Documentations complètes
Python3 : `https://docs.python.org/3/`
Nympy : `https://docs.scipy.org/doc/`
Matplotlib : `http://matplotlib.org/`
SQL : `http://sql.sh/`

7
Théorie des jeux

Sommaire

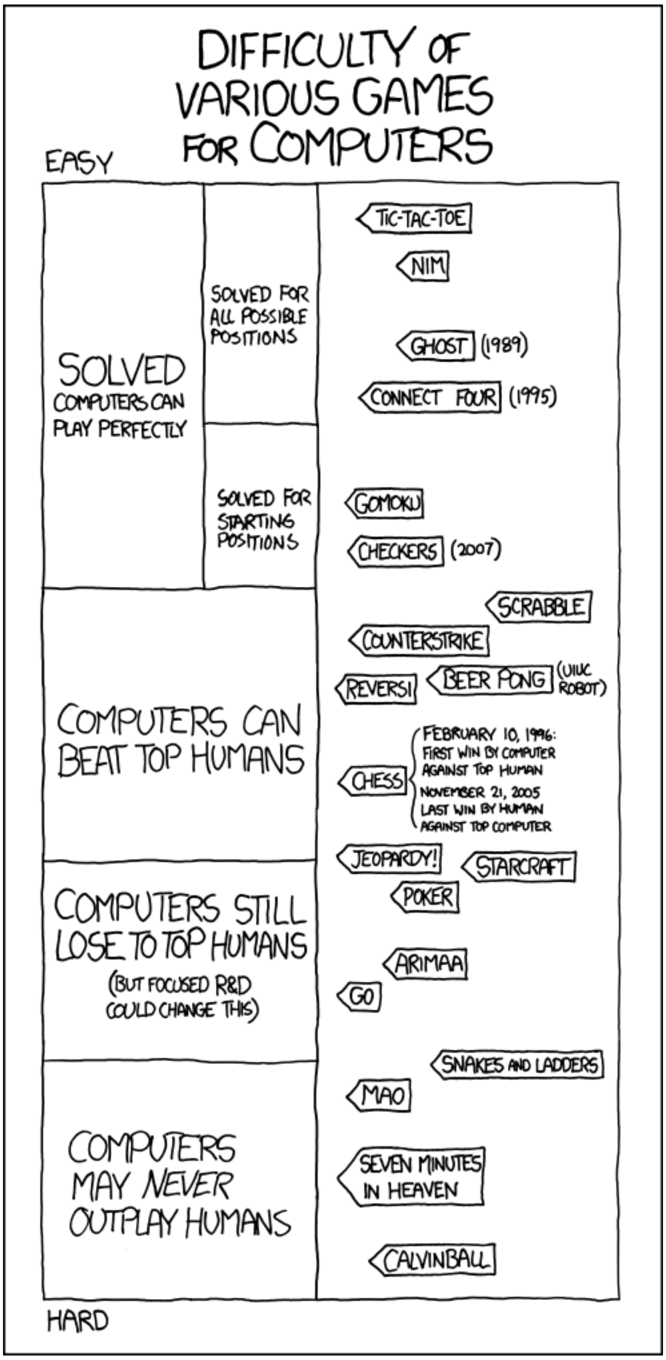


FIGURE 7.1 – From https ://xkcd.com/1002/

7.1 Jeux sur des graphes

7.1.1 Exemples de jeux

Le jeu de Nim

Le jeu de Nim est un jeu de stratégie mathématique pour deux joueurs, dont il existe plusieurs versions. Nous allons considérer la version suivante :

— On dispose sur une table un 12 allumettes.
— L'un après l'autre, chaque joueur retire de la table entre 1 à 3 allumettes.
— Le joueur qui retire la dernière allumette a perdu.

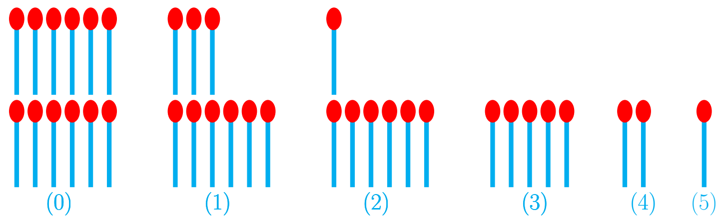


FIGURE 7.2 – Une partie de Nim.

La figure (**7.2**) représente une partie de NIm, pour passer de la configuration (0) à la configuration (1) le premier joueur a enlevé 3 allumettes, puis le second joueur est passé de la configuration (1) à la configuration (2) en retirant 2 allumettes, etc... Le premier joueur a gagné cette partie.

Le jeu de Chomp

Le jeu de Chomp est un jeu de stratégie à deux joueurs inventé par le mathématicien anglais David Gale en 1974. Il se joue avec une tablette de chocolat initialement rectangulaire, posée devant les joueurs et dont le coin supérieur gauche, de coordonnée $(0,0)$, est empoisonné. Chaque joueur choisit à tour de rôle un carré et le mange ainsi que tous les carrés situés à droite et sous ce carré. Autrement dit, si un joeur choisit le carré de coordonnées (i,j), il doit aussi manger tous les morceaux de coordonnées (k,ℓ) avec $k \geq i$ et $\ell \geq j$. Evidemment, le joueur qui est forcé de manger le dernier carré de chocolat empoisonné a perdu.

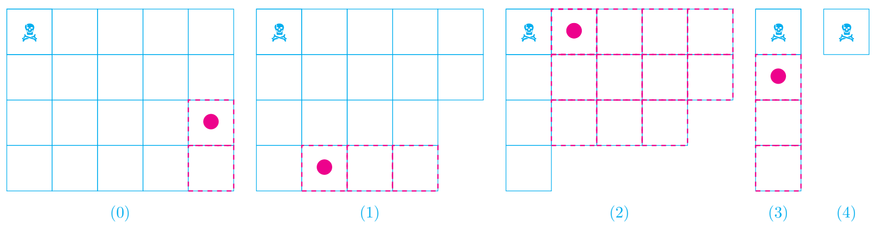


FIGURE 7.3 – Une partie du jeu de Chomp.

La figure (7.3) représente une partie du jeu de Chomp jouée avec une tablette initiale de taille 4×5, le carré empoisonné étant marqué d'un ☠. Le point • représente le carré choisi alternativement par les deux joueurs, et les carrés en pointillée représentent ceux qui doivent être mangé après ce choix. Cette partie est gagnée par le joueur qui a joué en premier.

Le jeu Hex

Le jeu Hex a été inventé indépendamment par deux mathématiciens différents : le mathématicien danois Piet Hein en 1942[i] et le mathématicien américain John Nash en 1947, lorsqu'il était étudiant à Princeton.

i. Piet Hein a inventé le jeu lorsqu'il était prisonnier pendant l'occupation allemande du Danemark pendant la Seconde Guerre mondiale. Il a créé le jeu pour distraire son esprit de la vie en captivité et pour apporter de la joie à ses compagnons

Le jeu Hex est un jeu de stratégie pour deux joueurs qui se joue sur un plateau en forme de losange composé de cases hexagonales. Les quatre sommets du losange sont le nord N, le sud S, l'est E et O l'ouest, et ses côtés sont NE, NO, SE, SO. À tour de rôle, chaque joueur pose un pion sur une des cases encore libres. Le premier joueur parvenant à connecter deux côtés opposés par une chaîne continue couvertes par ses pions, c'est-à-dire posés sur des cases adjacentes, a gagné.

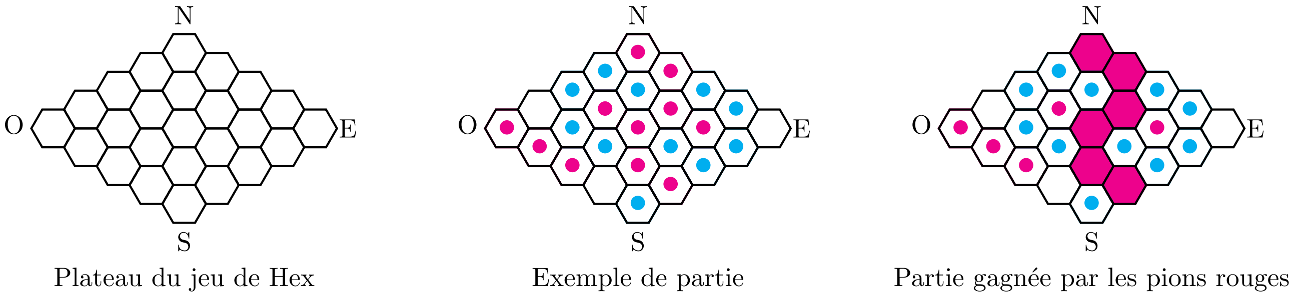


FIGURE 7.4 – Plateau du Hex et exemple d'une partie gagnée par les rouges.

La figure (7.4) présente un plateau de Hex de taille 5×5, soit 25 cases, ainsi qu'un exemple de partie où le joueur qui a les rouges a déposé 11 pions et celui qui a les bleus en a déposé 11 aussi. Cette partie est gagné par celui qui a les rouges puisqu'il a relié le côté NO au côté SE.

Points communs

Les trois jeux cités en exemple, ont un certain nombre de points communs :

- ce sont des jeux à deux joueurs antagonistes jouant alternativement, contrairement à des jeux coopératifs
- ce sont des jeux sont à information parfaite, c'est-à-dire qu'à tout instant d'une partie, chacun des joueurs a une vision complète de l'état du jeu, ce qui n'est pas le cas de la plus part des jeux de cartes, mais celui des échecs ou du jeu de dames.
- ce sont des jeux sans hasard, ce qui implique que seul les actions passées détermine l'état du jeu à un instant donné, ainsi que les « possibilités » de mouvement d'un joueur.
- ce sont des jeux qui se finissent en un nombre fini de coups.
- ce sont des jeux à somme nulle : si un joueur gagne l'autre perd.

7.1.2 Modélisation des jeux

Nous allons nous intéresser ici à des jeux qui ne présentent qu'un nombre fini de configurations possibles, ce qui est le cas des jeux cités en exemple.

Comme c'est la coutume nous appelerons Adam et Ève les deux joueurs.

A un tel jeu on peut associer un graphe orienté fini $G_c = (S_c, A_c)$,, dit , tous ses sommets représentent une configuration possible du jeu, et si deux sommets sont relié par une arrête c'est que l'on peut passer d'une configuration à une autre en un coup.
Le graphe des configurations d'un jeu ne permet pas de savoir quel joueur joue, ni quel mouvement, à un instant donné.

Aussi, pour modéliser un jeu on lui associe un triplet (G, S_a, S_e), que l'on appelle arène, où $G = (S, A)$ est graphe orienté fini dont tous les sommets ont un successeur, et deux sous-ensembles S_a et S_e de S tels que $S = S_a \cup S_e$ et $S_a \cap S_e = \emptyset$, les sommets de S_a, reps. S_e, sont dits appartenir à Adam, resp. à Ève. Comme un

prisonniers. Il a appelé son jeu "Con-tac-tix" et l'a présenté pour la première fois en 1942.

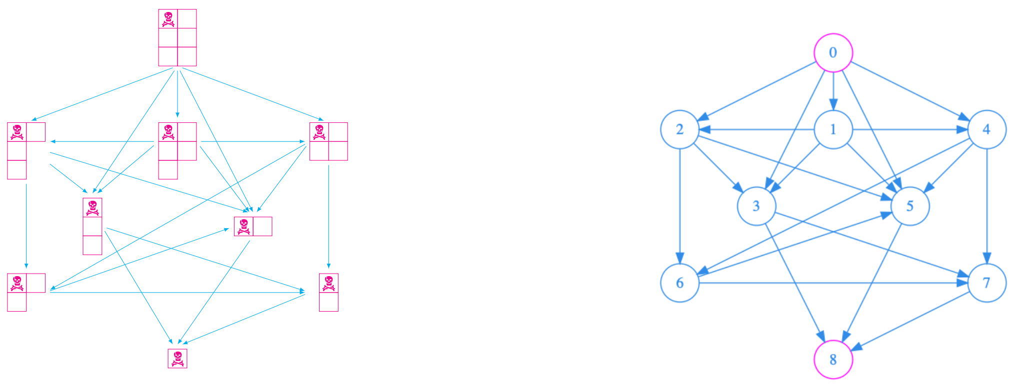


FIGURE 7.5 – Graphe des configurations du jeu de Chomp avec une tablette initiale de taille 4×2.

joueur ne peut pas jouer deux fois de suite un arc relie nécessairement un sommet de S_a et un sommet de S_e. AUtrement dit le graphe G est biparti.

Chaque sommet de G représente une configuration possible du jeu, que l'on appelle position, l'une d'elles, s_0, est appelée position initiale ou position de départ du jeu.

De manière heuristique, une partie se déroule ainsi : un jeton est placé sur le sommet s_0, et il est déplacé par chaque joueur de sommet en sommet, par une succession de mouvements. Un mouvement consiste à déplacer je jeton le long d'un arc de A, lorsque le jeton est dur un sommet s, le joueur à qui s appartient le déplace sur un successeur de s, et ainsi de suite. Une partie est le chemin parcouru par le jeton. Cela suppose que l'on choisisse quel est le joueur qui commence la partie.

La figure (7.6) montre l'arène du jeu de Nim si c'est Ève qui commence, les positions d'Ève sont indexées par un « e », celles d'Adam par un « a ».

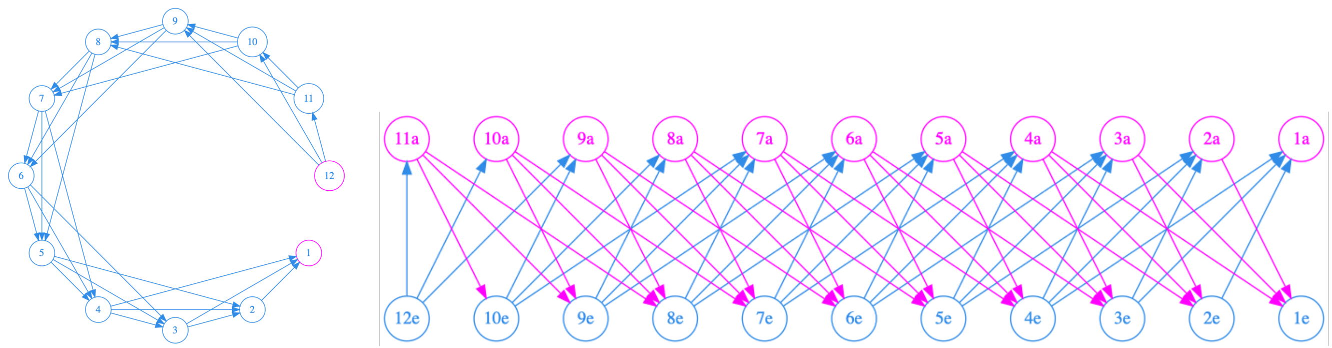


FIGURE 7.6 – À gauche le graphe des configurations du jeu de Nim. À droite l'arène du jeu si Ève commence.

Sur la figure (7.7) on a tracé une partie du jeu de Nim, commencée par Ève et gagnée par Ève.

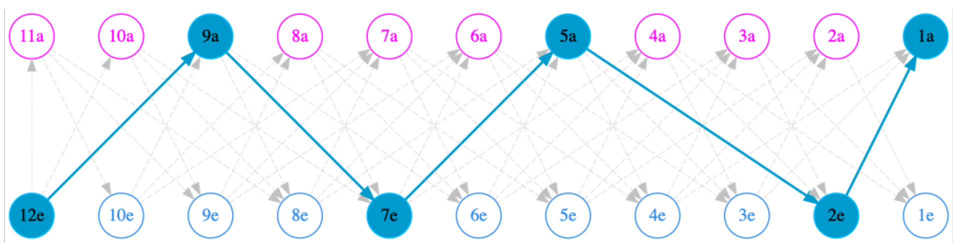


FIGURE 7.7 – Une partie du jeu de Nim.

Pour compléter la description du jeu, on se donne un ensemble $\mathcal{W}$ de chemins du graphe de l'arène. Une partie est gagnée par Ève si elle est dans $\mathcal{W}$, sinon elle est gagnée par Adam. Le jeu est donc complètement décrit par $(G, S_a, S_e, s_0, \mathcal{W})$.

Une stratégie positionnelle pour Ève est une application $\varphi : S_e^{>0} \to S_a$, où $S_e^{>0} \subset S_e$ est formé de sommets dont le demi-degré extérieur est non nul, telle que $(s, \varphi(s)) \in A$ pour tout $s \in S_e^{>0}$. Si φ est une stratégie pour Ève, on dit qu'une partie $s_0 \to s_1 \to s_2 \to \cdots \to s_n$ est jouée selon φ si pour tout couple (s_k, s_{k+1}) de positions consécutives dans la partie avec $s_k \in S_e$, on a $s_{k+1} = \varphi(s_k)$.

Suivre une stratégie consiste donc à choisir à l'avance, pour chaque position, le mouvement à jouer. Une stratégie φ pour Ève est dite gagnante si toute partie qui est jouée selon φ est gagnée par Ève. On définit de même une stratégie positionnelle et une stratégie gagnante pour Adam.

Dans le jeu de Nim à 12 allumettes une stratégie gagnante est donnée par :

$$\varphi(12e) = 9a, \varphi(8e) = \varphi(7e) = \varphi(6e) = 5a, \varphi(4e) = \varphi(3e) = \varphi(2e) = 1_a.$$

La figure (7.8) présente en gras les arrêtes et les sommets de cette stratégie :

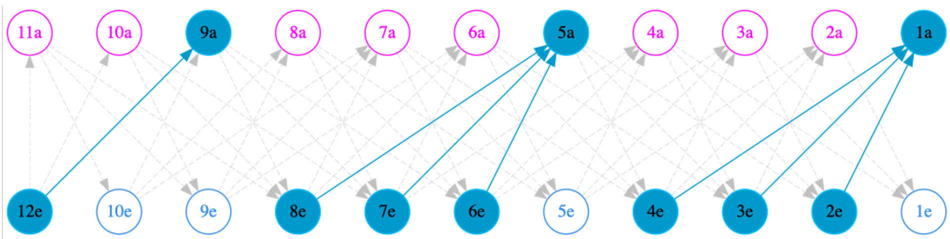


FIGURE 7.8 – Stratégie gagnate pour Ève.

On appelle position gagnante pour Ève un sommet $s \in S_e$, tel qu'il existe une stratégie gagnante pour elle qui débute en s. Sur la figure (7.8) on voit que les postions $10e$ et $9e$ ne sont pas gagnantes pour Ève, on dit qu'elles sont perdantes, alors que les position $12e$, $8e$, $7e$ et $6e$ le sont.

Un autre exemple. Sur le figure (7.9), l'arène du jeu de Chomp pour une tablette 2×2 est représenté, les positions d'Ève sont en rouge et celles d'Adam en bleu. On voit que les positions $0e$, $4e$ et $6e$ sont gagnantes pour Ève.

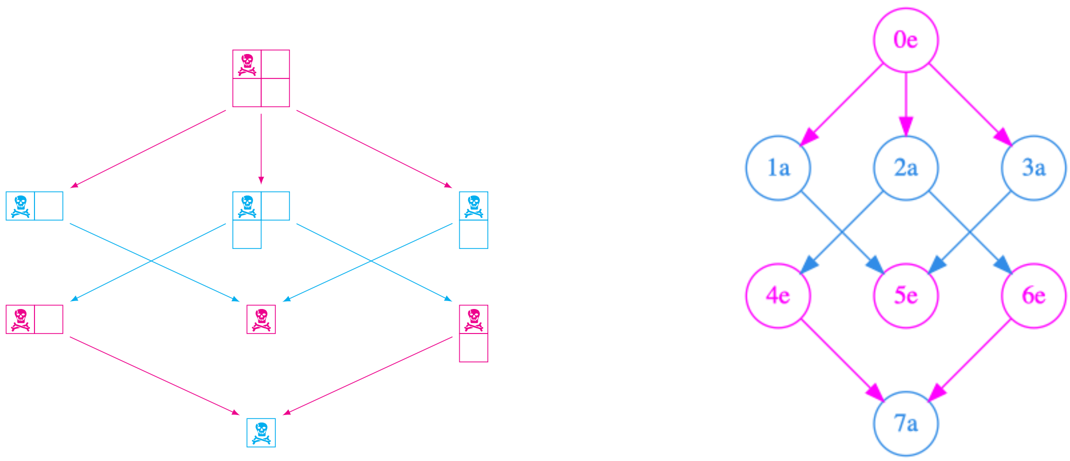


FIGURE 7.9 – Positions gagnantes pour le jeu de Chomp 2×2.

Théorème 7.1 (Jhon Nash)**.**

Il existe une stratégie gagnante pour le joueur qui commence la partie, mais il n'en existe pas pour le second joueur.

La démonstration de Nash n'est pas constructive [ii] , c'est-à-dire qu'elle ne donne pas la stratégie gagnante mais prouve juste sont existence. C'est une démonstration par l'absurde qui repose essentiellement sur le principe de « vol de stratégie » [iii] . Aujourd'hui on connait des stratégies gagnantes pour des plateaux de taille inférieure à 9×9.

Théorème 7.2.

Il existe une stratégie gagnante pour le premier joueur au jeu de Chomp si la tablette est au moins de taille 2×2.

Là encore la preuve de ce théorème repose sur le « vol de stratégie », et est malheureusement non constructive.

7.1.3 Jeux d'accessibilité

On s'intéresse dans cette partie aux arènes dont le graphe est acyclique, ce qui impose aux parties d'être finies. Dans un jeu, où on ne peut revenir en arrière si l'arène acyclique, c'est le cas des jeux cités en exemple.

Une condition de victoire, ou d'atteignabilité pour Ève, est un sous-ensemble $V_e \subset S$, tel que si Ève atteint une de ces positions elle remporte la partie. On définit de même une condition de victoire pour Adam. Par exemple pour le jeu de Nim on a $V_e = \{1a\}$ et $V_a = \{1e\}$.

Pour résoudre un jeu, c'est-à-dire trouver une stratégie gagnante dans un jeu d'accessibilité à deux joueurs, on utilise la notion d'attracteur. On va s'intéresser aux stratégie gagnantes d'Ève. Pour un graphe $G = (S, A)$, avec $S = S_e \cup S_a$, et $S_a \cap S_e = \emptyset$, on commence par définir par induction la suite d'ensembles :

$$\left\{\begin{array}{lcl} Attr_0 & = & V_e \\ Attr_{i+1} & = & Attr_i \cup \underbrace{\{s \in S_e \mid t \in Attr_i,\ (s,t) \in A\}}_{\text{Les prédecesseurs } s \in S_e \text{ des sommets de } Attr_i} \cup \underbrace{\{s \in S_a \mid \forall t \in S,\ (s,t) \in A \Rightarrow t \in Attr_i\}}_{\text{Les sommets } s_a \text{ dont tous les successeurs sont dans } Attr_i.} \end{array}\right.$$

L'ensemble $Attr_{i+1}$ est donc constitué :

— des sommets de $Attr_i$;

— les positions d'Ève pour lesquelles il existe au moins un arc permettant de rejoindre $Attr_i$;

— des positions d'Adam pour lesquelles tous les arcs sortant ont pour extrémité un sommets de $Attr_i$.

La suite $(Attr_i)_{i \in \mathbb{N}}$ étant croissante et bornée, elle converge vers $Attr = \bigcup_{i=0}^{\infty} Attr_i$. L'ensemble $Attr$ est appelé Attracteur pour Ève, son complémentaire est qualifié de piège pour Ève.

Théorème 7.3.

Les sommets de $Attr$ sont exactement les positions gagnantes pour Ève.

ii. Il existe une autre démonstration basée sur un théorème combinatoire de Hales-Jewett, mais qui n'est pas non plus constructive.

iii. Vous pourrez la lire ici `https://images.math.cnrs.fr/Le-jeu-de-Hex.html`

Preuve 7.1.

On commence par définir l'application $rg : S \to \mathbb{N}$ par $\text{rg}(s) = \min\{i \in \mathbb{N} \mid s \in Attr_i\}$, avec la convention que $\min \emptyset = \infty$. On appelle rang de s la valeur de $rg(s)$.

Montrons par récurrence forte sur le rang que les sommets de $Attr$ sont des positions gagnantes pour Éve :

— tous les sommets de rang nul sont dans V_e, par définition de V_e le résultat est acquis.

— soit $i \in \mathbb{N}$, supposons le résultat vrai pour tous les sommets t de $Attr$ de rang $rg(t) \leq i$. Soit alors un sommet $s \in Attr_{i+1} \setminus Attr_i$ de rang $i+1$. ALors de deux choses l'une :

Soit $s \in S_e$ **:** par définition de $Attr_{i+1}$, il existe alors $t \in Attr_i$ tel que $(s,t) \in A$, donc Ève peut se déplacer suivant l'arc (s,t), et comme la position t est gagnate, il en est de même de la position s.

Soit $s \in S_a$ **:** par définition de $Attr_{i+1}$, tous les arcs issus de s ont une extrémité t dans $Attr_i$, qui par hypothèse de récurrence est une position gagnate pour Ève, donc il en est de même pour s.

Maintenant, si s n'est pas un attracteur pour ve, alors de deux choses l'une encore une fois :

Soit $s \in S_e$ **:** alors quelque soit le choix d'Ève , il mènera à une position $t \notin Attr$, par définition.

Soit $s \in S_a$ **:** alors Adam, peut se déplacer sur un sommet $t \notin Attr$.

De cette manière Adam, peut éviter toutes les positions gagnantes d'Ève.

L'algorithme suivant permet de calculer l'attracteur.

Algorithm 6: Calcul de l'attracteur

def $parcours(s)$**:**
  if $s \notin \mathcal{A}$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{s\}$
    for $t \in \Gamma^+_{G^T}(s)$ **do**
      $d^+_G(s) \leftarrow d^+_G(s) - 1$
      if $t \in V_e$ **or** $d^+_G(s) == 0$ **then**
        $parcours(t)$

def $attracteur(G, S_a, V_a)$**:**
  Data: $G = (S, A)$, $S_a V_a$
  Result: L'attracteur associé à V_e

  $\mathcal{A} \leftarrow \emptyset$
  $G^T = transpose(G)$
  for $s \in V_e$ **do**
    $parcours(s)$

Cet algorithme a une complexité en $O(\mid S \mid + \mid A \mid)$. Vous traiterez plus en détail la question de la construction des attracteurs en TD.

Exercice 7.1. Déterminer l'attracteur de V_e du graphe de Nim (figure (**7.6**)).

7.2 Algorithme Min-Max

Nous continuons dans cette partie à nous intéresser à des jeux à information complète, de somme nulle, dont les parties sont finis et pour lesquels les joueurs jouent l'un après l'autre.

Avant tout chose, rappelons, à toute fin utiles, qu'un arbre orienté enraciné[iv] est un graphe orienté connexe acyclique dont un des sommets est appelé racine, rappelons aussi un peu de vocabulaire sur les arbres :

Définition 7.1.

Si G est un arbre orienté, on appellera :

- racine de l'arbre : le sommet qui n'a pas de prédécesseur
- feuilles de l'arbre : les sommets qui n'ont pas de successeur
- nœuds de l'arbre : tous les autres sommets
- branche de l'arbre : tout chemin de la racine vers une feuille
- descendant de s : les successeurs de s
- ascendant de s : les prédécesseurs de s
- père de s : le prédécesseur direct de s
- fils de s : les successeurs directes de s
- la profondeur d'un noeud est la longueur de l'unique chemin qui le relie à la racine
- la hauteur de l'arbre est le nombre maximal d'arcs contenu dans une branche de l'arbre.

On peut représenter une partie d'un jeu à l'aide d'un arbre, qu'on appelle arbre de jeu. Les nœuds représentent les positions du jeu. Les nœuds d'un même niveau sont contrôles par un même joueur. À la racine de l'arbre on trouve la situation initiale du jeu, position qui appartient au premier joueur.

iv. Dans la suite on dira simplement arbre.

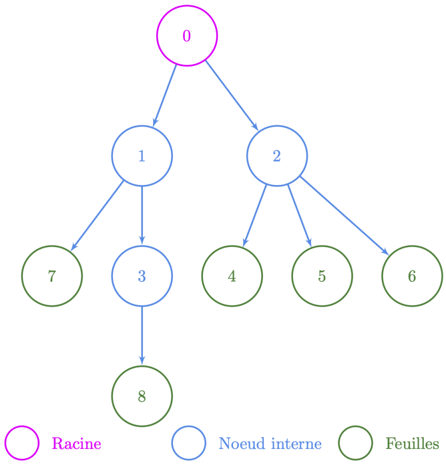


FIGURE 7.10 – Un peu de vocabulaire sur les arbres.

La figure (7.11) présente une, petite, partie de l'arbre du jeu Hex pour un plateau 3×3, les positions d'Ève sont en rouge et celle d'Adam en bleu. Cet arbre est déjà d'une taille assez importante pour un petit tableau.

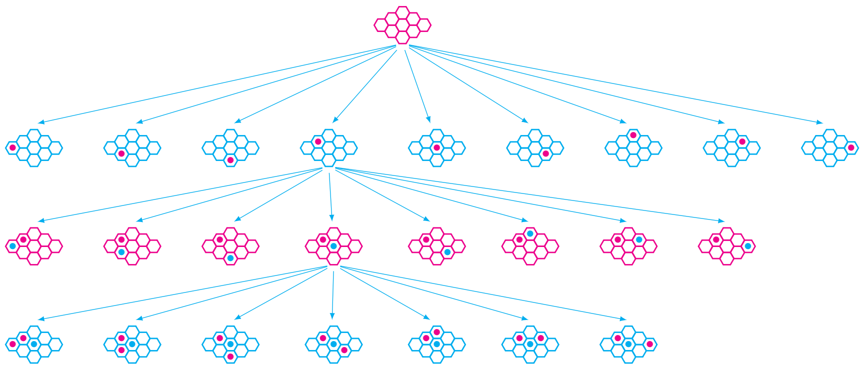


FIGURE 7.11 – Une représentation très partielle de l'arbre de jeu de Hex pour un plateau 3×3.

Une partie est finie lorsque la position actuelle est une feuille de l'arbre.

À un arbre de jeu on peut associer une fonction de score pour Ève, si f est une feuille de l'arbre on notera ce score $s(f) \in \mathbb{R} \cup \{+\infty, -\infty\}$. Une position perdante à un score de $-\infty$, et une position gagnante un score de $+\infty$. Le but d'Ève est évidement de maximiser son score, alors que celui d'Adam est de minimiser ce score.

On prolonge ensuite, de manière inductive, cette fonction aux noeuds internes de l'arbre en posant :

$$s(p) = \begin{cases} s(p) & \text{si } n \text{ est une feuille.} \\ \max\{s(f) \mid f \text{ fils de } p\} & \text{si } p \text{ est une position d'Ève.} \\ \min\{s(f) \mid f \text{ fils de } p\} & \text{si } p \text{ est une position d'Adam.} \end{cases}$$

On peut montrer par récurrence, que si Ève en position p effectue les choix le choix d'un fils f tel que $s(p) = s(f)$, alors à la fin de la partie le score d'Ève sera supérieur à $s(p)$, de même si Adam est sur la position p et qu'il joue un fils f tel que $s(p) = s(f)$ alors le score d'Ève sera inférieur à $s(p)$.

Aussi, si Ève joue en essayant de maximiser la valeur de son score, alors celui-ci sera à la fin de la partie la valeur de $s(r)$, où r est la racine de l'arbre. Donc, si $s(r) > 0$ c'est Ève qui gagne, si $s(r) = 0$ il y a match nul et sinon c'est Adam qui gagne.

Exercice 7.2. La figure (7.12), présente la cime d'un arbre de jeu. Les positions d'Ève sont en rouge et celles d'Adam en bleu. En fonction des valeurs de s sur les noeuds de profondeur 3 de l'arbre, calculer la valeur de s en chacun des noeuds internes de profondeur inférieur. Est-ce que Ève gagne ou perd à ce jeu en commençant.

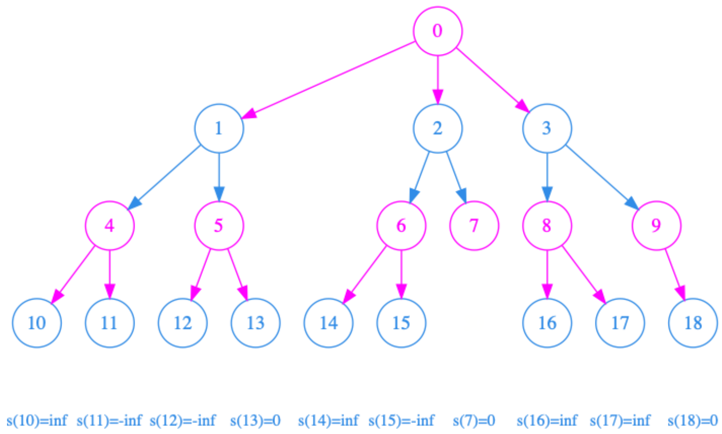


FIGURE 7.12 – Exemple de calcul de s.

7.2.1 Des problèmes de différentes tailles

Les algorithmes de calcul de l'attracteur pour un joueur ont une complexité qui les rend impossible à utiliser pour des jeux comme les échecs ou le go. En effet les arènes associées ont un nombre de sommets compris entre 10^{40} et 10^{50} pour les échec et 10^{100} [v] pour le go.

La taille des arbres de jeu nous empêche d'en faire une exploration exhaustive, mais pas d'en faire une exploration locale.

Lors de l'exploration locale d'un arbre de jeu, la partie est peut-être très loin d'être finie, donc les feuilles sont peut-être encore très loin. C'est pourquoi l'algorithme que l'on va voir se base sur une heuristique capable d'estimer le gain d'une position sans pour autant disposer de l'intégralité de l'arbre du jeu.

Les heuristiques servent à partir d'une connaissance incomplète à aboutir à une « bonne » solution d'un problème, dans un temps acceptable, même si la solution apportée n'est pas optimale. Nous avons déjà rencontré des algorithmes utilisant des heuristiques, par exemple A^*.

7.2.2 L'algorithme Min-max ou Minimax

Lorsque l'arbre de jeu ne peut être explorer en entier, l'algorithme min-max, ou minimax, consiste à fournir la « meilleure » stratégie à partir d'une position $p \in S$. Pour ça, il a recourt à une fonction $h : S \to \mathbb{R} \cup \{-\infty, +\infty\}$ qui doit être une bonne approximation de la fonction de score sur les noeuds qui ne sont pas des feuilles, et à explorer le sous-arbre de racine p d'hauteur inférieure à h_{max}, un paramètre fixé.

Sur la figure (**??**), on a représenté le sous-arbre de racine p d'un arbre de jeu. Comme toujours les positions d'Ève portent des numéros rouge, celles d'Adam des bleus, et en vert ce sont des feuilles de l'arbre de jeu complet. Les valeurs de la fonction h sur les noeuds qui ne sont pas des feuilles permettent de déterminer parmi les positions 1, 2 et 3 celle qui est la plus favorable à Ève.

v. C'est un nombre plus grand que le nombre d'atomes dans l'univers observable.

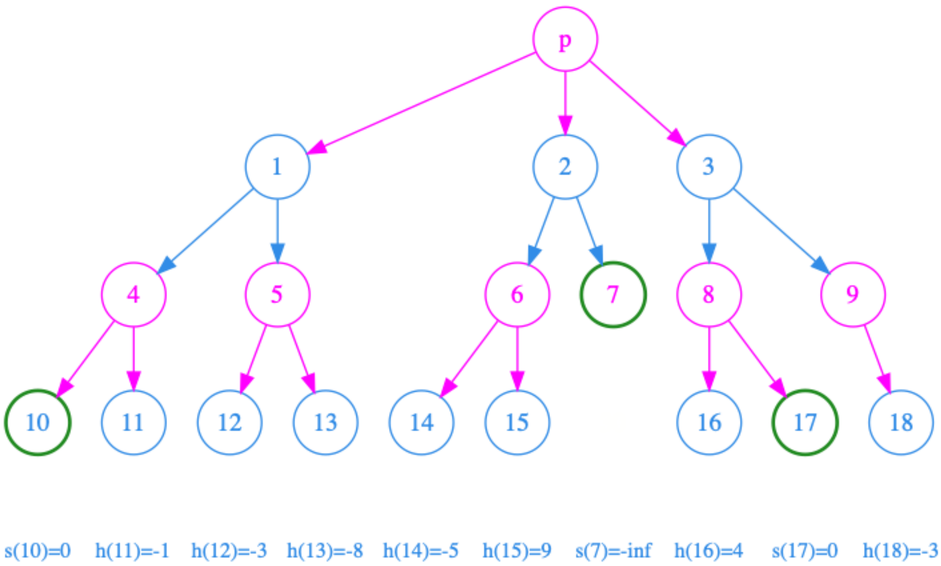


FIGURE 7.13 – Principe de l'algorithme minimax.

Algorithm 7: MinMax

def $min\_max(p, h_{max}, s, h)$**:**

Data: Une position p, la hauteur h_{max} d'exploration, les fonctions s et h

Result: Une estimation de $s(p)$

if *p est une feuille* **then**

  return $s(p)$

if *Si p est une position d'Ève* **then**

  return $\max\{min\_max(f, h_{max} - 1, s, h) \mid f$ fils de $p\}$

else

  return $\min\{min\_max(f, h_{max} - 1, s, h) \mid f$ fils de $p\}$

7.2.3 Exemples d'heuristique

Le Tic-Toc-Toe

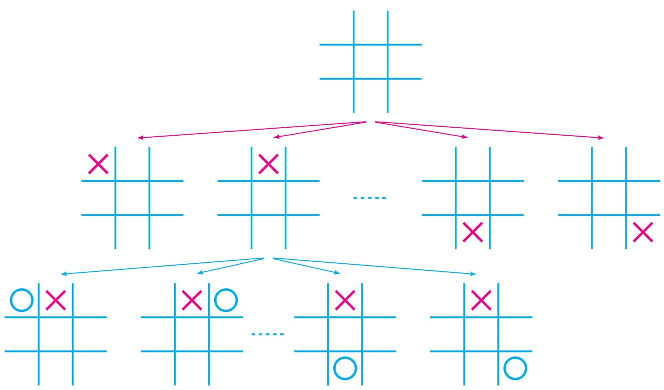


FIGURE 7.14 – Arbre de jeu du morpion.

Vous connaissez tous le jeu du Tic-Toc-Toe. Même si l'arbre du jeu n'est pas d'une hauteur très importante, comparée à celui des échecs, on peut utiliser une heuristique pour déterminer à partir d'une position quel coup doit jouer Ève ensuite.

$$\begin{aligned} h(p) \quad = & \quad \text{nombre de lignes ouvertes pour Ève} \\ + & \quad \text{nombre de colonnes ouvertes pour Ève} \\ + & \quad \text{nombre de diagonales ouvertes pour Ève} \\ - & \quad \text{nombre de lignes ouvertes pour Adam} \\ - & \quad \text{nombre de colonnes ouvertes pour Adam} \\ - & \quad \text{nombre de diagonales ouvertes pour Adam.} \end{aligned}$$

Une ligne, une colonne ou une diagonale est ouverte si Adam n'y a pas de pions. Par exemple, pour la position p représentée par la figure **(7.15)**, on a :

$$h(p) = 2 + 2 + 2 - 1 - 1 - 0 = 4.$$

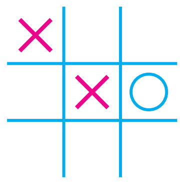


FIGURE 7.15 – Arbre de jeu du morpion.

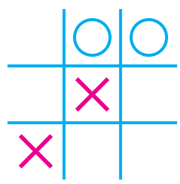
Bien sûr, sur les feuilles h vaut soit $+\infty$, soit $-\infty$ en cas de victoire d'Ève ou d'Adam, et 0 en cas de match nul.

Exercice 7.3. 1. Si Ève commence la partie :

(a) Quelle est le meilleur premier coup à jouer donné par `min_max`, si $h_{max} = 1$?

(b) Même question avec $h_{max} = 2$?

2. À partir de la position suivante que préconise `min_max` comme coup avec $h_{max} = 2$?



Le puissance 4

Là encore, vous connaissez tous le jeu du puissance 4. L'arbre du jeu est plus conséquent que pour le Tic-Tac-Toe.

Il est de coutume de construire une heuristique en commençant par attribuer à chaque case du plateau de jeu une valeur qui correspond au nombre d'alignements potentiels de quatre pions lorsqu'on place un pion à cet emplacement, puis pour évaluer une position p de poser :

$$\begin{aligned} h(p) \quad = & \quad \text{somme des valeurs des cases contenant un pion d'Ève} \\ & - \text{somme des valeurs des cases contenant un pion d'Adam.} \end{aligned}$$

3	4	5	7	5	4	3
4	6	8	10	8	6	4
5	8	11	13	11	8	5
5	8	11	13	11	8	5
4	6	8	10	8	6	4
3	4	5	7	5	4	3

FIGURE 7.16 – Valeurs des cases.

Exercice 7.4.

1. Si Ève joue les rouges, à partir de la position suivante que préconise `min_max` comme coup avec $h_{max} = 1$? Et avec $h_{max} = 2$?
2. Et si elle joue les jaunes ?