which when taking into account only the shared knowledge, could act as anchors for the given description. For instance, given the description *the cook* and assuming that it is shared knowledge that restaurants have cooks, the set of hearer anchors would contain all cooks and all restaurants mentioned in the shared knowledge (independently of whether they are in fact related to the target). Since the hearer does not know the target, what he considers possible anchors is based on the description only. The set of hearer anchors is, therefore, defined with respect to the context and the property given by the description.

DEFINITION 3. **(Hearer Anchors)** *Given a property P, and a context C, the set of* direct hearer anchors *(*dHA*) is defined as*

$$\mathsf{dHA}(P, C) = \{a \mid C_{shared} \models P(a)\},$$

the set of indirect hearer anchors *(*iHA*) is defined as*

$$\mathsf{iHA}(P, C) = \{a \mid C_{shared} \models \exists x[bridge(x, a) \wedge P(x)]\},$$

and the set of hearer anchors *(*HA*) is*

$$\mathsf{HA}(P, C) = \mathsf{dHA}(P, C) \cup \mathsf{iHA}(P, C).$$

Definition 3 says that all entities of which the hearer knows that the description holds are direct hearer anchors. So, the book b in Figure 5 is a hearer anchor for entity b given the property *book*. In addition, a discourse old entity can be an indirect hearer anchor if the hearer knows that this discourse old entity is related to an entity of which the property holds. For this reason, entity b in Figure 5 is a hearer anchor for entity a given the property *author*. The entity b is not a hearer anchor of c given property *cockroach*, though, as $\exists x[bridge(x, b) \wedge cockroach(x)]$ does not follow from the shared knowledge.

This definition for hearer anchors captures cases of bridging or indirectly coreferential definite descriptions where the existence of an entity fitting the description is entailed by the context (Examples (7a) and (8a)). To also capture cases where the property used to describe the target is a hyponym of a concept for which the existence of an entity belonging to that concept is entailed (Examples (7b), (7c), and (8c), we need to extend our definition of hearer anchors a bit.

Recall the reasoning process that we proposed to explain how the link between the room and the chandelier in Example (7c) is established: From world knowledge we know that rooms are related to certain objects which we call room accessory ($room_a cc$) ($\forall x[room(x) \rightarrow \exists y[room\_acc(y) \wedge bridge(x,y)]]$), and we know that chandeliers are a kind of room accessory ($\forall x[chandelier(x) \rightarrow room\_acc(x)]$). Similarly, we know that Volvos are a kind of car, which allows the hearer to consider the car mentioned in the first sentence of Example (8c) as a possible anchor for the Volvo mentioned in the second sentence.

To handle such cases the definition of hearer anchors is modified to involve not just the property P given by the description but also an adequately generalized version of this property ($P[Q/N]$). $P[Q/N]$ is derived from P by replacing the part of P that represents the meaning of the head noun, let's call it N, with a more general concept Q. (Note that this reference to the head noun means that this strategy only works if surface realization is interleaved with the content planning for definite descriptions.) Furthermore, we restrict Q to be the most specific property for which C_{shared} entails that a possible anchor has property Q (for direct hearer anchors) or for which C_{shared} entails that there is a bridging relation between a possible anchor entity and an entity with property Q (for indirect hearer anchors). So, this extended version of the definition of hearer anchors looks like this:

DEFINITION 4. **(Hearer Anchors — revised)** *Given a property P, and a context C, the set of* direct hearer anchors *(*dHA*) is defined as*

$$\mathsf{dHA}(P,C) = \{a \mid \exists P[Q/N] \text{ such that } C_{shared} \models P[Q/N](a)\},$$

the set of indirect hearer anchors *(*iHA*) is defined as*

$$\begin{aligned}\mathsf{iHA}(P,C) \; = \; \{a \mid \; & \exists P[Q/N] \text{ such that} \\ & C_{shared} \models \exists x[bridge(x,a) \wedge P[Q/N](x)] \text{and} \\ & Q \text{ is the most specific concept for which} \\ & C_{shared} \models \exists x[bridge(x,a) \wedge Q(x)]\},\end{aligned}$$

and the set of hearer anchors *(*HA*) is*

$$\mathsf{HA}(P,C) = \mathsf{dHA}(P,C) \cup \mathsf{iHA}(P,C).$$

5.4. *Defining Familiarity and Uniqueness*

We can now define the familiarity condition (given the definite description the hearer should be able to find at least one of the anchors that the speaker intended) in terms of hearer and speaker anchors. Namely, the intersection of these two sets should not be empty.

DEFINITION 5. **(Familiarity)** *An entity t described using property P is familiar in a context C if* $(\mathsf{dSA}(t) \cap \mathsf{dHA}(P, C)) \cup (\mathsf{iSA}(t, C) \cap \mathsf{iHA}(P, C)) \neq \emptyset$.
We will also call $(\mathsf{dSA}(t) \cap \mathsf{dHA}(P, C)) \cup (\mathsf{iSA}(t, C) \cap \mathsf{iHA}(P, C))$ *the set of* familiar anchors *(*FA*(t,P,C)).*

This definition allows for entities to have several familiar anchors based on the same property. If there are several entities to which the target is related and for which lexical or world knowledge provides a relation to the target, all of these entities count as familiar anchors. This fits the findings of Spenader (2002).

The first part of the uniqueness condition says that the definite description should be such that it does not allow the hearer to consider entities as possible anchors which the speaker does not intend to be anchors. That is, the set of hearer anchors should be a subset of the speaker anchors.

DEFINITION 6. **(Uniqueness Condition I)** *Property P correctly identifies the anchors of target t in context C iff*

$$\mathsf{dHA}(P, C) \subseteq \mathsf{dSA}(t) \textit{ and } \mathsf{iHA}(P, C) \subseteq \mathsf{iSA}(t, C).$$

The set of speaker anchors only contains entities which are, so to speak, real anchors, because the speaker knows that they are related to the target. By requiring that the set of hearer anchors does not contain any additional entities we ensure that the hearer does not consider any entities as anchors that are not related to the target. By way of illustration consider the context represented in Figure 6a. The property *cook* does not correctly identify the anchors of entity c in that context, since $\mathsf{iHA}(cook, C) = \{r_1, r_2\}$ while $\mathsf{iSA}(c, C) = \{r_1\}$. However, the property $\lambda x.cook(x) \wedge \exists y[restaurant(y) \wedge italian(x) \wedge of(x, y)]$ would correctly identify c's anchors.

The second part of the uniqueness condition requires that it should be plausible to assume that the anchor is related to only one entity which fits the description. What we have to check is whether it follows

(a)
$$\begin{array}{|ll|}\hline C_{shared}: & restaurant(r_1),\ italian(r_1), \\ & restaurant(r_2),\ chinese(r_2) \\ & \forall x[restaurant(x) \rightarrow \exists y[cook(y) \wedge of(y,x)]] \\ & \forall xy[of(x,y) \rightarrow bridge(x,y)] \\ C_{private}: & cook(c),\ of(c,r_1) \\ \hline \end{array}$$

(b)
$$\begin{array}{|ll|}\hline C_{shared}: & book(b), \\ & \forall x[book(x) \rightarrow \exists yz[page(y) \wedge of(y,x) \wedge page(z) \wedge of(z,x) \\ & \qquad \wedge y \neq z]] \\ & \forall xy[of(x,y) \rightarrow bridge(x,y)] \\ C_{private}: & page(p),\ of(p,b) \\ \hline \end{array}$$

Figure 6. Example contexts to illustrate the definition of uniqueness.

from the context that there is more than one entity of which the property holds and which is related to the anchor. If this is the case, then we have evidence that the second uniqueness condition is *not* satisfied and that, therefore, a definite description cannot be used. If, however, the context does not provide any evidence that there is more than one entity which fits the description and is related to the anchor, then it is possible to assume that the target entity is the only such entity. So, we are checking for consistency. We are not requiring that uniqueness with respect to the anchor has to follow from the context; it just has to be consistent with it.

DEFINITION 7. **(Uniqueness Condition II)** *Property P uniquely identifies target t with respect to its anchors in the context C iff*

$$C_{private} \cup C_{shared} \not\models \exists x[P(x) \ \wedge \ bridge(x,a) \ \wedge \ x \neq t]$$

holds for all entities $a \in \mathsf{FA}(t,P,C)$

Given the target c and the property *cook*, this condition is satisfied in context (a) of Figure 6. There is nothing in the context which enforces that r_1 or r_2 necessarily have more than one cook. In Figure 6b, in contrast, the property *page* does not suffice to uniquely identify p with respect to its anchors, since it follows from the shared knowledge that b has more than one page.

6. An Algorithm for Generating Bridging Anaphora

In this section, we show how to extend the base algorithm introduced in Section 2 to bridging and indirect coreferential uses of definites. We illustrate the workings of the algorithm by means of some examples and describe an implementation of it using a description logic reasoning system for carrying out the necessary inferences on the background knowledge.

6.1. *Extending the Standard Algorithm*

We will now present an extension of the base algorithm that generates bridging definites in addition to coreferring definites. The main idea behind this extension is to use the relation between hearer and speaker anchors to control the algorithm. Thus, the proposed algorithm starts with an empty description and keeps adding properties as long as the Familiarity Condition *is* satisfied and the Uniqueness Condition *is not* satisfied. In terms of anchors this means that the algorithm proceeds until the set of hearer anchors is a subset of the set of speaker anchors and while the intersection of the two sets is not empty.

As in the base algorithm the description is represented by a set of literals. To be able to use the definitions of the previous section we have to derive the property that is attributed to an entity by a set of literals. Given a set of literals Γ, let $\bigwedge\Gamma$ be the conjunction of elements of Γ. If $\bigwedge\Gamma$ mentions entities $t_1, \ldots, t_n$ then

$$P(t_i, \Gamma) \;=\; \lambda x_i \exists x_1, \ldots, x_{i-1}, x_{i-1}, \ldots, x_n \bigwedge \Gamma[t_1/x_1, \ldots, t_n/x_n].$$

Given, for instance, the set of literals $\Gamma = \{cook(c), of(c, r), restaurant(r), italian(r)\}$ the property that is attributed to entity c by this set is

$$P(c, \Gamma) = \lambda x_c \exists x_r [cook(x_c) \wedge of(x_c, x_r) \wedge restaurant(x_r) \wedge italian(x_r)].$$

Assuming a given target entity t, we will now also write $\mathsf{HA}(\Gamma, C)$ to mean $\mathsf{HA}(P(t, \Gamma), C)$ and similarly for the direct (dHA) and indirect (iHA) hearer anchors as well as for the familiar anchors (FA).

The search problem for the extended algorithm is shown in Figure 7. As in the original version of the algorithm, states contain a) the description $\mathsf{Description}(i)$ that has been built up to that state and b) the set of those entities mentioned in $\mathsf{Description}(i)$ which have not yet been uniquely identified. Descriptions are sets of literals. In the beginning, the description is empty and the target list only contains the entity

input: a context C consisting of C_{shared} and $C_{private}$
a target entity t

state $s(i)$**:** $\mathsf{Description}(i)$, a set of literals
$\mathsf{Targets}(i)$, a list of target objects

initial state: $\mathsf{Description}(0) = \emptyset$
(state $s(0)$) $\mathsf{Targets}(0) = \{t\}$

goal state: State $s(i)$ is a goal state if
$\mathsf{Targets}(i) = \emptyset$

operator: 1) Select a literal L from $C_{private} \cup C_{shared}$. L has to

1. mention at least one entity in $\mathsf{Targets}(i)$,
2. rule out at least one distracting anchor for at least one entity in $\mathsf{Targets}(i)$ or mentioned in L, and
3. $\mathsf{FA}(a, \mathsf{Description}(i) \cup \{L\}, C)$ should be non-empty for all entities a mentioned in $\mathsf{Description}(i) \cup \{L\}$. (Definition 5).

2) $\mathsf{Description}(i+1) = \mathsf{Description}(i) \cup \{L\}$

3) $\mathsf{Targets}(i+1)$ is derived from $\mathsf{Targets}(i)$ by updating it with L: first, those entities mentioned in L that are not mentioned in $\mathsf{Description}(i)$ are added and then, all those entities for which $\mathsf{Description}(i+1)$ satisfies the uniqueness condition (Definitions 6 and 7) are deleted.

Figure 7. Searching a uniquely identifying description.

that was specified in the input. We have found a solution if the target list is empty. A new search state is computed by the following three steps:

Step 1: Select a literal. Pick a literal L such that $C_{private} \cup C_{shared} \models L$. L has to satisfy the following conditions.

– L has to mention at least one entity t which is an element of $\mathsf{Targets}(i)$.

- The addition of L should rule out at least one distracting anchor for at least one entity mentioned in $\mathsf{Targets}(i)$ or in L.
- The resulting description has to be such that all entities mentioned in this description are familiar given the description. More formally, for all entities t mentioned in $\mathsf{Description}(i) \cup \{L\}$ the set of familiar anchors $\mathsf{FA}(t, \mathsf{Description}(i) \cup \{L\}, C)$ should not be empty.

If there is more than one viable literal, heuristics are used to choose one. As before different heuristics are possible. For instance, literals which rule out the most distractors could be preferred (as in Dale and Reiter's greedy heuristics) or literals could be added according to some predefined order (as in Dale and Reiter's incremental algorithm).

Step 2: Update the description. The literal L that is chosen in the first step is added to the description:

$$\mathsf{Description}(i+1) \;=\; \mathsf{Description}(i) \cup \{L\}.$$

Step 3: Update the target list. All entities mentioned in L which are not elements of $\mathsf{Description}(i)$ are added to $\mathsf{Targets}(i)$. Then all those entities for which $\mathsf{Description}(i+1)$ satisfies the uniqueness condition are eliminated from the list.

6.2. *Examples*

In the examples that follow, we will assume a simple search strategy following Dale and Reiter's incremental algorithm. We will prefer sortal information to any other kind of property, and unary literals to binary ones.

Example 1. In this example, the algorithm builds an implicitly anchored bridging description. Let's assume the following context:

$$\begin{array}{ll} C_{shared}\text{:} & \mathit{restaurant}(r), \\ & \forall x[\mathit{restaurant}(x) \rightarrow \exists y[\mathit{cook}(y) \wedge \mathit{of}(y,x)]] \\ & \forall xy[\mathit{of}(x,y) \rightarrow \mathit{bridge}(x,y)] \\ C_{private}\text{:} & \mathit{cook}(c),\ \mathit{of}(c,r) \end{array}$$

Suppose the goal is to build an expression referring to entity c. Row one of the table below shows the initial search state (description and target list) as well as the hearer and speaker anchors of c. In the first step, the algorithm could add either *cook(c)* or *of(c,r)*. Dale and Reiter's incremental heuristics prefers the former. The set of hearer anchors of c now equals the set of speaker anchors, and hence, the first part of the uniqueness condition is satisfied. The second part of the uniqueness condition is also satisfied, as there is no indication that r has more than one cook. Entity c can therefore be eliminated from the target set, which then is empty. Hence, the algorithm has found a goal state and stops.

	Description	Targets	HA	SA
1.	$\emptyset$	$\{c\}$	c: all entities of C_{shared}	$c : \{r\}$
2.	$\{cook(c)\}$	$\emptyset$	$c : \{r\}$	$c : \{r\}$

Example 2. In this example, the bridging relation has to be made explicit in the description. We assume the following context, which is very much like the context of the previous example, but there are two restaurants now – an Italian one and a Chinese one.

$$\begin{aligned} C_{shared}\!: &\; restaurant(r_1),\ italian(r_1), \\ &\; restaurant(r_2),\ chinese(r_2) \\ &\; \forall x[restaurant(x) \rightarrow \exists y[cook(y) \wedge of(y,x)]] \\ &\; \forall xy[of(x,y) \rightarrow bridge(x,y)] \\ C_{private}\!: &\; cook(c),\ of(c,r_1) \end{aligned}$$

Suppose that the target entity is again c. The search starts out as in the previous example (as shown by rows 1 and 2 of the table below). In contrast to the previous example, though, adding $cook(c)$ to the description does not lead to a goal state. The set of hearer anchors of c still contains both restaurants r_1 and r_2 and is therefore not a subset of the speaker anchors. The only thing that can be added in the next step is $of(c, r_1)$, which adds a new entity, r_1, to the target list. r_1 is of type *restaurant*. However, adding the literal $restaurant(r_1)$ to the description would not reduce the set of hearer anchors for either c or r_1. Assuming that the algorithm prefers literals which rule out distractors, it chooses to add $italian(r_1)$ instead. Now, the set of hearer anchors of both c and r_1 are equal to the respective sets of familiar anchors. It is also consistent with $C_{private} \cup C_{shared}$ to assume that r_1 has only one cook. Hence, both parts of the uniqueness condition are fulfilled.

Note that there is no straightforward way of realizing the description as an English noun phrase. It corresponds to something like *the cook of the Italian ...*, where a noun is missing in the embedded noun phrase. So, again, we see that syntactic constraints should be taken into account when building definite descriptions.

	Description	Targets	HA	SA
1.	$\emptyset$	$\{c\}$	c: all entities of C_{shared}	$c : \{r_1\}$
2.	$\{cook(c)\}$	$\{c\}$	$c : \{r_1, r_2\}$	$c : \{r_1\}$
3.	$\{cook(c), of(c, r_1)\}$	$\{c, r_1\}$	$c : \{r_1, r_2\}$ $r_1 : \{r_1, r_2\}$	$c : \{r_1\}$ $r_1 : \{r_1\}$
4.	$\{cook(c), of(c, r_1),$ $italian(r_1)\}$	$\emptyset$	$c : \{r_1\}$ $r_1 : \{r_1\}$	$c : \{r_1\}$ $r_1 : \{r_1\}$

Example 3. Here is an example in which the algorithm fails because the familiarity condition cannot be satisfied. Consider the following situation:

$$
\begin{aligned}
C_{shared}: \ & book(b), \\
& \forall x[book(x) \rightarrow \exists yz[page(y) \wedge page(z) \wedge of(y, x) \wedge of(z, x) \\
& \qquad \wedge\ y \neq z]] \\
& \forall xy[in(x, y) \rightarrow bridge(x, y)] \\
C_{private}: \ & cockroach(c),\ in(c, b)
\end{aligned}
$$

Imagine that we need to refer to entity c, the cockroach. c is not hearer old, but c is related to a hearer old entity, namely the book b. So, the set of speaker anchors is $\{b\}$. However, the shared knowledge does not include any information which would tell the hearer that all books contain cockroaches or that all cockroaches are related to books. So, there is no literal that could be added to the description while preserving the familiarity condition. This means that it is not possible to build a definite description and the algorithm fails. Other mechanisms have to be used to build an appropriate referring expression.

6.3. *Implementation*

There is a proof of concept implementation of the extended algorithm in the functional language Mozart Oz. It uses RACER, an automated reasoning system for description logics (DL), to carry out the necessary inferences on the discourse context. In this section, we will look at how these inferences are formulated as queries to the DL reasoning system. We will first give a brief introduction to description logics and describe how the different components of the discourse context can be represented as a DL knowledge base. Then, we will show how the functionality provided by typical DL reasoning systems nicely supports the reasoning tasks required by the extended algorithm.

Description Logics

Description logic (DL) is a family of logics in the tradition of knowledge representation formalisms such as KL-ONE (Woods and Schmolze, 1992). DL is a fragment of first-order logic which only allows unary and binary predicates (*concepts* and *roles*) and only very restricted quantification. A knowledge base consists of a *T-Box*, which contains axioms relating the concepts and roles, and one or more *A-Boxes*, which state that individuals belong to certain concepts, or are related by certain roles.

T-Box statements have the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept expressions. Concepts denote sets of individuals and the statement means that denotation of C_1 is a subset of the denotation of C_2, that is, C_2 subsumes C_1. So, we can, for example, write $cook \sqsubseteq human$ to express *cooks are human.* Concepts can be combined by the Boolean connectives $\sqcap$ (and), $\sqcup$ (or), and $\neg$. For example, $animal \sqsubseteq \neg human$. Finally, we can use roles (binary predicates) and their inverses in combination with the quantifiers $\forall$ and $\exists$ to relate two concepts: $restaurant \sqsubseteq \exists part\_of^{-1}.cook$ expresses that every restaurant is related to a cook via an (inverse) *part_of* relation. More expressive DLs furthermore allow number restrictions on roles such as $horse \sqsubseteq (= 4\ part\_of^{-1}).leg$, which means *horses have exactly four legs.*

That was the T-Box. The A-Box contains statements such as $rabbit(a)$ or $love(b, c)$ to express that the individual a is an instance of the concept rabbit, and the individual b and c are related through the role *love.*

We will represent the world knowledge as a T-Box and the discourse model and the speaker's model as A-Boxes.

Theorem provers for description logics support a range of different reasoning tasks. Among the basic reasoning tasks are, for example, *sub-*

sumption checking (Does one concept subsume another?), and *instance* and *relation checking* (Does a given individual belong to a certain concept?/Are two individuals related through a certain relation?). In addition, description logic systems usually provide some *retrieval functionality* which, for example, allows to compute all atomic concepts that a given individual belongs to or all individuals that belong to a given concept. This is very useful for our purposes as retrieval of concepts allows easy access to all properties of an object and instance retrieval provides an elegant way for computing the speaker and hearer anchors.

There is a wide range of different description logics which add different extensions to a common core. Of course, the more expressive these extensions become, the more complex the reasoning problems are. In the last few years, new systems such as FaCT (Horrocks) and RACER (Haarslev and Möller) have shown that it is possible to achieve surprisingly good average-case performance for very expressive (but still decidable) logics. In this paper, we employ the RACER system because it allows for A-Box inferences.

DL Reasoning for the Extended Algorithm

The core reasoning task in the extended algorithm is to compute the set of hearer anchors. In a DL setting, this is straightforwardly implemented by using the instance retrieval mechanism. Recall that this mechanism returns all instances of a given concept. We first create a DL concept which approximates the characterization of hearer anchors given in Section 5. That is, given a target t and a description Γ, we create concepts which approximates the following formulas

$$\begin{array}{cl} P(t,\Gamma) & \text{(direct hearer anchors)} \\ \lambda x.\exists y[bridge(x,y) \wedge P(t,\Gamma)(y)] & \text{(indirect hearer anchors)} \end{array}$$

(Remember that $P(t,\Gamma)$ is the property that is attributed to entity t by the set of literals Γ.) Using instance retrieval, we can then gather all objects belonging to this concept or which are related to an instance of this concept via a bridging relation.

To construct the approximating DL concept from the set Γ of literals representing the semantic content of the definite description, we use the following strategy. Assuming that we want to compute the hearer anchors of object o, we first collect all unary properties of o in Γ and conjoin the predicate symbols to form a concept expression. These properties are deleted from the set Γ. Then, we take one by one the binary properties relating o to some other object o' via a relation R, we (recursively) build a concept expression $C_{o'}$ for o' and conjoin $\exists R.C_{o'}$ with the previously constructed part. For example, for $\Gamma = \{cook(c), part\_of(c,r), restaurant(r), italian(r)\}$ and target c we will build the following DL concept:

$$cook \sqcap \exists part\_of.(restaurant \sqcap italian)$$

The resulting concept is only an approximation of the corresponding first order formula. This is due to the restricted expressive power of DL, which cannot capture reflexivity: $\lambda x(R(x,x))$ would be rendered as $\exists T.\top$. Similarly, if the same two objects are related in two different ways, this information is lost in the DL concept: $\lambda x(\exists y(R_1(x,y) \wedge R_2(x,y))$ becomes $(\exists R_1.\top)\wedge(\exists R_2.\top)$. As we only generate descriptions expressing a set of positive facts about an entity, however, these two are the only cases in which the DL concept is not equivalent to the first order formula corresponding to the description.

To check the second part of the Uniqueness Condition, we employ number restrictions and test that it is consistent with the shared knowledge to assume that the anchor is related to exactly one instance of the DL concept corresponding to the description. For example, to test whether it is *consistent* with world knowledge that an entity a has exactly one page, we test whether the *negation follows* from world knowledge. If so, it is not consistent, otherwise it is. To test the entailment, we send the query "is a an instance of the concept $\neg(= 1\ part\_of^{-1}).page$" to the DL prover.

Finally, we use RACER's functionality for retrieving properties (concepts, most specific concepts, roles) of a given instance to collect all potentially applicable properties.

The inferences that are necessary in our approach to generating definite descriptions could also be carried out by automated theorem provers for first order logic. As first order logic provers do not provide the kind of knowledge base management and retrieval functionality that comes with DL systems, however, one would need additional mechanisms for selecting applicable properties and for retrieving and maintaining the set of hearer anchors.

7. Conclusion

In this paper, we have shown how the basic incremental algorithm for generating definite descriptions proposed by Dale and Reiter, can be extended to handle definite descriptions which depend on world knowledge in such a way that their processing requires knowledge based reasoning. Specifically, we have shown how it can be integrated with reasoning to generate bridging and to a certain extent, indirect coreferential uses of definite descriptions.

But as seen in Section 3, bridging and coreferential uses do not exhaust the usage spectrum of definite descriptions. Larger situation and unfamiliar uses are also very frequent. Provided the context is extended to encode world and situational knowledge, the proposed algorithm naturally extends to larger situation uses – these are just uses where the entity is familiar because it is hearer old.

The unfamiliar class is more problematic. Recall that it includes definite descriptions with sentential complements (*the fact that John's father is bald*) and containing inferables, that is, entities that are familiar by virtue of being related to some discourse old entity (*the man John met yesterday, the Iran/Iraq war*). The first subclass (descriptions with sentential complements) can be viewed as a kind of event anaphora (the speaker is referring to John's father's baldness) and should probably be treated as such. The second case (containing inferables) raises the question of how familiarity should be defined. Conceivably, the definition of hearer anchors could be further weakened to encompass such cases. However, doing so might lead to overgeneration. It would therefore be important to first have a better understanding of the distribution and form of containing inferables and of when explicit bridging is acceptable.

In this paper, we are mainly interested in the knowledge based reasoning necessary for generating inference based definite descriptions. There are other factors which also play important roles for the generation of anaphoric expressions. In particular, our account needs to be supplemented with a notion of salience (such as the one suggested by Krahmer and Theune (2001), for example) and possibly other pragmatic mechanisms for explaining anaphora resolution preferences.

Acknowledgments

This work was partially supported by the Project InDiGen in SPP–Sprachproduktion, grant by the Deutsche Forschungsgemeinschaft to the University of Saarbrücken and by the Lorraine Region within the project "Ingénierie des langues, du document et de l'Information scientifique Technique et Culturelle" of the Plan Etat-Région 2000-2004.

References

Clark, H. H.: 1977, 'Bridging'. In: P. N. Johnson-Laird and P. C. Wason (eds.): *Thinking: Readings in Cognitive Science*. Cambridge University Press, Cambridge.

Dale, R.: 1989, 'Cooking up referring expressions'. In: *Proc. of the 27th ACL*. pp. 68–75.

Dale, R.: 1992, *Generating Referring Expressions: Building Descriptions in a Domain of Objects and Processes*. MIT Press.

Dale, R. and N. Haddock (1991). Content Determination in the Generation of Referring Expressions. *Computational Intelligence 7*(4), 252–265.

Dale, R. and E. Reiter: 1995, 'Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions'. *Cognitive Science* **18**, 233–263.

Gardent, C., H. Manuélian, and E. Kow (2003). Which bridges for bridging definite descriptions? In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*.

Haarslev, V. and R. Möller, 'RACER: Renamed ABox and Concept Expression Reasoner'. `http://www.fh-wedel.de/~mo/racer/index.html`.

Hawkins, J. A.: 1978, *Definiteness and Indefiniteness*. London: Croom Helm.

Heim, I.: 1982, 'The Semantics of Definite and Indefinite Noun Phrases'. Ph.D. thesis, University of Massachusetts.

Horacek, H.: 1997, 'An Algorithm for Generating Referential Descriptions With Flexible Interfaces'. In: *Proc. of the 35th ACL*. pp. 206–213.

Horrocks, I., 'The FaCT System'. `http://www.cs.man.ac.uk/~horrocks/FaCT/`.

Kleiber, G. (1997). Des anaphores associatives méronymiques aux anaphores associatives locatives. *Verbum 1–2*, 25–66.

Krahmer, E. and M. Theune: 2001, 'Efficient context-sensitive generation of referring expressions'. In: K. van Deemter and R.Kibble (eds.): *Information Sharing: Givenness and Newness in Language Processing*. CSLI Publication.

Poesio, M. and R. Vieira: 1998, 'A Corpus-based Investigation of Definite Description Use'. *Computational Linguistics* **24**(2), 183–216.

Prince, E.: 1981, 'Towards a taxonomy of given-new information'. In: P. Cole (ed.): *Radical Pragmatics*. New York: Academic Press, pp. 223–256.

Russell, B.: 1906, 'On denoting'. *Mind* **14**, 479–493.

Spenader, J. (2002). *Presupposition in Spoken Discourse*. Ph. D. thesis, Stockholm University.

Stone, M.: 1998, 'Modality in Dialogue: Planning, Pragmatics and Computation'. Ph.D. thesis, Department of Computer & Information Science, University of Pennsylvania.

Woods, W. and J. Schmolze: 1992, 'The KL-ONE Family'. *Computer and Mathematics with Applications* **23**(2–5).