

Foundations of Agent Programming

Prof. Michael Thielscher, Sebastian Voigt

International Master Program in Computational Logic — summer term 2009

25.06.2009

Exercise 8.1

Consider a room which has only one door and contains three agents: a claustrophobic (`claus`), a paranoid (`para`) and a porter (`porter`). `claus` always wants the door to be unlocked, `para` always wants it to be locked and only `porter` is able to lock and unlock the door upon requested to do so. `claus` is issuing a request for unlocking as soon as the door is locked, `para` does it the other way around. The exercise is supposed to lead to a Jason encoding of the given domain.

- a) Download the AgentSpeak interpreter Jason via the link which is provided on the web page. Furthermore download the files `Room.mas2j` (the project file) and `RoomEnv.java` (the Java file which simulates the environment) and create three empty files with the names `claus.asl`, `para.asl` and `porter.asl`, respectively.
- b) Specify procedures in the agent `porter` such that the triggering event `!locked(door)` results in the door being locked. Similarly the triggering event `!~locked(door)` should cause an unlocked door.
- c) Specify procedures in the agent `claus` such that getting to know a locked door causes the agent to ask the porter to unlock the door. Agent `para` should send a similar request.
Hint: The internal action `.send(ag,achieve,goal)` causes a triggering event `!goal` in the agent with name `ag`.
- d) Use the internal action `.print(String)` to let `para` thank the porter with some printed message as soon as the door is locked. `claus` should do so similarly when it gets unlocked.

Exercise 8.2

Consider the AgentSpeak Gold Mining Domain from the lecture and solve the following tasks with regard to SPARK.

- a) Translate the three AgentSpeak procedures from Slide 6a/16 to suitable SPARK procedures.
- b) Specify three further procedures concerning the triggers `Do(Pick(Gold))`, `Do(Move(x,y))` and `Do(Drop(Gold))`, respectively.
- c) Construct corresponding finite state machines for the procedures specified in a) and b).