



2nd Practical Assignment in Artificial Intelligence (WS 2006/2007)

Issued: November 7, 2006

Due: November 21, 2006

Submit your solutions electronically to your tutor (via email). Groups of up to four persons are allowed.

The following exercises use the **AIchhörnchen** environment. The environment defines a game board with fields, which may be occupied by trees or nuts. The agents are represented by squirrels, which can be moved from one field to another (which is associated with a specific cost).

You can find information about the environment and how to set it up in the **AIchhörnchen** manual¹. This can be found on the lecture webpage². If you encounter any problems you can ask for help in the lecture forum³.

To program an agent some constraints have to be fulfilled. An agent needs to have a lambda function. The lambda function is called by the client who passes the output from the server. The return value of this lambda function is the move calculated by the agent. Here is a template⁴ of an agent lisp file:

```
(let (
  (variable1 "Hello")
  (variable2 12345)
  ...
)

(labels (
  (example-function1 (x y) (+ x y))
  (example-function2 (x y) (- y x))
  (calculate-next-move ...)
  ...
)

#' (lambda (output-from-server)
  (example-function1 variable2 4321)
  (example-function2 1 3)
  ...
  (calculate-next-move)
)))
```

First the variables (here: variable1 and variable2) are defined within **let**. Then the functions (here: example-function1+2, calculate-next-move) are defined within **labels**. The lambda

¹<https://xantippe.cs.uni-sb.de/~dudel/files/manual.pdf>

²<http://www.ags.uni-sb.de/~omega/teach/KI0607/>

³<http://www.ags.uni-sb.de/~schiller/phpBB2>

⁴This template can also be downloaded from <https://xantippe.cs.uni-sb.de/~dudel/files/agent-template.lisp>

function is called by the client. The variable `output-from-server` is a s-expression with the output the server sends to the agent. The lambda function is like the main function in Java or C++ Programs. In this function you can call your functions defined in `labels` to calculate the next move. The last expression within the lambda function is the return value.

Important is that after the nut has been found by the agent and the server outputs STOP your agent has to output the final path it has found beginning with the field it started on and ending with the field where the nut lies, e.g. (K F S T P U).

Exercise 2.1 (*Uniform Cost Search*) **(60 P)**

Define an agent that implements uniform cost search using the `AIchhörnchen` environment. By default, the environment provides a game field with exactly one nut. Your task is to write an agent that finds the path to the nut with the smallest cost.

Fields are represented by random letters. Your agent can see the fields which are adjacent to the field it stands on. However, it does not see the coordinates of these fields. Your agent can jump to all fields adjacent to already explored fields, thus your agent has to maintain a list of the unexplored fields it can jump to. Once your agent has tried to move on a field that contains a tree this will cause a “soft error”. If it tries again to access such a field this will raise a hard error.

To configure the client which loads your agent file open `main.lisp` in the `lisp-client` directory with an editor and adjust the variables in the first couple of lines. Important is that `gametype` is set to `"uis"`. To let the client load your own agent set `agent` to the name of your agent.

Exercise 2.2 (*A* Search*) **(40 P)**

Now we consider informed search. To change the gametype to informed search set the variable `gametype` in `main.lisp` to `"is"`. Positions on the game board are identified by their coordinates, and the position of the nut is known. Write an agent that implements A* search with a heuristic that guarantees the optimality of the solution found by your agent. Use the graphical user interface to compare the A* algorithm with the Uniform Cost algorithm. Try both algorithms with different maps and different seeds (which can be changed in the configuration file of the map). Which observations can you make with regard to the number of explored fields?