Are computers intelligent? Will they ever be?

Write down a short paragraph (15min) on whether or not you think that computers can think/are intelligent and why you think so.

I want you to hand those paragraphs in, but you don't need to put your name on them.

How to contact me

Kristina Striegnitz

- email: striegnk@union.edu
- office hours: Steinmetz 233
  Mon 11:45-12:45
  Wed 1-2
  Fri 2-3
- phone: 518 388 6554

About this course

- Intro to computer sciences

  Studies theoretical foundations, implementation and applications of computing.
- Intro to programming

  An important tool for computer scientist.
- Intro to artificial intelligence

  Study and design of intelligent systems.

Course Website

http://antipasto.union.edu/~striegnk/courses/cancomputersthink

Getting your D4NEW accounts set up

- Log on using your Union College user name and the word '*union*' followed by your Union College user name. For example: `striegnk` and `unionstriegnk`. This is your *samba password*.
- Use ssh to log onto antipasto.union.edu. The password is the word '*union*' followed by your Union College ID number. For example: `union2172678`. This is your *linux password*.
- Type `smbpasswd` to change your samba password.
- Then type `yppasswd` to change your linux password.
- It may be a good idea to change them to be the same.

Pair and Group Work

- Work TOGETHER. - Discuss ideas, explain solutions, ask questions.
- Make sure everybody participates and contributes. - Do both talking and listening. Take turns at the keyboard.
- Don't always work with the same partner. I will often randomly assign partners.
- Grading: 0, 1, or 2 points.

ELIZA

Eliza is a program that simulates a psychotherapist. Go to this online version http://www.chayden.net/eliza/Eliza.html and try it out.

Discuss the following questions and come up with answers that all members of your group can agree on:

- Is the program convincing? Why or why not?
- Try to test the limits. What works; what doesn't?
- Is Eliza intelligent? Why or why not?

Write down brief answers to these questions and hand them in after all members of your group have signed this paper.

Discussion Rules

- Be polite.
- The current speaker chooses the next speaker.
- The current speaker gets to decide whether to continue the current topic or to change the direction of the discussion.
- The overall goal of the discussion should be to find answers to the discussion questions.

Discussion Questions

- What features of intelligence does Eliza have? What is missing?
- What properties does an intelligent system need to have?

How does ELIZA work?

Can you see patterns in the way it answers?

Can you write down some rules that predict how ELIZA will react to certain types of input?

Algorithm

An algorithm is a list of instructions for solving a task.

ELIZA's Algorithm

- ELIZA contains a list of sentence patterns associated with reassembly rules. E.g.

```
(X1 you X2 me) => (what makes you think I X2 you)
                  (why do you think I X2 you)
                  (you like to think I X2 you – don't
                                                  you)
```

- Given the user input, ELIZA finds a pattern that matches the input and chooses one of the reassembly rules to build a reply.

```
it seems that you hate me
                    => what makes you think I hate you
it seems that you like making fun of me
          => why do you think I like making fun of you
```

Algorithm

An algorithm is a list of instructions for solving a task.

Algorithms - history

- 1900. Hilbert's 10th problem: Devise *a process, which can determine by a finite number of operations* whether a given mathematical equation (of a certain kind) has a solution (of a certain kind).
- This triggers interest in algorithms (processes which answer a problem by a finite number of operations).
- 1936: Alan Turing, Alonzo Church, and Emil Post propose formalizations of the notion of algorithm.
- They also show: there are problems which cannot be solved algorithmically.

Alan Turing

Peanut-Butter-Jelly-Sandwich Algorithm

Write down an algorithm for making a peanut butter jelly sandwich.

PBJ Algorithm (version 1)

1) Spread peanut butter on a slice of bread.
2) Spread jelly on top.
3) Put a second slice of bread on top.

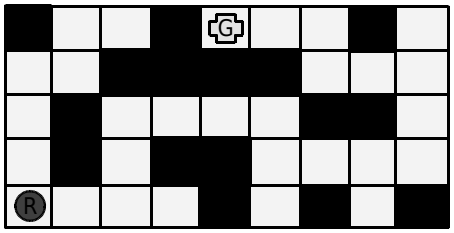
PBJ Algorithm (version 2)

1) Get two slices of bread, a jar of peanut butter, a jar of jelly, and a knife.
2) Open the jar of peanut butter.
3) Use the knife to scoop up some peanut butter from the jar.
4) Spread the peanut butter on one slice of bread.
5) Open the jar of jelly.
6) Use the knife to scoop up some jelly from the jar.
7) Spread the jelly onto the slice of bread on top of the peanut butter.
8) Put the second slice of bread on top.

PBJ Algorithm (version 3)

1) Get two slices of bread, a jar of peanut butter, a jar of jelly, and a knife.
2) Take the jar of peanut butter into one hand and grab the lid with the other hand.
3) Rotate the lid of the jar of peanut butter counter-clockwise until you can lift it off.
4) Put the lid aside and take the knife instead.
5) Insert the knife vertically into the jar of peanut butter.
6) Tilt the knife about 30°.
7) Lift the knife out of the jar.
8) ...

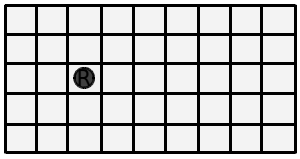
Robot-in-a-Maze Algorithm



Write down an algorithm that guides the robot through the maze to the goal.

Primitives that the robot understands: "go one step to the north/east/west/south"

Wall-following-Robot Algorithm



Primitives:

- "go one step to the N/E/S/W"
- "check whether there is a wall to the N/E/S/W"

1) Write down an algorithm that guides the robot to the north-east corner. (no matter where the robot starts in the grid)
2) Write down an algorithm that first guides the robot to a wall and then makes it walk along the wall.

In both cases your algorithm should work for any starting position of the robot.

Self-check

Can you say what the following are?

- algorithm
- primitives
- conditional/if statements
- loops

Self-check

Can you say what the following are?

- Python interpreter
- interactive Python shell
- objects/values (integers, floats, strings)
- expressions
- operators (+, -, /, *)
- [functions (type, float, int, len, str)]

Turing – Computing Machinery and Intelligence

Discussion questions:

- Is the imitation game proposed by Turing a good test for determining whether a machine is intelligent?
- Could a machine pass the test without being intelligent?
- Could the test fail to detect intelligence?

Review - Algorithms

- Algorithms: a list of instructions for solving a task.
- The primitives depend on who we are talking to (PBJ example) or what programming language we use.
- Control flow is from top to bottom, unless changed by control-flow primitives like if-statements or loops (robot examples)
- Programming languages: languages for specifying algorithms such that a computer can "understand" them.

Exercise – print statements

1) Write a program that prints a pyramid of *x*'s onto the screen.

```
   x
  xxx
 xxxxx
xxxxxxx
```

2) Write an algorithm in English (not Python) that asks the user for an odd number *n* and then "draws" a pyramid with *n* *x*'s at the bottom. You will need a loop for this algorithm.

Exercise – variable assignment

1) Start out with the following variable assignments.

```
x = 1
y = 0
```

Then write a sequence of statements that switches the values of `x` and `y`. That is, at the end `x` should refer to value 0 and `y` to value 1.

Exercises

1) Write a program that asks the user to type in the radius of a circle (a number) and then uses this number to calculate area and diameter of the circle and prints out the result. (Hint: the math module – `import math` – provides a name referring to the value of pi – `math.pi`.)
2) Write a program that calculates the cost per square inch of a circular pizza, given its diameter and price.
3) Write a program that determines the distance to a lightning strike based on the time elapsed between the flash and the sound of thunder. The speed of sound is approximately 1100 ft/sec and 1 mile is 5280 ft.
4) A coffee shop sells coffee at $10.50 a pound plus the cost of shipping. Each order ships for $0.86 per pound + $1.50 fixed cost for overhead. Write a program that calculates the cost of an order.