

Inteligenta artificiala - Knowledge representation

Micul vrajitor - documentatie

S-au folosit in plus metodele:

- `update_bootsnstone` (metoda de updatare a ghetelor si a ghiozdanului in momentul in care micul vrajitor intra intr-o incapere noua);
- `genereazaSuccesori` (metoda genereaza succesorii care verifica daca vrajitorul poate merge in sus/jos/stanga/dreapta pe matrice avand in vedere ghetele cu care este incaltat si ghetele pe care le are in ghiozdan);
- `read` (functie care ne ajuta cu citirea inputului);
- `refresh_files` (functie care da refresh la output de fiecare data cand rulam programul deoarece toate celelalte functii dau append si outputul nu s-ar sterge niciodata).

Algoritmii ruleaza dinamic de 2 ori pentru a face posibila intoarcerea vrajitorului la iesirea din peștera după ce a luat piatra.

Euristici

1. Euristica banala($euristica = 0$);
2. Distanța Manhattan calculată până la $(start+finish)/2$;
3. Distanța Euclidiană calculată până la $(start+finish)/2$;
4. Euristica neadmisibilă (până la suma dintre X-ul și Y-ul punctului curent).

Exemplu de rulare

python little_wizard.py timeout euristica nsol folder input

python little_wizard.py 1 2 3 input 3

Explicație: există 4 euristici ($1 \leq e \leq 4$) și 4 inputuri ($1 \leq i \leq 4$). Dintre toate tipurile de input, nu putem avea starea finală = starea inițială deoarece nu putem pune caracterele '*' și '@' pe același patratel. Singurul input care poate fi apelat cu $nsol > 2$ este inputul 3 deoarece inputul 1 are o singură soluție, inputul 4 nu are soluții și inputul 2 are un singur pas (încercare de stare finală=inițială).