CHAPITRE 4

Théorie des jeux combinatoires

Jérôme VON BUHREN
http://vonbuhren.free.fr

CPGE - Filière Scientifique - 2e année

Dans ce chapitre, on étudie les jeux présentant l'ensemble des propriétés suivantes.

- Deux joueurs antagonistes jouent alternativement sans pouvoir passer leur tour : le jeu se termine soit par la victoire d'un seul des deux joueurs, soit par un match nul.
- Il y a un nombre fini de configurations du jeu.
- Le jeu est à information complète : chaque joueur a une vision complète de l'état du jeu à tout instant.
- Le jeu est déterministe : dans une situation donnée, une décision amène toujours à la même situation.

Exemple 1

Les hypothèses ci-dessus excluent le shifumi, le Monopoly et la plupart des jeux de cartes, mais elles permettent d'étudier le morpion, le jeu Puissance 4, les dames, les échecs ou le go.

Remarque 1

Nous appelons les deux joueurs J_0 et J_1 et nous supposons que le joueur J_0 est le premier à jouer.

Dans ce chapitre, nous allons étudier comment modéliser ce type de jeux par des graphes et comment déterminer une stratégie pour l'un ou l'autre des joueurs.

Rappelons qu'un graphe G est un couple (S, A) où S est l'ensemble des sommets du graphe et $A \subset S^2$ décrit l'ensemble des arêtes de G.

Définition (Graphe biparti)

Un graphe (S, A) est biparti s'il existe une partition de S en deux parties S_0 et S_1 telle qu'aucune arête de S ne relie deux sommets de S_0 ou deux sommets de S_1.

Un jeu vérifiant les hypothèses de l'énoncé peut naturellement se représenter par un graphe biparti (S, A) de la manière suivante :

- l'ensemble S_0 des sommets représentant les états du jeu contrôlés par le joueur J_0 ;
- l'ensemble S_1 des sommets représentant les états du jeu contrôlés par le joueur J_1 ;
- l'ensemble A des arêtes représentant les transitions possibles entre les états du jeu, i.e. les choix que les joueurs peuvent effectuer à chacun de leur tour.

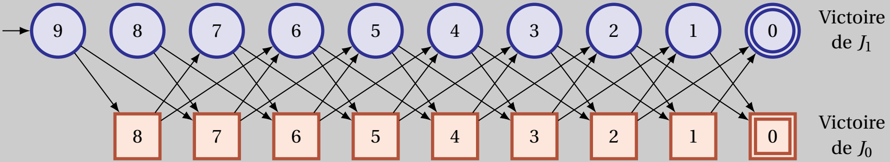
Le sommet initial représente le début du jeu tandis que les sommets finaux (les sommets sans successeurs) représentent les différentes fins de partie.

Exemple 2 (Le jeu des allumettes)

On installe 9 allumettes sur un plateau du jeu. Les joueurs J_0 et J_1 jouent alternativement : durant son tour, chaque joueur peut retirer 1 ou 2 allumettes de la table. Le joueur qui ne peut plus jouer a perdu et l'autre joueur a gagné la partie.

Exemple 2

On peut représenter ce jeu par un graphe biparti : le numéro de chaque sommet indique le nombre d'allumettes restantes sur le plateau. On représente les états contrôlés par J_0 par des ronds bleus et les états contrôlés par J_1 par des carrés rouges.



On remarque qu'on aurait pu supprimer le sommet 8 pour le joueur J_0 : il correspond à une situation impossible.

Remarque 2

L'ensemble des sommets finaux F du graphe biparti (S, A) se partitionne en trois parties :

- l'ensemble F_0 des sommets finaux représentant une victoire de J_0 ;
- l'ensemble F_1 des sommets finaux représentant une victoire de J_1 ;
- l'ensemble F_N des sommets finaux représentant un match nul.

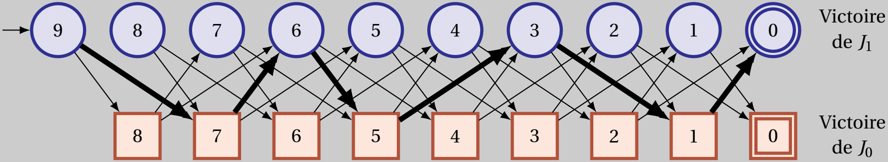
Exemple 3

Dans l'exemple 2, les ensembles F_0 et F_1 sont constitués chacun d'un seul sommet tandis que F_N est vide : le match nul n'est pas possible dans ce jeu.

Chaque partie du jeu est représentée naturellement par un chemin $(s_0, s_1, \dots, s_n)$ dans le graphe (S, A) commençant au sommet initial s_0 et se terminant sur un sommet final $s_n \in F$.

Exemple 4

En reprenant l'exemple 2, voici un exemple de partie gagnée par le joueur J_1.



Dans cette sous-partie, on considère un jeu modélisé par un graphe biparti (S, A) avec $S = S_0 \cup S_1$. On note F l'ensemble de ses sommets finaux.

Définition (Stratégie)

Soit $k \in \{0, 1\}$. Une stratégie pour le joueur J_k est une application $\varphi_k : S_k \setminus F \to S_{1-k}$ telle que pour tout $s \in S_k \setminus F$, le couple $(s, \varphi_k(s))$ est une arête du graphe biparti.

Remarques 3

a) Une stratégie pour J_k est la donnée du choix effectué par J_k pour chaque état du jeu qu'il contrôle.

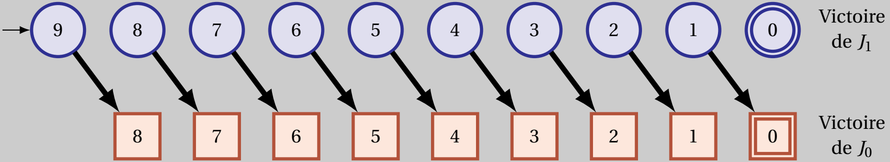
b) Dans le programme, on ne considère que des stratégies **sans mémoires** : le choix effectué par un joueur ne dépend que de l'état actuel du jeu (et pas de tout l'historique de la partie en cours).

c) Une partie $(s_0, s_1, \ldots, s_n)$ du jeu est dite jouée suivant la stratégie φ_k si

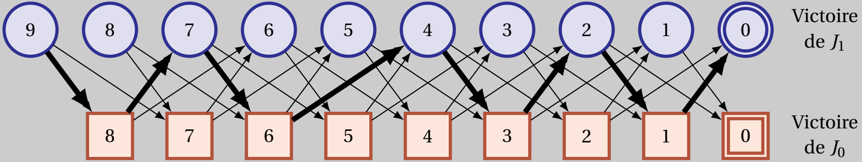
$$\forall i \in [\![0, n-1]\!], \quad s_i \in S_k \quad \Rightarrow \quad s_{i+1} = \varphi_k(s_i).$$

Exemple 5

En reprenant l'exemple 2, on peut représenter la stratégie « stupide » du joueur J_0 consistant à ne prendre qu'une allumette à chacun de ses tours.



Voici ci-dessous une partie jouée où le joueur J_0 a joué suivant la stratégie décrite ci-dessus.

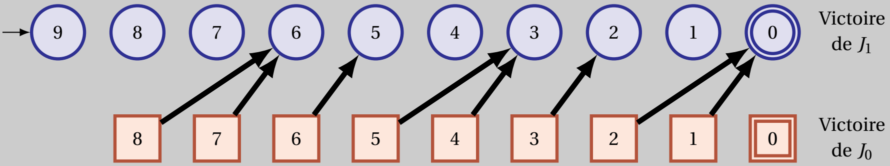


Définition (Stratégie gagnante)

Soit $k \in \{0,1\}$. Une stratégie est dite gagnante pour J_k si toute partie jouée en suivant cette stratégie est gagnante pour J_k.

Exemple 6

En reprenant l'exemple 2, on peut vérifier que J_0 n'a pas de stratégie gagnante et que la stratégie représentée ci-dessous pour J_1 est gagnante.

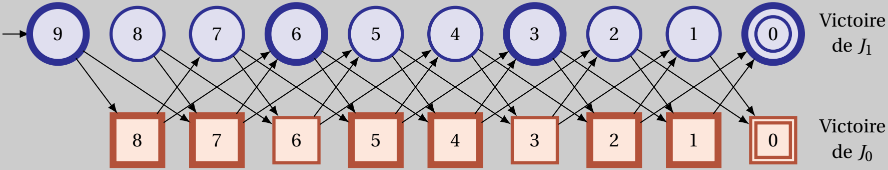


Définition (Position gagnante)

Soit $k \in \{0, 1\}$. Une position du jeu $s \in S$ est dite gagnante pour J_k lorsqu'il existe une stratégie gagnante pour J_k pour une partie débutant au sommet s.

Exemple 7

En reprenant l'exemple 2, on peut vérifier que les positions gagnantes pour le joueur J_1 sont celles mises en valeur ci-dessous.



Remarques 4

a) L'ensemble des sommets du graphe biparti (S, A) se partitionne en trois parties :
   - l'ensemble des sommets représentant une position gagnante pour J_0 ;
   - l'ensemble des sommets représentant une position gagnante pour J_1 ;
   - l'ensemble des sommets pour lesquels aucun joueur n'a de stratégie gagnante : en partant d'un de ces sommets, si les deux joueurs choisissent une stratégie optimale, la partie se finira par un match nul.

b) Nous verrons dans les parties suivantes comment déterminer les positions gagnantes et une stratégie gagnante.

Dans cette sous-partie, on considère un jeu modélisé par un graphe biparti (S, A) avec $S = S_0 \cup S_1$. Pour $k \in \{0, 1\}$, on note F_k l'ensemble de ses sommets finaux représentant une victoire de J_k. Nous allons présenter un algorithme pour déterminer l'ensemble des positions gagnantes d'un joueur donné.

Pour déterminer les positions gagnantes de J_0, l'idée est relativement élémentaire : on part des sommets de F_0 qui sont tous gagnants pour J_0, puis on « remonte » dans le graphe biparti (S, A) en suivant les règles suivantes :

- un sommet de S_0, menant vers un sommet gagnant pour J_0, est lui-même gagnant pour J_0, car le joueur J_0 pourra faire ce choix;
- un sommet de S_1, dont toutes les arêtes en partant mènent vers un sommet gagnant pour J_0, est lui-même gagnant (car le joueur J_1 ne pourra que faire un choix gagnant pour J_0).

Pour formaliser cette idée, on introduit la notion d'attracteur.

Définition (Attracteur)

Soit $k \in \{0,1\}$. On définit une suite d'ensemble $\left(\mathscr{A}_i^{(k)}\right)_{i\in\mathbb{N}}$ par $\mathscr{A}_0^{(k)} = F_k$ et en posant pour tout $i \in \mathbb{N}$ que

$$\begin{aligned}\mathscr{A}_{i+1}^{(k)} = \mathscr{A}_i^{(k)} \cup \left\{s \in S_k \;\middle|\; \exists s' \in \mathscr{A}_i^{(k)},\ (s,s') \in A\right\} \\ \cup \left\{s \in S_{1-k} \;\middle|\; \forall s' \in S_k,\ (s,s') \in A \Rightarrow\ s' \in \mathscr{A}_i^{(k)}\right\}.\end{aligned}$$

On appelle attracteur pour le joueur J_k l'ensemble $\mathscr{A}^{(k)} = \bigcup_{i\in\mathbb{N}} \mathscr{A}_i^{(k)}$.

Remarques 5

a) Autrement dit, l'ensemble des sommets de $\mathscr{A}_{i+1}^{(k)}$ est construit à partir de $\mathscr{A}_i^{(k)}$ en ajoutant :

- les sommets contrôlés par J_k dont au moins un successeur est dans $\mathscr{A}_i^{(k)}$;
- les sommets contrôlés par J_{1-k} dont tous les successeurs sont dans $\mathscr{A}_i^{(k)}$.

b) Par définition, les éléments de $\mathscr{A}_i^{(k)}$ sont les positions gagnantes pour le joueur J_k lui permettant de gagner le jeu en au plus i coups.

c) L'attracteur $\mathscr{A}^{(k)}$ est l'ensemble des positions gagnantes pour le joueur J_k.

Remarques 6

d) Le joueur J_k admet une stratégie gagnante si et seulement si le sommet initial est dans l'attracteur $\mathscr{A}^{(k)}$. Dans ce cas, en notant F l'ensemble des sommets finaux du graphe biparti (S, A), on peut définir une stratégie gagnante $\varphi : S_k \setminus F \to S_{1-k}$ en définissant $\varphi(s)$ pour tout $s \in S_k \setminus F$ de la manière suivante :
- si $s \in \mathscr{A}^{(k)}$, alors on considère un indice $i \in \mathbb{N}^*$ minimal tel que $s \in \mathscr{A}_i^{(k)}$ et on définit $\varphi(s)$ comme un successeur de s dans $\mathscr{A}_{i-1}^{(k)}$;
- si $s \notin \mathscr{A}^{(k)}$, on définit $\varphi(s)$ comme un successeur quelconque de s (le choix n'a aucune incidence, car cette portion de la stratégie ne sera pas utilisée au cours de la partie).

Remarques 6

e) Comme la suite $\left(\mathscr{A}_i^{(k)}\right)_{i\in\mathbb{N}}$ est croissante pour l'inclusion et que tous ses éléments sont des parties de l'ensemble fini des sommets S, on en déduit qu'elle est stationnaire :

$$\exists i_0 \in \mathbb{N}, \quad \mathscr{A}^{(k)} = \mathscr{A}_{i_0}^{(k)}.$$

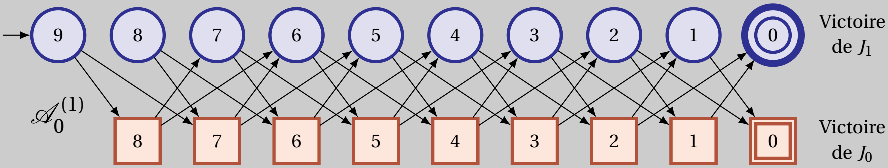
En pratique, comme la suite $\left(\mathscr{A}_i^{(k)}\right)_{i\in\mathbb{N}}$ est définie par récurrence, on peut considérer

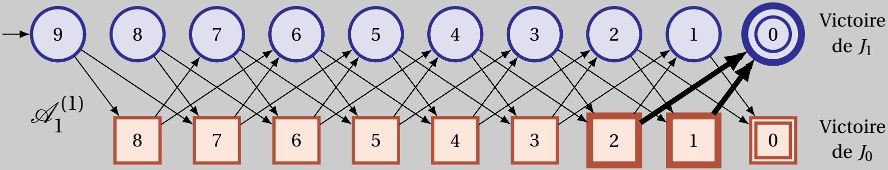
$$i_0 = \min\left\{i \in \mathbb{N} \mid \mathscr{A}_i^{(k)} = \mathscr{A}_{i+1}^{(k)}\right\}.$$

f) Le calcul de l'attracteur $\mathscr{A}^{(k)}$ peut se faire en utilisant un parcours en largeur du graphe transposé de (S, A), donc avec une complexité en $O\big(\mathrm{Card}(S) + \mathrm{Card}(A)\big)$.

Exemple 8

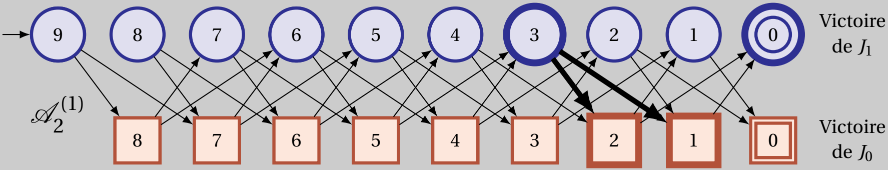
Calculons l'attracteur du joueur J_1 avec le jeu de l'exemple 2.

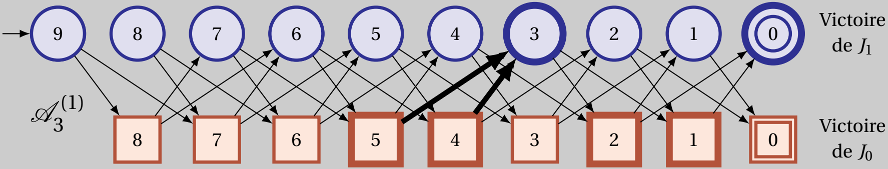




Exemple 8

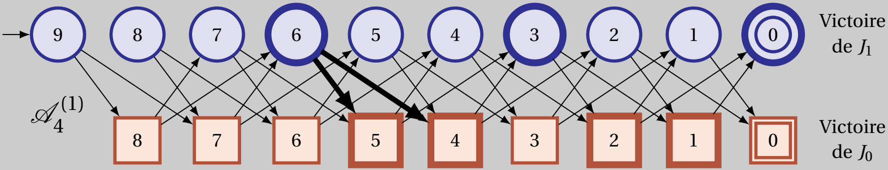
Calculons l'attracteur du joueur J_1 avec le jeu de l'exemple 2.

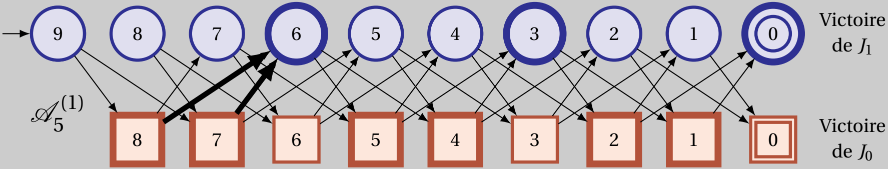




Exemple 8

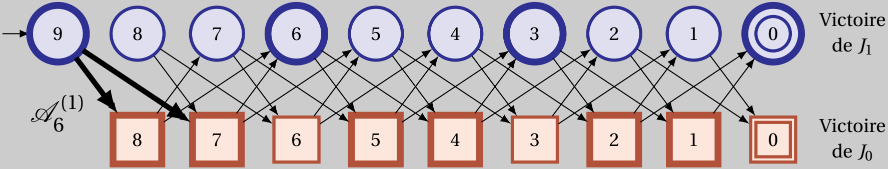
Calculons l'attracteur du joueur J_1 avec le jeu de l'exemple 2.





Exemple 8

Calculons l'attracteur du joueur J_1 avec le jeu de l'exemple 2.



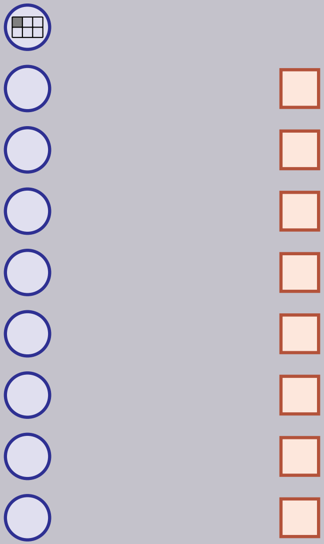
Finalement, on remarque que $\mathscr{A}_7^{(1)} = \mathscr{A}_6^{(1)}$, donc on conclut que $\mathscr{A}^{(1)} = \mathscr{A}_6^{(1)}$. Comme le sommet initial se trouve dans $\mathscr{A}^{(1)}$, le joueur J_1 admet une stratégie gagnante (représentée dans l'exemple 6).

Exercice 1 (Le jeu de Chomp)

Deux joueurs J_0 et J_1 jouent avec une tablette en chocolat composée de 2 lignes et 3 colonnes de carreaux. Chaque joueur choisit à tour de rôle un carré et le mange, ainsi que tous les morceaux situés à sa droite ou en dessous du carré choisi. Le carreau du coin supérieur gauche est empoisonné : le joueur qui se trouve dans l'obligation de le manger perd la partie.

1. Représenter ci-dessous le graphe biparti associé à ce jeu.
2. Déterminer les positions gagnantes pour le joueur J_0.
3. Le joueur J_0 admet-il une stratégie gagnante ? Si oui, en déterminer une.

Exercice 1 (Le jeu de Chomp)



Dans cette partie, on considère un jeu modélisé par un graphe biparti (S, A) avec $S = S_0 \cup S_1$.

En théorie, le calcul des attracteurs présenté dans la partie précédente permet de déterminer les stratégies gagnantes et de jouer de manière parfaite. En pratique, l'algorithme permettant de les déterminer n'est pas utilisable pour des jeux complexes, i.e. lorsque le graphe associé (S, A) est trop gros. Par exemple, on estime que le nombre d'états du jeu est de l'ordre de 10^{32} pour les dames, d'au moins 10^{46} pour les échecs et de 10^{100} pour le go.

Pour contourner cette difficulté, nous allons présenter l'algorithme min-max qui ne nécessite pas une exploration complète du graphe du jeu. En contrepartie, la stratégie obtenue via cette approche ne sera plus parfaite en général.

L'algorithme min-max repose sur une fonction permettant d'évaluer la qualité de chaque position du jeu.

Définition (Heuristique)

Une heuristique pour le jeu étudié est une fonction $h : S \to \mathbb{R} \cup \{-\infty, +\infty\}$ de sorte que pour toute position $p \in S$ du jeu :

- plus $h(p)$ est grand, meilleure est la position pour J_0 ;
- plus $h(p)$ est petit, meilleure est la position pour J_1.

Remarques 6

a) Le mot « heuristique » signifie « qui sert à la découverte » ou « qui est propre à guider une recherche ».

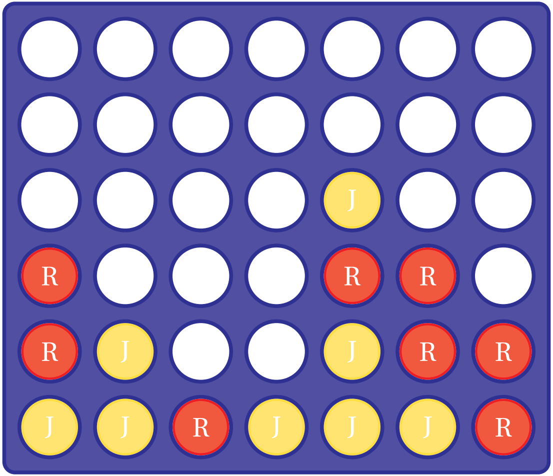
b) En pratique, si $s \in S$ est un sommet final représentant une victoire de J_0, alors on pose $h(s) = +\infty$; tandis que si $s \in S$ est un sommet final représentant une victoire de J_1, alors on pose $h(s) = -\infty$.

c) En pratique, une heuristique est souvent construite de manière expérimentale.

d) Plus l'heuristique utilisée sera pertinente, meilleure sera la qualité de la stratégie obtenue.

Pour illustrer cette notion, on s'intéresse au jeu Puissance 4. Le but du jeu est d'aligner une suite de 4 pions de même couleur dans une grille rectangulaire composée de 6 lignes et 7 colonnes. Chaque joueur dispose de 21 pions d'une même couleur. Les deux joueurs placent tour à tour un pion dans la colonne de leur choix, le pion coulisse alors jusqu'à la position la plus basse possible dans ladite colonne, à la suite de quoi c'est à l'adversaire de jouer. Le vainqueur est le joueur qui réalise en premier un alignement (horizontal, vertical ou diagonal) consécutif d'au moins 4 pions de sa couleur. Si toutes les cases de la grille de jeu sont remplies et qu'aucun des deux joueurs n'a réalisé un tel alignement, la partie est déclarée nulle.

Dans la suite, on suppose que J_0 joue avec des pions jaunes et que J_1 joue avec des pions rouges.



Une position du jeu Puissance 4

On commence par construire une heuristique pour ce jeu. Pour ce faire, on attribue à chaque case du jeu le nombre d'alignements possibles de quatre jetons contenant cette case. Les valeurs attribuées sont reportées dans le tableau ci-dessous.

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

Tableau des valeurs attribuées à chaque case

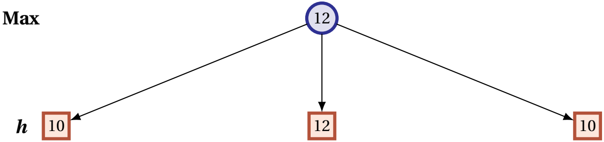
Ensuite, pour calculer l'heuristique d'une position (non finale) du jeu, on calcule la somme des valeurs des cases contrôlées par J_0 à laquelle on retranche la somme des valeurs des cases contrôlées par J_1. Par exemple, l'heuristique de la position s représentée au début de cette partie est

$$h(s) = (3+4+7+5+4+6+8+11) - (5+3+4+6+4+5+11+8) = 2.$$

Dans cette sous-partie, on suppose que l'on dispose d'une heuristique $h : S \to \mathbb{R} \cup \{-\infty, +\infty\}$.

Pour illustrer le principe de l'algorithme min-max, on reprend l'exemple du jeu Puissance 4. On représente les états contrôlés par J_0 par des ronds bleus et les états contrôlés par J_1 par des carrés rouges. Afin d'avoir des arbres de taille raisonnable, on limite les coups possibles des deux joueurs à placer un pion dans une des trois colonnes centrales.

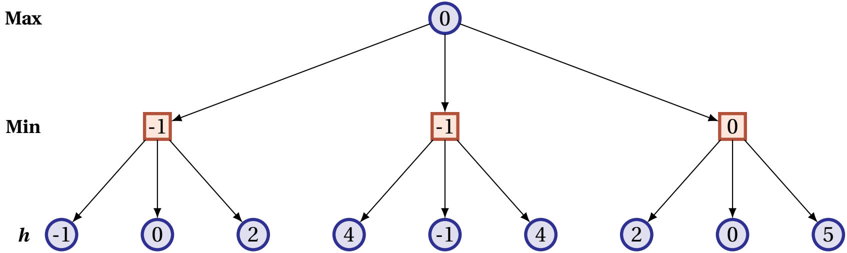
On part de la position du jeu Puissance 4 représentée précédemment : c'est au joueur J_0 de jouer. Une première possibilité pour effectuer le choix du coup à jouer est de calculer l'heuristique de chacune des positions atteignables, puis de rejoindre celle dont l'heuristique est maximale : le joueur J_0 cherche à se trouver dans une position avec une grande heurisitique. On peut représenter la situation avec l'arbre ci-dessous.



Exploration de l'arbre avec une profondeur de 1

En utilisant cette stratégie, le joueur J_0 choisit de jouer le coup au milieu.

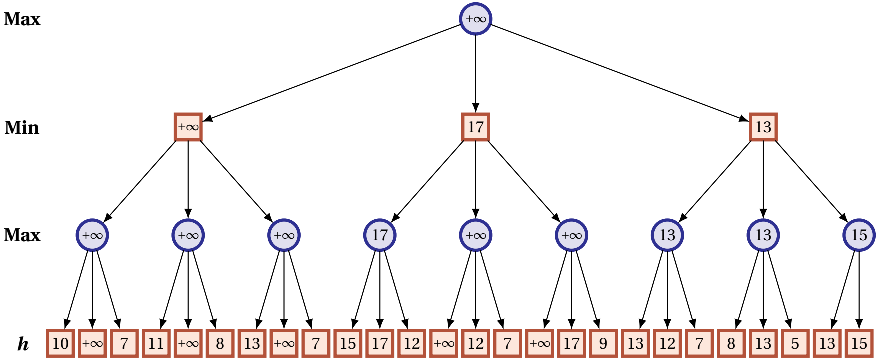
Une seconde possibilité pour effectuer le choix du coup à jouer pour J_0 est de tenir en plus compte du coup suivant qui sera joué par le joueur J_1 : le joueur J_0 peut calculer l'heuristique des positions atteignables en deux coups, puis de choisir le coup lui permettant d'obtenir une heuristique maximale (en supposant que le joueur J_1 va jouer de sorte à minimiser l'heuristique).



Exploration de l'arbre avec une profondeur de 2

En utilisant cette stratégie, le joueur J_0 choisit soit de jouer à gauche, soit de jouer le coup à droite.

Une troisième possibilité pour effectuer le choix du coup à jouer pour J_0 est d'explorer toutes les possibilités pour les trois coups suivants. En reprenant la démarche précédente, on obtient l'arbre ci-dessous.



Exploration de l'arbre avec une profondeur de 3

En utilisant cette stratégie, le joueur J_0 choisit de jouer à gauche. De plus, on observe qu'il gagnera la partie dans trois coups.

Remarques 7

a) La démarche précédente se généralise : l'algorithme min-max consiste en partant d'une position donnée du jeu à explorer toutes les possibilités après $p \in \mathbb{N}^*$ coups. L'objectif de J_0 est de choisir les positions maximisant l'heuristique, tandis que l'objectif de J_1 est de choisir les positions minimisant l'heuristique.

b) Le nombre de positions à examiner a tendance à croitre exponentiellement par rapport à la profondeur d'exploration de l'arbre, donc il est nécessaire de limiter la taille de l'entier $p \in \mathbb{N}^*$.

L'algorithme ci-dessous permet d'implémenter la méthode décrite ci-dessus à l'aide d'une fonction récursive. On suppose que l'on dispose d'une fonction `h` permettant de calculer l'heuristique de toute position $s \in S$ du jeu.

Algorithme min-max

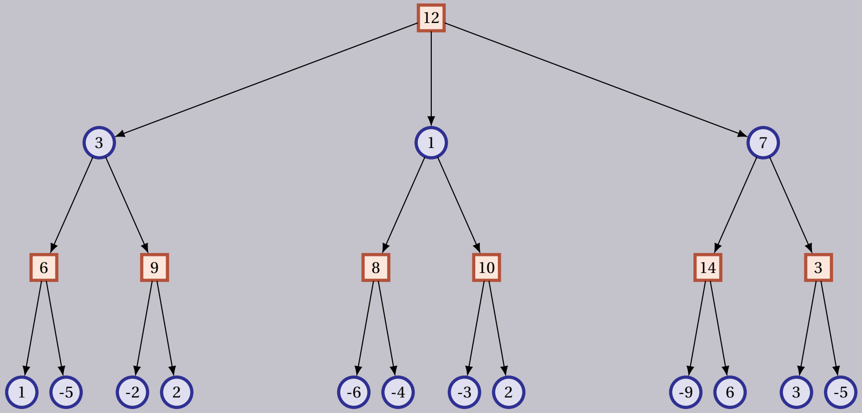
```
Fonction minMax(s, p)
    Entrée : Une position s ∈ S, une profondeur p ∈ ℕ.
    Sortie : La meilleure valeur possible après p coups, le coup à jouer
             pour l'atteindre.

    Calculer la liste lstSucc des successeurs de s

    si p = 0 ou lstSucc est vide alors
        retourner (h(p), -1)
    sinon
        Calculer la liste lstValSucc des éléments minMax(f,p-1)
        où f parcourt lstSucc

    si la position s est contrôlée par le joueur J0 alors
        Déterminer la valeur maximale vMax et sa position indMax
        dans la liste lstValSucc
        retourner (vMax, indMax)
    sinon
        Déterminer la valeur minimale vMin et sa position indMin
        dans la liste lstValSucc
        retourner (vMin, indMin)
```

Exercice 2

On se trouve au milieu d'une partie d'un jeu : c'est au joueur J_1 de jouer. L'arbre ci-dessous représente toutes les possibilités pour les 3 coups suivants. La valeur indiquée dans chaque noeud est l'heuristique de la position utilisée par le joueur J_1 pour sa stratégie.

1. Quel coup doit jouer le joueur J_1 s'il applique l'algorithme min-max avec une profondeur de 1 ?
2. Quel coup doit jouer le joueur J_1 s'il applique l'algorithme min-max avec une profondeur de 2 ?
3. Quel coup doit jouer le joueur J_1 s'il applique l'algorithme min-max avec une profondeur de 3 ?

Exercice 2



Arbre représentant les évolutions possibles de la partie au cours des trois prochains coups