

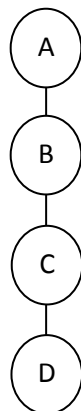
A(E)3M33UI — Exercise 9:

Hierarchical planning 2

Martin Macaš

2015

1. Before you will start to implement this exercise, find in the JSHOP2.pdf how to force JSHOP2 to return only the first plan found for the particular domain.
2. Having the environment specified in the planning seminar work (graph implemented by predicate `link`), the goal is to plan a trip of soldier from one node to another. The particular environment looks like this:



Start with the problem specification using predicates `link` and `atS`. Position a soldier into node A and define the goal node as D:

```
(defproblem demo_problem demo_domain
  (
    (link A B) (link B C) (link C D)
    (link B A) (link C B) (link D C)
    (atS Soldier A)
  )
  (
    (moveto Soldier D)
  )
)
```

Create a domain specification, which contains method `moveto ?soldier ?loc2` and operator `!GOTO ?soldier ?loc1 ?loc2` and which can be used for solving the problem defined above. The method can have three branches corresponding to the position of the soldier with respect to the goal node. (1-soldier stays at the goal node; 2-soldier stays at a neighbor of the goal node; 3-otherwise)

3. In the problem specification change the order of the `link` predicates and run the planning

```
(defproblem demo_problem demo_domain
  (
    (link A B) (link B A)
    (link B C) (link C B)
    (link C D) (link D C)
    (atS Soldier A)
  )
  (
    (moveto Soldier D)
  )
)
```

Axioms

Axioms are defined using dash symbol -

```
(:- a [name1] L1 [name2] L2 . . . [namen] Ln)
```

L are logical preconditions. Axiom is true if L1 is true or L1 is false but L2 is true or

How to compare two objects

```
(:- (same ?x ?x) Nil)
```

```
(:- (different ?x ?y) ((not (same ?x ?y))))
```

3. Modify the code to get bi-directional links by defining axiom `linked ?loc1 ?loc2`, which means that there is a link from `loc1` to `loc2` or back. Use only one tail (logical expression) with `or`.

4. Now, try to find a path from V11 to V55 and back in the following example environment. See what happens if you add `atS` predicate into branch 3 of the method.

(link V11 V21) (link V21 V11)	(link V12 V13) (link V13 V12)
(link V12 V22) (link V22 V12)	(link V13 V14) (link V14 V13)
(link V13 V23) (link V23 V13)	(link V14 V15) (link V15 V14)
(link V14 V24) (link V24 V14)	(link V15 V16) (link V16 V15)
(link V15 V25) (link V25 V15)	(link V21 V22) (link V22 V21)
(link V16 V26) (link V26 V16)	(link V22 V23) (link V23 V22)
(link V21 V31) (link V31 V21)	(link V23 V24) (link V24 V23)
(link V22 V32) (link V32 V22)	(link V24 V25) (link V25 V24)
(link V23 V33) (link V33 V23)	(link V25 V26) (link V26 V25)
(link V24 V34) (link V34 V24)	(link V31 V32) (link V32 V31)
(link V25 V35) (link V35 V25)	(link V32 V33) (link V33 V32)
(link V26 V36) (link V36 V26)	(link V33 V34) (link V34 V33)
(link V31 V41) (link V41 V31)	(link V34 V35) (link V35 V34)
(link V32 V42) (link V42 V32)	(link V35 V36) (link V36 V35)
(link V33 V43) (link V43 V33)	(link V41 V42) (link V42 V41)
(link V34 V44) (link V44 V34)	(link V42 V43) (link V43 V42)
(link V35 V45) (link V45 V35)	(link V43 V44) (link V44 V43)
(link V36 V46) (link V46 V36)	(link V44 V45) (link V45 V44)
(link V41 V51) (link V51 V41)	(link V45 V46) (link V46 V45)
(link V42 V52) (link V52 V42)	(link V51 V52) (link V52 V51)
(link V43 V53) (link V53 V43)	(link V52 V53) (link V53 V52)
(link V44 V54) (link V54 V44)	(link V53 V54) (link V54 V53)
(link V45 V55) (link V55 V45)	(link V54 V55) (link V55 V54)
(link V46 V56) (link V56 V46)	(link V55 V56) (link V56 V55)
(link V11 V12) (link V12 V11)	