

Artificial intelligence - Project 3

- Planning -

Nicoară Marusea Ioana

Popescu Ovidiu Ștefan

Grupa 30234

1 Introducere

Pornind de la stresul pe care îl are un pompier atunci când vrea să salveze vieți, ne-am gândit să rezolvăm o problemă simplă din viața cotidiană utilizând Planning Domain Definition Language (PDDL).

2 Definirea problemei

Motănelul Felix s-a urcat într-un copac ca să-și prindă prada, dar a rămas blocat acolo. Pompierul Dan a primit un apel telefonic prin care a aflat că motănelul Felix, a făcut o năzbâtie. Acesta a pornit de îndată către locul cu pricina, pentru a salva pisica din copac.

Dan trebuie să-l hrănească și să-i ofere apă lui Felix după ce îl salvează.

3 Soluție conceptuală

Primordială este salvarea pisici. Pentru a realiza această misiune, pompierul trebuie să găsească o scară și să o sprijine de trunchiul copacului pentru a se urca după pisică. După salvarea pisici, pe sol, trebuie să existe o pernă pe care pompierul să pună pisica.

Pisica este înfometată și deshidratată. Pompierul trebuie să găsească un pește pentru a hrăni pisica, un bol pentru apă și o sursă de apă.

Acțiunile pe care le poate îndeplini pompierul:

- Se poate deplasa dintr-o poziție în alta (GO-TO)
- Se poate cățăra pe scară (CLIMB)
- Poate coborî de pe scară (DESCEND)
- Poate pune scara pe trunchiul copacului (PUSH-LADDER)
- Poate să prindă peștele (CATCH-FISH)
- Poate să ia un bol pentru apă (PICK-BOWL)
- Poate să umple bolul cu apă (FILL-BOWL)
- Poate salva pisica din copac (SAVE-CAT)
- Poate să elibereze pisica (FREE-CAT)
- Poate să hrănească pisica (FEED-CAT)
- Poate să îi ofere pisicii apă (GIVE-WATER)

Fiecare acțiune se poate realiza doar dacă înainte sunt îndeplinite câteva precondiții cu privire la poziția obiectelor și posesia lor.

4 Implementare

În implementarea soluției au fost folosite tipuri pentru obiectele din domeniul problemei. Acțiunile care intră în componența domeniului vor avea parametri de anumite tipuri. Astfel, problemele vor putea fi extinse ușor pentru mai multe obiecte.

De exemplu, vom putea avea mai mulți pompieri, mai multe animale ce trebuiesc îngrijite, mai multe produse alimentare etc.

Domeniu:

```
1 (define (domain firefighter)
2   (:requirements :strips :typing)
3   (:types
4     firefighter animal food hydrant ladder tree pillow bowl - object
5     firefighter
6     animal
7     food
8     hydrant
9     ladder
10    tree
11    pillow
12    bowl
13    position)
14   (:predicates
15     (at ?obj - object ?pos - position)
16     (on-floor ?x - (either firefighter animal))
17     (on-ladder ?x - firefighter ?y - ladder)
18     (has-food ?x - firefighter ?y - food)
19     (has-animal ?x - firefighter ?y - animal)
20     (has-bowl ?x - firefighter ?y - bowl)
21     (has-water ?x - bowl)
22     (is-fed ?x - animal)
23     (is-hydrated ?x - animal))
24   ;; actiunea de depalsare
25   (:action GO-TO
26     :parameters (?x - (either firefighter animal) ?pos - position ?new-pos - position)
27     :precondition (and (at ?x ?pos) (on-floor ?x))
28     :effect (and (at ?x ?new-pos) (not (at ?x ?pos))))
29   ;; actiunea de catarare
30   (:action CLIMB
31     :parameters (?x - firefighter ?y - ladder ?pos - position)
32     :precondition (and (at ?x ?pos) (at ?y ?pos) (on-floor ?x))
33     :effect (and (on-ladder ?x ?y) (not (on-floor ?x))))
34   ;; actiunea de coborare
35   (:action DESCEND
36     :parameters (?x - firefighter ?y - ladder ?pos - position)
37     :precondition (and (on-ladder ?x ?y) (at ?x ?pos) (at ?y ?pos)
38       (not (on-floor ?x)))
39     :effect (and (not (on-ladder ?x ?y)) (on-floor ?x)))
```

```

1 ;; actiunea de impingere a scarii (pana la copac)
2 (:action PUSH-LADDER
3   :parameters (?x - firefighter ?y - ladder ?pos - position ?new-pos - position)
4   :precondition (and (at ?x ?pos) (at ?y ?pos)
5     (on-floor ?x))
6   :effect (and (at ?x ?new-pos) (not (at ?x ?pos))
7     (at ?y ?new-pos) (not (at ?y ?pos))))
8 ;; actiunea de prindere a pestelui
9 (:action CATCH-FOOD
10  :parameters (?x - firefighter ?y - food ?pos - position)
11  :precondition (and (at ?x ?pos) (at ?y ?pos))
12  :effect (and (has-food ?x ?y) (not (at ?y ?pos))))
13 ;; actiunea de ridicare a bolului
14 (:action PICK-BOWL
15  :parameters (?x - firefighter ?y - bowl ?pos - position)
16  :precondition (and (at ?x ?pos) (at ?y ?pos))
17  :effect (and (has-bowl ?x ?y) (not (at ?y ?pos))))
18 ;; actiunea de umplere a bolului cu apa
19 (:action FILL-BOWL
20  :parameters (?x - firefighter ?y - bowl ?z - hydrant ?pos - position)
21  :precondition (and (at ?x ?pos) (has-bowl ?x ?y)
22    (at ?z ?pos) (on-floor ?x) (not (has-water ?y)))
23  :effect (has-water ?y))
24 ;; actiunea de salvare a animalului
25 (:action SAVE-ANIMAL
26  :parameters (?x - firefighter ?y - animal ?z - ladder ?t - tree ?pos - position)
27  :precondition (and (at ?x ?pos) (at ?y ?pos) (at ?z ?pos) (at ?t ?pos)
28    (on-ladder ?x ?z) (not (on-floor ?y)))
29  :effect (and (has-animal ?x ?y) (not (at ?y ?pos))))
30 )
31 ;; actiunea de eliberare a animalului
32 (:action FREE-ANIMAL
33  :parameters (?x - firefighter ?y - animal ?z - pillow ?pos - position)
34  :precondition (and (at ?x ?pos) (at ?z ?pos) (on-floor ?x)
35    (has-animal ?x ?y))
36  :effect (and (on-floor ?y) (at ?y ?pos) (not (has-animal ?x ?y))))
37 ;; actiunea de hranire a animalului
38 (:action FEED-ANIMAL
39  :parameters (?x - firefighter ?y - animal ?z - food ?pos - position)
40  :precondition (and (at ?x ?pos) (at ?y ?pos) (has-food ?x ?z)
41    (on-floor ?x) (on-floor ?y))
42  :effect (and (is-fed ?y) (not (has-food ?x ?z))))
43 ;; actiunea de hidratare a animalului
44 (:action GIVE-WATER
45  :parameters (?x - firefighter ?y - animal ?z - bowl ?pos - position)
46  :precondition (and (at ?x ?pos) (at ?y ?pos) (has-bowl ?x ?z)
47    (has-water ?z) (on-floor ?x) (on-floor ?y))
48  :effect (and (is-hydrated ?y) (not (has-bowl ?x ?z)) (not (has-water ?z))))

```

5 Probleme

Problema 1:

```
1 (define (problem free-cat-pb)
2   (:domain firefighter)
3
4   (:objects
5     p1 p2 p3 p4 - position
6     Dan - firefighter
7     arbor - tree
8     cat - animal
9     ladder-1 - ladder
10    pillow-1 - pillow
11    bowl-1 - bowl
12    hydrant-1 - hydrant
13    fish - food
14  )
15
16  (:init
17    (at Dan p1)
18    (on-floor Dan)
19    (at ladder-1 p2)
20    (at arbor p3)
21    (at cat p3)
22    (at pillow-1 p4)
23  )
24
25  (:goal
26    (and (on-floor cat))
27  )
28 )
```

Problema 2:

```
1 (define (problem free-and-feed-cat-pb)
2   (:domain firefighter)
3
4   (:objects
5     p1 p2 p3 p4 p5 - position
6     Dan - firefighter
7     ladder-1 - ladder
8     arbor - tree
9     cat - animal
10    fish - food
11    pillow-1 - pillow
12  )
13
14  (:init
15    (at Dan p1)
16    (on-floor Dan)
17    (at ladder-1 p2)
18    (at arbor p5)
19    (at cat p5)
20    (at fish p3)
21    (at pillow-1 p4)
22  )
23
24  (:goal
25    (and (on-floor cat) (is-fed cat))
26  )
27 )
```

Problema 3:

```
1 (define (problem save-fed-and-give-water-to-the-cat-pb)
2   (:domain firefighter)
3
4   (:objects
5     p1 p2 p3 p4 p5 p6 p7 p8 p9 - position
6     Dan - firefighter
7     ladder-1 - ladder
8     arbor - tree
9     cat - animal
10    dog - animal
11    fish - food
12    bowl-1 - bowl
13    bowl-2 - bowl
14    pillow-1 - pillow
15    hydrant-1 - hydrant
16  )
17
18  (:init
19    (at Dan p1)
20    (on-floor Dan)
21    (at ladder-1 p2)
22    (at arbor p5)
23    (at cat p5)
24    (at fish p3)
25    (at pillow-1 p4)
26    (at hydrant-1 p6)
27    (at bowl-1 p7)
28    (at dog p8)
29    (on-floor dog)
30    (at bowl-2 p9)
31  )
32
33  (:goal
34    (and (on-floor cat) (is-fed cat) (is-hydrated cat) (is-hydrated dog))
35  )
36 )
```

6 Concluzii

În cazul acestei probleme, pompierul trebuie să salveze pisica din copac și s-o elibereze pe o pernă aflată la sol.

The screenshot displays a Prolog environment with a problem definition on the left and a solution plan on the right.

Problem Definition (Left Panel):

```

src > [a] problem.pddl > {} problem
1 (define (problem free-cat-pb)
2   (:domain firefighter)
3
4   Show hierarchy
5   (:objects
6     p1 p2 p3 p4 - position
7     Dan - firefighter
8     arbor - tree
9     cat - animal
10    ladder-1 - ladder
11    pillow-1 - pillow
12    bowl-1 - bowl
13    hydrant-1 - hydrant
14    fish - food
15  )
16
17  View
18  (:init
19    (at Dan p1)
20    (on-floor Dan)
21    (at ladder-1 p2)
22    (at arbor p3)
23    (at cat p3)
24    (at pillow-1 p4)
25  )
26
27  (:goal
28    (and (on-floor cat))
29  )
30 )

```

Solution Plan (Right Panel):

The solution plan is a sequence of actions represented by colored squares with labels:

- GO-TO Dan p1 p4 (Red)
- GO-TO Dan p4 p2 (Red)
- PUSH-LADDER Dan ladder-1 p2 p3 (Pink)
- CLIMB Dan ladder-1 p3 (Green)
- SAVE-ANIMAL Dan cat ladder-1 arbor p3 (Orange)
- DESCEND Dan ladder-1 p3 (Blue)
- GO-TO Dan p3 p4 (Red)
- FREE-ANIMAL Dan cat pillow-1 p4 (Green)

Object State Table:

Object	State
firefighter	Dan
position	p1, p2, p3, p4
tree	arbor
animal	cat
ladder	ladder-1
pillow	pillow-1
bowl	bowl-1
hydrant	hydrant-1
food	fish
cat	cat
pillow-1	pillow-1
bowl-1	bowl-1
position	p1, p2, p3, p4

Fig 6.1. Rezultate problema 1

În cazul acestei probleme, pompierul trebuie să salveze pisica din copac și s-o elibereze pe o pernă aflată la sol, apoi el trebuie să hrănească pisica.

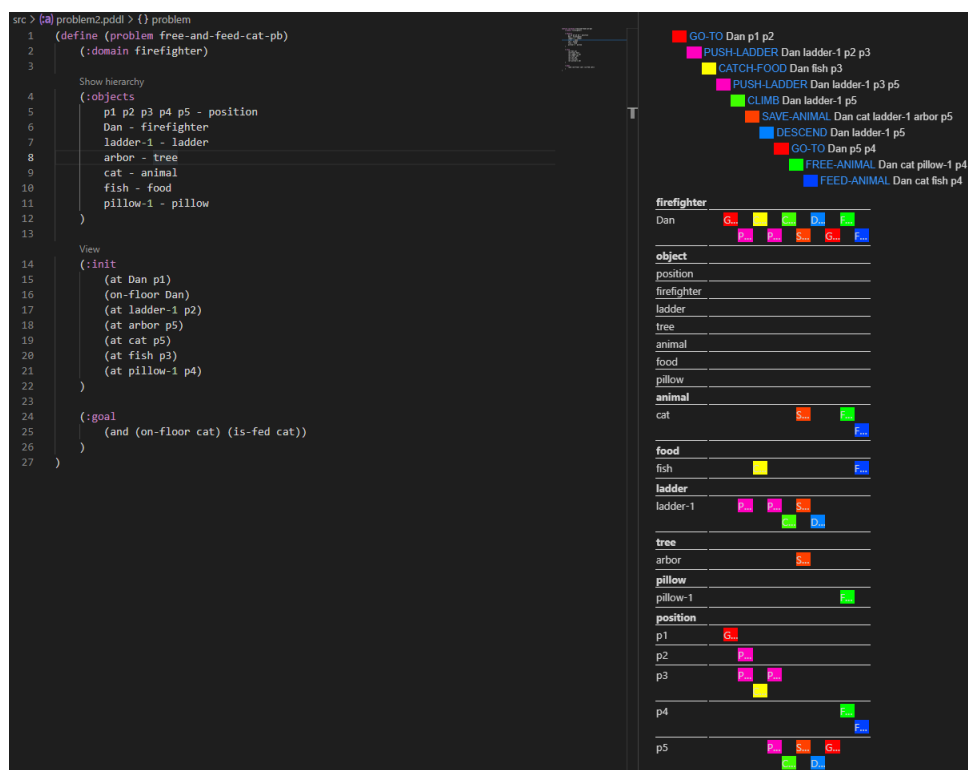


Fig 6.2. Rezultate problema 2

În cazul acestei probleme, pompierul trebuie să salveze pisica din copac și s-o elibereze pe o pernă aflată la sol, apoi el trebuie să hrănească pisica și să îi ofere apă.

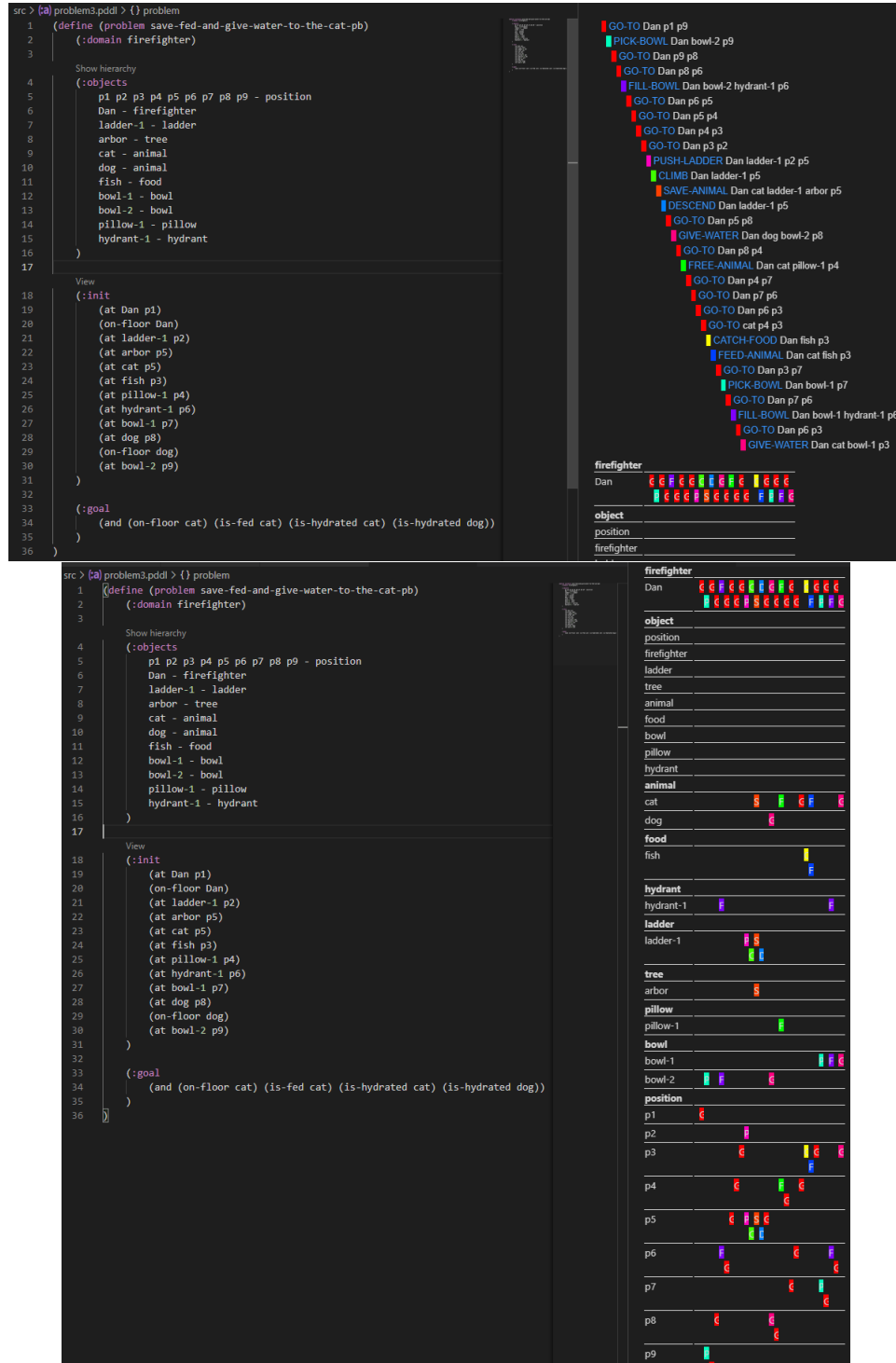


Fig 6.3. Rezultate problema 3