

*Groupe Informatique*

*IREM de Lille*

*Asli Grimaud*



# IA et Apprentissage Automatique

*Avril 2026*

`asli.grimaud@ac-lille.fr`

# Algorithme min-max et le jeu de Nim

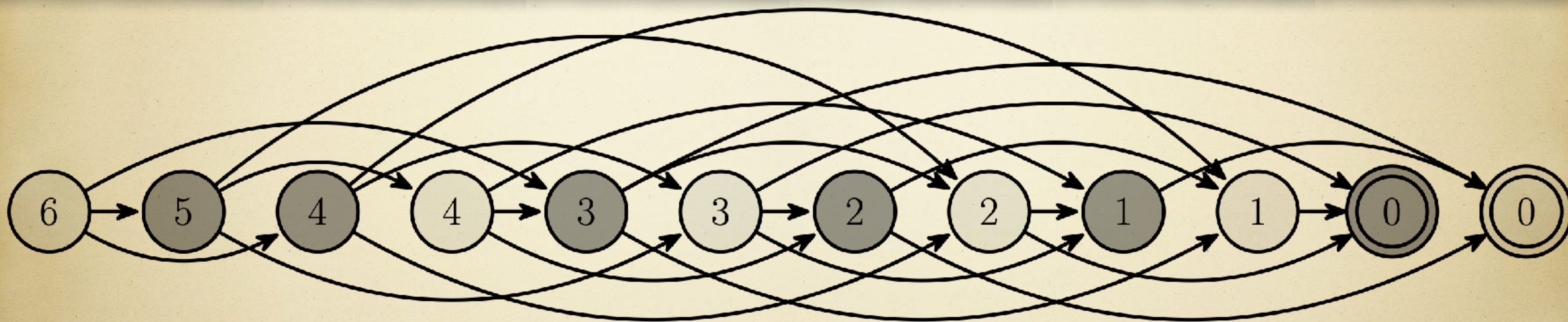


- Jeu d'accessibilité à 2 joueurs  $P_1$  et  $P_2$
- un seul tas de  $n=6$  pierres
- à chaque tour, on tire 1, 2 ou 3 pierres
- $P_1$  commence
- version : le joueur qui prend la dernière pierre a gagné



# Algorithme min-max et le jeu de Nim

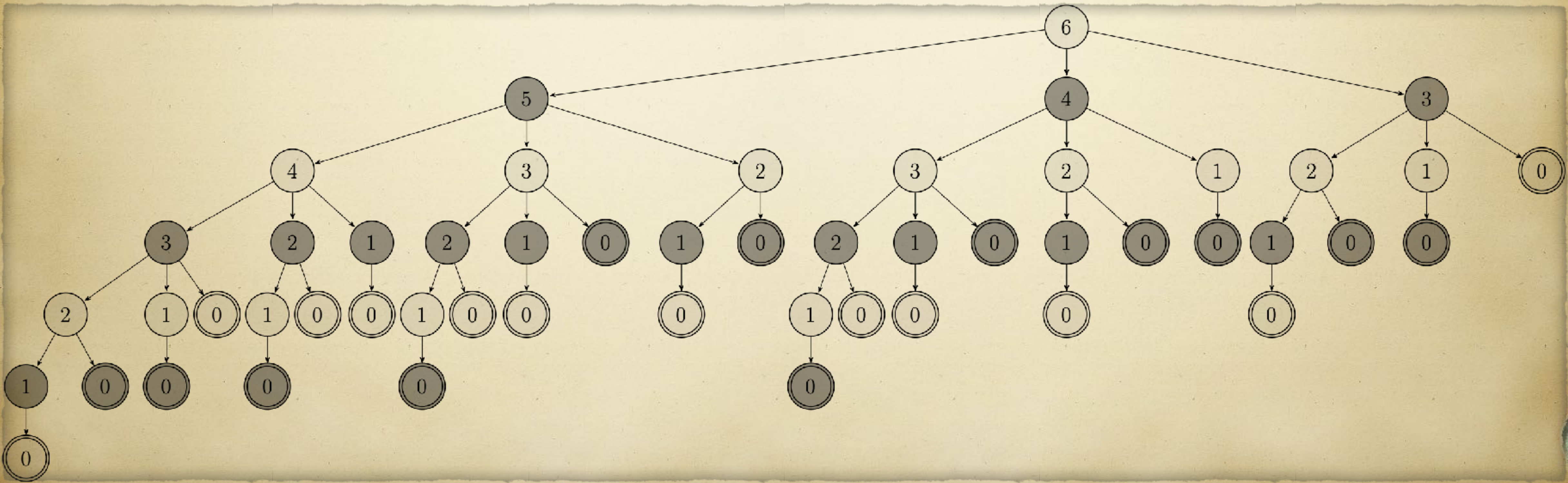
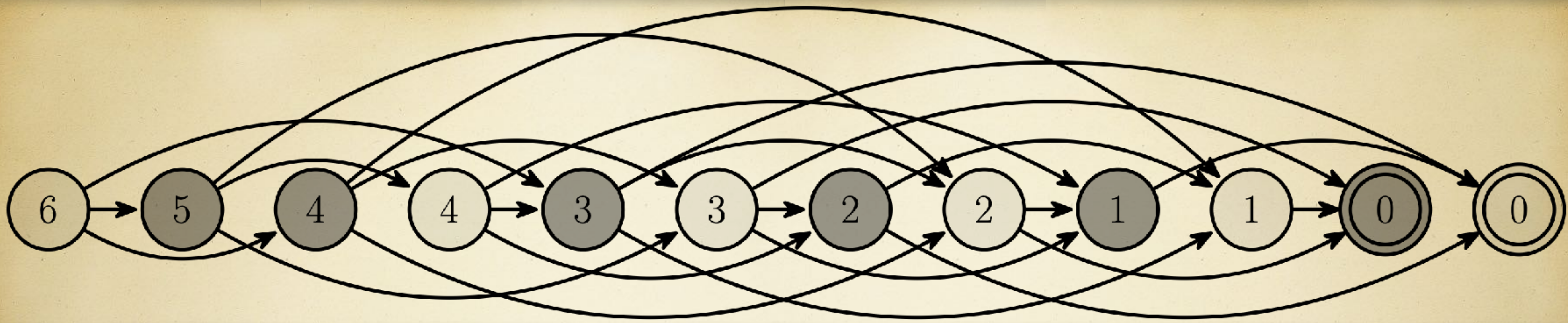
Représentation sous forme de graphe biparté  $G = (V = V_1 \oplus V_2, E)$



état gagnant pour le joueur  $P_1$



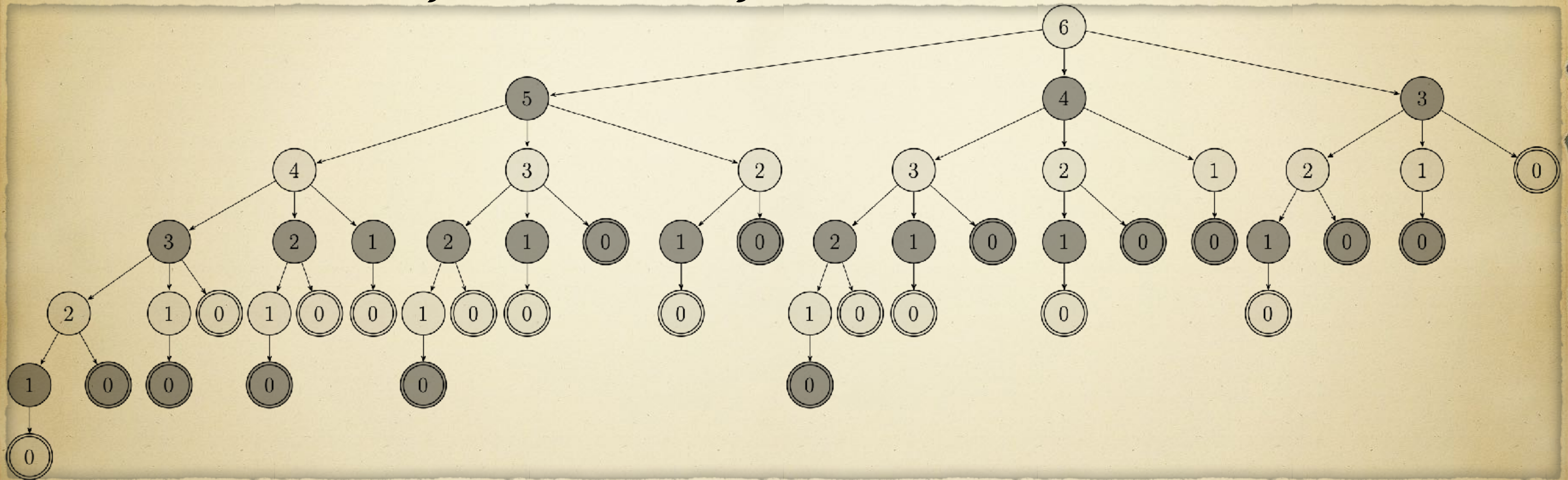
état gagnant pour le joueur  $P_2$





# Algorithme min-max et le jeu de Nim

Représentation sous forme d'arbre de décision

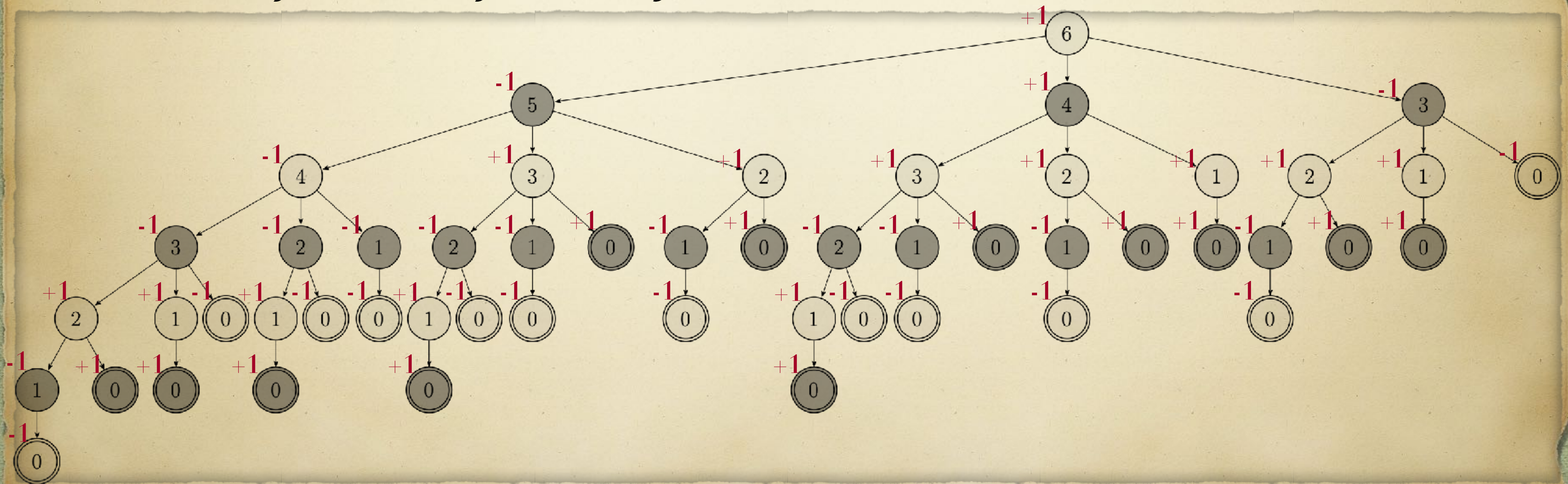


Pour  $n = 6$ , il y a 24 façons de jouer.



# Algorithme min-max et le jeu de Nim

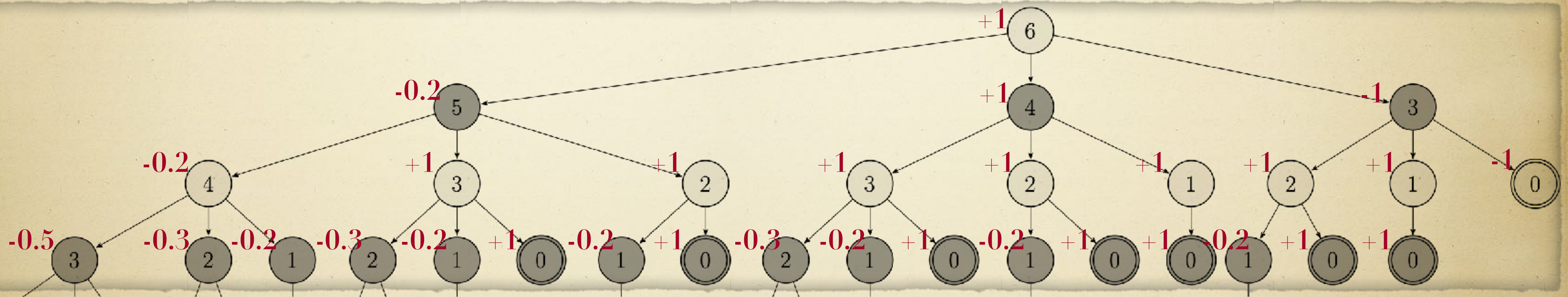
- Calculer un score pour chaque état
- Minimiser le score pour le joueur  $P_2$ , maximiser le score pour le joueur  $P_1$
- Exploration à partir des feuilles





# Algorithme min-max et le jeu de Nim

- Calculer un score pour chaque état
- Minimiser le score pour le joueur  $P_2$ , maximiser le score pour le joueur  $P_1$
- Exploration à partir des feuilles



• *parité des tas*

Heuristiques

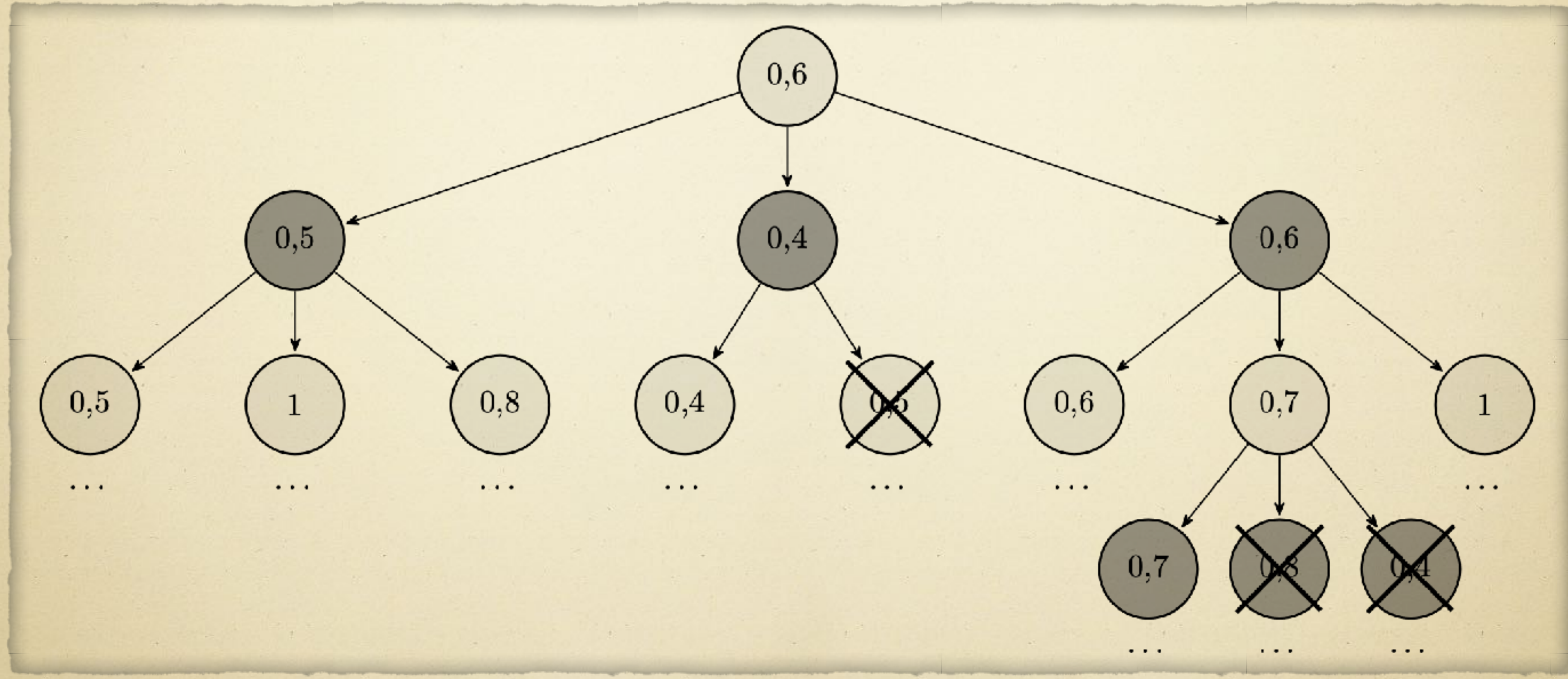
• *somme des pierres restantes*

Valeur normalisée dans  $[-1, 1]$

• *taille du plus grand tas*



# Algorithme min-max et le jeu de Nim Élagage alpha-bêta



# Algorithme min-max et le jeu de Nim

## Implémentation

**human**



$n_1$

$n_2$

$n_3$

$n_4$

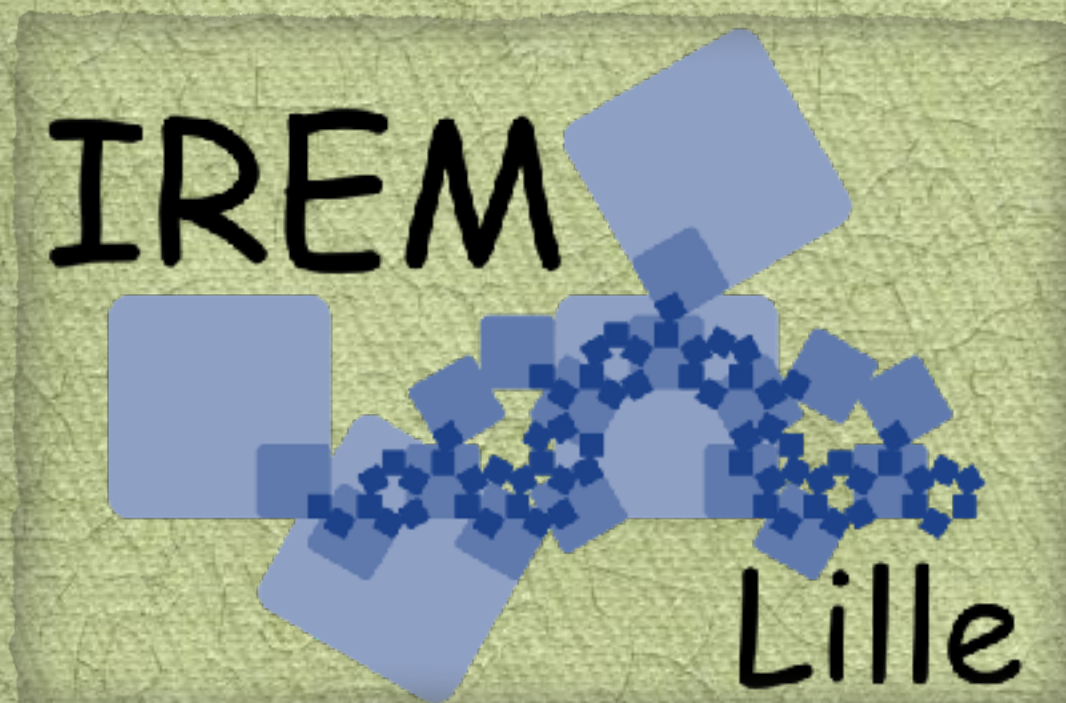
**computer**

random\_strategy

min\_max\_strategy

min\_max\_ab\_strategy

**main\_loop**



*Groupe Informatique*

*IREM de Lille*

*Asli Grimaud*



*IA et Apprentissage Automatique*

*MERCI*

*Avril 2026*

*asli.grimaud@ac-lille.fr*