Jeux à deux joueurs

Quentin Fortier

January 21, 2026

Jeux à deux joueurs

Un puits est un sommet sans arête sortante (de degré sortant nul).

Définition

Un jeu à deux joueurs est un quadruplet (G, s_0, T_0, T_1) où :

- $G = (S, A)$ est un graphe orienté biparti tel que $S = S_0 \sqcup S_1$ et $A \subset S_0 \times S_1 \cup S_1 \times S_0$, où S_i est l'ensemble des configurations du jeu où c'est le joueur i qui doit jouer.

Jeux à deux joueurs

Un puits est un sommet sans arête sortante (de degré sortant nul).

Définition

Un jeu à deux joueurs est un quadruplet (G, s_0, T_0, T_1) où :

- $G = (S, A)$ est un graphe orienté biparti tel que $S = S_0 \sqcup S_1$ et $A \subset S_0 \times S_1 \cup S_1 \times S_0$, où S_i est l'ensemble des configurations du jeu où c'est le joueur i qui doit jouer.
- $s_0 \in S$ est l'état initial.

Jeux à deux joueurs

Un puits est un sommet sans arête sortante (de degré sortant nul).

Définition

Un jeu à deux joueurs est un quadruplet (G, s_0, T_0, T_1) où :

- $G = (S, A)$ est un graphe orienté biparti tel que $S = S_0 \sqcup S_1$ et $A \subset S_0 \times S_1 \cup S_1 \times S_0$, où S_i est l'ensemble des configurations du jeu où c'est le joueur i qui doit jouer.
- $s_0 \in S$ est l'état initial.
- Pour $i \in \{0, 1\}$, $T_i \subset S$ est l'ensemble des puits qui sont les états gagnants pour le joueur i. On suppose $T_0 \cap T_1 = \emptyset$.

Jeux à deux joueurs

Un puits est un sommet sans arête sortante (de degré sortant nul).

Définition

Un jeu à deux joueurs est un quadruplet (G, s_0, T_0, T_1) où :

- $G = (S, A)$ est un graphe orienté biparti tel que $S = S_0 \sqcup S_1$ et $A \subset S_0 \times S_1 \cup S_1 \times S_0$, où S_i est l'ensemble des configurations du jeu où c'est le joueur i qui doit jouer.
- $s_0 \in S$ est l'état initial.
- Pour $i \in \{0, 1\}$, $T_i \subset S$ est l'ensemble des puits qui sont les états gagnants pour le joueur i. On suppose $T_0 \cap T_1 = \emptyset$.
  On note T l'ensemble des puits de G, qu'on appelle aussi états finaux. Les états finaux qui ne sont pas des éléments de T_0 ni T_1 sont appelés états nuls.

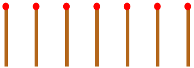
Les joueurs peuvent être nommés 0 et 1, A et B, Alice et Bob...

Exercice

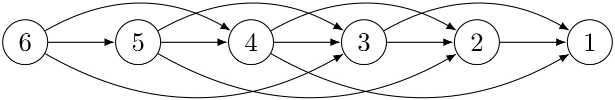
Montrer que tout graphe orienté acyclique possède un puits.

Jeux à deux joueurs

Exemple (jeu de Nim) : il y a n allumettes. Chaque joueur peut retirer 1, 2 ou 3 allumettes. Le joueur qui retire la dernière allumette a perdu.

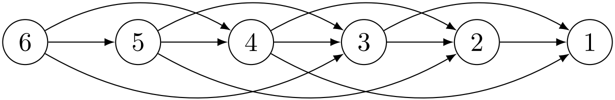


On peut représenter le jeu par un graphe où les sommets sont les configurations et les arêtes sont les coups possibles :



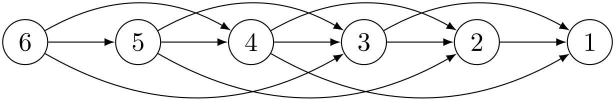
C'est un graphe acyclique, chaque chemin correspondant à une séquence de coups.

Jeux à deux joueurs

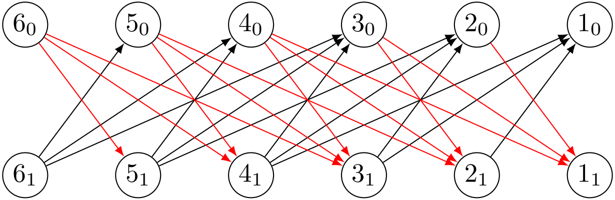


Pour caractériser parfaitement une situation de jeu, il faut savoir quel joueur doit jouer.

Jeux à deux joueurs



Pour caractériser parfaitement une situation de jeu, il faut savoir quel joueur doit jouer. On peut donc considérer le graphe biparti où les sommets sont dupliqués, pour chaque joueur :



Les arêtes rouges correspondent aux coups possibles pour le joueur 0.

Stratégie

Définition

- Une partie est un chemin d'un état initial vers un puits.

Stratégie

Définition

- Une partie est un chemin d'un état initial vers un puits.
- Une stratégie pour le joueur i est une fonction $f : S_i \setminus T \to S_{1-i}$ telle que $\forall s \in S_i, (s, f(s)) \in A$.

Stratégie

Définition

- Une partie est un chemin d'un état initial vers un puits.
- Une stratégie pour le joueur i est une fonction $f : S_i \setminus T \to S_{1-i}$ telle que $\forall s \in S_i, (s, f(s)) \in A$.
- Une stratégie f pour le joueur i est gagnante si quelle que soit la stratégie g pour l'autre joueur $1 - i$, la partie obtenue en jouant alternativement avec les stratégies f et g aboutie en un état final gagnant pour le joueur i.

Stratégie

Définition

- Une partie est un chemin d'un état initial vers un puits.
- Une stratégie pour le joueur i est une fonction $f : S_i \setminus T \to S_{1-i}$ telle que $\forall s \in S_i, (s, f(s)) \in A$.
- Une stratégie f pour le joueur i est gagnante si quelle que soit la stratégie g pour l'autre joueur $1 - i$, la partie obtenue en jouant alternativement avec les stratégies f et g aboutie en un état final gagnant pour le joueur i.

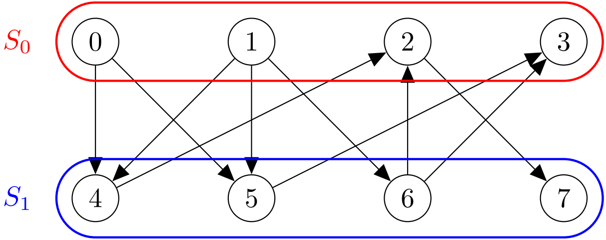
Remarques :

- Une stratégie pour le joueur i consiste à choisir, pour chaque position de S_i, le prochain coup à jouer.
- Une stratégie est sans mémoire : le choix du successeur ne dépend que de l'état actuel et pas du chemin suivi depuis l'état initial.

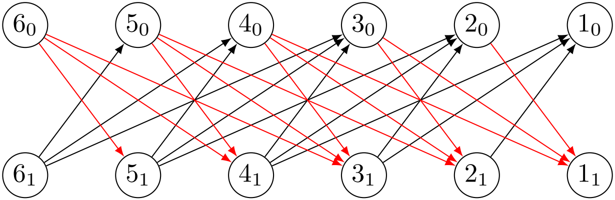
Stratégie

Exercice

Donner une stratégie gagnante pour le joueur 0 dans le jeu suivant (G, s_0, T_0, T_1) où $s_0 = 0,\ T_0 = \{7\},\ T_1 = \{3\}$.



Stratégie



Exercice

On considère le jeu de Nim avec initialement n allumettes et où le joueur 0 commence.

1. Montrer que si $n \equiv 1[4]$ alors le joueur 1 a une stratégie gagnante.
2. Montrer que si $n \not\equiv 1[4]$ alors le joueur 0 a une stratégie gagnante.

Attracteur

Soit (G, s_0, T_0, T_1) un jeu à deux joueurs où $G = (S, A)$.

Définition

Soit (G, s_0, T_0, T_1) un jeu à deux joueurs où $G = (S, A)$. Soit $s \in S$.

- s est une position gagnante pour le joueur i s'il existe une stratégie gagnante pour une partie qui commence en s (c'est-à-dire pour la partie (G, s, T_0, T_1)).
- L'attracteur $A(i)$ du joueur i est l'ensemble des positions gagnantes pour le joueur i.

Exemple : dans le jeu de Nim, $A(0)$ est l'ensemble des configurations où il reste n allumettes avec $n \not\equiv 1[4]$.

Attracteur

Exercice

Montrer que $A(0) \cap A(1) = \emptyset$.

Attracteur

On peut calculer l'attracteur $A(i)$ du joueur 0 par récurrence :

- Si $s \in S_i$ et qu'il existe $(s,t) \in A$ tel que $t \in A(i)$ alors $s \in A(i)$ (le joueur i peut choisir d'aller en t pour gagner).
- Si $s \in S_{1-i} \setminus T$ et que pour tout $(s,t) \in A$, $t \in A(i)$ alors $s \in A(i)$ (quel que soit le coup du joueur $1-i$, il fera gagner le joueur i).

Attracteur

Définition

On définit par récurrence $A_n(i)$ pour $i \in \{0,1\}$ et $n \in \mathbb{N}$ par :

- $A_0(i) = T_i$.
- pour $n \in \mathbb{N}$,

$$\begin{aligned} A_{n+1}(i) = A_n(i) \; & \cup \{s \in S_i \mid \exists t \in A_n(i), (s,t) \in A\} \\ & \cup \{s \in S_{1-i} \setminus T \mid \forall (s,t) \in A, t \in A_n(i)\} \end{aligned}$$

Attracteur

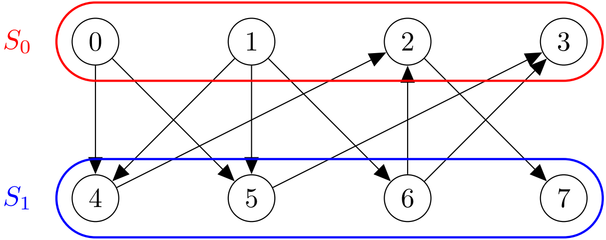
Théorème

$A_n(i)$ converge vers l'attracteur $A(i)$ du joueur i : il existe N tel que pour tout $n \geq N$, $A_n(i) = A(i)$.

Attracteur

Exercice

Calculer l'attracteur du joueur 0 pour le jeu ci-dessous (G, s_0, T_0, T_1) où $s_0 = 0,\ T_0 = \{7\},\ T_1 = \{3\}$.



Attracteur

Une fois calculé $A(i)$ pour $i \in \{0, 1\}$, on peut définir une stratégie optimale f pour le joueur i :

- Si $s \in A(i)$, on considère le plus petit n tel que $s \in A_n(i)$ et on choisit $f(s)$ parmi les $t \in A_{n-1}(i)$ tels que $(s, t) \in A$.
- Sinon, on choisit $f(s)$ quelconque.

Théorème

f est une stratégie gagnante pour le joueur i si et seulement si $s_0 \in A(i)$. Dans ce cas, f permet au joueur i de gagner avec le nombre minimum de coups.

Attracteur

Exercice

Soit (G, s_0, T_0, T_1) un jeu où il n'y a pas d'état nul ni de cycle. Montrer qu'il existe une stratégie gagnante pour le joueur 0 ou le joueur 1.

Attracteur

Calcul des attracteurs en Python avec mémoïsation pour éviter de faire plusieurs appels récursifs sur la même configuration :

```python
def attracteurs(G, S0, T0):
    d = {} # d[v] = True si v est un attracteur
    def aux(v): # détermine si v est un attracteur
        if v not in d:
            voisins = [aux(w) for w in G[v]]
            if G[v] == []:
                d[v] = v in T0
            elif v in S0:
                d[v] = any(voisins) # il existe un true ?
            else:
                d[v] = all(voisins) # tous les éléments sont true
        return d[v]
    return [v for v in G if aux(v)]
```

Complexité :

Attracteur

Calcul des attracteurs en Python avec mémoïsation pour éviter de faire plusieurs appels récursifs sur la même configuration :

```python
def attracteurs(G, S0, T0):
    d = {} # d[v] = True si v est un attracteur
    def aux(v): # détermine si v est un attracteur
        if v not in d:
            voisins = [aux(w) for w in G[v]]
            if G[v] == []:
                d[v] = v in T0
            elif v in S0:
                d[v] = any(voisins) # il existe un true ?
            else:
                d[v] = all(voisins) # tous les éléments sont true
        return d[v]
    return [v for v in G if aux(v)]
```

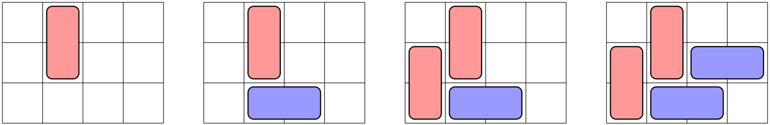
Complexité : $O(|S| + |A|)$ avec des tables de hachage pour `d`, `S0` et `T0`.

Le nombre de configurations $|S|$ peut être très grand

Exemple : Jeu du domineering

Le jeu du domineering est un jeu de plateau où le joueur 0 place un domino vertical et le joueur 1 un domino horizontal. Un joueur qui ne peut plus jouer perd.

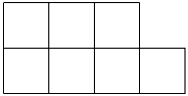
Exemple de partie :



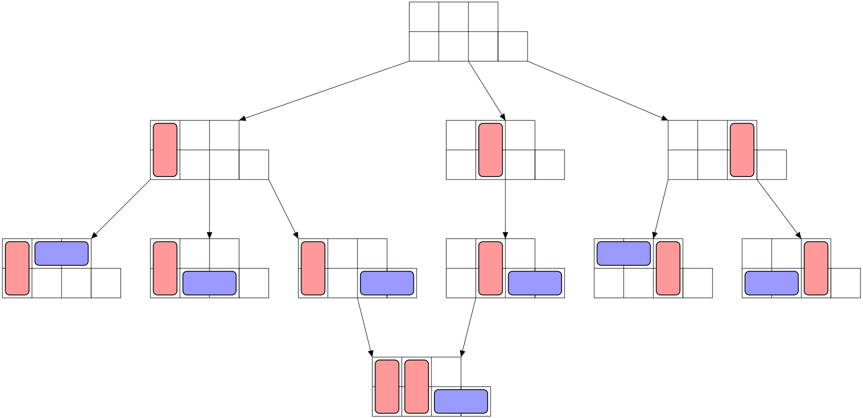
Exemple : Jeu du domineering

Exercice

1. Dessiner le graphe des configurations pour le jeu de domineering sur le plateau ci-dessous. Il n'est pas nécessaire de dupliquer les configurations car on sait quel joueur doit joueur (en comptant le nombre de dominos).
2. Montrer que le joueur 0 a une stratégie gagnante.
3. Trouver l'attracteur du joueur 0.



Exemple : Jeu du domineering



Graphe des configurations