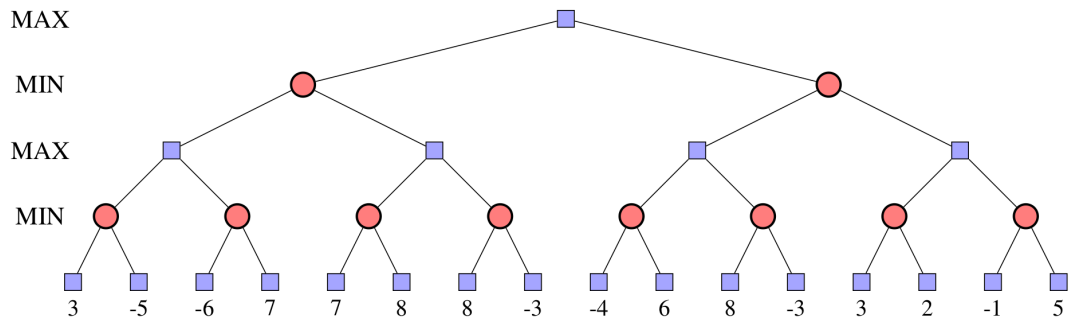


# [IA02] TD Jeux – Algorithme min-max et élagage $\alpha$ - $\beta$

## Exercice 1

On considère l'arbre de jeu suivant :



1. Appliquer l'algorithme min-max sur cet arbre.
2. Appliquer l'algorithme  $\alpha$ - $\beta$  sur cet arbre.

## Exercice 2

On considère le jeu du Tic-tac-toe (morpion). Rappel des règles (source wikipedia) :

Deux joueurs s'affrontent. Ils doivent remplir chacun à leur tour une case de la grille avec le symbole qui leur est attribué : O ou X. Le gagnant est celui qui arrive à aligner trois symboles identiques, horizontalement, verticalement ou en diagonale. Il est coutume de laisser le joueur jouant X effectuer le premier coup de la partie.

### Question 1

Soit l'état de jeu suivant.

3	X		O
2	O	O	X
1			X
	A	B	C

1. À qui est-ce de jouer ?
2. Utiliser l'algorithme min-max pour déterminer quel est le(s) coup(s) optimal(aux) pour ce joueur.

## Question 2

3			
2	O	X	
1			
	A	B	C

On considère l'état suivant et une recherche limitée à 2 coups.

Proposer différentes fonctions d'évaluation et les appliquer à cet état.

## Exercice 3 : Chou-fleur (final P22)

Ce jeu de cour de récréation se joue à deux. Les joueurs se font face et sont à une distance  $d$  l'un de l'autre. Ils avancent chacun leur tour d'une certaine distance qui est retranchée de la distance initiale. Le premier joueur dit *chou* en avançant et le deuxième dit *fleur* et ainsi de suite. Le but du jeu est de marcher sur le pied de l'autre, ce qui arrive quand  $d < 0$ .

Dans le cadre de cet exercice, on considère que chaque joueur peut avancer d'uniquement 3 distances différentes d'une unité arbitraire.

**Exemple :** la distance initiale est  $d = 5$ , le joueur *chou* et le joueur *fleur* ne peuvent avancer que de 1, 2 ou 3. Le joueur *chou* avance de 3, puis le joueur *fleur* de 2, puis le joueur *chou* de 1. Le joueur *chou* gagne.

On supposera pour la suite que :

- Le joueur *chou* peut avancer des trois distances  $\{8, 6, 5\}$ .
- Le joueur *fleur* peut avancer des trois distances  $\{6, 5, 4\}$ .
- La distance initiale entre les 2 joueurs est  $d = 14$ .

## Questions

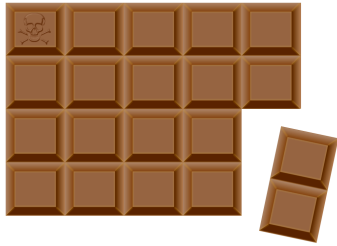
1. Modéliser le jeu chou-fleur comme un jeu à somme nulle.
2. Dans quel cas obtient-on un match nul ?
3. Résoudre ce jeu formel à l'aide de l'algorithme Min-Max. On veillera à construire et parcourir l'arbre du jeu en considérant les actions des joueurs dans l'**ordre de l'énoncé**.
4. Résoudre à l'aide de l'algorithme Alpha-Beta. *Même contrainte*.

*NB :* une autre façon, plus efficace, de traiter ce problème serait d'utiliser la programmation dynamique. Ce n'est pas ce qui est demandé ici.

# Problème – Le jeu du Chomp (final P23)

N.B. Les parties 2 et 3 sont indépendantes.

## Préliminaires



([https://commons.wikimedia.org/wiki/File:Chomp\\_game.png#/media/File:Chomp\\_game.png](https://commons.wikimedia.org/wiki/File:Chomp_game.png#/media/File:Chomp_game.png)).

Par [Lord Belbury](https://commons.wikimedia.org/wiki/User:Lord_Belbury) ([https://commons.wikimedia.org/wiki/User:Lord\\_Belbury](https://commons.wikimedia.org/wiki/User:Lord_Belbury)). — Travail personnel, [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0) (<https://creativecommons.org/licenses/by-sa/4.0>).

D'après : [https://fr.wikipedia.org/wiki/Chomp\\_\(jeu\)](https://fr.wikipedia.org/wiki/Chomp_(jeu)) ([https://fr.wikipedia.org/wiki/Chomp\\_\(jeu\)](https://fr.wikipedia.org/wiki/Chomp_(jeu))).

Le jeu du Chomp est un jeu à deux joueurs, joué sur un rectangle composé de blocs carrés représentés comme une tablette de chocolat. Chaque joueur choisit un carré à tour de rôle, et le mange, ainsi que tous les carrés situés à sa droite ou plus bas. Le carré en haut à gauche est empoisonné et celui qui le mange perd la partie.

## Exemple



Le joueur A commence. On notera les coordonnées (ligne, colonne) en partant de (1,1) pour le carré empoisonné, en haut à gauche. Il choisit le carré en (3, 5) et mange les deux carrés en bas à droite, en (3, 5) et (4, 5). Le joueur B choisit le carré en (4, 2) et mange alors 3 carrés, (4, 2), (4, 3) et (4, 4). Puis le joueur A choisit le carré en (1, 2) et mange 11 carrés. Le joueur B choisit le carré en (2, 1) et mange trois carrés de la seule colonne restante. Enfin le joueur A est obligé de manger le seul carré restant, celui en (1, 1) qui est empoisonné. Le joueur A perd donc la partie.

Sauf mention contraire, on se restreindra dans la suite de ce problème à une tablette de chocolat de **taille 2x2**.



## Partie 1 : Modélisation du jeu

1. Comment modéliser un état de jeu ? Quel est l'état de départ ?
2. Énumérer toutes les actions possibles au premier coup.
3. Donner une fonction de gain possible. Dans quel cas peut-on obtenir un match nul ?

## Partie 2 : Fonction de hachage de Zobrist pour le jeu du Chomp

On souhaite utiliser la fonction de hachage de Zobrist pour stocker un état du jeu.

On considère pour cela la suite aléatoire de nombres [3, 8, 1, 5] .

1. Donner la décomposition binaire de chacun de ces nombres. Combien de bits sont nécessaires ?
2. Combien existe-t-il d'états possibles différents dans le jeu du Chomp  $2 \times 2$  ? Les énumérer sous forme de petits schémas.
3. Pour chacun de ces états de jeu, donner une valeur de hachage de Zobrist. Donner les détails de votre modélisation et de vos calculs.

### Partie 3 : Résolution du jeu

1. Appliquer l'algorithme min-max en classant les coups (le carré croqué) les plus en bas et les plus à droite (c.-à-d. priorité au numéro de ligne le plus élevé puis au numéro de colonne le plus élevée). Dessiner l'arbre entier. Que peut-on en déduire ?
2. Appliquer l'algorithme alpha-beta en utilisant le même ordre de coups. Redessiner l'arbre en entier en donnant les valeurs successives des variables `value`, `alpha` et `beta` .
3. Proposer une fonction d'évaluation pour le jeu du Chomp en taille  $n \times n$ . Justifier le choix fait.