Joao Filipe Costa da Quinta

TP3 Intelligence Artificielle

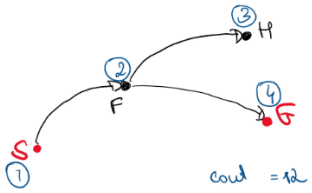
Université de Genève – 20/10/2020

TP_3

exo 1 ①

On commence à l'état S on veut atteindre G

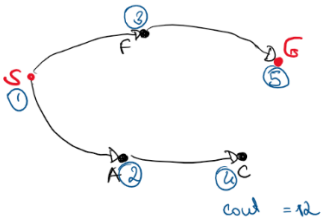
DFS



① [ ] + [A, F] = [A, F]
② [A] + [A, G, H] = [A, A, G, H]
③ [A, A, G] + [ ] = [A, A, G]

T(S) = [A, F]
T(A) = [C, F]
T(F) = [A, G, H]
T(H) = [ ]
T(C) = [D, G]
T(D) = [G]

DFS



① [ ] + [A, F] = [A, F] (on ne met pas le dossier)
② [F,] + [C, G] = [F, C] (déjà exploré)
③ [C] + [A, G, H] = [C, G, H]
④ [G, H] + [D, G] = [G, H, D]

1.2 ①

etat	S	A	C	D	F	G	H
h	10	5	4	3	4	0	2

heuristique : ① ∀ noeud V, h(V) ≥ 0 ✓
② soit V' le noeud à atteindre ⇒ h(V') = 0 ✓

c'est une heuristique, est-elle admissible?

admissible : ① soit $h^*(V)$ la fonction qui retourne le coût réel du noeud V à la solution

etat	S	A	C	D	F	G	H
h	10	5	4	3	4	0	2
h^*	11	9	5	3	5	0	

⇒ $\forall$ noeud V, $0 \leq h(V) \leq h^*(V)$ ✓

→ oui elle est admissible

(2) greedy best first search

greedy prend la meilleure décision possible à chaque noeud V_i sans compter avec les étapes déjà faites,

soit $f(v)$ la fonction d'evaluation $\forall$ noeud v,

$\rightsquigarrow f(v) = h(v)$

A*

fonctionne comme greedy, mais prend en compte le cout total depuis l'origine

$\rightsquigarrow f(v) = h(v) + g(v)$

où $g(v)$ est le cout depuis origine à v

(3) Greedy

① crée la racine de l'arbre de recherche N_0 avec l'état S et $f(N_0) = 10$, ensuite il atteint les noeuds suivants successivement:

② N_0 → F ⇝ $f(F) = 4$
→ A ⇝ $f(A) = 5$

N_0 crée N_1 avec l'état F ⇝ $f(N_1) = 4$

③ N_1 → H ⇝ $f(H) = 2$
→ G ⇝ $f(G) = 0$
→ A ⇝ $f(A) = 5$

N_1 crée N_2, avec l'état G ⇝ $f(N_2) = 0$

Solution :

N_0 (S) —4→ N_1 (F) —5→ N_2 (G)

cout = 12

A*

① crée la racine de l'arbre de recherche N_0 avec l'état S et $f(N_0) = 10 + 0$, ensuite il atteint les noeuds suivants successivement:

② $N_0 \rightarrow A \rightsquigarrow f(A) = h(A) + g(A) = 5 + 2 = 7$

$N_0 \rightarrow F \rightsquigarrow f(F) = h(F) + g(F) = 4 + 7 = 11$

N_0 crée N_1 avec l'état A $\rightsquigarrow f(N_1) = 7$

③ $N_1 \rightarrow F \rightsquigarrow f(F) = h(F) + g(F) = 4 + 10 = 14$

$N_1 \rightarrow C \rightsquigarrow f(C) = h(C) + g(C) = 4 + 6 = 10$

N_1 crée N_2 avec l'état C $\rightsquigarrow f(N_2) = 10$

④ $N_2 \rightarrow D \rightsquigarrow f(D) = h(D) + g(D) = 3 + 9 = 12$

$N_2 \rightarrow G \rightsquigarrow f(G) = h(G) + g(G) = 0 + 11 = 11$

N_2 crée N_3 avec l'état G $\rightsquigarrow f(N_3) = 11$

Solution:

N_0 N_1 N_2 N_3

S $\xrightarrow{2}$ A $\xrightarrow{4}$ C $\xrightarrow{5}$ G coût = 11

④ cf code python