Propriétés de A*

- Si le graphe est fini, A* termine toujours
- Si un chemin vers le but existe, A* va en trouver un
- Si la fonction heuristique *h* retourne toujours un **estimé inférieur ou égal au coût réel à venir**, on dit que *h* est **admissible** :
  - dans ce cas, **A* retourne toujours un chemin optimal**
- *Parfois, on entend par A* la version de l'algorithme avec la condition additionnelle que h soit admissible*
  - A* est alors un *Best-First-Search* où $f(n) = g(n) + h(n)$ **et** $h(n)$ est admissible

Propriétés de A* : recherche en largeur

- En utilisant des coûts des arcs uniformément égaux et strictement positifs (par exemple, tous égaux à 1) et $h(n)$ retournant toujours 0 quelque soit le nœud n, A* devient une recherche en largeur

- *Open* devient une queue LILO (*last in, last out),* en d'autre termes « dernier entré, dernier sorti »

Propriétés de A* : Dijkstra

- En utilisant une heuristique $h(n)$ retournant toujours, A* est équivalent à l'algorithme de Dijkstra
  - cependant, A* s'arrête lorsque le chemin optimal vers le but a été trouvé
  - Dijkstra continuerait jusqu'à avoir trouvé le chemin optimal vers tous les nœuds

Propriétés de A*

- Soit $f^*(n)$ le coût exact (pas un estimé) **du chemin optimal** du nœud initial au nœud but, **passant par *n***
- Soit $g^*(n)$ le coût exact du **chemin optimal du nœud initial au nœud *n***
- Soit $h^*(n)$ le coût exact du **chemin optimal du nœud *n* au nœud but**
- On a donc que $f^*(n) = g^*(n) + h^*(n)$
- Si l'heuristique est admissible, pour chaque nœud *n* exploré par A*, on peut montrer que l'on a toujours $f(n) \leq f^*(n)$

Propriétés de A*

- Si quelque soit un nœud n_1 et son successeur n_2, nous avons toujours

$$h(n_1) \leq c(n_1,n_2) + h(n_2)$$

  où $c(n_1,n_2)$ est le coût de l'arc (n_1,n_2).

  On dit alors que h est **cohérente** (on dit aussi parfois **monotone** – mais c'est en réalité f qui devient monotone). Dans ce cas :

  - h est aussi admissible
  - chaque fois que A* choisit un nœud au début de *open*, A* a alors trouvé le chemin optimal vers ce nœud
    - » le nœud ne sera plus jamais revisité!

Hugo Larochelle et Froduald Kabanza

Propriétés de A*

- Si on a deux heuristiques *admissibles* h_1 et h_2, tel que $h_1(n) < h_2(n)$, alors $h_2(n)$ conduit plus vite au but
  - avec h_2, A* explore moins ou autant de nœuds avant d'arriver au but qu'avec h_1

- Si h n'est pas admissible, soit b la borne supérieure sur la surestimation du coût, c-à-d. on a toujours $h(n) \leq h^*(n) + b$:
  - A* retournera une solution dont le coût est au plus b de plus que le coût optimal, c-à-d., A* ne se trompe pas plus que b sur l'optimalité.

Propriétés de A*

- Si $h(n) = h^*(n)$, pour tout état n, l'optimalité de A* est garantie

- Étant donné une fonction heuristique non admissible, l'algorithme A* donne toujours une solution lorsqu'elle existe, mais il n'y a pas de certitude qu'elle soit optimale

Définition générique de f

- Selon le poids que l'on veut donner à l'une ou l'autre partie, on définie *f* comme suit :

$$f(n) = (1-w)*g(n) + w*h(n)$$

  où *w* est un nombre réel supérieur ou égal à 0 et inférieur ou égal à 1

- Selon les valeurs qu'on donne à *w*, on obtient des algorithmes de recherche classique :
  - ***Dijkstra*** : *w* = 0 ($f(n) = g(n)$)
  - ***Greedy best-first search*** : *w* = 1 ($f(n) = h(n)$)
  - ***A**** : *w* = 0.5 ($f(n) = g(n) + h(n)$)