

Artificial intelligence - Project 3
- Planning -

Alexandru Madalina-Elena

13/12/2021

1 Planning

In this project the solution for the following will be presented:

We will have a sliding-tiles puzzle on a 5x5 grid. Three tiles, 11, 15, and 24 will never change their initial place..

1.1 Domain File

Cod:

```
1 (define (domain sliding-tile)
2
3   (:requirements :strips :action-costs)
4
5   (:predicates
6     (is-tile ?x)
7     (is-col ?x)
8     (is-row ?x)
9     (tile-at ?t ?r ?c)
10    (is-blank ?r ?c)
11    (next-row ?r1 ?r2)
12    (next-column ?c1 ?c2)
13    (is-glued ?t)
14  )
15
16  (:functions
17    (WEIGHT ?t) - number
18    (total-cost) - number)
19
20  (:action unstick
21    :parameters(?tile)
22    :precondition (is-glued ?tile)
23    :effect (not (is-glued ?tile))
24  )
25
26  (:action move-tile-down
27    :parameters (?tile ?old-row ?new-row ?col)
28    :precondition (and (is-tile ?tile)
29                      (not (is-glued ?tile))
30                      (is-row ?old-row)
31                      (is-row ?new-row)
32                      (is-col ?col)
33                      (next-row ?old-row ?new-row)
34                      (tile-at ?tile ?old-row ?col)
35                      (is-blank ?new-row ?col))
36    :effect (and (not (tile-at ?tile ?old-row ?col))
37                (not (is-blank ?new-row ?col))
38                (tile-at ?tile ?new-row ?col)
39                (is-blank ?old-row ?col)
40                (increase (total-cost) (WEIGHT ?tile))))
41
42  (:action move-tile-up
43    :parameters (?tile ?old-row ?new-row ?col)
44    :precondition (and (is-tile ?tile)
45                      (not (is-glued ?tile))
```

```

46             (is-row ?old-row)
47             (is-row ?new-row)
48             (is-col ?col)
49             (next-row ?new-row ?old-row)
50             (tile-at ?tile ?old-row ?col)
51             (is-blank ?new-row ?col))
52 :effect (and (not (tile-at ?tile ?old-row ?col))
53             (not (is-blank ?new-row ?col))
54             (tile-at ?tile ?new-row ?col)
55             (is-blank ?old-row ?col)
56             (increase (total-cost) (WEIGHT ?tile))))
57
58 (:action move-tile-right
59  :parameters (?tile ?row ?old-col ?new-col)
60  :precondition (and (is-tile ?tile)
61                    (not (is-glued ?tile))
62                    (is-col ?old-col)
63                    (is-col ?new-col)
64                    (is-row ?row)
65                    (next-column ?old-col ?new-col)
66                    (tile-at ?tile ?row ?old-col)
67                    (is-blank ?row ?new-col))
68  :effect (and (not (tile-at ?tile ?row ?old-col))
69              (not (is-blank ?row ?new-col))
70              (tile-at ?tile ?row ?new-col)
71              (is-blank ?row ?old-col)
72              (increase (total-cost) (WEIGHT ?tile))))
73
74 (:action move-tile-left
75  :parameters (?tile ?row ?old-col ?new-col)
76  :precondition (and (is-tile ?tile)
77                    (not (is-glued ?tile))
78                    (is-col ?old-col)
79                    (is-col ?new-col)
80                    (is-row ?row)
81                    (next-column ?new-col ?old-col)
82                    (tile-at ?tile ?row ?old-col)
83                    (is-blank ?row ?new-col))
84  :effect (and (not (tile-at ?tile ?row ?old-col))
85              (not (is-blank ?row ?new-col))
86              (tile-at ?tile ?row ?new-col)
87              (is-blank ?row ?old-col)
88              (increase (total-cost) (WEIGHT ?tile))))
89
90 )

```

Explicatii:

- Domeniul problemei contine predicatele si actiunile definite.
- Am ales 8 predicate:
 - is-tile ?t: t este tile;
 - is-col ?c: c este coloana;
 - is-row ?r: r este rand;
 - tile-at ?t ?r ?c: primul parametru, t, este localizat la randul r si coloana c;

- is-blank ?r ?c: spatiul localizat la randul r si coloana c este liber;
 - nex-row ?r1 ?r2: r2 este randul care urmeaza dupa r1;
 - next-column ?c1 ?c2: c2 este coloana care urmeaza dupa c1;
 - is-glued ?t: t este glued, nu poate fi miscata;
- Fiecare tile va avea o anumita greutate.
 - Exista 4 miscari posibile pentru fiecare tile, move-tile-up, move-tile-down, move-tile-right, move-tile-left, fiecare mutand piesa in partea precizata.
 - Pentru fiecare actiune, se verifica daca piesa este lipita, daca este posibila miscarea dorita si daca parametrii transmisi sunt de tipul dorit, folosindu-se predicatele precizate mai sus. Daca se indeplinesc toate aceste conditii, atunci se muta piesa de la locatia sa anterioara.
 - O alta actiune posibila este aceea de a dezlipi o piesa pentru ca aceasta sa poata fi mutata din nou.

1.2 Problem File

Cod

```

1 (define (problem twenty-four-puzzle)
2   (:domain sliding-tile)
3   (:objects
4     tile1 tile2 tile3 tile4 tile5
5     tile6 tile7 tile8 tile9 tile10
6     tile11 tile12 tile13 tile14 tile15
7     tile16 tile17 tile18 tile19 tile20
8     tile21 tile22 tile23 tile24
9
10    row1 row2 row3 row4 row5
11    col1 col2 col3 col4 col5)
12   (:init
13     (is-tile tile1)
14     (is-tile tile2)
15     (is-tile tile3)
16     (is-tile tile4)
17     (is-tile tile5)
18     (is-tile tile6)
19     (is-tile tile7)
20     (is-tile tile8)
21     (is-tile tile9)
22     (is-tile tile10)
23     (is-tile tile11)
24     (is-tile tile12)
25     (is-tile tile13)
26     (is-tile tile14)
27     (is-tile tile15)
28     (is-tile tile16)
29     (is-tile tile17)
30     (is-tile tile18)
31     (is-tile tile19)
32     (is-tile tile20)
33     (is-tile tile21)
34     (is-tile tile22)
35     (is-tile tile23)
36     (is-tile tile24)
37

```

```

38      (is-row row1)
39      (is-row row2)
40      (is-row row3)
41      (is-row row4)
42      (is-row row5)
43
44
45      (is-col col1)
46      (is-col col2)
47      (is-col col3)
48      (is-col col4)
49      (is-col col5)
50
51      (next-row row1 row2)
52      (next-row row2 row3)
53      (next-row row3 row4)
54      (next-row row4 row5)
55
56      (next-column col1 col2)
57      (next-column col2 col3)
58      (next-column col3 col4)
59      (next-column col4 col5)
60
61      (is-glued tile15)
62      (is-glued tile11)
63      (is-glued tile24)
64
65      (tile-at tile10 row1 col1)
66      (tile-at tile12 row1 col2)
67      (tile-at tile2 row1 col3)
68      (tile-at tile3 row1 col4)
69      (tile-at tile9 row1 col5)
70
71      (tile-at tile21 row2 col1)
72      (tile-at tile20 row2 col2)
73      (tile-at tile4 row2 col3)
74      (tile-at tile16 row2 col4)
75      (tile-at tile14 row2 col5)
76
77      (tile-at tile11 row3 col1)
78      (tile-at tile17 row3 col2)
79      (tile-at tile23 row3 col3)
80      (tile-at tile5 row3 col4)
81      (tile-at tile15 row3 col5)
82
83      (tile-at tile22 row4 col1)
84      (tile-at tile19 row4 col2)
85      (tile-at tile13 row4 col3)
86      (tile-at tile7 row4 col4)
87      (tile-at tile8 row4 col5)
88
89      (tile-at tile18 row5 col1)
90      (tile-at tile1 row5 col2)
91      (tile-at tile6 row5 col3)

```

```

92      (tile-at tile24 row5 col4)
93      (is-blank row5 col5)
94
95      (= (total-cost) 0)
96
97      (= (WEIGHT tile1) 1)
98      (= (WEIGHT tile2) 2)
99      (= (WEIGHT tile3) 3)
100     (= (WEIGHT tile4) 1)
101     (= (WEIGHT tile5) 2)
102     (= (WEIGHT tile6) 3)
103     (= (WEIGHT tile7) 1)
104     (= (WEIGHT tile8) 2)
105     (= (WEIGHT tile9) 3)
106     (= (WEIGHT tile10) 1)
107     (= (WEIGHT tile11) 2)
108     (= (WEIGHT tile12) 3)
109     (= (WEIGHT tile13) 1)
110     (= (WEIGHT tile14) 2)
111     (= (WEIGHT tile15) 3)
112     (= (WEIGHT tile16) 1)
113     (= (WEIGHT tile17) 2)
114     (= (WEIGHT tile18) 3)
115     (= (WEIGHT tile19) 1)
116     (= (WEIGHT tile20) 2)
117     (= (WEIGHT tile21) 3)
118     (= (WEIGHT tile22) 1)
119     (= (WEIGHT tile23) 2)
120     (= (WEIGHT tile24) 3)
121   )
122
123   (:goal
124     (and
125       (tile-at tile1 row1 col1)      (tile-at tile2 row1 col2)
126       (tile-at tile3 row1 col3)      (tile-at tile4 row1 col4)
127       (tile-at tile5 row1 col5)      (tile-at tile6 row2 col1)
128       (tile-at tile7 row2 col2)      (tile-at tile8 row2 col3)
129       (tile-at tile9 row2 col4)      (tile-at tile10 row2 col5)
130       (tile-at tile11 row3 col1)     (tile-at tile12 row3 col2)
131       (tile-at tile13 row3 col3)     (tile-at tile14 row3 col4)
132       (tile-at tile15 row3 col5)     (tile-at tile16 row4 col1)
133       (tile-at tile17 row4 col2)     (tile-at tile18 row4 col3)
134       (tile-at tile19 row4 col4)     (tile-at tile20 row4 col5)
135       (tile-at tile21 row5 col1)     (tile-at tile22 row5 col2)
136       (tile-at tile23 row5 col3)     (tile-at tile24 row5 col4)))
137
138   (:metric minimize (total-cost))
139 )

```

Explicatii:

- Fisierul problemei descrie starea initiala in care se afla obiectele problemei si starea "goal" ale acestora.
- Initial se stabileste domeniul problemei. In continuare stabilim obiectele si anule cele 24 de piese, cele 5 randuri si cele 5 coloane.

- In partea de initializare, este declarata starea pieselor, randurilor si a coloanelor, dar si ordinea coloanelor si a randurilor. De asemenea, se asociaza o greutate pentru fiecare piesa, 1, 2 sau 3. Se scriu locatiile fiecărei piese; spatiul de la randul 5, coloana 5 este gol.
- Starea finala este cea in care toate cele 24 de piese sunt ordonate.
- Am inclus in proiect 3 fisiera problema.

1.3 Comenzi:

- `./fast-downward.py sliding-tiles.pddl problem.pddl`
- `./fast-downward.py sliding-tiles.pddl problem1.pddl`
- `./fast-downward.py sliding-tiles.pddl problem2.pddl`

1.4 Rezultate comparate

	P1		P2		P3	
Algorithm	Eager greedy	Lazy greedy	Eager greedy	Lazy greedy	Eager greedy	Lazy greedy
Plan length	415	669	624	960	1021	887
Plan cost	722	1118	1117	1717	1756	1630
Expanded states	4550	22153	6159	98576	21068	18386
Time	1.32s	2.34s	1.57s	7.14s	6.56s	2.04s

1.5 Uncertainty

1.5.1 Comenzi:

- `./Contingent-FF -o contff-tests/sliding_tiles_uncertainty/uncertainty.pddl -f contff-tests/sliding_tiles_uncertainty/problem_uncertainty.pddl`

1.6 Explicatii:

- Pentru aceasta parte am adaugat in problema faptul ca una dintre cateva piese definite ca "unknown" ar putea fi lipita.
- Astfel am folosit contingent planning pentru a rezolva problema.
- Fisierul de domeniu pentru problema de incertitudine este uncertainty.pddl, iar cel ce contine problema este problem_uncertainty.pddl