5x5 Tile Slide

Vlad Ursache

Cuprins

Rezumat

Aceasta este documentația pentru mini-proiectul de planificare în cadrul disciplinei Inteligență Artificială. Acesta constă într-o problemă de tip Tile Slide 5x5. La final putem vedea rezultatele (cel puțin până rămân fără RAM).

1 Introducere

Am ales acest proiect pentru a testa limitele planner-ului Fast Downward (dar și ale PC-ului meu). Puzzle-ul funcționează prin mutarea pieselor prin glisare. Ordinea pieselor trebuie restaurată.

Figura 1: Puzzle

Vom folosi exemplul de mai sus.

2 Mod de funcționare

Vom rula proiectul astfel: python3 fast-downward.py domain.pddl task.pddl –search "astar(ff())"
.
Am putea folosi alt algoritm de căutare și altă euristică. Nu suntem constrânși. De dragul exemplului, vom rula comanda de mai sus.

2.1 Domeniu

```
(define (domain sliding-tile)

  (:predicates
   (tile-at ?t ?r ?c)
   (is-blank ?r ?c)
   (next-row ?r1 ?r2)
   (next-column ?c1 ?c2)
  )

  (:action move-tile-down
    :parameters (?tile ?old-row ?new-row ?col)
    :precondition (and (next-row ?old-row ?new-row)
                       (tile-at ?tile ?old-row ?col)
                       (is-blank ?new-row ?col))
    :effect (and (not (tile-at ?tile ?old-row ?col))
                 (not (is-blank ?new-row ?col))
                 (tile-at ?tile ?new-row ?col)
                 (is-blank ?old-row ?col)))

  (:action move-tile-up
    :parameters (?tile ?old-row ?new-row ?col)
    :precondition (and (next-row ?new-row ?old-row)
                       (tile-at ?tile ?old-row ?col)
                       (is-blank ?new-row ?col))
    :effect (and (not (tile-at ?tile ?old-row ?col))
                 (not (is-blank ?new-row ?col))
                 (tile-at ?tile ?new-row ?col)
                 (is-blank ?old-row ?col)))

  (:action move-tile-right
    :parameters (?tile ?row ?old-col ?new-col)
    :precondition (and (next-column ?old-col ?new-col)
                       (tile-at ?tile ?row ?old-col)
                       (is-blank ?row ?new-col))
    :effect (and (not (tile-at ?tile ?row ?old-col))
                 (not (is-blank ?row ?new-col))
                 (tile-at ?tile ?row ?new-col)
                 (is-blank ?row ?old-col)))

  (:action move-tile-left
    :parameters (?tile ?row ?old-col ?new-col)
    :precondition (and (next-column ?new-col ?old-col)
                       (tile-at ?tile ?row ?old-col)
                       (is-blank ?row ?new-col))
    :effect (and (not (tile-at ?tile ?row ?old-col))
                 (not (is-blank ?row ?new-col))
                 (tile-at ?tile ?row ?new-col)
                 (is-blank ?row ?old-col))))
```

Predicatele tile-at, is-blank, next-row și next-column, alături de parametrii lor ajută la compunerea acțiunilor posibile (a parametrilor, precondițiilor și efectelor lor). Aceste acțiuni sunt cele de glisare ale pieselor. Acest fișier este împrumutat din laborator.

2.2 Problemă

```
(define (problem eight-puzzle)
  (:domain sliding-tile)
  (:objects
     tile1 tile2 tile3 tile4 tile5
     tile6 tile7 tile8 tile9 tile10
     tile11 tile12 tile13 tile14 tile15
     tile16 tile17 tile18 tile19 tile20
     tile21 tile22 tile23 tile24
     row1 row2 row3 row4 row5
     col1 col2 col3 col4 col5)
  (:init
    (next-row row1 row2)     (next-column col1 col2)
    (next-row row2 row3)     (next-column col2 col3)
    (next-row row3 row4)     (next-column col3 col4)
    (next-row row4 row5)     (next-column col4 col5)

    (tile-at tile23 row1 col1)     (tile-at tile3 row1 col2)
    (tile-at tile9 row1 col3)      (tile-at tile17 row1 col4)
    (tile-at tile14 row1 col5)     (tile-at tile19 row2 col1)
    (tile-at tile12 row2 col2)     (tile-at tile15 row2 col3)
    (tile-at tile7 row2 col4)      (tile-at tile20 row2 col5)
    (tile-at tile18 row3 col1)     (tile-at tile13 row3 col2)
    (tile-at tile21 row3 col3)     (tile-at tile8 row3 col4)
    (tile-at tile22 row3 col5)     (tile-at tile6 row4 col1)
    (tile-at tile10 row4 col2)     (tile-at tile2 row4 col3)
    (tile-at tile11 row4 col4)     (tile-at tile1 row4 col5)
    (tile-at tile24 row5 col1)     (tile-at tile16 row5 col2)
    (tile-at tile4 row5 col3)      (tile-at tile5 row5 col4)
    (is-blank row5 col5))
  (:goal
   (and
    (tile-at tile1 row1 col1)      (tile-at tile2 row1 col2)
    (tile-at tile3 row1 col3)      (tile-at tile4 row1 col4)
    (tile-at tile5 row1 col5)      (tile-at tile6 row2 col1)
    (tile-at tile7 row2 col2)      (tile-at tile8 row2 col3)
    (tile-at tile9 row2 col4)      (tile-at tile10 row2 col5)
    (tile-at tile11 row3 col1)     (tile-at tile12 row3 col2)
    (tile-at tile13 row3 col3)     (tile-at tile14 row3 col4)
    (tile-at tile15 row3 col5)     (tile-at tile16 row4 col1)
    (tile-at tile17 row4 col2)     (tile-at tile18 row4 col3)
    (tile-at tile19 row4 col4)     (tile-at tile20 row4 col5)
    (tile-at tile21 row5 col1)     (tile-at tile22 row5 col2)
    (tile-at tile23 row5 col3)     (tile-at tile24 row5 col4)
    (is-blank row5 col5))))
```

Am declarat domeniul, starea inițială și starea finală a puzzle-ului.

3 Concluzii

Am reușit să ajung la un număr minim de mutări egal cu 37. Progresul înainte de a rămâne aproape fără RAM îl putem regăsi mai jos. Soluția suboptimă a fost identificată după mai bine de 2 ore și 10GB de RAM.
O soluție mai scurtă a fost găsită utilizând euristica cg. Cu ajutorul acesteia am identificat un joc de 26 de mutări înainte de a rămâne din nou fără RAM.

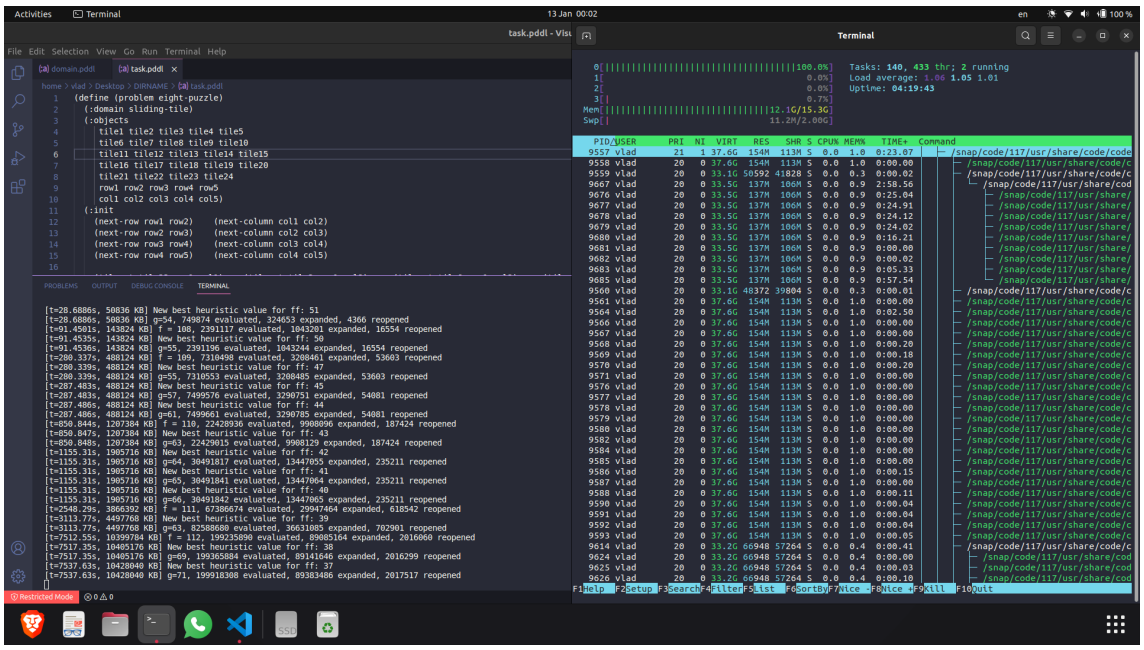

Figura 2: Progres ff

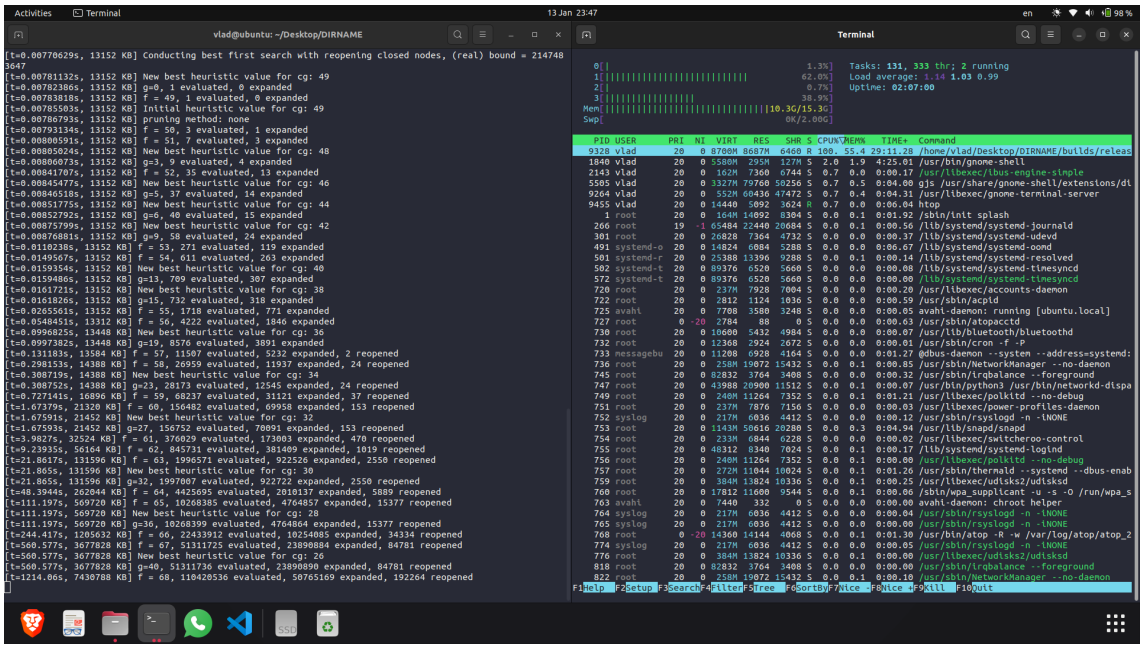

Figura 3: Progres cg