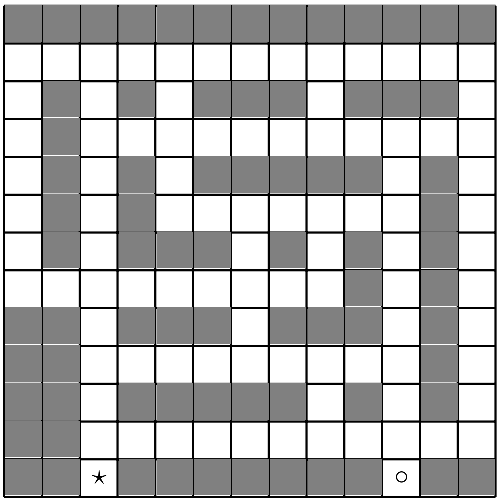
Série 1 : Recherche Heuristique

Exercice 1: Modélisation de graphes d'états

On s'intéresse pour commencer à un labyrinthe proposé par Robert Abbott. Comme vous l'aurez deviné, il s'agit d'entrer par la case représentée par $\circ$ et de sortir par la case représentée par $\star$. Facile... mais l'histoire va un peu se compliquer.
Je vous invite à consulter son site (`http://www.logicmazes.com/`) pour de nombreux autres exemples.



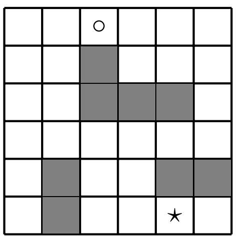
1. Quel modélisation de ce problème pouvez-vous proposer ? Quel est le facteur de branchement de votre problème ?

2. Résoudre le problème en utilisant une stratégie de recherche *greedy best-first* (expansion du noeud de la frontière avec la plus petite valeur de h) en prenant pour h la distance de Manhattan.

3. A présent, changeons légèrement les règles : il est désormais interdit de tourner à gauche (et de faire demi-tour). Quelle modification devez vous apporter à votre modélisation ? Quelle est la conséquence sur le nombre d'état ? L'heuristique employée reste-t-elle admissible ? (Au fait, avez-vous trouvé la solution ?)

Exercice 2: Recherche de chemins

Abordons à présent un environnement moins contraint. L'objectif est encore une de vous déplacer de $\circ$ jusqu'à $\star$. Les déplacements autorisés sont verticaux et horizontaux, pour commencer.

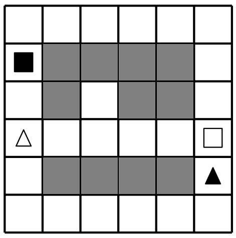


1. Vous souhaitez utiliser l'algorithme A^* pour résoudre ce problème : vaut-il mieux utiliser la distance de Manhattan ou la distance euclidienne ?

2. Déroulez l'algorithme A^*, en indiquant à chaque étape l'état de la *frontière* courante et de la *réserve*. Lorsque plusieurs choix s'offrent à vous pour le noeud à étendre, vous être libres de faire le choix le plus "malin".

3. Pour aller plus loin : imaginons maintenant que notre agent ne se déplace pas de manière discrète sur une grille, mais comme il le souhaite dans le plan. Quelle technique pourriez-vous imaginer pour trouver le plus court chemin dans ce cas ?

4. On considère enfin que deux agents souhaitent se déplacer : l'agent triangle souhaite atteindre le triangle noir, tandis que l'agent carré souhaite atteindre le carré noir. Quel sera le comportement observé si chaque agent planifie indépendamment selon A^*, et replanifie en cas de blocage ?



Exercice 3: Autour des heuristiques

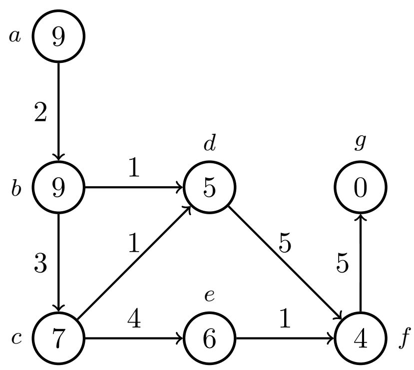
1. Montrer que, comme nous l'avons affirmé en cours, sous l'hypothèse de consistance de l'heuristique, les valeurs de $f(n)$ sont croissantes (au sens large) sur les noeuds n composant n'importe quel chemin vers un noeud but.

2. Montrer que consistance implique l'admissibilité d'une heuristique.

3. Que se passe-t-il si l'heuristique est *parfaite*, c'est-à-dire qu'elle évalue toujours exactement la longueur du chemin jusqu'à l'état but ?

4. Dans de nombreuses situations, la garantie d'optimalité n'est pas cruciale et on peut alors utiliser des heuristiques non-admissibles qui permettent d'accélérer le traitement. Pourriez-vous en tester une sur un des exemples précédents ?

5. Sur l'exemple suivant, la valeur indiquée dans le noeud est la valeur de l'heuristique. La valeur sur les arcs est la coût du déplacement d'un noeud à l'autre. L'heuristique donnée est-elle admissible ? Est-elle consistante ?



6. Une technique connue permettant d'assurer que les valeurs de f sont croissantes (au sens large) d'un nœud quelconque à un nœud but est de simplement mettre à jour les heuristiques en appliquant l'ajustement de valeur suivant (avec n nœud de père p) :

$$h'(n) = \max(h(n), h'(p) - cost(p,n))$$

Cette méthode est communément appelée le *MaxPath trick.* Appliquez là sur l'exemple précédent.