Assignment 2 - Adversarial Search

Artificial Intelligence

WS 2015

Due: 30th November 2015, 23:55

General Information

Before you start, **read the whole assignment sheet**! Make sure you download the game framework version released for the second assignment from the moodle site. Empty classes and methods of the algorithms to be implemented are available in the package `at.jku.cp.ai.search.algorithms`. Please write your code into these dummy classes.

To hand in your solutions, please create a **ZIP** file containing your implementations and the report, as a **PDF** file for the non-programming assignments. The **ZIP** file should contain the following items, **and nothing else**.

```
assignment2
├── report.pdf
└── src
    └── at
        └── jku
            └── cp
                └── ai
                    └── search
                        └── algorithms
                            ├── MinMaxSearch.java
                            └── AlphaBetaSearch.java
```

Make sure that the **ZIP** file also contains the whole directory structure, so when you unzip it, it unzips into a folder named `assignment2`, containing the rest of the tree outlined above. Upload the **ZIP** file via MOODLE.

You can test your implementations on a set of unit tests available in the package `at.jku.cp.ai.tests.assignment2`. Note however that passing all these tests does not necessarily mean that your implementation has no flaws, since the tests do not cover all possible invariants! You are welcome to create your own tests and submit them together with the rest of your code (these would go in an optional package `at.jku.cp.ai.tests.assignment2` in the **ZIP** file)

1 Adversarial Search

Implement the following adversarial search algorithms within the provided framework. **For the sake of simplicity, do not avoid expanding already visited nodes**. Refer to the source code comments in the empty classes (and their parents) of the algorithms for detailed information on their parameters and return values.

(A) MinMax Search (8 points)
`at.jku.cp.ai.search.algorithms.MinMaxSearch`

(B) Alpha-Beta Search (8 points)
`at.jku.cp.ai.search.algorithms.AlphaBetaSearch`

2 Logic

Choose the appropriate inference method (resolution, forward chaining, backward chaining) for the following exercises. **Show all necessary steps** to prove the statements, and if you believe you cannot prove one of them (or any of them), give a detailed explanation why.

Given this ancient knowledge base about unicorns:

> If the unicorn is magical or goes sailing on rainbows, then it can plant rainbow seeds. If the unicorn is mythical, then it is immortal and goes sailing on rainbows, but if it is not mythical, then it is just a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Can you prove that...

(A) ...the unicorn goes sailing on rainbows? (2 Points)

(B) ...the unicorn can plant rainbow seeds? (2 Points)

3 Logic and Games

Choose the appropriate inference method (resolution, forward chaining, backward chaining) for the following exercises. **Show all necessary steps** to prove the statements, and if you believe you cannot prove one of them (or any of them), give a detailed explanation why.

Given the following level:

U 0,0	0,1
1,0	1,1
2,0	S 2,1
F 3,0	3,1

$U_{i,j}^{(t)}$ stands for:
"Unicorn at position i, j, at time t"

$S_{i,j}^{(t,f)}$ stands for:
"Seed at position i, j, at time t, with fuse f left"

$R_{i,j}^{(t)}$ stands for:
"Rainbow at position i, j, at time t"

$F_{i,j}$ stands for:
"Fountain at position i, j"

$Right_{i,j}/Left_{i,j}/Up_{i,j}/Down_{i,j}/Stay_{i,j}$ stand for:
"Every direction is possible on each square"

Example: if your knowledgebase contains:
$U_{0,0}^{(0)}$ and $Right_{0,0}$
then you can conclude $U_{0,1}^{(1)}$ via rule 9.

And this knowledge base:

1	$U_{0,0}^{(0)}$	
2	$S_{2,1}^{(0,2)}$	
3	$F_{3,0}$	
4	$Right_{i,j}$	
5	$Left_{i,j}$	
6	$Up_{i,j}$	
7	$Down_{i,j}$	
8	$Stay_{i,j}$	
9	$U_{i,j}^{(t)} \wedge Right_{i,j} \Rightarrow$	$U_{i,j+1}^{(t+1)}$
10	$U_{i,j}^{(t)} \wedge Left_{i,j} \Rightarrow$	$U_{i,j-1}^{(t+1)}$
11	$U_{i,j}^{(t)} \wedge Up_{i,j} \Rightarrow$	$U_{i-1,j}^{(t+1)}$
12	$U_{i,j}^{(t)} \wedge Down_{i,j} \Rightarrow$	$U_{i+1,j}^{(t+1)}$
13	$U_{i,j}^{(t)} \wedge Stay_{i,j} \Rightarrow$	$U_{i,j}^{(t+1)}$
14	$S_{i,j}^{(t,f)} \Rightarrow$	$S_{i,j}^{(t+1,f-1)}$
15	$S_{i,j}^{(t,0)} \Rightarrow$	$R_{i,j}^{(t)}$
16	$S_{i,j}^{(t,0)} \Rightarrow$	$R_{i,j+1}^{(t)}$
17	$S_{i,j}^{(t,0)} \Rightarrow$	$R_{i,j-1}^{(t)}$
18	$S_{i,j}^{(t,0)} \Rightarrow$	$R_{i-1,j}^{(t)}$
19	$S_{i,j}^{(t,0)} \Rightarrow$	$R_{i+1,j}^{(t)}$
20	$U_{i,j}^{(t)} \wedge R_{i,j}^{(t)} \Rightarrow$	$Sailing$
21	$U_{i,j}^{(t)} \wedge F_{i,j} \Rightarrow$	$Fountain$
22	$Sailing \wedge Fountain \Rightarrow$	$False$

There is **one additional assumption** that comes with this knowledge base: if **one or both** of the indices i, j on any new fact would take on a value that takes you out of the level, this new fact is **not** part of the knowledge base!

Can you provide a proof that:

(A) ... unicorn U can reach the fountain F without going sailing? (6 Points)