# Jeux à deux joueurs

Thibaut Cantaluppi

February 10, 2025

# Rappels sur les graphes : Définition

Dans ce cours, on considère des graphes orientés. Revoir aussi le cours de première année.

## Définition

Un graphe orienté est un couple $(V, E)$ où $V$ est un ensemble fini de sommets et $E \subseteq V \times V$ est un ensemble fini d'arêtes.

# Rappels sur les graphes : Représentation

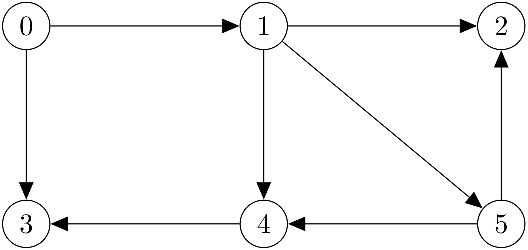
Représentations possibles d'un graphe $G = (V, E)$ :

1. Matrice d'adjacence : par une matrice carrée `A` telle que `A[i][j]` vaut 1 si $(i, j) \in E$ et 0 sinon.
2. Liste d'adjacence : par une liste `L` telle que `L[i]` est la liste des sommets adjacents à $i$.
3. Dictionnaire d'adjacence : par un dictionnaire `D` telle que `D[i]` est la liste des sommets adjacents à $i$.

# Rappels sur les graphes : Représentation

## Exercice

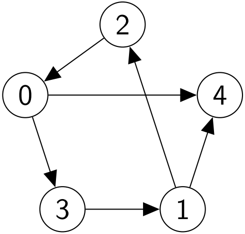
Représenter le graphe suivant des trois manières possibles (matrice d'adjacence, liste d'adjacence, dictionnaire d'adjacence).



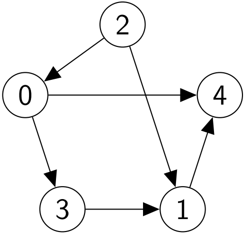
# Rappels sur les graphes : Graphe acyclique

## Définition

- Un **cycle** est un chemin revenant au sommet de départ.
- Un graphe est **acyclique** s'il ne contient pas de cycle.



Graphe ayant un cycle



Graphe acyclique

# Rappels sur les graphes : Graphe acyclique

**Définition**

Un **puit** $t$ est un sommet qui n'a pas de successeur (il n'existe pas d'arête sortant de $t$).

**Exercice**

Montrer que tout graphe acyclique possède un puit.

# Rappels sur les graphes : Graphe biparti

## Définition

Un graphe $G = (V, E)$ est **biparti** s'il existe une partition $V = V_A \sqcup V_B$ telle que toute arête de $E$ a une extrémité dans $V_A$ et une extrémité dans $V_B$.

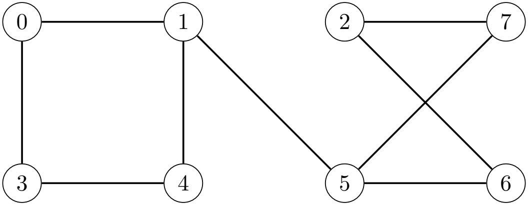
# Rappels sur les graphes : Graphe biparti

## Définition

Un graphe $G = (V, E)$ est **biparti** s'il existe une partition $V = V_A \sqcup V_B$ telle que toute arête de $E$ a une extrémité dans $V_A$ et une extrémité dans $V_B$.

## Question

Montrer que le graphe ci-dessous est biparti, en donnant une partition de ses sommets.



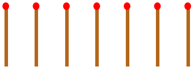
# Jeux à deux joueurs

On s'intéresse à un jeu à deux joueurs (Alice et Bob), qui se joue chacun son tour. On suppose qu'Alice commence. Un joueur a perdu lorsqu'il n'a plus de coup possible.

# Jeux à deux joueurs

On s'intéresse à un jeu à deux joueurs (Alice et Bob), qui se joue chacun son tour. On suppose qu'Alice commence. Un joueur a perdu lorsqu'il n'a plus de coup possible.

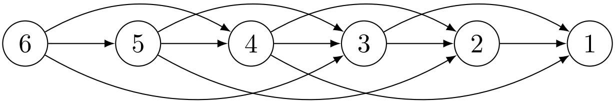
<u>Exemple (jeu de Nim)</u> : il y a $n$ allumettes. Chaque joueur peut retirer 1, 2 ou 3 allumettes. Le joueur qui retire la dernière allumette a perdu.



# Jeux à deux joueurs

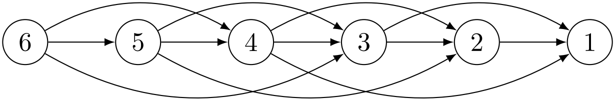
Exemple (jeu de Nim) : il y a $n$ allumettes. Chaque joueur peut retirer 1, 2 ou 3 allumettes. Le joueur qui retire la dernière allumette a perdu.

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



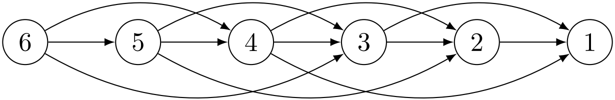

C'est un graphe acyclique, chaque chemin correspondant à une séquence de coups.

# Jeux à deux joueurs

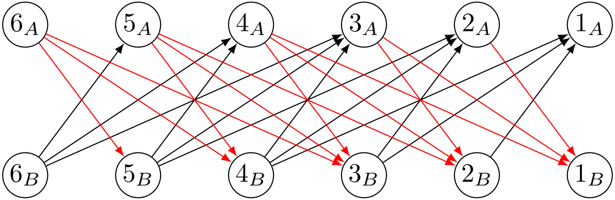



Pour caractériser parfaitement une situation de jeu, il faut savoir aussi qui est le joueur qui doit jouer.

# Jeux à deux joueurs




Pour caractériser parfaitement une situation de jeu, il faut savoir aussi qui est le joueur qui 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 Alice.

# Jeux à deux joueurs : Stratégie

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

## Définition

- Le jeu commence en un **sommet initial** $v \in A$.

# Jeux à deux joueurs : Stratégie

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

## Définition

- Le jeu commence en un **sommet initial** $v \in A$.
- Une **partie** est un chemin commençant en $v$ dont les arêtes sont choisies alternativement par Alice et Bob.

# Jeux à deux joueurs : Stratégie

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

### Définition

- Le jeu commence en un **sommet initial** $v \in A$.
- Une **partie** est un chemin commençant en $v$ dont les arêtes sont choisies alternativement par Alice et Bob.
- L'ensemble $P_A$ des puits de $V_A$ sont les situations où Alice perd.

# Jeux à deux joueurs : Stratégie

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

## Définition

- Le jeu commence en un **sommet initial** $v \in A$.
- Une **partie** est un chemin commençant en $v$ dont les arêtes sont choisies alternativement par Alice et Bob.
- L'ensemble $P_A$ des puits de $V_A$ sont les situations où Alice perd.
- Une **stratégie** pour Alice est une fonction $f : V_A \setminus P_A \to V_B$ telle que $\forall v \in V_A \setminus P_A$, $(v, f(v)) \in E$.
  On définit une stratégie pour Bob de façon symétrique.

# Jeux à deux joueurs : Stratégie

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

## Définition

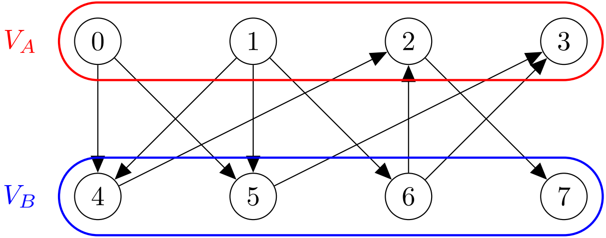
- Le jeu commence en un **sommet initial** $v \in A$.
- Une **partie** est un chemin commençant en $v$ dont les arêtes sont choisies alternativement par Alice et Bob.
- L'ensemble $P_A$ des puits de $V_A$ sont les situations où Alice perd.
- Une **stratégie** pour Alice est une fonction $f : V_A \setminus P_A \to V_B$ telle que $\forall v \in V_A \setminus P_A$, $(v, f(v)) \in E$.
  On définit une stratégie pour Bob de façon symétrique.
- Une **stratégie gagnante** pour Alice est une stratégie $f$ qui permette à Alice de gagner, quel que soit la stratégie de Bob.

Les définitions sont similaires pour Bob, en échangeant $A$ et $B$.

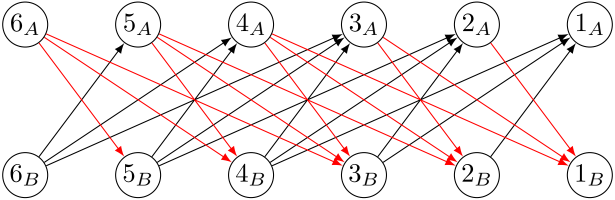
# Jeux à deux joueurs : Stratégie

**Exercice**

Donner une stratégie gagnante pour Alice dans le jeu suivant, où le sommet initial est $0$.



# Jeux à deux joueurs : Stratégie




**Exercice**

On considère le jeu de Nim avec initialement $n$ allumettes.

1. Montrer que si $n \equiv 1[4]$ alors Bob a une stratégie gagnante.
2. Montrer que si $n \not\equiv 1[4]$ Alice a une stratégie gagnante.

# Jeux à deux joueurs : Attracteur

Soit $G = (V, E)$ un graphe biparti acyclique, avec $V = V_A \sqcup V_B$.

**Définition**

1. $v \in V$ est une **position gagnante** pour Alice si elle possède une stratégie gagnante pour une partie qui commence en $v$.
2. L'**attracteur** $A$ d'Alice est l'ensemble des position gagnantes pour Alice.

Les définitions sont similaires pour Bob.

Exemple : l'attracteur d'Alice dans le jeu de Nim est l'ensemble des configurations où il reste $n$ allumettes avec $n \not\equiv 1[4]$.

# Jeux à deux joueurs : Attracteur

On peut calculer l'attracteur d'Alice de proche en proche :

- Soit $A_0 = P_B$ (gagnants pour Alice).

# Jeux à deux joueurs : Attracteur

On peut calculer l'attracteur d'Alice de proche en proche :

- Soit $A_0 = P_B$ (gagnants pour Alice).
- Soit $A_1 = \{u \in V_A \mid \exists (u, v) \in E, v \in A_0\}$ l'ensemble des sommets de $A$ qui permettent à Alice d'aller sur un sommet de $A_0$.

# Jeux à deux joueurs : Attracteur

On peut calculer l'attracteur d'Alice de proche en proche :

- Soit $A_0 = P_B$ (gagnants pour Alice).
- Soit $A_1 = \{u \in V_A \mid \exists (u, v) \in E, v \in A_0\}$ l'ensemble des sommets de $A$ qui permettent à Alice d'aller sur un sommet de $A_0$.
- Soit $A_2 = \{u \in V_B \mid \forall (u, v) \in E, v \in A_1\}$ l'ensemble des sommets de $B$ qui mènent forcément vers un sommet de $A_1$.

# Jeux à deux joueurs : Attracteur

On peut calculer l'attracteur d'Alice de proche en proche :

- Soit $A_0 = P_B$ (gagnants pour Alice).
- Soit $A_1 = \{u \in V_A \mid \exists (u, v) \in E, v \in A_0\}$ l'ensemble des sommets de $A$ qui permettent à Alice d'aller sur un sommet de $A_0$.
- Soit $A_2 = \{u \in V_B \mid \forall (u, v) \in E, v \in A_1\}$ l'ensemble des sommets de $B$ qui mènent forcément vers un sommet de $A_1$.
- ...

# Jeux à deux joueurs : Attracteur

De façon générale :

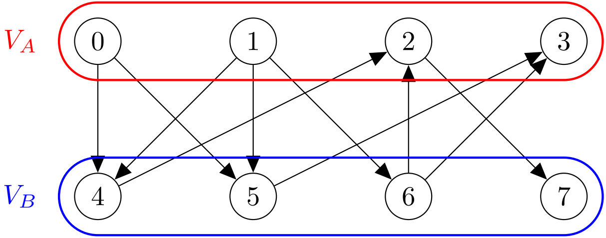
- Si $k$ est pair : $A_{k+1} = \{u \in V_A \mid \exists (u,v) \in E, v \in A_k\}$
- Si $k$ est impair : $A_{k+1} = \{u \in V_B \mid \forall (u,v) \in E, v \in A_k\}$

($A_k$ est l'ensemble des positions gagnantes pour Alice en $k$ coups)
S'il y a $n$ sommets, un chemin possède au plus $n-1$ arêtes d'où :

$$A = \bigcup_{k=0}^{n-1} A_k$$

## Exercice

Donner l'attracteur d'Alice pour le jeu ci-dessous, où $V_A = \{0, 1, 2, 3\}$ et $P_A = \{3\}$.



# Jeux à deux joueurs : Attracteur

Un sommet $v$ est un attracteur dans l'un des trois cas suivants :

- $v \in P_B$.
- $v \in V_A$ et il existe un attracteur $w$ tel que $(v, w) \in E$.
- $v \in V_B$ et pour tout $(v, w) \in E$, $w$ est un attracteur.

# Jeux à deux joueurs : Attracteur

Un sommet $v$ est un attracteur dans l'un des trois cas suivants :

- $v \in P_B$.
- $v \in V_A$ et il existe un attracteur $w$ tel que $(v, w) \in E$.
- $v \in V_B$ et pour tout $(v, w) \in E$, $w$ est un attracteur.

## Exercice

Écrire une fonction `attracteurs(G, fA)` renvoyant la liste des attracteurs d'Alice dans le graphe `G` représenté par dictionnaire d'adjacence, où `fA` est une fonction indiquant si un sommet appartient à $V_A$.

# Jeux à deux joueurs : Attracteur

```python
def attracteurs(G, fA):
    d = {} # d[v] = True si v est un attracteur
    def aux(v): # détermine si v est un attracteur
        if v not in d:
            succ = [aux(w) for w in G[v]]
            if len(G[v]) == 0:
                d[v] = not fA(v)
            elif fA(v):
                # test si il existe (v, w) € E avec w attracteur
                d[v] = False
                for w in G[v]:
                    if aux(w):
                        d[v] = True
            else:
                # test si pour tout (v, w) € E, w est attracteur
                d[v] = True
                for w in G[v]:
                    if not aux(w):
                        d[v] = False
        return d[v]
    return [v for v in G if aux(v)]
```

# Jeux à deux joueurs : Attracteur

On peut aussi détailler la syntaxe par compréhension suivante :

```
return [v for v in G if aux(v)]
```

Qui signifie plus simplement :

```
res_list = []
for v in G:
    if aux(v):
        res_list.append(v)
return res_list
```

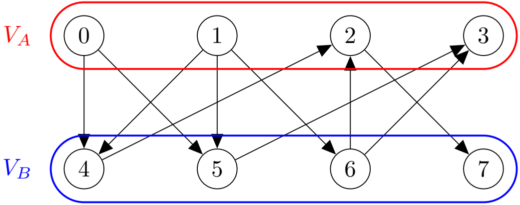
# Jeux à deux joueurs : Attracteur

On peut aussi utiliser `any` et `all` :

```python
def attracteurs(G, fA):
    d = {} # d[v] = True si v est un attracteur
    def aux(v): # détermine si v est un attracteur
        if v not in d:
            succ = [aux(w) for w in G[v]]
            if len(G[v]) == 0:
                d[v] = not fA(v)
            elif fA(v):
                d[v] = any(succ)
            else:
                d[v] = all(succ)
        return d[v]
    return [v for v in G if aux(v)]
```

## Exercice

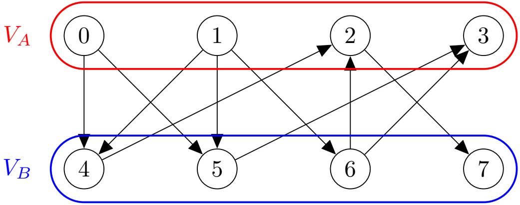
Utiliser la fonction `attracteurs(G, fA)` pour trouver les attracteurs d'Alice dans le jeu ci-dessous.



# Jeux à deux joueurs : Attracteur

## Exercice

Utiliser la fonction `attracteurs(G, fA)` pour trouver les attracteurs d'Alice dans le jeu ci-dessous.




```python
G = {
    0: [4, 5], 1: [4, 5, 6], 2: [7], 3: [],
    4: [2], 5: [3], 6: [3], 7: []
}
attracteurs(G, lambda v: v < 4)
```

# Jeux à deux joueurs : Attracteur

**Exercice**

Utiliser la fonction `attracteurs(G, fA)` pour trouver les attracteurs d'Alice dans le jeu de Nim (avec $n$ allumettes).

**Exercice**

Utiliser la fonction `attracteurs(G, fA)` pour trouver les attracteurs d'Alice dans le jeu de Nim (avec $n$ allumettes).

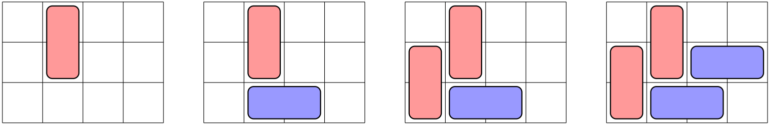
```python
def nim(n):
    G = {}
    for i in range(1, n + 1):
        for j in ['A', 'B']:
            if (i, j) not in G:
                G[(i, j)] = []
            for k in range(i + 1, i + 4):
                if k <= n:
                    G[(i, j)].append((k, 'A' if j == 'B' else 'B')
    return G

attracteurs(nim(9), lambda v: v[1] == 'A')
```

# Jeux à deux joueurs : Attracteur

Remarque : en l'absence de partie nulle, chaque sommet est attracteur pour Alice ou Bob.
Depuis la position initiale, il existe donc soit une stratégie gagnante pour Alice, soit une stratégie gagnante pour Bob.

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

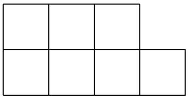
Exemple de partie :



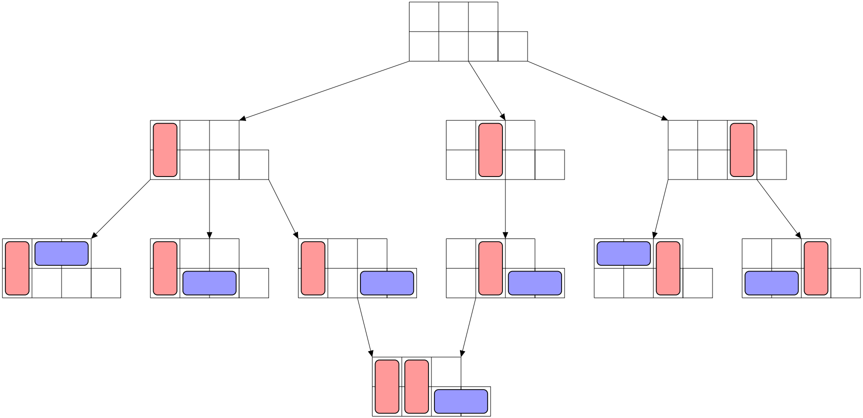
# Jeux à deux joueurs : Jeu du domineering

## 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 qu'Alice a une stratégie gagnante.
3. Trouver l'attracteur d'Alice.



# Jeux à deux joueurs : Jeu du domineering



Graphe des configurations

# Jeux à deux joueurs : Match nul

Dans certains jeux (morpion...), il peut y avoir match nul.
Il faut alors spécifier trois types de puits : gagnants pour Alice, gagnants pour Bob et match nul.

Un sommet $v$ est alors attracteur si un des cas suivants est vérifié :

- $v$ est un puit gagnant pour Alice.
- $v \in V_A$ et il existe un attracteur $w$ tel que $(v, w) \in E$.
- $v \in V_B$ et pour tout $(v, w) \in E$, $w$ est un attracteur.