

# Exercisesheet No.1

Alexander Diete      Magnus Müller      Martin Pfannemüller

September 26, 2014

## Exercise 1

The agent is rational because it ends in a finite state. At most the agent uses three operations to behave correctly. To prove the rationality in this simple case it is possible to enumerate all possible outcomes of the world:

- Case: both squares are dirty
  - Robot starts left: *clean, right, clean*
  - Robot starts right: *clean, left, clean*
- Case: left square is dirty
  - Robot starts left: *clean, right*
  - Robot starts right: *left, clean*
- Case: right square is dirty
  - Robot starts left: *right, clean*
  - Robot starts right: *clean, left*
- Case: no square is dirty
  - Robot starts left: *right*
  - Robot starts right: *left*

By definition a rational agent is supposed to maximize its performance measure. It is easy to see that the agent always acts correctly. Thus, it maximizes its performance and there is no agent that can do better. Therefore we can sufficiently say the agent is rational.

## Exercise 2

a)

No it is not a rational agent as it doesn't stop once every square is cleaned. Instead it keeps moving, thus earning penalty points by every move it takes.

b)

The cleaner stops once it visited all squares as it has a map of the squares. As there is no cleaner that performs better under this environment it is a rational cleaner. (The only assumption is that there exists no cleaner that knows the distribution of the dirty squares a priori)

c)

the agent knows the state of the environment then the agent knows exactly what to do next. For the agent there is a mapping of actions to each of the eight states. Thus this simple reflex agent is rational in this environment. As the model-based agent/ reflex agent with state is a more sophisticated agent than the simple reflex agent it is rational as well.

### Exercise 3

a)

No, the simple reflex agent will be trapped in a corner forever once it hits a wall.

b)

Yes, the randomized agent can outperform the simple reflex agent. The agent function could look like this:

**function** randomizedAgentFunction (percept) **returns** an action

**if** *dirty* **then**

        return *clean*

**else**

        return *randomDirection*

**end if**

Eventually the agent using this function will clean all dirty squares.

c)

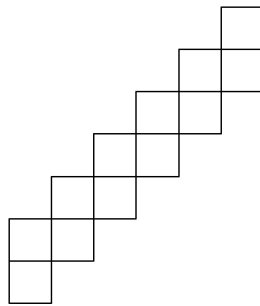


Figure 1: Hard environment

The randomized agent will perform poor in environments with many corners and small paths or in general environments with many edges where you can not go.

## Exercise 4

a)

*There exists task environment in which no pure reflex agent can behave rationally.*

This statement is **true**. We consider for instance the vacuum-cleaner example with an environment that is not fully observable and also consists of obstacles and other traps. A reflex based agent is not able to act rationally without having some sort of state.

b)

*There exists task environment in which every agent is rational.*

This statement is **false**. For every environment an agent is able to behave the exact opposite of what it is supposed to do.

c)

*The input to an agent function is the same as the input to the agent problem.*

This statement is **false**. The agent function may be working on a partially solved problem and may be called multiple times in the program. The input of the agent problem on the other hand is the starting point of the agent.

d)

*There can be more than one agent program that implements a given agent function.*

This statement is **true**. The agent function is a formal description mapping any percept sequence to an action. There can be multiple different implementations of this formal description.

e)

*An agent that senses only partial information about the state cannot be perfectly rational.*

This statement is **false**. Missing (sensorial) information can e.g. be compensated by saving an internal state. This means, although an agent cannot sense every information needed, it can act rationally.

## Exercise 5

We can simply solve this exercise by using truth tables and counting the positive Results. It is important to note, that if a Variable is not mentioned in the preposition, its value is not important for the model and can be either **true** or **false**.

a)

$$(A \wedge B) \vee (B \wedge C)$$

A	B	C	Result
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	1
1	1	1	1

We thus have 3 positive outcomes in the truth-table. As D is not considered in the propositions, its value is not important and can be either true or false. This doubles the result-size to **6 possible models**.

b)

$$A \vee B$$

A	B	Result
0	0	0
1	0	1
0	1	1
1	1	1

This preposition also has three positive outcomes but in contrast to a) both C and D are not bound in the formula. Therefore any combination of assignment of C and D is valid. This results in  $3 \cdot 2^2 = \mathbf{12}$  possible models

c)

$$A \Leftrightarrow B \Leftrightarrow C$$

A	B	C	Result
0	0	0	1
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	1

The results are similar to *a)* as D is unbound. This results in  $2 \cdot 2 = 4$  **possible models**.

d)

$$A \wedge B \wedge \neg D$$

A	B	D	Result
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

The results are similar to *a)* or *c)*. But in this case C is unbound. This results in  $1 \cdot 2 = 2$  **possible models**.

## Exercise 6

a)

If  $\alpha$  is valid, then it is true. Therefore true always is a subset of/entails  $\alpha$  (see definition).

b)

If  $\alpha$  is false, then  $\text{false} \models \text{false}$  which is obviously true. If  $\alpha$  is true, then  $\text{false} \models \text{true}$  which is true as well. Thus the assertion is true.

c)

$\alpha$	$\beta$	$\neg\alpha \vee \beta$	$\alpha \subseteq \beta$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

It follows from the table that the assertion is true.

d)

$\alpha$	$\beta$	$\alpha \Leftrightarrow \beta$	$\alpha \equiv \beta$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

It follows from the table that the assertion is true.

e)



$\alpha$	$\beta$	$\alpha \wedge \neg \beta$	$\neg \alpha \vee \beta$	$\alpha \models \beta$
0	0	0	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	1	1

It follows from the table that the assertion is true.