

TD10. Théorie des jeux

Exercice 1. Le jeu du tic-tac-toe

Dans le jeu du tic-tac-toe (aussi appelé jeu du morpion), deux joueurs s'affrontent sur une grille 3×3 initialement vide. Ils doivent remplir chacun à leur tour une case de la grille avec le symbole qui leur est attribué : $\times$ ou $\circ$. Le gagnant est celui qui arrive à aligner trois symboles identiques, horizontalement, verticalement ou en diagonale. C'est le joueur $\times$ qui commence.

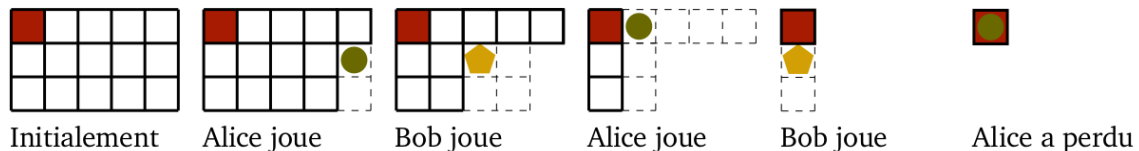
On considère la position suivante, dans laquelle c'est au tour du joueur $\times$.

$\circ$		$\times$
		$\times$
		$\circ$

1. Exhiber une partie depuis cette position remportée par le joueur $\circ$.
2. À l'aide du calcul des attracteurs, montrer que le joueur $\times$ a une stratégie gagnante depuis cette position.

Exercice 2. Le jeu de Chomp

Chomp est un jeu se jouant avec une tablette de chocolat de dimensions $n \times m$ (i.e. à n lignes et m colonnes). Chaque joueur, à son tour, doit choisir un carreau non encore mangé de la tablette, et doit manger tous les carrés se trouvant en dessous à droite de celui-ci (au sens large). Le carreau en haut à gauche est pimenté, celui qui doit le manger a perdu. Ci-dessous un exemple de déroulé d'une partie de Chomp sur une tablette 3×5 .



1. Donner une stratégie gagnante dans le cas d'une tablette de dimensions $1 \times m$ avec $m > 1$.
2. À l'aide d'un graphe qui décrit tous les coups possibles, trouver une stratégie gagnante pour une tablette 2×2 .
3. Même question avec une tablette 2×3 .
4. On se propose d'étudier le cas $2 \times m$. Déterminer une stratégie gagnante.
5. Montrer que quelque soit la tablette initiale, l'un des deux joueurs possède une stratégie gagnante.
6. Montrer que les stratégie gagnantes le sont pour le joueur 1.
7. Déterminer une stratégie gagnante si la tablette est de dimension $n \times n$.

Exercice 3. Le jeu des allumettes généralisé

On considère le jeu des allumettes paramétré par deux entiers $n \in \mathbb{N}^*$ et $k \in \{1, \dots, n\}$.

Le jeu se déroule avec n allumettes initiales, et à chaque tour un joueur doit retirer entre 1 et k allumettes.

On représente un état du jeu par un couple $(i, j) \in \{0, \dots, n\} \times \{1, 2\}$, où i représente le nombre d'allumettes, et j le joueur dont c'est le tour.

1. Montrer que le jeu des allumettes généralisé n'admet pas de partie nulle ni de partie infinie.
2. Démontrer que $Att^1 = \{(i, 1) \mid i \not\equiv 1[k+1]\} \cup \{(i, 2) \mid i \equiv 1[k+1]\}$ et que $Att^2 = \{(i, 2) \mid i \not\equiv 1[k+1]\} \cup \{(i, 1) \mid i \equiv 1[k+1]\}$.
3. En déduire qui du joueur 1 ou 2 admet une stratégie gagnante selon la valeur de n . Donner cette stratégie.

Exercice 4. Algorithme Min-Max en OCaml

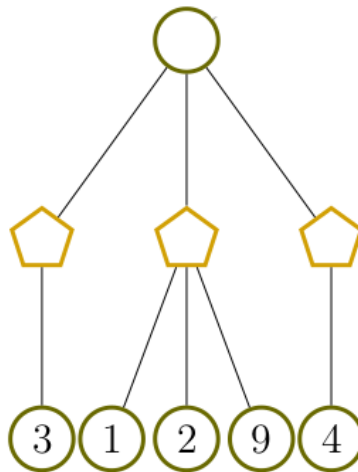
On souhaite fournir une implémentation OCaml de l'algorithme Min-Max.

On définit les types suivants pour représenter les arbres de Min-Max.

```
1 type joueur = J1 | J2
2
3 type arbre_jeu =
4   Feuille of joueur * int | Noeud of joueur * arbre_jeu list
```

On remarque que dans de tels arbres les nœuds internes ne contiennent pas de valeurs. En effet seules les feuilles sont annotées avec une valeur, que l'on peut imaginer être celle fournie par une fonction d'heuristique.

1. Donner une représentation OCaml de l'arbre suivant, où les nœuds ronds correspondent aux états où c'est au joueur 1 de jouer et les nœuds pentagonaux ceux où c'est au joueur 2 de jouer.



2. Proposer une fonction `min_max : arbre_jeu -> int` prenant en argument un arbre de jeu et renvoyant le score de la racine du point de vue du joueur 1.
3. Écrire une fonction `alpha_beta : arbre_jeu -> int -> int -> int` qui implémente l'élagage α - β et renvoie le score de la racine du point de vue du joueur 1.