

Reasoning Agents

Prof. Michael Thielscher, Sebastian Voigt

International Masters Programme in Computational Logic — summer term 2009

10.07.2009

Assignment 5.1

Consider a two dimensional grid environment and a vacuum cleaner agent with the following actions and percepts:

left, right, up, down	$\hat{=}$	move one position left (right, up, down, respectively)
suck	$\hat{=}$	clean the current location
dirty, clean	$\hat{=}$	the current position is dirty (clean, respectively)
pos(X)	$\hat{=}$	the agent currently is at position $X \in \{1, 2, 3, 4\}$ (1 - upper left, 2 - upper right, 3 - lower left, 4 - lower right)

As from time to time positions become dirty, the (endless) task of the agent is to keep the environment clean. This scenario should be programmed using Jason.

- Download Jason and the vacuum cleaner project files via the links on the course web page and run them: you should get some graphic output window representing the environment.
- Specify procedures for the triggering event `!clean_house` such that the agent cleans its current position if it is dirty and moves one step otherwise. The aim to have a clean environment should be triggered initially (by stating `!clean_house.`) and be pursued forever.
- Implement simple move procedures which cause the agent to run in circles.
- Change the file `VCWorld.java` such that the grid is enlarged to 5×5 and does not have dirty positions initially. Moreover dirt should also be generated at the new positions and moving to them should be possible.

Hints:

- Change the global array variable `dirty` which contains the state of the environment.
 - Have a look at constructor `VCWorld()` for dirt generation.
 - The public method `executeAction(...)` takes care of movement possibilities.
- Change the private method `createPercept()` in `VCWorld.java` such as to issue a percept `pos(X,Y)` (instead of `pos(X)`) if the agent is at position (x,y) currently. Adapt the move procedures of the agent accordingly.
 - To enable a more involved behaviour, the agent should now get to know a new dirty position via percept `dirtAt(X,Y)`. Add a private method `createDirtPercept(int dirtX,int dirtY)` to `VCWorld.java` that issues this percept. Call the method each time after generating a new dirty position.

Make sure the agent remembers new dirty positions: add a procedure which is triggered by this new percept and adds an atom `remDirt(X,Y)` to the belief base of the agent. Also account for deleting this belief atom as soon as position (x,y) is cleaned by the agent.

- Change the move procedures of the agent such as to move one step towards the closest position that has been marked via `remDirt(X,Y)`. If there is no such position, the agent should head for $(2,2)$ and wait there until he gets to know another dirty position.