Algorithme min-max et heuristique

Quentin Fortier

April 5, 2023

Algorithme min-max

L'algorithme de calcul des attracteurs que nous avons vu demande de parcourir chaque sommet du graphe des configurations possibles.

Il est donc beaucoup trop lent pour des jeux comme les échecs ou le go où le nombre de configurations est très grand.

Algorithme min-max

Une **heuristique** est une fonction qui à une configuration associe une valeur dans $\mathbb{R}$.

Algorithme min-max

Une **heuristique** est une fonction qui à une configuration associe une valeur dans $\mathbb{R}$.

Exemple : l'algorithme A^* vu en 1ère année utilise une heuristique pour estimer la distance entre un sommet et le sommet de destination.

Algorithme min-max

Une **heuristique** est une fonction qui à une configuration associe une valeur dans $\mathbb{R}$.

Exemple : l'algorithme A* vu en 1ère année utilise une heuristique pour estimer la distance entre un sommet et le sommet de destination.

Dans le cas d'un jeu, on va utiliser une heuristique h qui estime à quel point la configuration v est favorable à un joueur : plus $h(v)$ est grand, plus v est favorable à Alice et inversement.

Algorithme min-max

Une **heuristique** est une fonction qui à une configuration associe une valeur dans $\mathbb{R}$.

Exemple : l'algorithme A* vu en 1ère année utilise une heuristique pour estimer la distance entre un sommet et le sommet de destination.

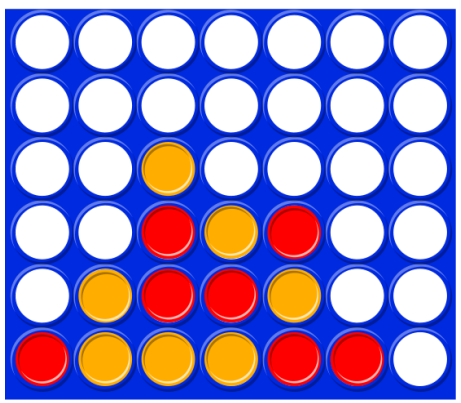
Dans le cas d'un jeu, on va utiliser une heuristique h qui estime à quel point la configuration v est favorable à un joueur : plus $h(v)$ est grand, plus v est favorable à Alice et inversement.

Attention : Aussi bien dans A* que dans les jeux, utiliser une heuristique permet d'accélerer la recherche, mais le résultat n'est pas forcément optimal.

Algorithme min-max

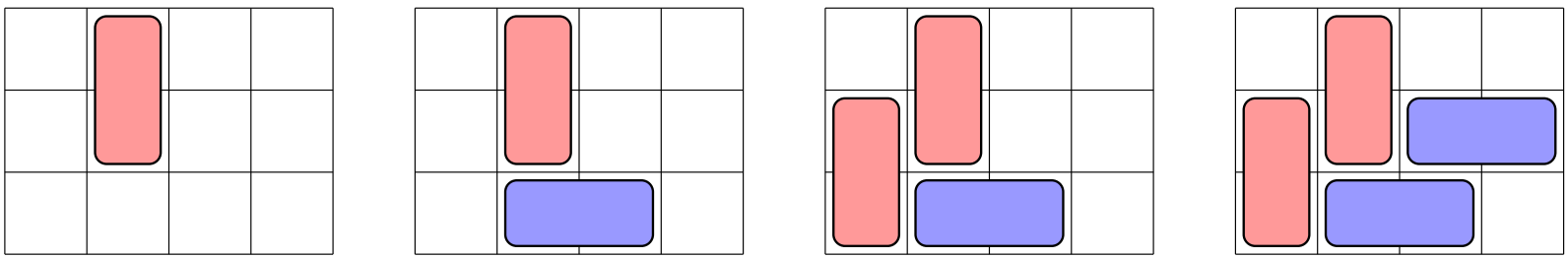
Question

Proposer une heuristique pour le puissance 4.



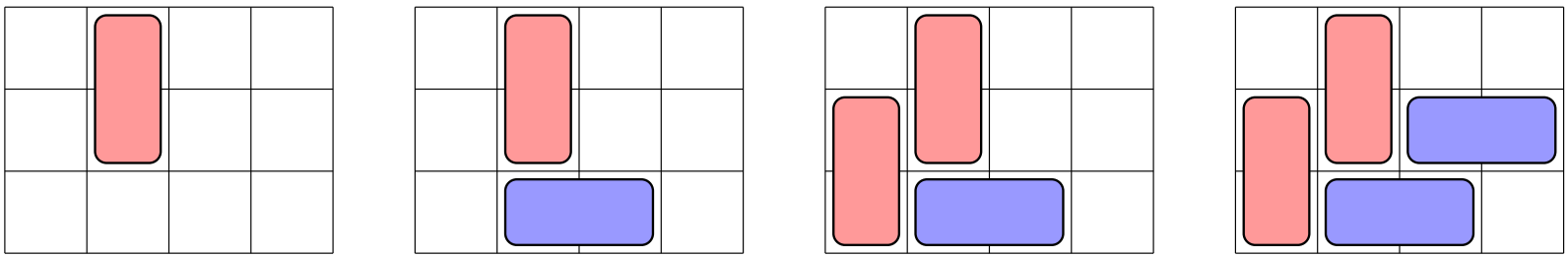
Algorithme min-max

Le 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. Une configuration est représentée par une matrice (-1 = vide, 0 = Alice, 1 = Bob)



Algorithme min-max

Le 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. Une configuration est représentée par une matrice (-1 = vide, 0 = Alice, 1 = Bob)



Exercice

1. Proposer une heuristique pour le domineering.
2. L'implémenter sous forme d'une fonction `h(v)` où `v` est une matrice représentant la configuration.

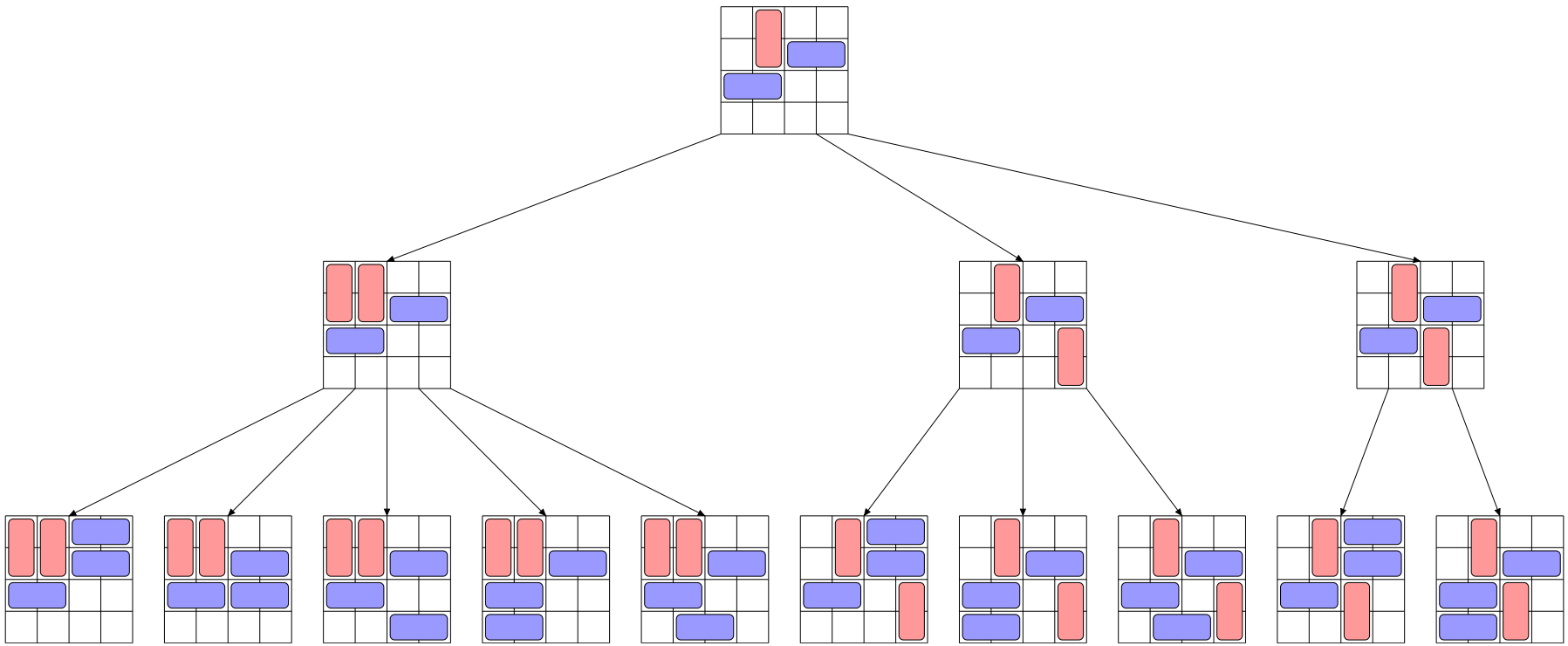
Algorithme min-max

```
def h(v):
    n1, n2 = 0, 0
    for i in range(len(v)):
        for j in range(len(v[0])):
            if i < len(v) - 1 and v[i][j] == v[i + 1][j] == -1:
                n1 += 1
            if j < len(v[0]) - 1 and v[i][j] == v[i][j + 1] == -1:
                n2 += 1
    if n1 == 0:
        return -float('inf')
    if n2 == 0:
        return float('inf')
    return n1 - n2
```

Algorithme min-max

On fixe une profondeur $p \in \mathbb{N}$.
L'**algorithme min-max** consiste à regarder, depuis la position en cours, toutes les positions atteignables après p coups et conserver celle ayant la meilleure heuristique.

Exemple : arbre des positions atteignables après $p = 2$ coups.



Algorithme min-max

L'algorithme min-max donne une valeur à chaque sommet de l'arbre de proche en proche :

1. Calcul de l'heuristique des sommets à profondeur p et ceux sans successeurs.

Algorithme min-max

L'algorithme min-max donne une valeur à chaque sommet de l'arbre de proche en proche :

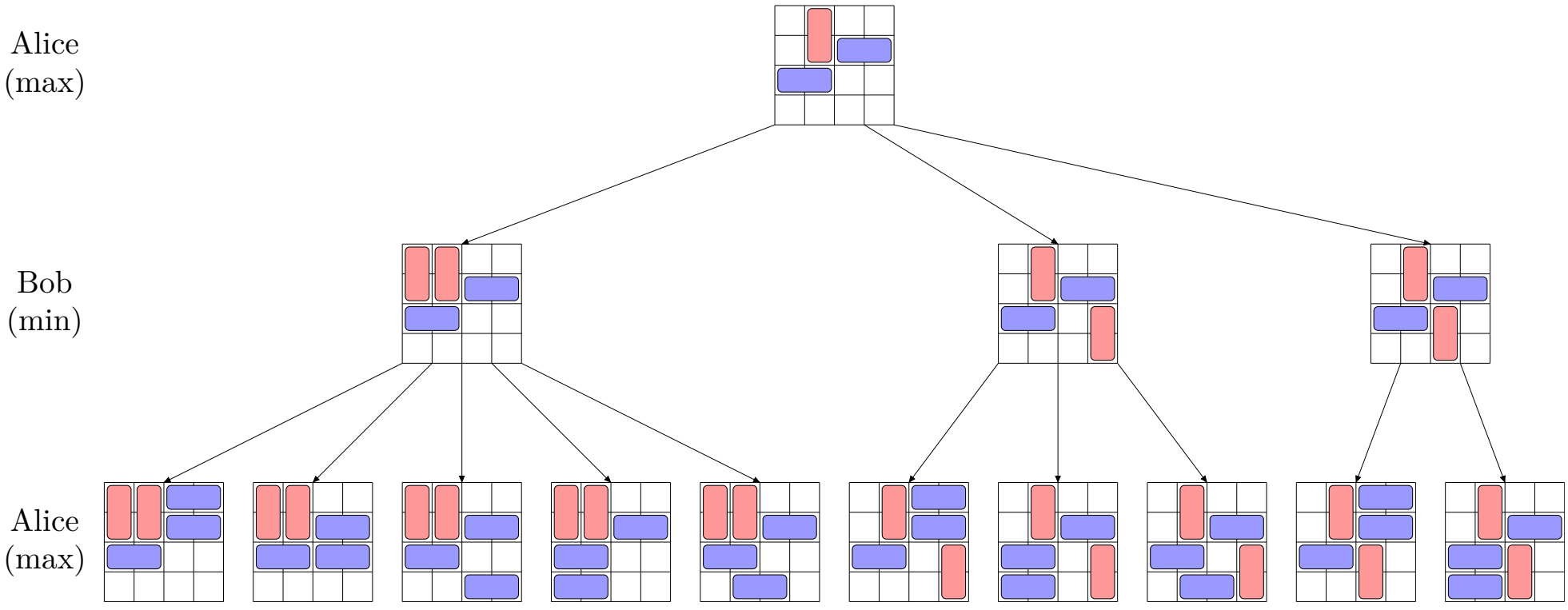
1. Calcul de l'heuristique des sommets à profondeur p et ceux sans successeurs.
2. Calcul de la valeur des sommets à profondeur $p-1$ en prenant le maximum (pour Alice) ou le minimum (pour Bob) des valeurs des successeurs.

Algorithme min-max

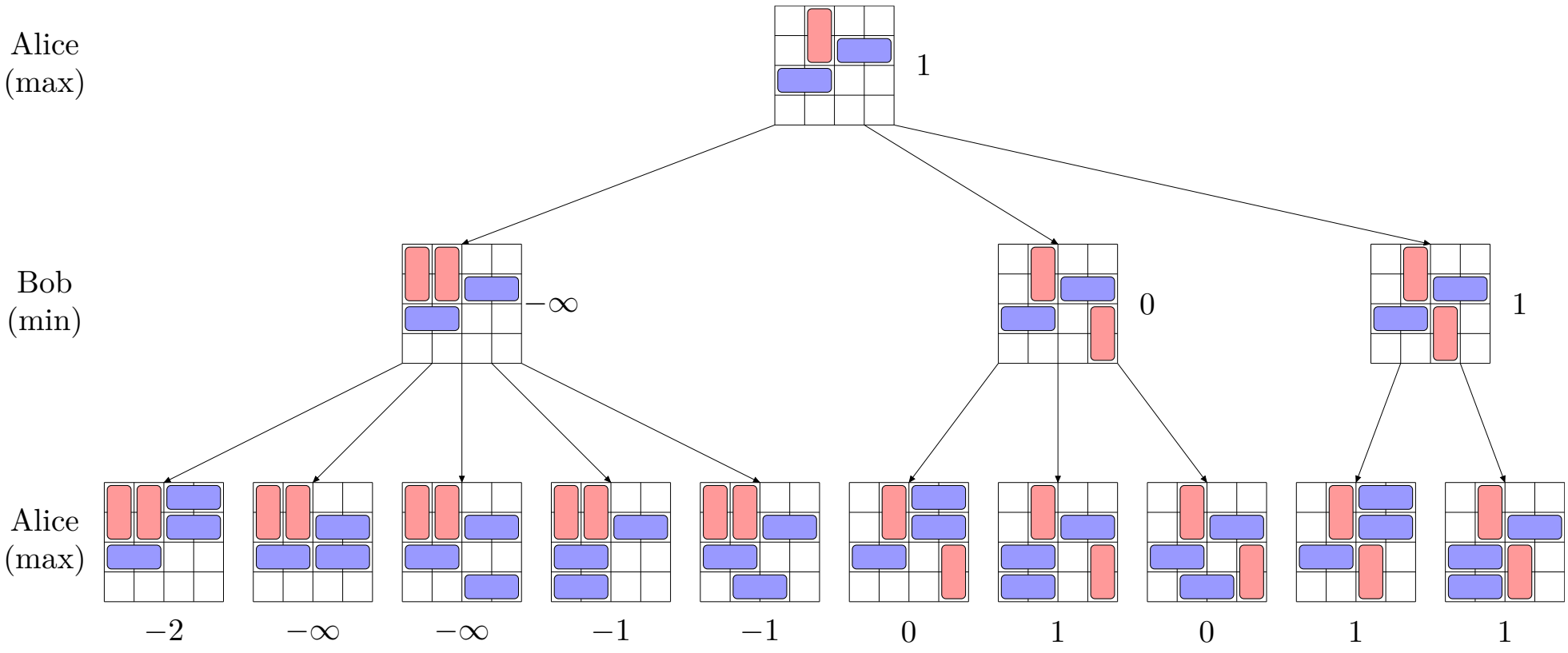
L'algorithme min-max donne une valeur à chaque sommet de l'arbre de proche en proche :

1. Calcul de l'heuristique des sommets à profondeur p et ceux sans successeurs.
2. Calcul de la valeur des sommets à profondeur $p-1$ en prenant le maximum (pour Alice) ou le minimum (pour Bob) des valeurs des successeurs.
3. ...
4. Calcul de la valeur de la racine.

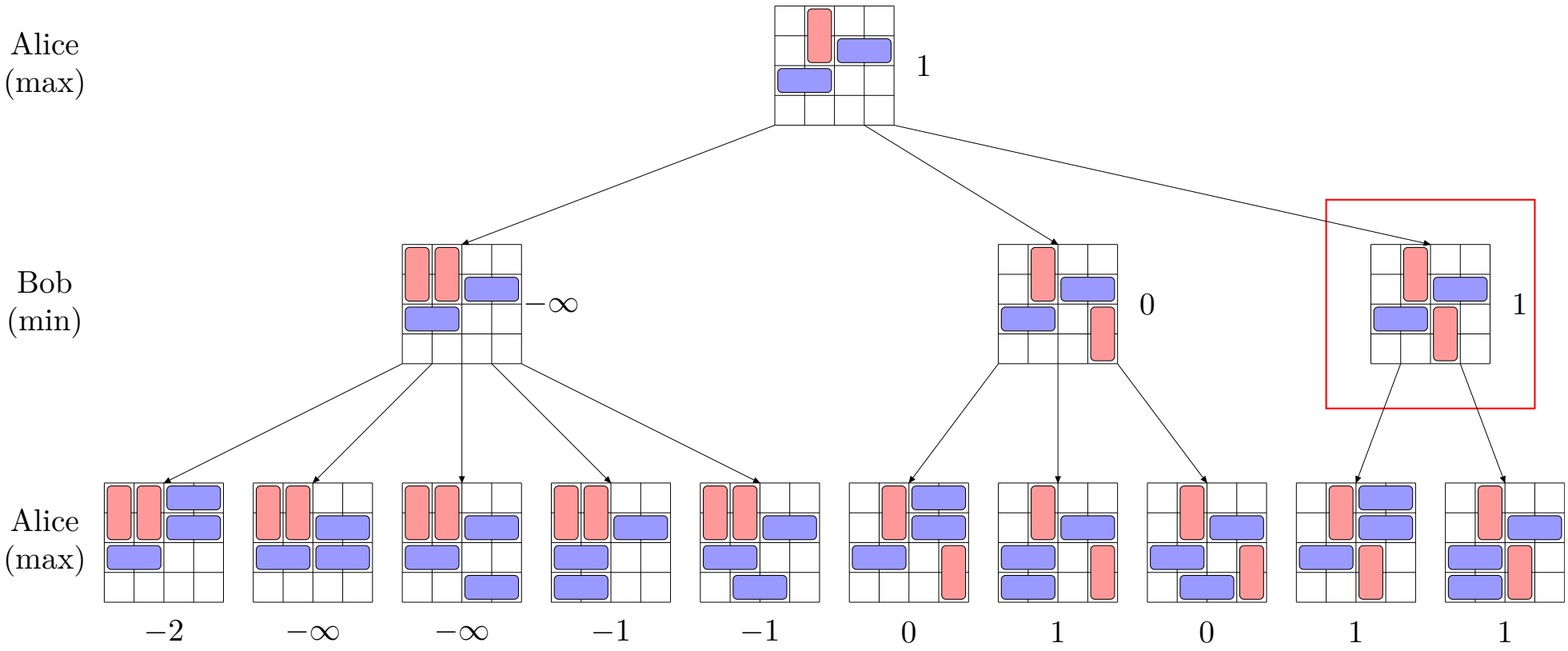
Algorithme min-max



Algorithme min-max



Algorithme min-max

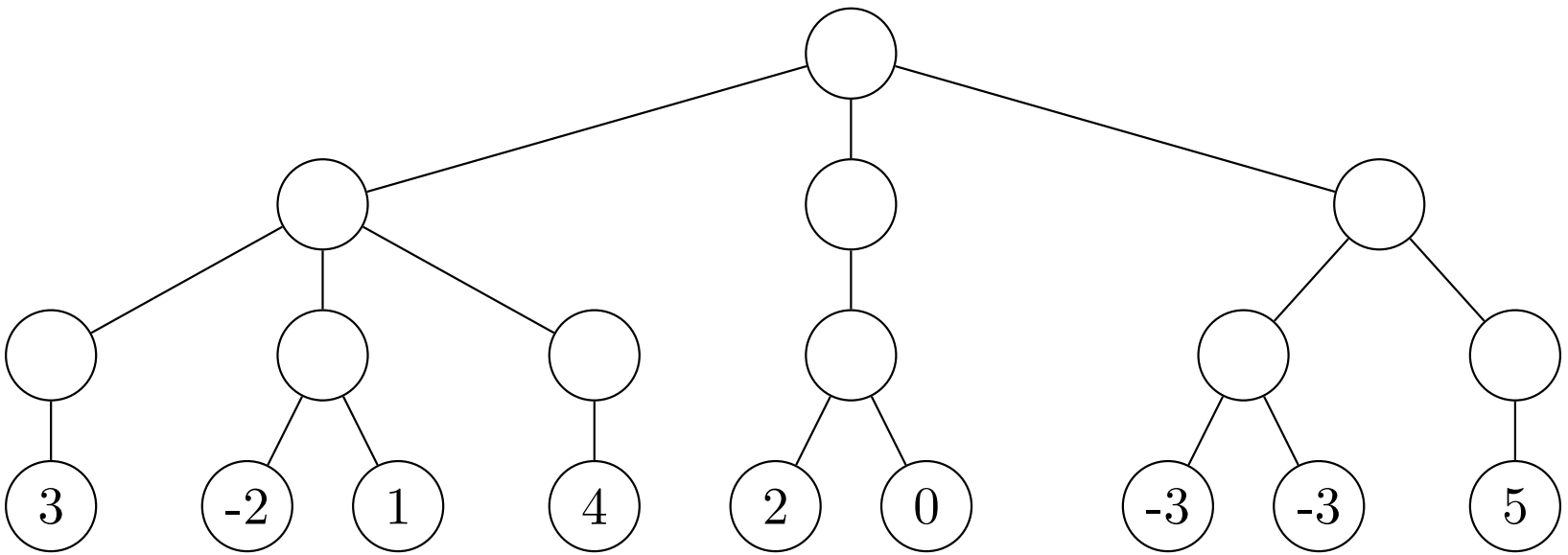


Alice choisit alors le coup maximisant l'heuristique (Bob choisirait le coup minimisant l'heuristique).

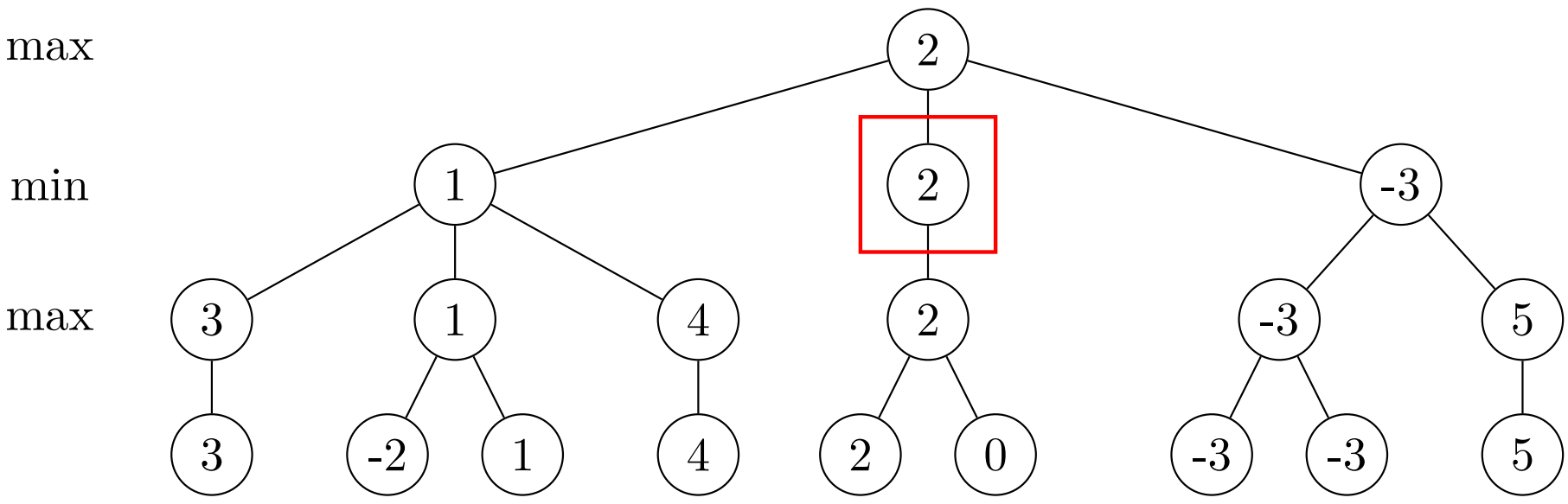
Algorithme min-max

Question

Compléter l'arbre mixmax ci-dessous, où le joueur qui joue en premier souhaite maximiser l'heuristique.



Algorithme min-max



Algorithme min-max

Question

Écrire une fonction récursive `minmax(s, h, v, p, j)` où :

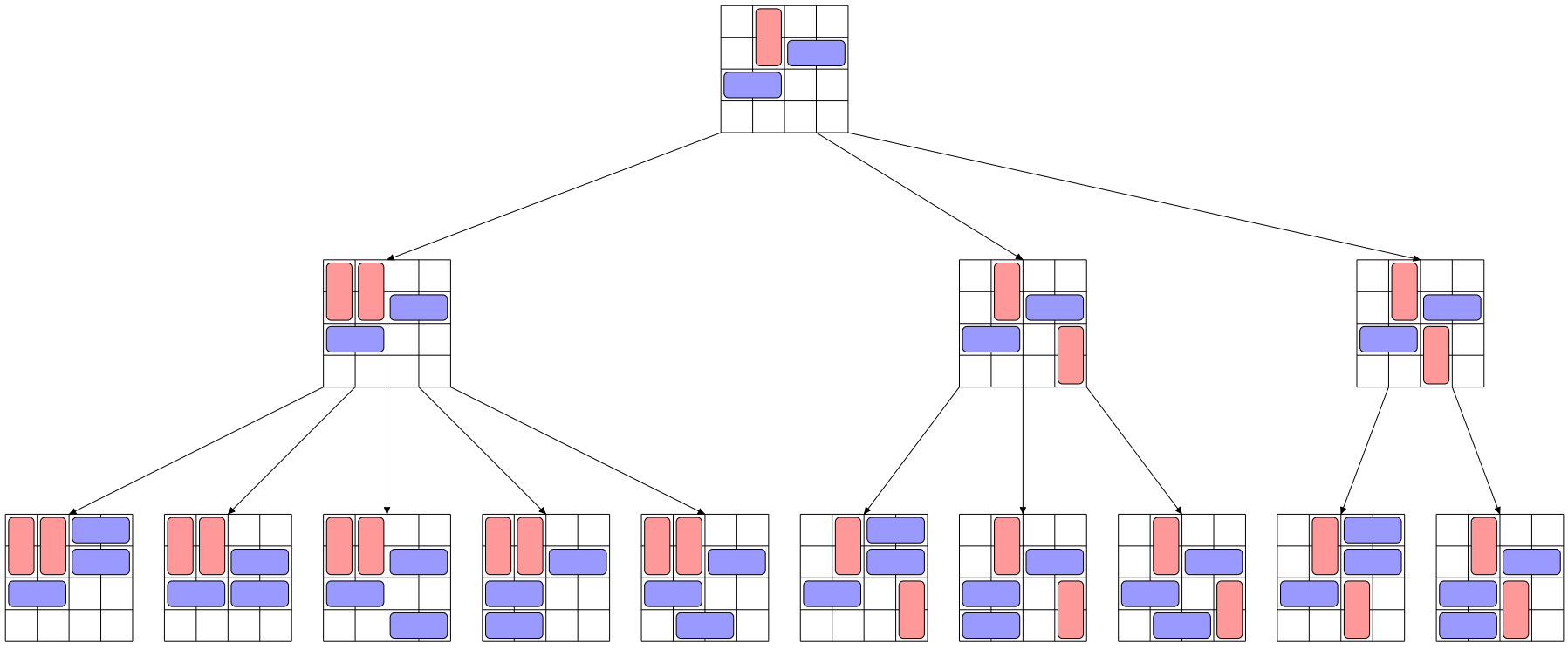
- `s` est une fonction telle que `s(v, j)` donne la liste des configurations atteignables depuis `v` après un coup du joueur `j`,
- `h` est une fonction heuristique,
- `v` est la configuration actuelle,
- `p` est la profondeur maximum,
- `j` est le joueur qui doit jouer.

`minmax(s, h, v, p, j)` renvoie la valeur de la position `v`, en appliquant l'algorithme min-max à la profondeur `p`.

Algorithme min-max

```
def minmax(s, h, v, p, j):
    succ = [minmax(s, h, w, p - 1, 1 - j) for w in s(v, j)]
    if succ == [] or p == 0:
        return h(v)
    if j == 0:
        return max(succ)
    else:
        return min(succ)
```

Algorithme min-max



Question

Pour le jeu du domineering, implémenter la fonction `s(v, j)` qui donne la liste des configurations atteignables depuis `v` après un coup du joueur `j`.

Algorithme min-max

Question

Adapter l'algorithme min-max de façon à renvoyer aussi le prochain coup à jouer.

`minmax(s, h, v, p, j)` va renvoyer un couple `(valeur, coup)`.