TDT4136 Logic and Reasoning Systems

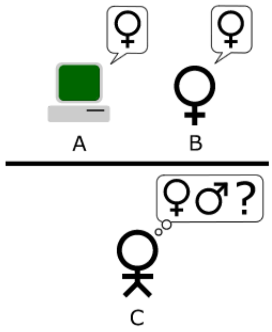
Jørgen Grimnes
Assignment 1

Fall 2013

1 What is Turing test, how it is conducted?

The Turing test is a traditional artificial intelligence test to separate a computer program from a human. The test was first introduced by Alan Turing in 1950 in *Computing Machinery and Intelligence*[1]. The test is executed by three parts: a computer, a human and a interrogator. The computer will pass the test if the interrogator isn't able to differ between the program and the human being.

Figure 1: An illustration of the Turing test.



2 What is the relationship between thinking rationally and acting rationally? Is rational thinking an absolute condition for actig rationally?

Whilest *thinking rationally* is a concept we humans favor, it creates tremendous algorithmic challenges to represent knowledge in a generic manner. Rational thinking could be refered to as logic(-al thinking). *Acting rationally* refers to reaching your goal and acting through ones beliefs. To reach a conclusion to act upon, we would use logical interference. There is still situations where such extensive meditation would be overkill: if you are about to lose you footing on a narrow ledge, you do not have time to ponder over your options. Rational thinking is thus not always a necessity to act rationally.

3 What is Tarski's "theory of reference" about?

The *Theory of Reference* describes how to link objects in logic to real world objects and instances.

```
Chair( X ) # refers X to a chair
Straw( Y ) # refers Y to a straw
```

4 Describe rationality. How is it defined?

Rationality is an action or belief that follows reason or logic. A rational decision should not only be a decision based upon correct reasoning, but also the most optimal solution to solve the given task.

[1]http://orium.pw/paper/turingai.pdf

5 Consider a robot of which task is to cross the road. Its action portfolio looks like this: look-back, look-forward, look-left-look-right, go-forward, go-back, go-left and go-right.

While crossing the road, a helicopter falls down on the robot and smashes it. Is the robot rational? Yes, there is no way neither humans nor computers could take everything into account. It is not rational to check with NASA each time we're crossing the street.

While crossing the road on a green light, a passing car crashes into the robot preventing it from crossing. Is the robot rational? This is debatable. We are teaching children to be observant when crossing the road even when their perceptions (the green *walking man* lights up) tells them they may act as intended. If this exercise is refering to a green light for the car, then the robots is not rational.

6 Consider the vacuum cleaner world described on page 38 of the textbook. Let us modify this vacuum environment so that the agent is penalized 1 point for each movement.

Can a simple reflex agent be rational for this environment? Explain your answer. Almost. For such a simple environment it would be rational. If the current location is dirty, the preceptions will lead the robot to decide to *suck*. Whilst if the location if clean, the robot would move to the next location. Still a problem occurs concerning movement. A simple reflex agent could be stuck in a infinite loop be trying to move through the wall.

Can a reflex agent with state be rational in this environment? Explain your answer. Yes, a reflex agent with a model or state representation would also be rational. If the program is able to track the previous states of the environment, it would be able to keep track of how partial environments and thus be able to act if it's sensors are malfunctioning too.

Assume now that the simple reflex agent (i.e., no internal state) can perceive the clean/dirty status of both locations at the same time. Can this agent be rational? Explain your answer. In case it can be rational, design the agent function. Given that the problem described in *a)* may not occur, then yes.

```
def agent_function( ):
        if percept = left_dirty:
                if wall_left: suck( )
                elif not right_dirty: # prevent inf. loop
                        move_left( )
        elif percept = right_dirty:
                if wall_right: suck( )
                else: move_right( )
```

7 Consider the vacuum cleaner environment shown in figure 2.3 in the text- book. Describe the environment using properties from chapter 2.3.2, e.g. episodic/sequential, deterministic/stochastic etc. Explain selected values for properties in regards to the vacuum cleaner environment.

Percept sequence	Action
[A, *Clean*]	*Right*
[A, *Dirty*]	*Suck*
[B, *Clean*]	*Left*
[B, *Dirty*]	*Suck*
[A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮
[A, *Clean*], [A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

This environment is episodic (the outcome of a action may not differ from a previous experience), partially observable (due to limited percepts), deterministic (if the dirty may not move), dynamic (as the agent cleans the world), descrete and involves a single agent.

8 Discuss the advantages and limitations of four basic kinds of agents:

Simple reflex agents are very simple to implement, but these agents often turn out to be dumb and there are few worlds where these agents may act rationally.

Model-based reflex agents keeps records of how the world may be[2] evolving. This is the most effective way to take partial observability into account, though it introduces uncertainty. We do not know exactly how the world looks, but the agent makes an educated guess and acts thereby.

Goal-based agents applies searching and planning to find a set of actions to reach its goal. By advancing the agent to plan its approach to solve the task, the agent function will be more generalized and able to withstand noise in the environment based on experience. The disadvantage from the previous agents is the efficiency and complexity.

[2] The agent only keeps a *model* of how the world is.

Utility-based agents applies searching and planning to find *an optimal set* of actions to reach its goal. It does so by applying a *utility function* and rating the different actions with a "happyness index".