Multiagent Systems

Prof. Dr. Bernhard Nebel — University of Freiburg
Dr. Christian Becker-Asano, Dr. Stefan Wölfl — Department of Computer Science
Summer term 2014

Exercise Sheet 2

Due: Friday, May 16, 2pm

Important: Each exercise sheet is to be solved in groups of **two students**. Thus, please note your names on each solution sheet and, if applicable, in the source code (as a comment on top of each source file).
After SVN-accounts have been created for you, the solutions are to be handed in as pdf or plain text files (UTF-8 encoded). We strongly suggest the use of LaTeX for typesetting your solutions.
You might complete your solutions in English or German.

Exercise 2.1 (Jason: Ping Pong agents; 2 points)

Model two agents `pingy` and `pongy`, which behave in the following way: When agent `pingy` sends a `ping` message containing an integer `counter` to the agent `pongy`, the agent `pongy` increases the `counter` by one and sends it back as an argument of his message `pong`. This is then used by `pingy` to update its own `counter` accordingly before sending the next `ping` message after some random delay.

Exercise 2.2 (Jason: Domestic Robot Supermarket extension; 2 points)

Improve the code of the Supermarket agent of the “Domestic Robot” example so that it manages its stock. Initially, it has, for instance, 100 bottles of beer and this value will decrease as it delivers beer to the robot. Of course, it can only deliver beer when it has enough beer in its stock, otherwise it should inform the robot that it has no more beer in its stock.

Exercise 2.3 (Jason, Domestic Robot order of plans; 2 points)

In the robot code, we have two plans for the goal `!at`. What happens if we change their order (`m2` before `m1`)? What happens if the second plan does not have a context, as in the following code?

```
@m1 +!at(robot,P) : at(robot,P) <- true.
@m2 +!at(robot,P) : true <- move_towards(P); !at(robot,P).
```

Is the behavior of the robot the same as with the original code? In this case, what happens if we change the order of the plans?

Exercise 2.4 (Jason, Domestic Robot economics; 4 points)

Create a new supermarket agent, initially with the same code as the supermarket of the example. Change the code of both supermarkets such that, once they have started, they send the following message to the robot, informing of their price for beer.

```
.send(robot,tell,price(beer,3))
```

The robot should then buy beer from the cheapest supermarket. To help the robot coding, write a (Prolog-like) rule that finds the cheapest supermarket according to the robot's belief base. Note that the robot will always have two beliefs of price from different sources (the two supermarkets).