Chapitre 8

Jeux

Hugo Salou MPI*

Dernière mise à jour le 13 mai 2023

Table des matières

DANS CE CHAPITRE, on s'intéresse à la théorie des jeux. L'objectifs est de modéliser des interactions (économie, coopération, ...). On s'intéressera, par contre, à une petite partie de la théorie des jeux : les jeux d'accessibilité. Par exemple, les échecs et le go sont deux jeux d'accessibilité, mais on s'intéressera à des jeux beaucoup plus simples. On cherchera les stratégies gagnantes pour ces jeux, des heuristiques. On raccrochera ce chapitre avec la théorème des graphes.

Par exemple, on peut coder un programme répondant, *plus ou moins correctement*, au jeu puissance 4. [Démonstration d'un jeu de puissance 4.]

Dans un premier temps, on s'intéresse à un jeu simple. On a 13 allumettes :

$$| \quad | \quad | \quad | \quad | \quad | \quad | \quad | \quad | \quad | \quad | \quad | \quad | .$$

On peut retirer une, deux ou trois allumettes. Le joueur retirant la dernière allumette perd. On peut représenter le jeu par un graphe, et *jouer* au jeu est se déplacer dans ce graphe. Ce graphe est *biparti*.

1 Jeux sur un graphe

Définition : On appelle *arène* la donnée

— d'un graphe biparti orienté $G = (V, E)$, avec $V = V_\mathrm{A} \uplus V_\mathrm{B}$ la séparation en deux sommets de ce graphes bipartis ;

— d'un sommet initial s.

Définition : Un état du jeu (un sommet de l'arène) est dit *terminal* lorsqu'il n'a pas de successeurs. On note, dans la suite, $V_\mathrm{A}^\star$ et $V_\mathrm{B}^\star$ les états non terminaux (notation non officielle).

REMARQUE :
Les états V_A sont les états où c'est à Alice de jouer. Les états V_B sont les états où c'est à Bob de jouer.

EXEMPLE :
On reprend l'exemple du jeu des allumettes. La figure ci-dessus représente les états du jeu. Les états i représentent les états de Alice (*i.e.* des éléments de V_A), les états $\boxed{j}$ représentent les états de Bob (*i.e.* des éléments de V_B). Les états doublement encadrés sont terminaux, les autres sont non terminaux.

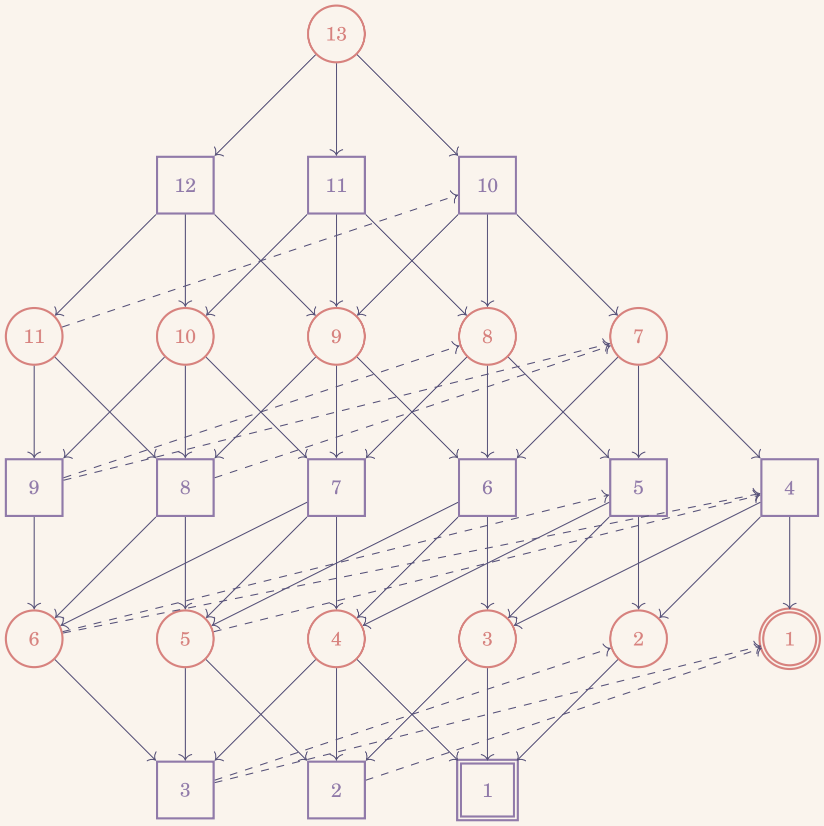

Figure 1 – États du jeu des allumettes

Définition : Un *jeu d'accessibilité* est déterminé par une arène $G = (V_{\mathrm{A}} \mathbin{\dot\cup} V_{\mathrm{B}}, E)$ de sommet initial s, et deux ensembles $\mathbb{O}_{\mathrm{A}}$ et $\mathbb{O}_{\mathrm{B}}$ de sommets terminaux tels que

- $\mathbb{O}_{\mathrm{A}} \cap \mathbb{O}_{\mathrm{B}} = \varnothing$,
- $\mathbb{O}_{\mathrm{A}} \subseteq V_{\mathrm{A}} \mathbin{\dot\cup} V_{\mathrm{B}}$,
- et $\mathbb{O}_{\mathrm{B}} \subseteq V_{\mathrm{A}} \mathbin{\dot\cup} V_{\mathrm{B}}$.

L'ensemble $\mathbb{O}_{\mathrm{A}}$ représente les sommets « *gagnants* » pour Alice. L'ensemble $\mathbb{O}_{\mathrm{B}}$ représente les sommets « *gagnants* » pour Bob.

Remarque :
Ils ne forment pas nécessairement une partition des sommets terminaux.

Exemple :
Dans la figure **??**, l'état gagnant pour Alice est 1 ; celui de Bob est $\boxed{1}$.

Définition : Une *partie depuis un sommet s* dans un jeu d'accessibilité est une suite (finie ou infinie) de sommets formant un chemin dans l'arène, partant de s, telle que, si la partie est finie, le dernier sommet est terminal.

On dit d'une partie

— qu'elle est *gagnée par Alice* si elle est finie et le dernier sommet est dans $\mathbb{O}_{\mathrm{A}}$;
— qu'elle est *gagnée par Bob* si elle est finie et le dernier sommet est dans $\mathbb{O}_{\mathrm{B}}$;
— qu'elle est nulle sinon.

Exemple :
À faire : Ajouter exemple ?

Définition : On appelle *stratégie pour Alice* une fonction $f : V_{\mathrm{A}}^{\star} \to V_{\mathrm{B}}$ telle que, pour tout état $s_{\mathrm{A}} \in V_{\mathrm{A}}^{\star}$, on a $\big(s_{\mathrm{A}}, f(s_{\mathrm{A}})\big) \in E$.

Soit $N \in \mathbb{N} \cup \{+\infty\}$ la longueur d'une partie $(s_1, s_2, \ldots, s_n, \ldots, s_N\ ?)$, cette partie est dite *jouée selon une stratégie* f si

$$\forall i \in [\![1, N+1[\![, \quad s_i \in V_{\mathrm{A}}^{\star} \implies s_{i+1} = f(s_i).$$

Une stratégie pour Alice est dit *gagnante depuis un état* s dès lors que toute partie depuis s jouée selon f est gagnée par Alice. Plus généralement, une stratégie est dit *gagnante* si elle est gagnante depuis l'état initial.

On appelle *stratégie pour Bob* une fonction $f : V_{\mathrm{B}}^{\star} \to V_{\mathrm{B}}$ telle que, pour tout état $s_{\mathrm{B}} \in V_{\mathrm{B}}^{\star}$, on a $\big(s_{\mathrm{B}}, f(s_{\mathrm{B}})\big) \in E$.

Soit $N \in \mathbb{N} \cup \{+\infty\}$ la longueur d'une partie $(s_1, s_2, \ldots, s_n, \ldots, s_N\ ?)$, cette partie est dite *jouée selon une stratégie* f si

$$\forall i \in [\![1, N+1[\![, \quad s_i \in V_{\mathrm{B}}^{\star} \implies s_{i+1} = f(s_i).$$

Une stratégie pour Bob est dit *gagnante depuis un état* s dès lors que toute partie depuis s jouée selon f est gagnée par Bob.

Exemple :
La stratégie $\Longrightarrow$ est gagnante pour Bob.

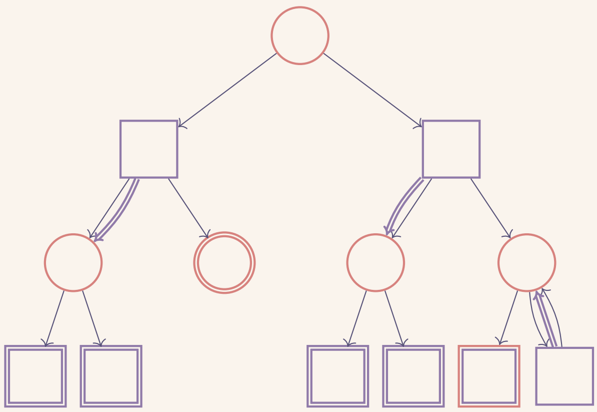

Figure 2 – Stratégie gagnante pour Bob

Interlude : représentation machine des jeux. On représente le graphe de manière implicite.

```
type joueur
type etat

val joueur : etat -> move
val possibilite_moves : etat -> etat list
val init : etat
val est_gagant : etat -> joueur option
```

Code 1 – Représentation machine des jeux, types

```
type joueur = Alice | Bob
type etat = { allu : int; joueur : joueur }

let autre = function
| Alice -> Bob
| Bob -> Alice

let joueur (e: etat) = e.joueur
let init = { allu = 13; joueur = Alice }
let est_gagnant (e: etat) =
  if e.allu = 1 then Some(autre e.joueur)
  else None
let possible_moves (e: etat) =
  (if e.allu > 3 then
    [{ allu = e.allu - 3; joueur = autre e.joueur }] else [])
  @ (if e.allu > 2 then
    [{ allu = e.allu - 2; joueur = autre e.joueur }] else [])
  @ (if e.allu > 1 then
    [{ allu = e.allu - 1; joueur = autre e.joueur }] else [])
```

Code 2 – Représentation machine du jeux des allumettes

Avec une représentation implicite du graphe, on ne stocke pas l'entièreté du graphe directement mais on ne récupère que les successeurs d'un sommet en particulier.

Par exemple, pour écrire un moteur de recherche, on utilise aussi une représentation implicite pour le graphe *internet* ; on ne peut pas *juste* télécharger l'entièreté du graphe.

Attracteur.

> Remarque :
> On suppose que les deux joueurs (Alice et Bob) jouent intelligemment. On se met à la place d'Alice. Bob joue le "mieux" pour lui, et le "pire" pour nous. Nous jouerons le "mieux" pour nous. Ainsi, Bob jouera vers un état de valeur minimale, nous jouerons vers un état de valeur maximale.

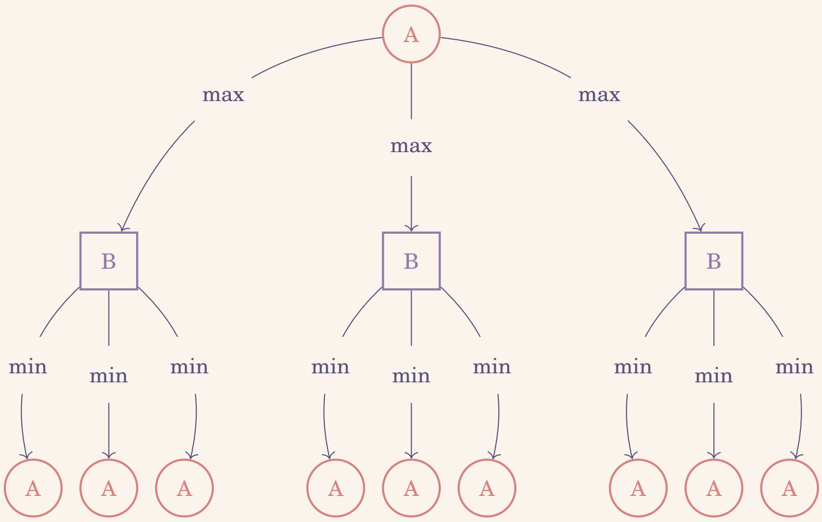

Figure 3 – Stratégie de minimisation pour Bob, et de maximisation pour Alice

Définition : On définit la suite d'ensembles $(\mathscr{A}_i)_{i\in\mathbb{N}}$ par $\mathscr{A}_0 = \mathscr{O}_{\mathrm{A}}$, et, pour tout $n \in \mathbb{N}$,

$$\begin{aligned}\mathscr{A}_{n+1} = \mathscr{A}_n &\cup \{s_{\mathrm{B}} \in V_{\mathrm{B}}^\star \mid \forall s_{\mathrm{A}} \in \mathrm{succ}(s_{\mathrm{B}}),\ s_{\mathrm{A}} \in \mathscr{A}_n\} \\ &\cup \{s_{\mathrm{A}} \in V_{\mathrm{A}}^\star \mid \exists s_{\mathrm{B}} \in \mathrm{succ}(s_{\mathrm{A}}),\ s_{\mathrm{B}} \in \mathscr{A}_n\}.\end{aligned}$$

On a appelle alors *attracteur d'Alice* les états $\mathscr{A} = \bigcup_{n\in\mathbb{N}} \mathscr{A}_n$.

Remarque :
La suite des $(\mathscr{A}_n)_{n\in\mathbb{N}}$ est croissante pour l'inclusion $\subseteq$, et ultimement stationnaire car le graphe est fini :

$$\exists N \in \mathbb{N},\ \forall n \geqslant N, \qquad \mathscr{A}_n = \mathscr{A}_N.$$

De même, on définit la suite d'ensembles $(\mathscr{B}_n)_{n\in\mathbb{N}}$ et l'ensemble $\mathscr{B}$ des attracteurs de Bob, en permutant les deux joueurs dans la définition précédente.

Exemple :
À faire : Ajouter exemple, *c.f. cahier de prépa* Si Bob joue intelligemment, il est garanti de gagner.

Remarque :
On remarque que les ensembles $\bigcup_{n\in\mathbb{N}} \mathscr{A}_n$ et $\bigcup_{n\in\mathbb{N}} \mathscr{B}_n$ ne forment pas une partition de l'ensemble des états : l'intersection est bien vide, mais l'union de ces deux ensembles ne couvre pas l'ensemble des états.

Définition (Stratégie induite par les attracteurs) **:** Soit $\mathscr{A}$ l'ensemble des attracteurs d'Alice. On définit la stratégie suivante

$$\begin{aligned} f : V_{\mathrm{A}}^{\star} &\longrightarrow V_{\mathrm{B}} \\ s_{\mathrm{A}} &\longmapsto \begin{cases} s_{\mathrm{B}} \in \mathrm{succ}(s_{\mathrm{A}}) \cap \mathscr{A} & \text{si } s_{\mathrm{A}} \in \mathscr{A} \\ s_{\mathrm{B}} \in \mathrm{succ}(s_{\mathrm{A}}) \text{ quelconque} & \text{sinon.} \end{cases} \end{aligned}$$

Cette stratégie est nommée *stratégie induite par les attracteurs*.

Définition : Soit $s \in \mathscr{A}$ un attracteur. On appelle *rang* de s, notée $\mathrm{rg}(s)$ l'entier

$$\mathrm{rg}(s) = \min\{n \in \mathbb{N} \mid s \in \mathscr{A}_n\}.$$

Lemme : Pour tout entier n, pour tout état $s \in \mathscr{A}_n$, un des trois cas est vrai :
- $s \in \mathscr{O}_{\mathrm{A}}$;
- $n > 0$, $s \in V_{\mathrm{A}}^{\star}$ et il existe $s_{\mathrm{B}} \in \mathrm{succ}(s)$ tel que $s_{\mathrm{B}} \in \mathscr{A}_{n-1}$;
- $n > 0$, $s \in V_{\mathrm{B}}^{\star}$ et pour tout $s_{\mathrm{A}} \in \mathrm{succ}(s)$, $s_{\mathrm{A}} \in \mathscr{A}_{n-1}$.

Preuve :
Par récurrence.
- Si $n = 0$, alors $\mathscr{A}_0 = \mathscr{O}_{\mathrm{A}}$.
- Supposons la propriété vraie au rang n. Soit $s \in \mathscr{A}_{n+1}$. Alors,
  - ou bien $s \in \mathscr{A}_n$, et on conclut par hypothèse de récurrence.
  - ou bien $s \in V_{\mathrm{A}}^{\star}$, et il existe $s_{\mathrm{B}} \in \mathrm{succ}(s)$ tel que $s_{\mathrm{B}} \in \mathscr{A}_n$, alors OK ;
  - ou bien $s \in B_{\mathrm{B}}^{\star}$, et pour tout $s_{\mathrm{A}} \in \mathrm{succ}(s)$, $s_{\mathrm{A}} \in \mathscr{A}_n$, alors OK.

□

Propriété : On a $\mathscr{A} \cap \mathscr{B} = \varnothing$.

Preuve :
Montrons par récurrence que, pour tout entier n, $\mathscr{A}_n \cap \mathscr{B}_n = \varnothing$. À faire : finir cette partie de la preuve. Soit $N \in \mathbb{N}$ tel que $\mathscr{A}_n = \mathscr{A}_N$, pour tout $n \geqslant N$. Soit $M \in \mathbb{N}$ tel que $\mathscr{B}_n = \mathscr{B}_M$, pour tout $n \geqslant M$. Ainsi, $\mathscr{A} = \mathscr{A}_N$ et $\mathscr{B} = \mathscr{B}_M$. On pose $K = \max(M, N)$, et on a donc

$$\mathscr{A} \cap \mathscr{B} = \mathscr{A}_K \cap \mathscr{B}_K = \varnothing.$$

□

Propriété : Si f est une stratégie induite par les attracteurs d'Alice $\mathscr{A}$, et que $s \in \mathscr{A}$, alors toute partie jouée selon f depuis s est gagnante pour Alice.

Preuve :
Soit $(s_n)_{n \in [\![1, N+1[\![}$, avec $N \in \mathbb{N} \cup \{+\infty\}$, une partie jouée selon f depuis s.
- Montrons $\forall n \in [\![1, N+1[\![$, $s_n \in \mathscr{A}$. Par récurrence. On a $s_1 = s \in \mathscr{A}$, d'où l'initialisation. Si la propriété est vraie au rang $n \in [\![1, N[\![$ (ainsi s_{n+1} existe),
  - si $s_n \in V_{\mathrm{A}}^{\star}$, alors $s_{n+1} = f(s_n) \in \mathrm{succ}(s_n) \cap \mathscr{A}$ donc $s_{n+1} \in \mathscr{A}$,
  - si $s_n \in V_{\mathrm{B}}^{\star}$, on a $s_n \in \mathscr{A}$ et $s_{n+1} \in \mathrm{succ}(s_n)$, d'où $s_{n+1} \in \mathscr{A}$.

  Concluant ainsi la récurrence.
- Montrons que $N \neq +\infty$. En effet, montrons que, $\forall n \in [\![1, N[\![$, $\mathrm{rg}(s_{n+1}) < \mathrm{rg}(s_n)$. L'état $s_n \in \mathscr{A}$, son rang est bien défini. De plus, $s_n \in \mathscr{A}_r$, où $r = \mathrm{rg}(s_n)$. Or, $s_n \notin \mathscr{O}_{\mathrm{A}}$. D'après le lemme,

$s_{n+1} \in \mathscr{A}_{r-1}$, donc $\mathrm{rg}(s_{n+1}) \leqslant r-1 < r$. Ainsi, $N \neq +\infty$ par existence du variant, comme l'ensemble ordonné $(\mathbb{N}, \leqslant)$ est bien fondé.

Ainsi, $(s_n)_{n \in [\![1, N+1[\![}$ est une partie finie, et $\forall n \in [\![1, N]\!]$, $s_n \in \mathscr{A}$. L'état s_N est terminal, et donc $s_N \in \mathscr{O}_{\mathrm{A}}$, d'où le résultat. □

Propriété : Soit s un sommet admettant une stratégie gagnante pour Alice. Alors, s est un attracteur : $s \in \mathscr{A}$.

Preuve :
Montrons par récurrence forte : « si s est un sommet admettant une stratégie gagnante f et que toute partie jouée depuis s selon f est de longueur au plus n, alors $s \in \mathscr{A}_n$. »

- Si le sommet s admet une stratégie gagnante, et toute partie depuis s est de longueur 0, alors $s \in \mathscr{O}_{\mathrm{A}} = \mathscr{A}_0$.
- Supposons, pour tout $k \in [\![1, n]\!]$, l'hypothèse de récurrence. Montrons la propriété pour $n+1$. Soit s admettant une stratégie gagnante f et tel que toute partie jouée depuis s selon f est de longueur au plus $n+1$.
  - Si $s \in V_{\mathrm{B}}$, alors
    - si s est terminal, $s \in \mathscr{O}_{\mathrm{A}} \subseteq \mathscr{A}$.
    - sinon, soit $s_1 \in \mathrm{succ}(s)$. Soit γ une partie jouée selon f depuis s_1. γ est une partie jouée selon f depuis s, elle est donc gagnante et de longueur au plus $n+1$. Alors γ est gagnante et de longueur au plus n donc s_1 vérifie les prémisses de l'hypothèse de récurrence, et donc $s_1 \in \mathscr{A}_n$. Ceci étant vrai pour tout $s_1 \in \mathrm{succ}(s)$, on a $s \in \mathscr{A}_{n+1}$.
  - Si $s \in V_{\mathrm{A}}$, alors
    - si s est terminal, alors $s \in \mathscr{O}_{\mathrm{A}} \subseteq \mathscr{A}$.
    - sinon, soit $s_1 = f(s)$, alors s_1 admet une stratégie gagnante (f), et toute partie jouée depuis s_1 est de longueur, au plus, n. L'état s_1 vérifie donc les prémisses l'hypothèse de récurrence. Ainsi $s_1 \in \mathscr{A}_n$, et donc s a un successeur dans $\mathscr{A}_n$. On en déduit $s \in \mathscr{A}_{n+1}$.

Soit alors s admettant une stratégie gagnante f. Ainsi, toute partie jouée depuis s selon f est finie. On peut donc conclure grâce à la démonstration par récurrence ci-avant. □

En pratique, la détermination des attracteurs nécessite une quantité bien trop importante de calculs. On souhaite revenir à une idée développée précédemment : réaliser un min-max. Mais, pour cela, il est nécessaire de donner une notion de « valeur » a chaque état du jeu. On définit donc une *fonction d'heuristique* qui, à un état, associé sa valeur. Le choix de cette fonction reste, cependant, très subjectif.

2 Résolution par heuristique

On suppose définie une *fonction d'heuristique* h de la forme :

$$h : \underset{\substack{\downarrow \\ \text{joueur}}}{\{\mathrm{A}, \mathrm{B}\}} \times \underset{\substack{\uparrow \\ \text{états}}}{V} \to \overbrace{\mathbb{Z} \cup \{+\infty, -\infty\}}^{\bar{\mathbb{Z}}}.$$

On représente informatiquement l'ensemble $\bar{\mathbb{Z}}$ par le type OCaml `int_bar` défini ci-dessous

```
type int_bar =
  | PInt
  | NInt
  | F of int
```

Code 3 – Type `int_bar` représentant un élément l'ensemble $\bar{\mathbb{Z}}$

On peut donc définir l'algorithme MinMax.

Algorithme 1 Algorithme MinMax

Entrée Un état e, un seuil de profondeur d, le joueur j
Sortie Un score

si $\text{succ}(e) = \varnothing$ **alors**
    si $e \in \mathcal{G}_j$ **alors**
        retourner $+\infty$
    sinon si $e \in \mathcal{G}_{\text{autre}(j)}$ **alors**
        retourner $-\infty$
    sinon
        retourner 0
sinon si $d = 0$ **alors**
    retourner $h(j, e)$
sinon
    si $j = \text{joueur}(e)$ **alors**
        retourner $\max\{\text{MinMax}(e', d-1, j) \mid e' \in \text{succ}(e)\}$
    sinon
        retourner $\min\{\text{MinMax}(e', d-1, j) \mid e' \in \text{succ}(e)\}$

Exemple :
c.f. cahier-de-prépa On applique l'algorithme MinMax pour Alice.

Exemple (td 13. Exercice 4) :

On peut améliorer l'algorithme MinMax avec « l'élagage α, β. »

Algorithme 2 Algorithme MinMax avec élagage α, β

Entrée Un état e, un seuil de profondeur d, le joueur j, et $\alpha, \beta \in \bar{\mathbb{Z}}$
Sortie Un score dans $[\alpha, \beta]$

si $\mathrm{succ}(e) = \varnothing$ **alors**
    si $e \in \mathcal{G}_j$ **alors**
        retourner β
    sinon si $e \in \mathcal{G}_{\mathrm{autre}(j)}$ **alors**
        retourner α
    sinon
        si $\beta \leqslant 0$ **alors retourner** β
        sinon si $\alpha \geqslant 0$ **alors retourner** α
        sinon retourner 0
sinon si $d = 0$ **alors**
    si $h(j, e) \geqslant \beta$ **alors retourner** β
    sinon si $h(j, e) \leqslant \alpha$ **alors retourner** α
    sinon retourner $h(j, e)$
sinon
    si $\mathrm{joueur}(e) = j$ **alors**
        $v \leftarrow \alpha$
        pour $e' \in \mathrm{succ}(e)$ **faire**
            $v \leftarrow \max(v, \textsc{MinMaxÉlagué}(e', d-1, j, v, \beta))$
            si $v \geqslant \beta$ **alors**
                retourner β
        retourner v
    sinon
        $u \leftarrow \beta$
        pour $e' \in \mathrm{succ}(e)$ **faire**
            $u \leftarrow \min(u, \textsc{MinMaxÉlagué}(e', d-1, j, \alpha, u))$
            si $u \leqslant \alpha$ **alors**
                retourner α
        retourner u