

Inteligenta artificiala

Tema 3

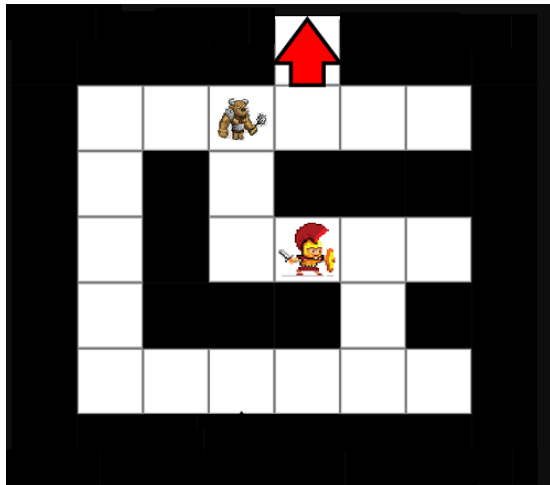
Proiect PDDL

Nume: Birladeanu Alexandru-Gabriel

Grupa: 302310

Minotaurul din Creta

Theseus a ajuns la curtea regelui Minos din Creta si a descoperit ca aici exista un labirint in care regele il tine ca prizonier pe fiul sau vitreg, un monstru pe jumatate om si pe jumatate taur: minotaurul. Theseus vrea sa il uida pe monstru, asa ca intra in labirint si lasa usa deschisa. Minotaurul simte ca parca vine curent de la iesire si, fiind satul de batai si de scandal, vrea sa se furiseze pe langa Theseus si sa iasa din labirint fara sa se mai lupte. Intre timp, Theseus nu cunoaste labirintul si va trece prin toate locurile din labirint, incercand sa gaseasca minotaurul. Acum ramane de vazut ce se va intampla, va reusi Theseus sa gaseasca si sa uida monstrul, sau va reusi acesta din urma sa evadeze, evitand confruntarea?



```
1 (define (domain labyrinth)
2   (:predicates
3     (isMinotaur ?character) 9
4     (isTheseus ?character) 9
5     (justMoved ?character) 8 8 8
6     (characterAt ?character ?r ?c) 10 8 8
7     (wallAt ?r ?c) 8
8     (nextr ?r1 ?r2) 10
9     (nextc ?c1 ?c2) 11
10    (theseusVisited ?r ?c) 25
11  )
```

```
1 (define (problem labyrinthProblem)
2   (:domain labyrinth)
3   Show hierarchy
4   (:objects minotaur theseus r1 r2 r3 r4 r5 r6 r7 c1 c2 c3 c4 c5 c6 c7 c8)
5   View
6   (:init (isMinotaur minotaur) (isTheseus theseus)
7     (nextr r1 r2) (nextr r2 r3) (nextr r3 r4) (nextr r4 r5) (nextr r5 r6) (nextr r6 r7)
8     (nextc c1 c2) (nextc c2 c3) (nextc c3 c4) (nextc c4 c5) (nextc c5 c6) (nextc c6 c7) (nextc c7 c8)
9     (wallAt r1 c1) (wallAt r1 c2) (wallAt r1 c3) (wallAt r1 c4) (wallAt r1 c5) (wallAt r1 c6) (wallAt r1 c7) (wallAt r1 c8)
10    (wallAt r2 c1) (wallAt r2 c2) (wallAt r2 c3) (wallAt r2 c4) (wallAt r2 c5) (wallAt r2 c6) (wallAt r2 c7) (wallAt r2 c8)
11    (wallAt r3 c1) (wallAt r3 c2) (wallAt r3 c3) (wallAt r3 c4) (wallAt r3 c5) (wallAt r3 c6) (wallAt r3 c7) (wallAt r3 c8)
12    (wallAt r4 c1) (wallAt r4 c2) (wallAt r4 c3) (wallAt r4 c4) (wallAt r4 c5) (wallAt r4 c6) (wallAt r4 c7) (wallAt r4 c8)
13    (wallAt r5 c1) (wallAt r5 c2) (wallAt r5 c3) (wallAt r5 c4) (wallAt r5 c5) (wallAt r5 c6) (wallAt r5 c7) (wallAt r5 c8)
14    (wallAt r6 c1) (wallAt r6 c2) (wallAt r6 c3) (wallAt r6 c4) (wallAt r6 c5) (wallAt r6 c6) (wallAt r6 c7) (wallAt r6 c8)
15    (wallAt r7 c1) (wallAt r7 c2) (wallAt r7 c3) (wallAt r7 c4) (wallAt r7 c5) (wallAt r7 c6) (wallAt r7 c7) (wallAt r7 c8)
16    (not(theseusVisited r2 c2))
17    (not(theseusVisited r2 c3))
18    (not(theseusVisited r2 c4))
19    (not(theseusVisited r2 c5))
20    (theseusVisited r2 c6)
21    (not(theseusVisited r2 c7))
22    (not(theseusVisited r3 c2))
23    (not(theseusVisited r3 c6))
24    (not(theseusVisited r4 c2))
25    (not(theseusVisited r4 c4))
26    (not(theseusVisited r4 c5))
27    (not(theseusVisited r4 c6))
28    (not(theseusVisited r4 c7))
29    (not(theseusVisited r5 c2))
30    (not(theseusVisited r5 c4))
31    (not(theseusVisited r6 c2))
32    (not(theseusVisited r6 c3))
33    (not(theseusVisited r6 c4))
34    (not(theseusVisited r6 c5))
35    (not(theseusVisited r6 c6))
36    (not(theseusVisited r6 c7))
37    (characterAt minotaur r6 c2)
38    (characterAt theseus r4 c4)
39    (justMoved minotaur)
40    (not(justMoved theseus))
41  )
```

Am construit labirintul din poza de mai sus cu ajutorul predicatelor `nextx`, `nexty` si `wallAt`. Numerotarea randurilor se face de jos in sus si numerotarea coloanelor se face de la stanga la dreapta. In problema am initializat coordonatele fiecarui zid, ordinea randurilor si coloanelor, precum si toate celulele prin care personajele pot trece ca fiind nevizitate de Theseus. Exceptie face celula din care el va porni. De asemenea, am initializat personajele cu predicatele `isMinotaur` si `isTheseus`. Voi explica predicatul `justMoved` ulterior.

Goal-ul este atins daca minotaurul ajunge la iesire sau daca Theseus viziteaza fiecare celula din labirint. Vom observa mai tarziu ca daca se indeplineste a doua conditie inseamna de fapt ca Theseus a ucis minotaurul, adica amandoi au ajuns in aceeasi pozitie.

Avem 4 actiuni de miscare pentru fiecare personaj (`minotaur-move-up`, `minotaur-move-down`, `minotaur-move-left`, `minotaur-move-right`, `theseus-move-up`, `theseus-move-down`, `theseus-move-right`, `theseus-move-left`) care sunt foarte similare. Voi explica detaliat doar actiunile pentru `move-up`.

```

38      (:goal (or (characterAt minotaur r7 c5)
39                (and
40                  (theseusVisited r2 c2)
41                  (theseusVisited r2 c3)
42                  (theseusVisited r2 c4)
43                  (theseusVisited r2 c5)
44                  (theseusVisited r2 c6)
45                  (theseusVisited r2 c7)
46                  (theseusVisited r3 c2)
47                  (theseusVisited r3 c6)
48                  (theseusVisited r4 c2)
49                  (theseusVisited r4 c4)
50                  (theseusVisited r4 c5)
51                  (theseusVisited r4 c6)
52                  (theseusVisited r4 c7)
53                  (theseusVisited r5 c2)
54                  (theseusVisited r5 c4)
55                  (theseusVisited r6 c2)
56                  (theseusVisited r6 c3)
57                  (theseusVisited r6 c4)
58                  (theseusVisited r6 c5)
59                  (theseusVisited r6 c6)
60                  (theseusVisited r6 c7)))
61      )
62    )
63  )

```

```

(:action minotaur-move-up
  :parameters (?minotaur ?theseus ?oldr ?c ?newr)
  :precondition (and (characterAt ?minotaur ?oldr ?c) (nextx ?oldr ?newr) (isMinotaur ?minotaur) (isTheseus ?theseus) (not (wallAt ?newr ?c)) (not (justMoved ?minotaur)))
  :effect (and (characterAt ?minotaur ?newr ?c) (not (characterAt ?minotaur ?oldr ?c)) (justMoved ?minotaur) (not (justMoved ?theseus))))
)

```

Preconditiile sunt ca paramentru `?character` sa fie un minotaur, sa se afle pe o anumita pozitie, randul nou sa aiba indexul cu 1 mai mare decat cel vechi si sa nu existe un zid pe pozitia unde vrem sa ajungem. La fiecare actiune de miscare ne folosim de predicatul `justMoved` pentru ca minotaurul si theseus sa se miste alternativ, odata minotaurului, odata theseus. Acest lucru functioneaza in felul urmatoare: dupa ce unul dintre ei face o miscare, `justMoved` pentru el devine adevarat si `justMoved` pentru celalalt caracter devine fals. Astfel, se poate misca doar caracterul care are `justMoved` fals. Daca nu am fi folosit acest predicat, atunci cand rulam proiectul, minotaurul s-ar fi deplasat numai el pana ajungea la iesire si theseus statea pe loc. In problema am initializat (`justMoved minotaur`) ca fiind adevarat, deci Theseus va face prima miscare.

```

(:action theseus-move-up
  :parameters (?minotaur ?theseus ?oldr ?c ?newr)
  :precondition (and (characterAt ?theseus ?oldr ?c) (nextx ?oldr ?newr) (isTheseus ?theseus) (isMinotaur ?minotaur) (not (wallAt ?newr ?c)) (not (justMoved ?theseus)))
  :effect (and (theseusVisited ?newr ?c) (characterAt ?theseus ?newr ?c) (not (characterAt ?theseus ?oldr ?c)) (justMoved ?theseus) (not (justMoved ?minotaur))))
)

```

Miscarea pentru theseus se face similar, doar ca la el setam ca `visited` celula in care ajunge.

```

(:action theseusFoundMinotaur
:parameters (?minotaur ?theseus ?r ?c ?r1 ?r2 ?r3 ?r4 ?r5 ?r6 ?r7 ?c1 ?c2 ?c3 ?c4 ?c5 ?c6 ?c7 ?c8)
:precondition (and(isMinotaur ?minotaur) (isTheseus ?theseus) (characterAt ?minotaur ?r ?c) (characterAt ?theseus ?r ?c)
  (nextc ?r1 ?r2) (nextc ?r2 ?r3) (nextc ?r3 ?r4) (nextc ?r4 ?r5) (nextc ?r5 ?r6) (nextc ?r6 ?r7)
  (nextc ?c1 ?c2) (nextc ?c2 ?c3) (nextc ?c3 ?c4) (nextc ?c4 ?c5) (nextc ?c5 ?c6) (nextc ?c6 ?c7) (nextc ?c7 ?c8)
)
:effect (and
  (theseusVisited ?r2 ?c2)
  (theseusVisited ?r2 ?c3)
  (theseusVisited ?r2 ?c4)
  (theseusVisited ?r2 ?c5)
  (theseusVisited ?r2 ?c6)
  (theseusVisited ?r2 ?c7)
  (theseusVisited ?r3 ?c2)
  (theseusVisited ?r3 ?c6)
  (theseusVisited ?r4 ?c2)
  (theseusVisited ?r4 ?c4)
  (theseusVisited ?r4 ?c5)
  (theseusVisited ?r4 ?c6)
  (theseusVisited ?r4 ?c7)
  (theseusVisited ?r5 ?c2)
  (theseusVisited ?r5 ?c4)
  (theseusVisited ?r6 ?c2)
  (theseusVisited ?r6 ?c3)
  (theseusVisited ?r6 ?c4)
  (theseusVisited ?r6 ?c5)
  (theseusVisited ?r6 ?c6)
  (theseusVisited ?r6 ?c7)
)
)
)

```

In actiunea theseusFoundMinotaur verificam daca cei 2 au ajuns in aceeași celulă din labirint, adică dacă se găsesc pe aceleași coordonate. În caz afirmativ, setăm toate celulele ca fiind vizitate de Theseus, chiar dacă el nu a ajuns prin unele dintre ele, dar astfel se încheie executia programului, fiind atins goal-ul.

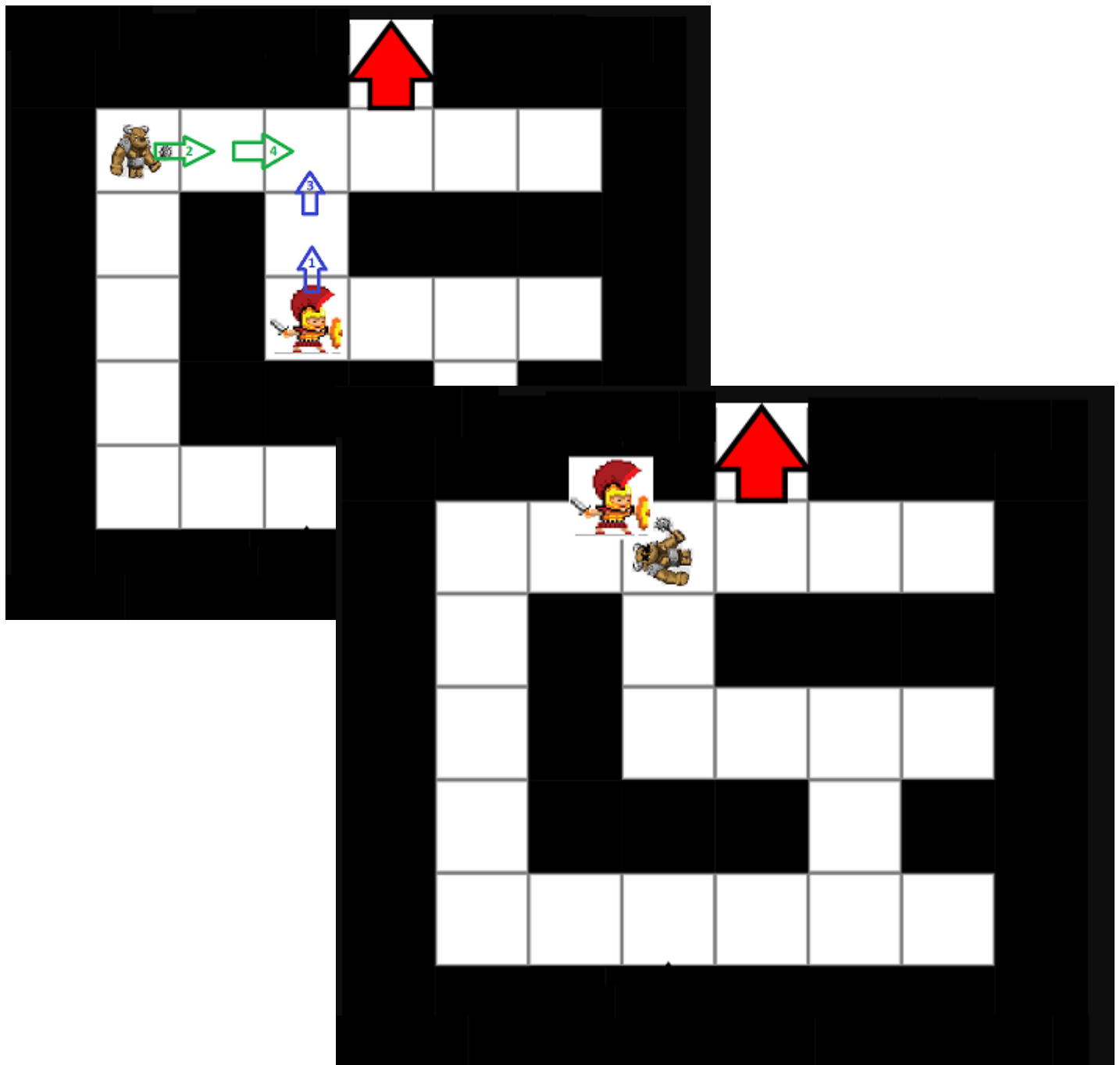
În concluzie, goal-ul poate fi activat în 2 moduri, depinzând de care dintre cele 2 argumente din funcția OR va avea valoarea de adevăr primă. Diferența se va face în funcție de pozițiile de început ale celor 2 personaje. Uneori, minotaurul va ajunge la destinație înainte ca Theseus să ajungă în aceeași celulă ca el, alteori cei 2 se vor întâlni și se va executa acțiunea theseusFoundMinotaur. Am implementat 3 probleme pentru a exemplifica algoritmul creat. În fiecare problemă, labirintul este construit la fel, goal-ul este același, dar diferă pozițiile de început ale celor 2 personaje.

Cazul 1:

Pozitii initiale: (characterAt minotaur r6 c2)
(characterAt theseus r4 c4)

Rezultat

```
[t=0.00667119s, 10232 KB] Solution found!  
[t=0.00669667s, 10232 KB] Actual search time: 0.000615832s  
theseus-move-up minotaur theseus r4 c4 r5 (1)  
minotaur-move-right minotaur theseus r6 c2 c3 (1)  
theseus-move-up minotaur theseus r5 c4 r6 (1)  
minotaur-move-right minotaur theseus r6 c3 c4 (1)  
theseusfoundminotaur minotaur theseus r6 c4 r1 r2 r3 r4 r5 r6 r7 c1 c2 c3 c4 c5 c6 c7 c8 (1)  
[t=0.00671121s, 10232 KB] Final length: 5 steps
```



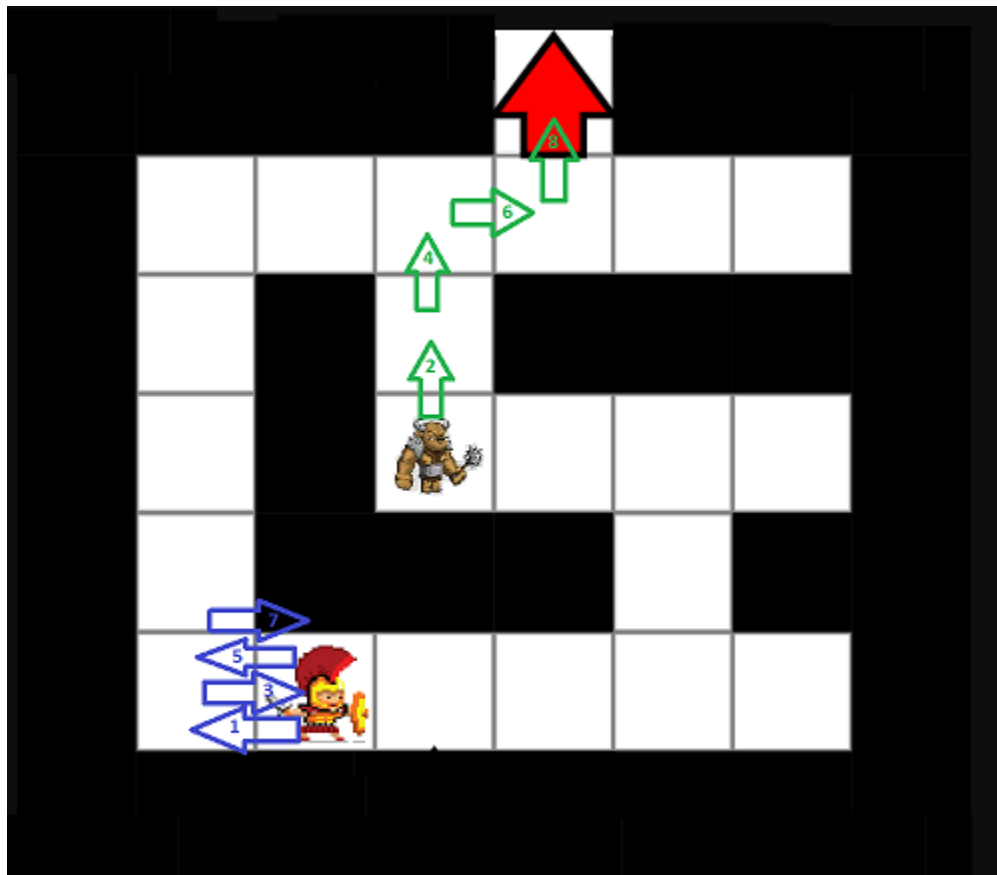
Cazul 2:

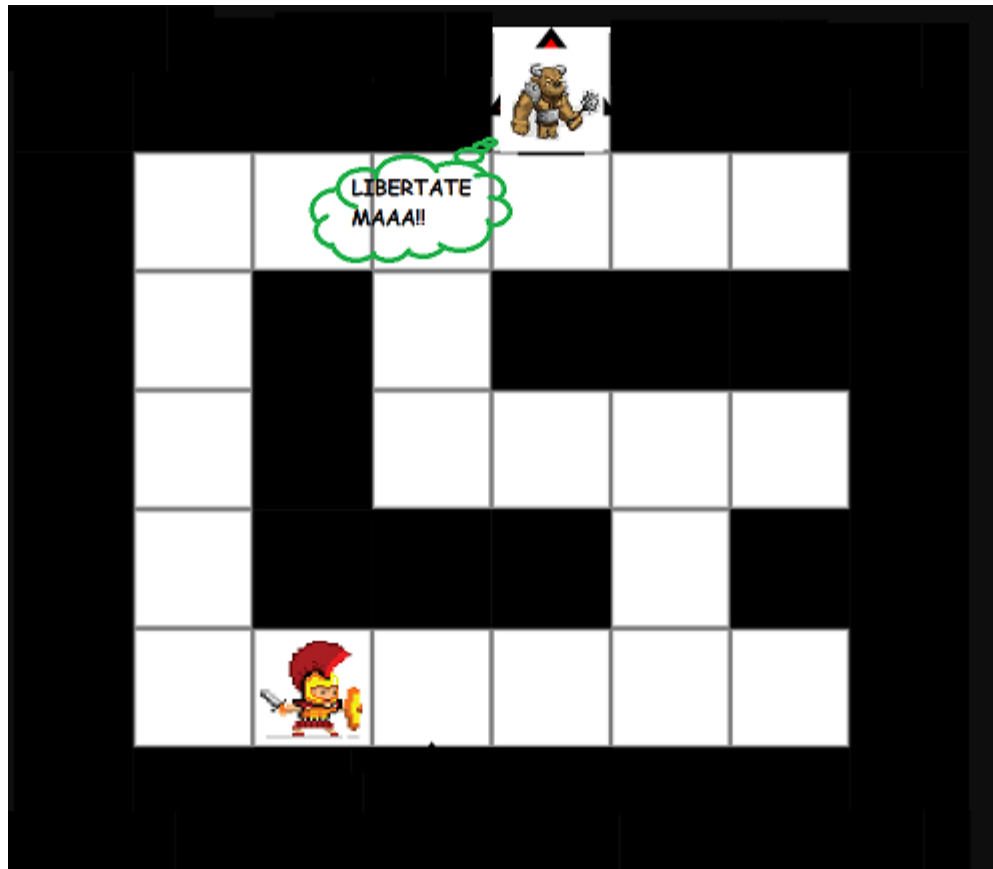
Pozitii initiale:

```
(characterAt minotaur r4 c4)
(characterAt theseus r2 c3)
```

Rezultat:

```
[t=0.00782073s, 10232 KB] Solution found!
[t=0.00784892s, 10232 KB] Actual search time: 0.00144968s
theseus-move-left minotaur theseus r2 c3 c2 (1)
minotaur-move-up minotaur theseus r4 c4 r5 (1)
theseus-move-right minotaur theseus r2 c2 c3 (1)
minotaur-move-up minotaur theseus r5 c4 r6 (1)
theseus-move-left minotaur theseus r2 c3 c2 (1)
minotaur-move-right minotaur theseus r6 c4 c5 (1)
theseus-move-right minotaur theseus r2 c2 c3 (1)
minotaur-move-up minotaur theseus r6 c5 r7 (1)
```





Cazul 3:

Pozitii initiale: `(characterAt minotaur r2 c7)`
`(characterAt theseus r7 c5)`

Rezultat:

```
[t=0.00530544s, 10232 KB] Solution found!
[t=0.00533318s, 10232 KB] Actual search time: 0.000803195s
theseus-move-down minotaur theseus r7 c5 r6 (1)
minotaur-move-left minotaur theseus r2 c7 c6 (1)
theseus-move-left minotaur theseus r6 c5 c4 (1)
minotaur-move-up minotaur theseus r2 c6 r3 (1)
theseus-move-down minotaur theseus r6 c4 r5 (1)
minotaur-move-up minotaur theseus r3 c6 r4 (1)
theseus-move-down minotaur theseus r5 c4 r4 (1)
minotaur-move-left minotaur theseus r4 c6 c5 (1)
theseus-move-right minotaur theseus r4 c4 c5 (1)
theseusfoundminotaur minotaur theseus r4 c5 r1 r2 r3 r4 r5 r6 r7 c1 c2 c3 c4 c5 c6 c7 c8 (1)
[t=0.00536811s, 10232 KB] Solution found!
[t=0.00539585s, 10232 KB] Actual search time: 0.00082874s
```

