Algorithmes MinMax

Clément PERNET

L3-MI, UFR IM^{2}AG, Université Grenoble Alpes

Plan

Algorithme Minmax

Élagage alpha-beta

Plan

Principe de l'algorithme MinMax

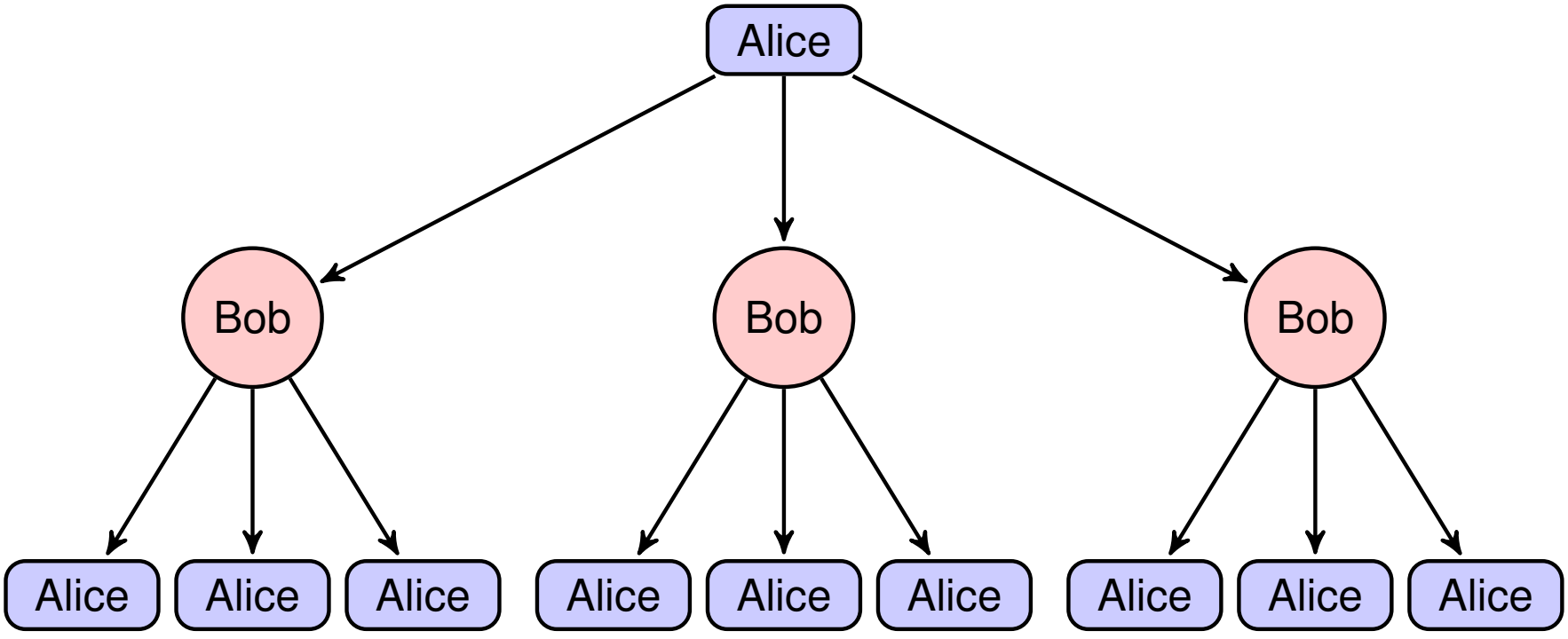
Contexte : théorie des jeux

- Jeu à deux joueurs (Alice et Bob)
- à somme nulle
- et à information complète

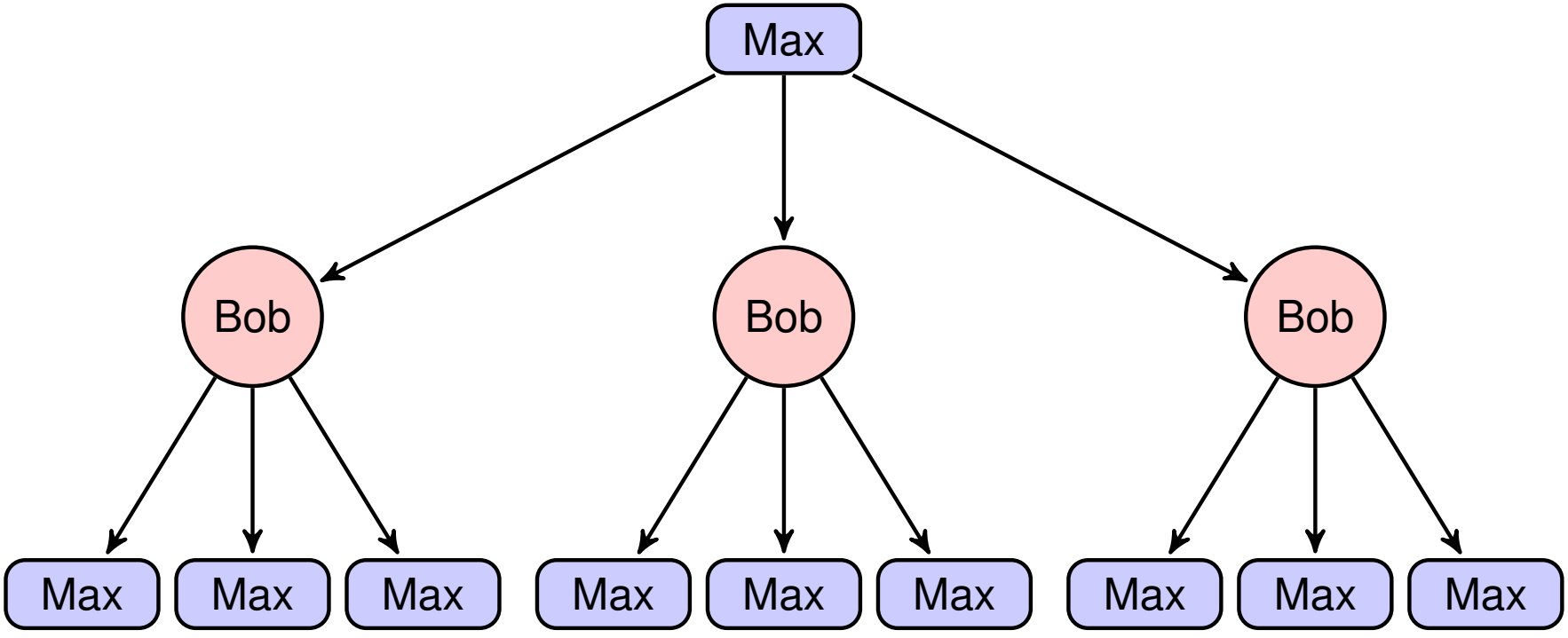
On travail sur l'arbre du jeu :

- chaque configuration du jeu correspond à un noeud.
  $\rightsquigarrow$ 2 types de noeuds
  - à Alice de jouer
  - à Bob de jouer
- Arêtes $=$ liste des coups possibles à jouer
  $\rightsquigarrow$ mène à des noeuds distincts

Principe de l'algorithme MinMax

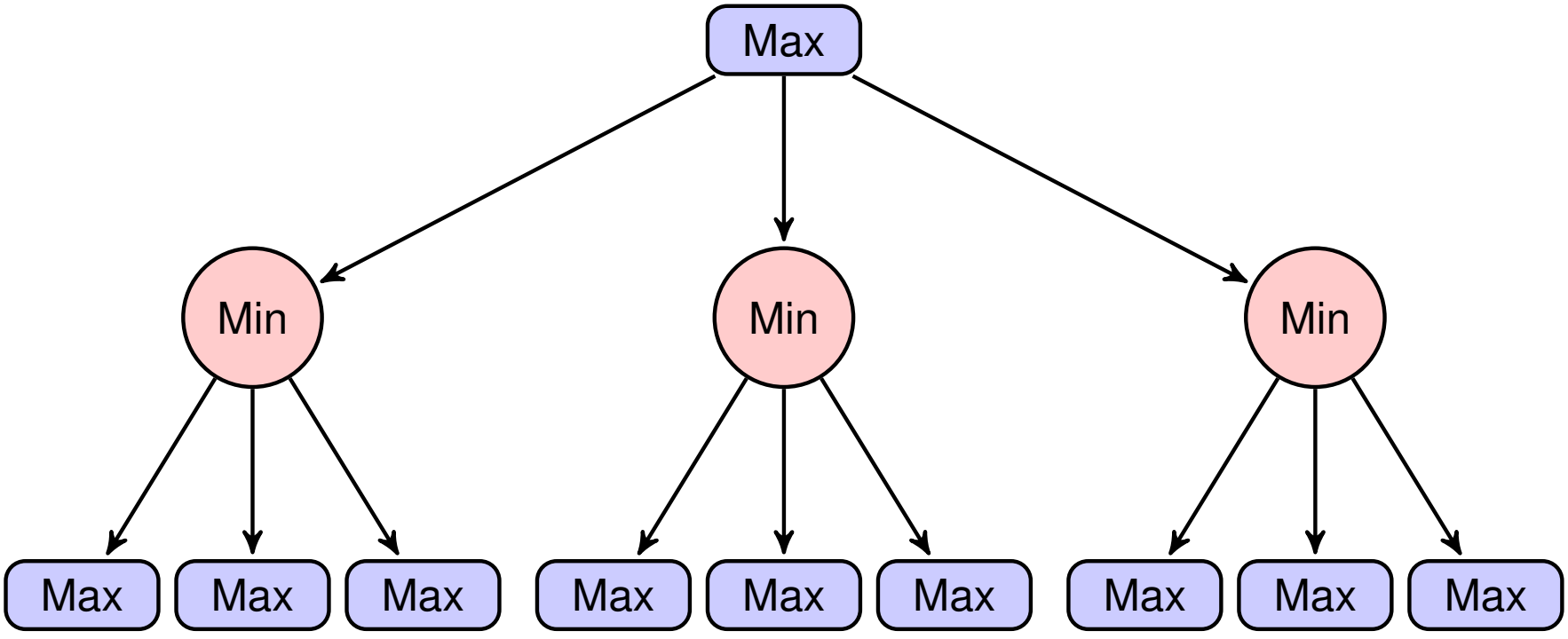


Principe de l'algorithme MinMax



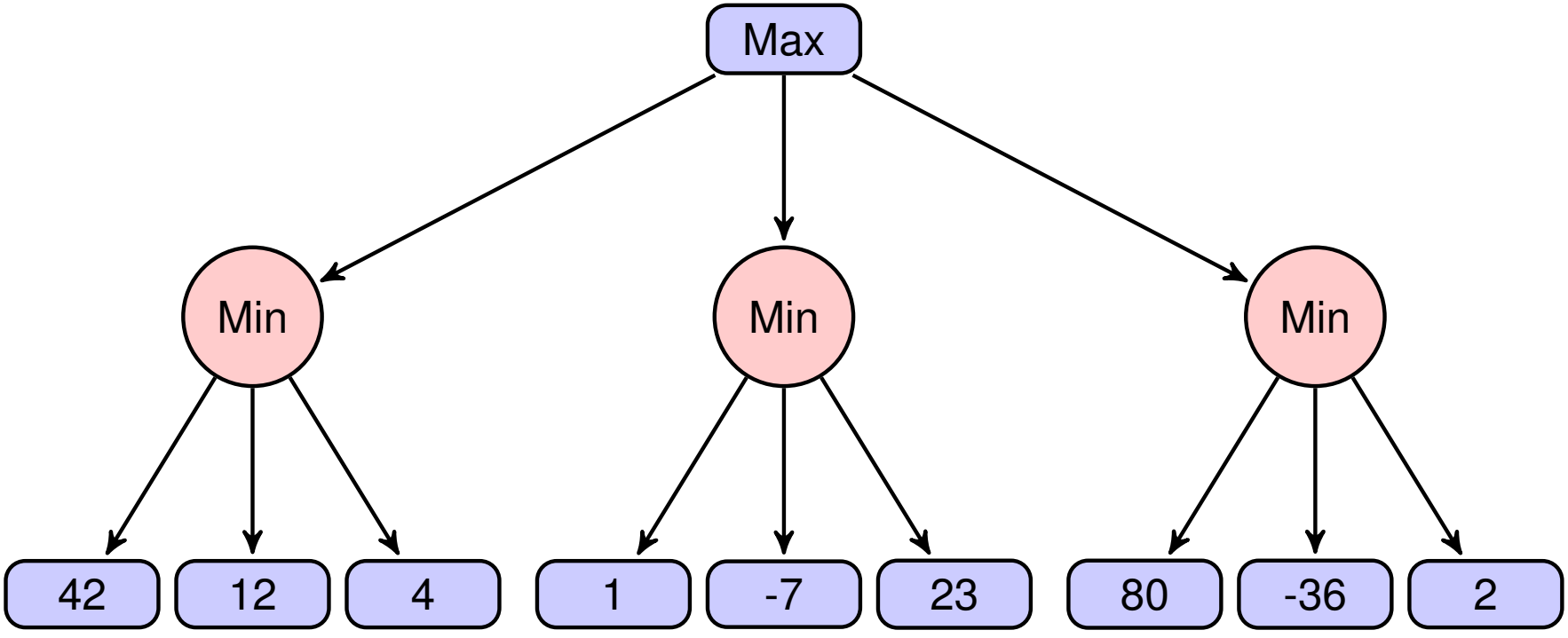
- Alice cherche à maximiser son score

Principe de l'algorithme MinMax



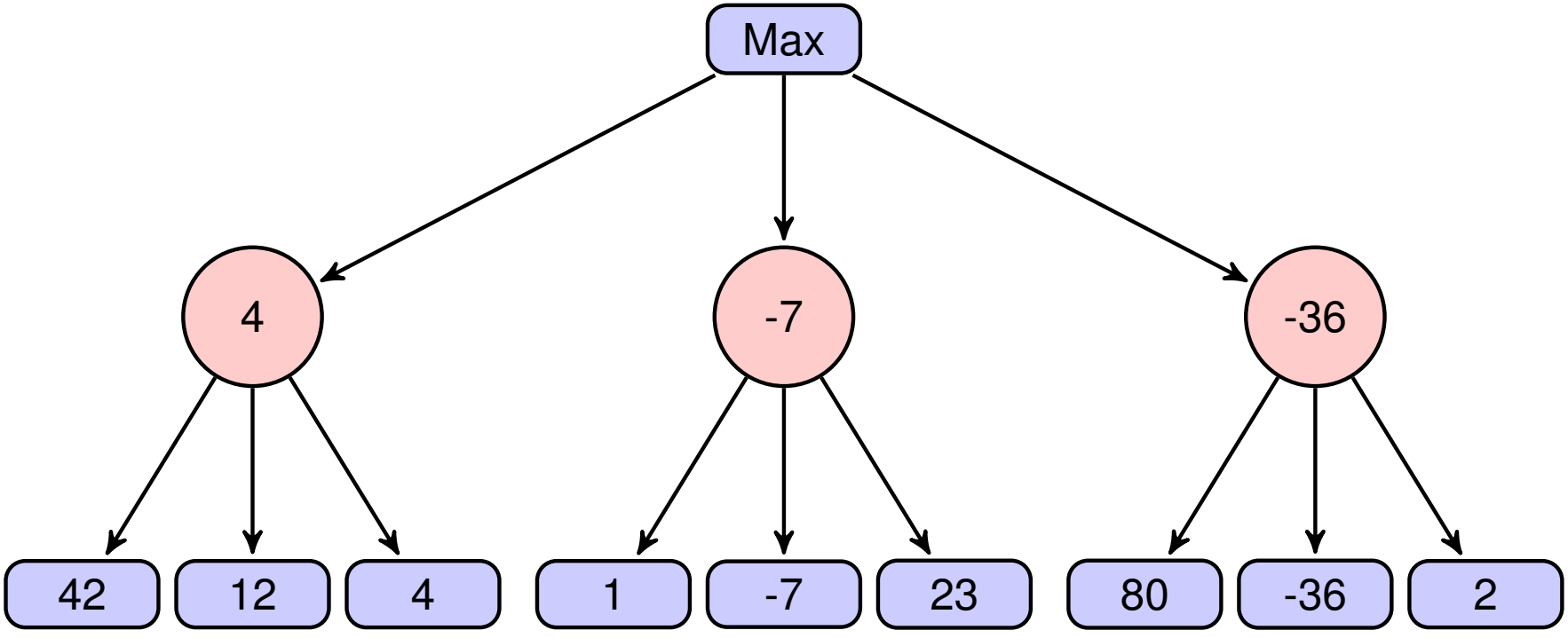
- Alice cherche à maximiser son score
- Bob aussi : $\rightsquigarrow$ minimiser le score de Alice

Principe de l'algorithme MinMax



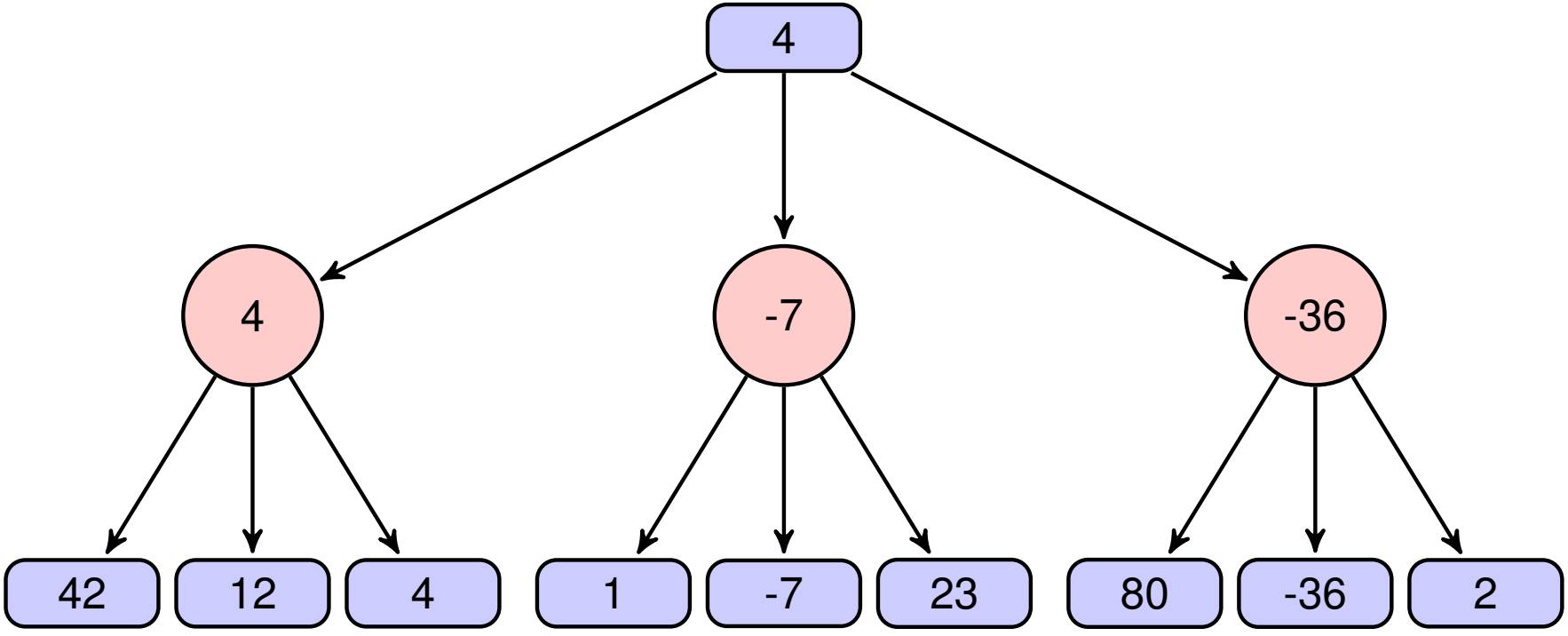
- Alice cherche à maximiser son score
- Bob aussi : $\rightsquigarrow$ minimiser le score de Alice
- A un niveau de récursion : on évalue le score (heuristique)

Principe de l'algorithme MinMax



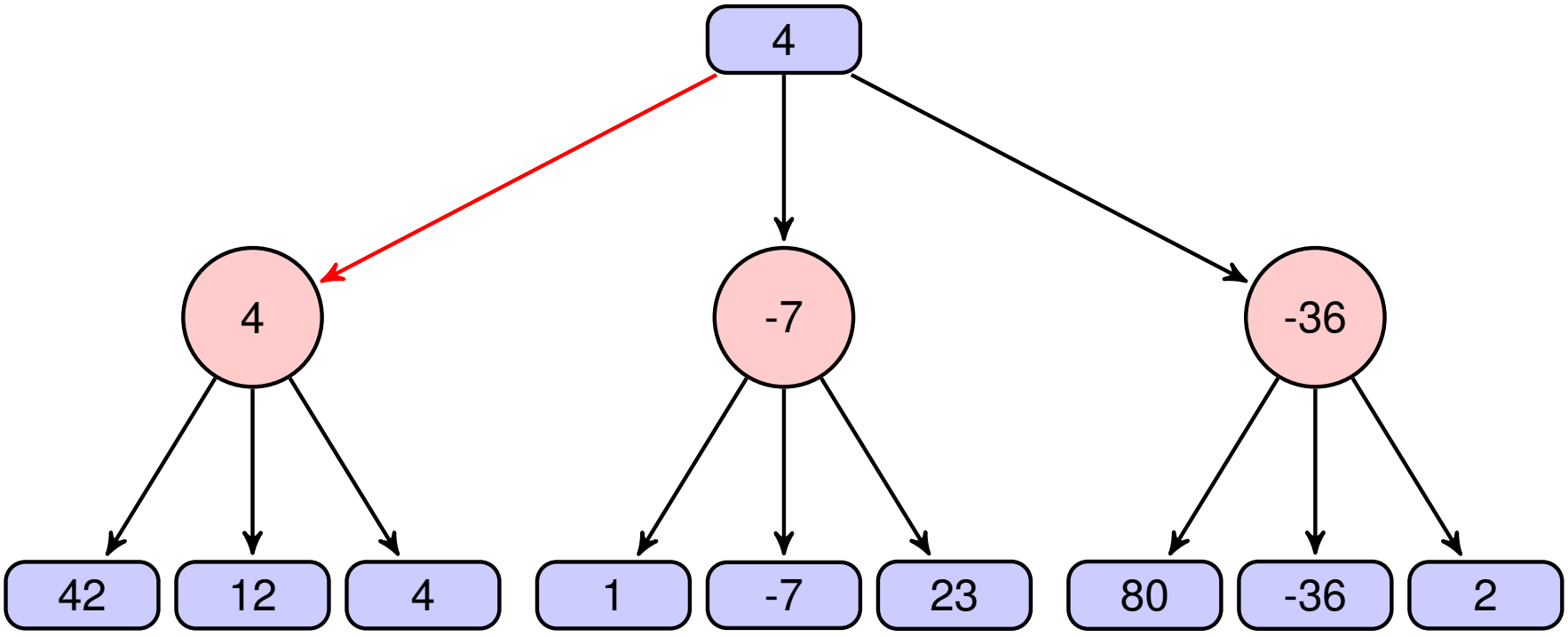
- Alice cherche à maximiser son score
- Bob aussi : $\rightsquigarrow$ minimiser le score de Alice
- A un niveau de récursion : on évalue le score (heuristique)
- On en déduit la valeur des noeud internes

Principe de l'algorithme MinMax



- Alice cherche à maximiser son score
- Bob aussi : $\rightsquigarrow$ minimiser le score de Alice
- A un niveau de récursion : on évalue le score (heuristique)
- On en déduit la valeur des noeud internes

Principe de l'algorithme MinMax



- Alice cherche à maximiser son score
- Bob aussi : $\rightsquigarrow$ minimiser le score de Alice
- A un niveau de récursion : on évalue le score (heuristique)
- On en déduit la valeur des noeud internes
- Donc le coup à jouer

Principe de l'algorithme MinMax

Quelques remarques :

- Si sur un noeud, la partie est gagnée $\rightsquigarrow$ score ∞
- Si sur un noeud, la partie est perdue $\rightsquigarrow$ score $-\infty$

Principe de l'algorithme MinMax

Quelques remarques :

- Si sur un noeud, la partie est gagnée $\rightsquigarrow$ score ∞
- Si sur un noeud, la partie est perdue $\rightsquigarrow$ score $-\infty$
- Idéalement, si on descend à une profondeur suffisamment grande, on connaît tout le jeu $\rightsquigarrow$ gagne toujours

Principe de l'algorithme MinMax

Quelques remarques :

- Si sur un noeud, la partie est gagnée $\rightsquigarrow$ score ∞
- Si sur un noeud, la partie est perdue $\rightsquigarrow$ score $-\infty$
- Idéalement, si on descend à une profondeur suffisamment grande, on connaît tout le jeu $\rightsquigarrow$ gagne toujours
- Mais coût exponentiel
  - profondeur limitée
  - importance d'avoir une évaluation fine des scores aux feuilles

Plan

Élagage alpha-beta

Idée

- couper des branches inutiles dans l'arbre pour réduire le coût
- donc permettre d'aller plus en profondeur et *voir plus loin*

Élagage alpha-beta

Idée

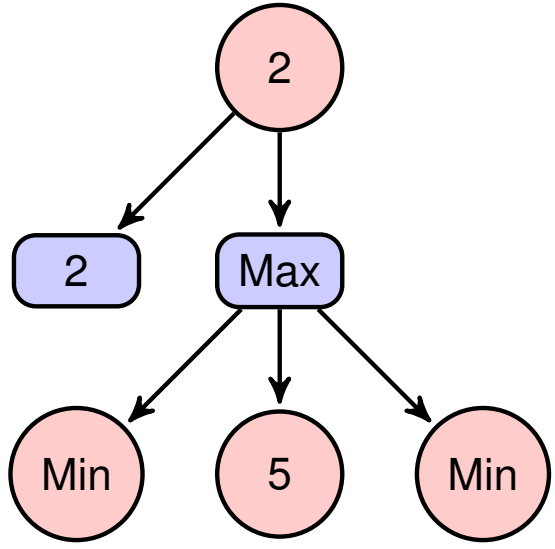
- couper des branches inutiles dans l'arbre pour réduire le coût
- donc permettre d'aller plus en profondeur et *voir plus loin*

Si un noeud MAX a :

- un ancêtre MIN avec un score partiel m
- un fils MIN de valeur $> m$

Alors :

- Pas de besoin de visiter les autres fils



Élagage alpha-beta

Idée

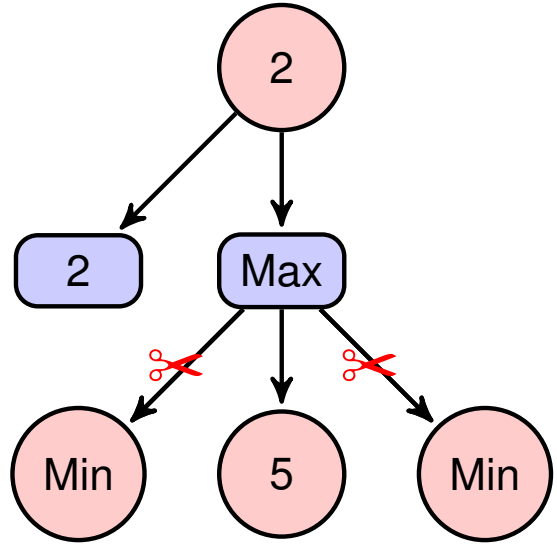
- couper des branches inutiles dans l'arbre pour réduire le coût
- donc permettre d'aller plus en profondeur et *voir plus loin*

Si un noeud MAX a :

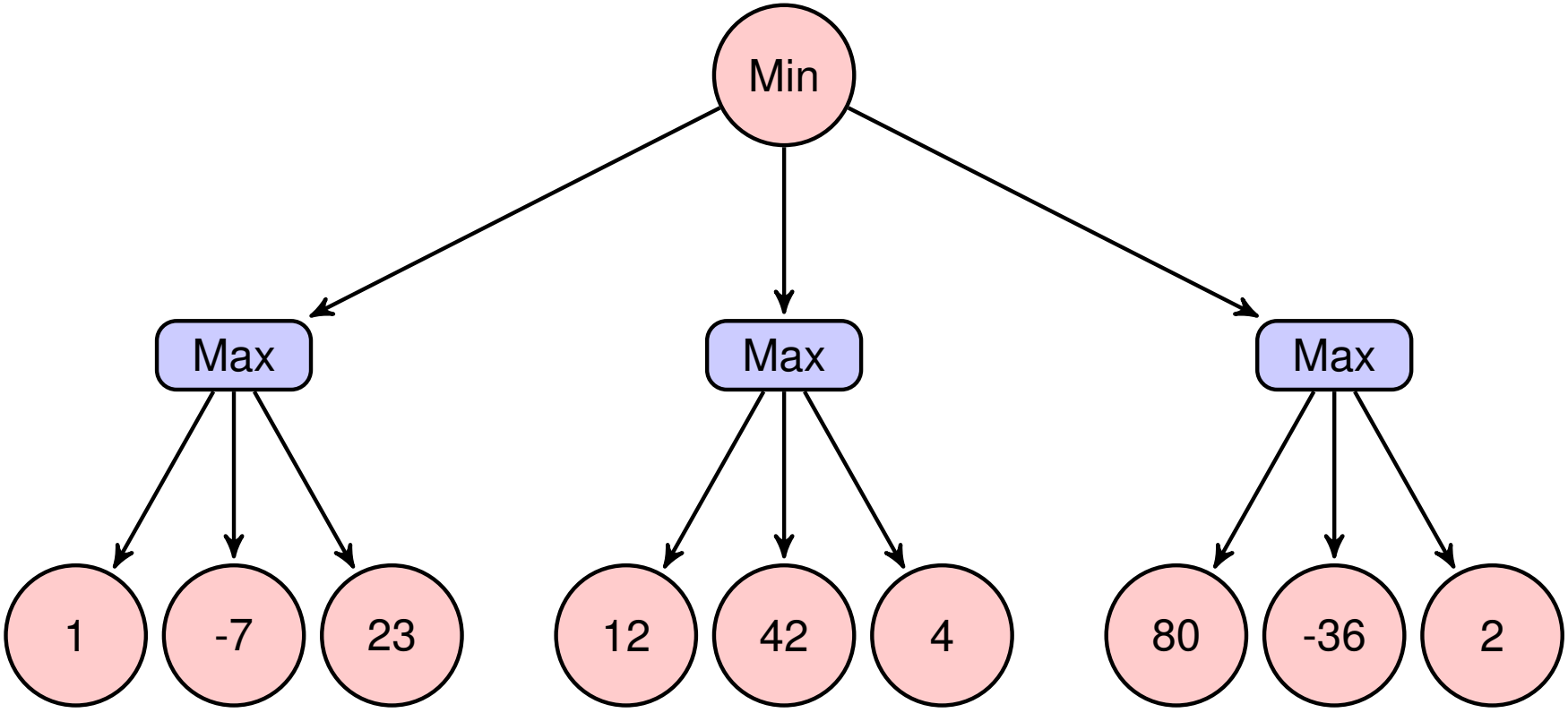
- un ancêtre MIN avec un score partiel m
- un fils MIN de valeur $> m$

Alors :

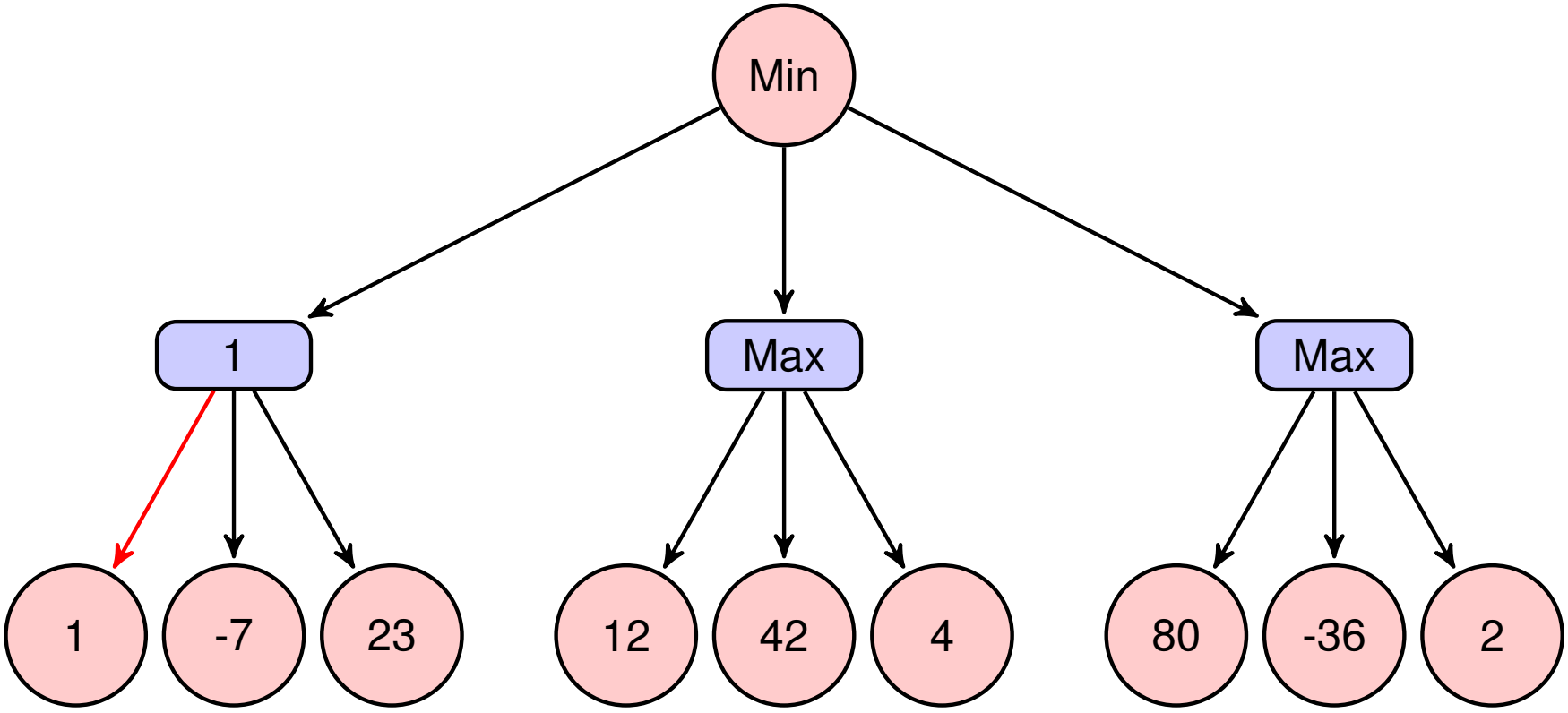
- Pas de besoin de visiter les autres fils



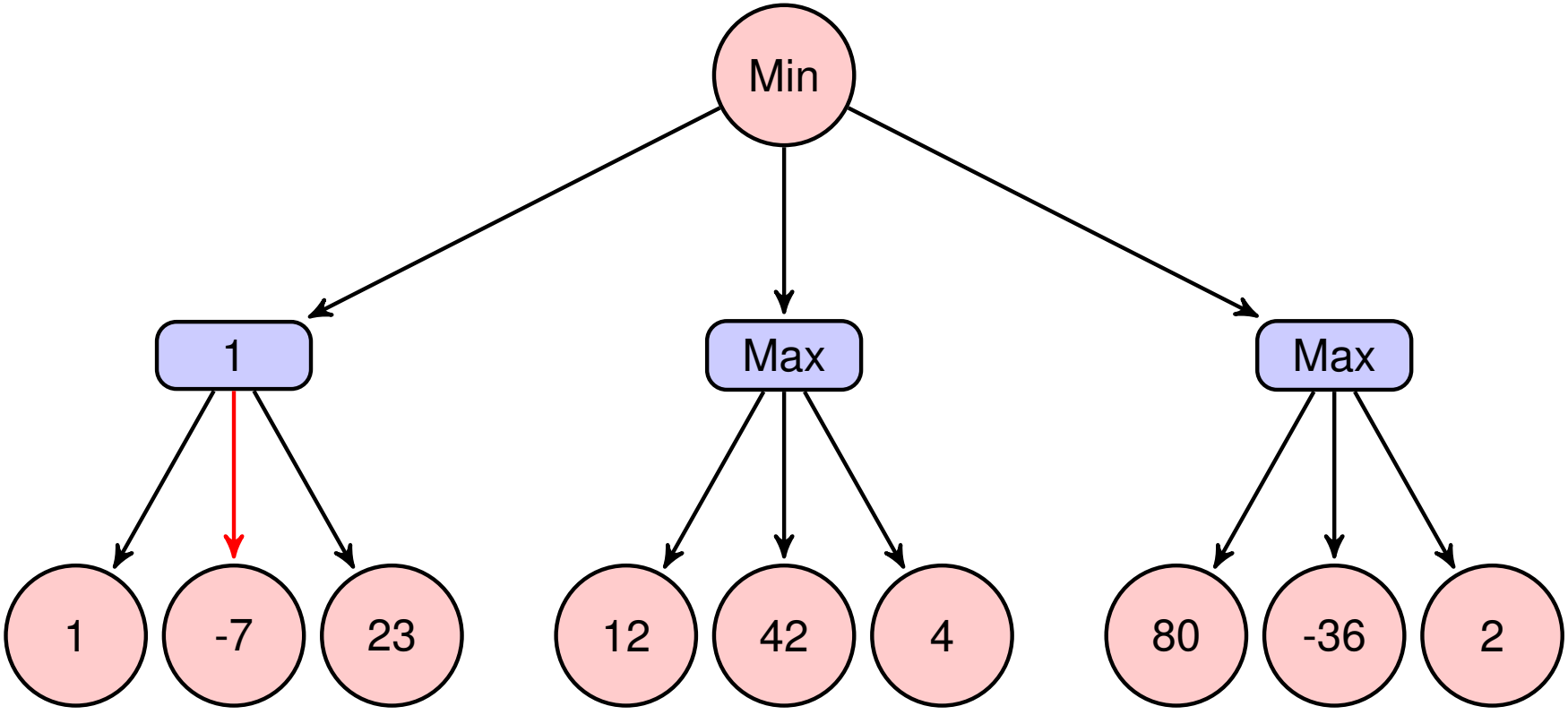
Exemple



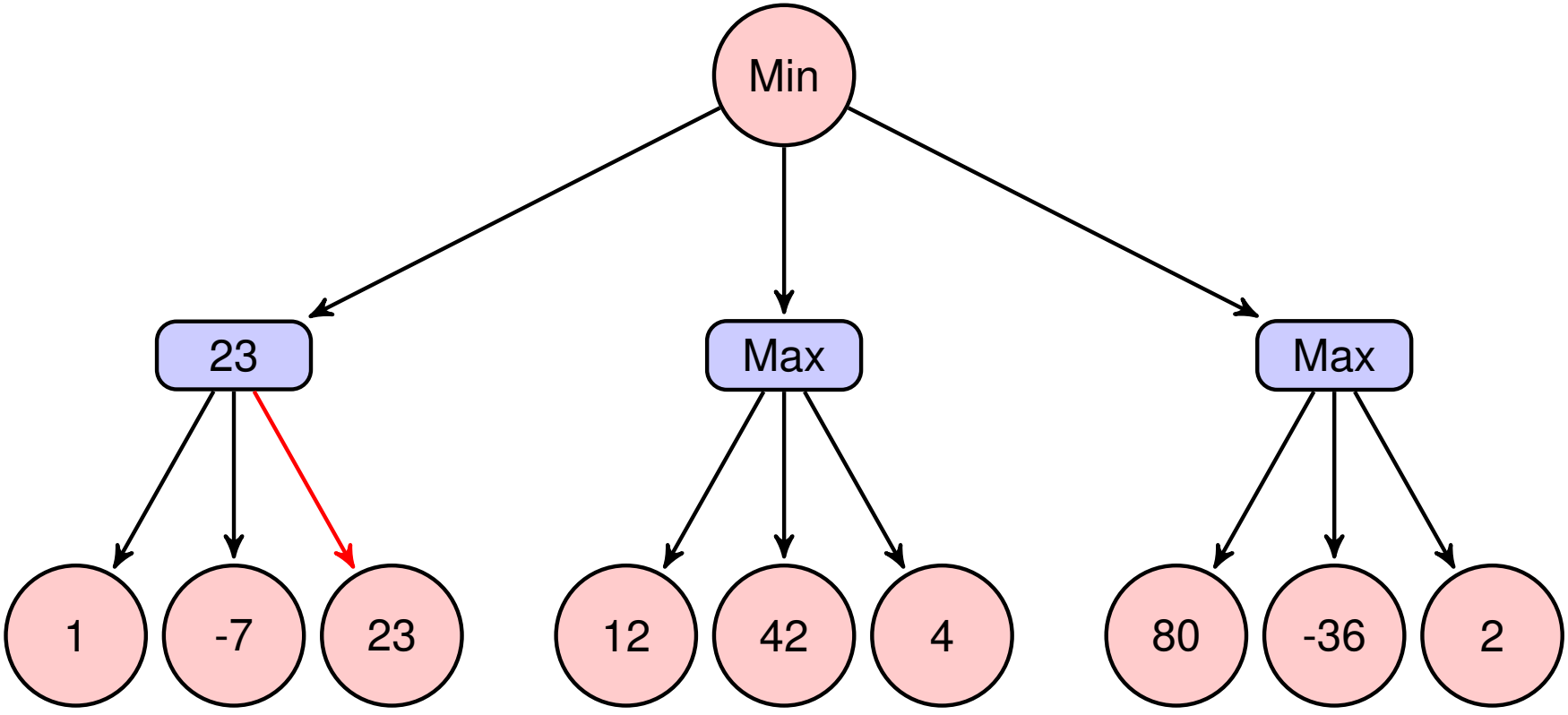
Exemple



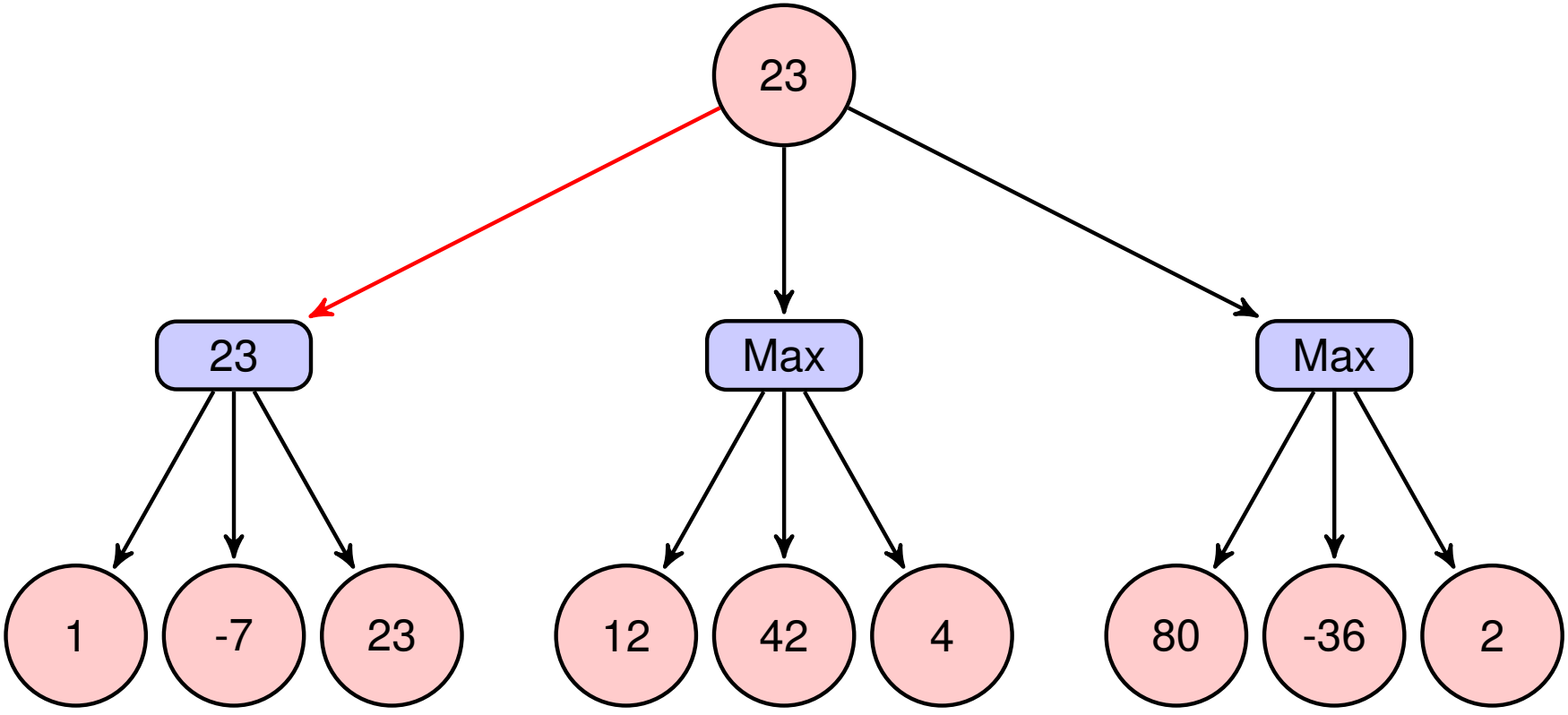
Exemple



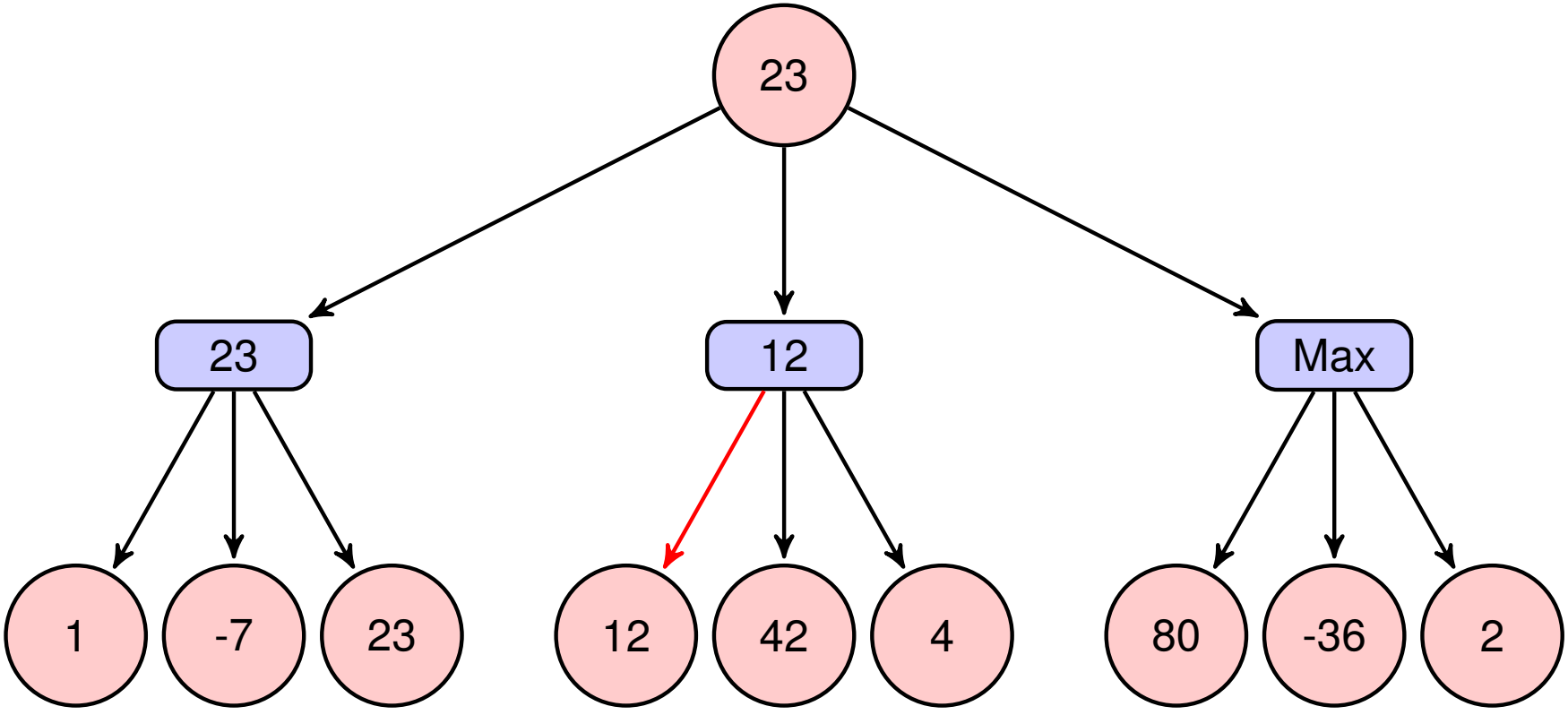
Exemple



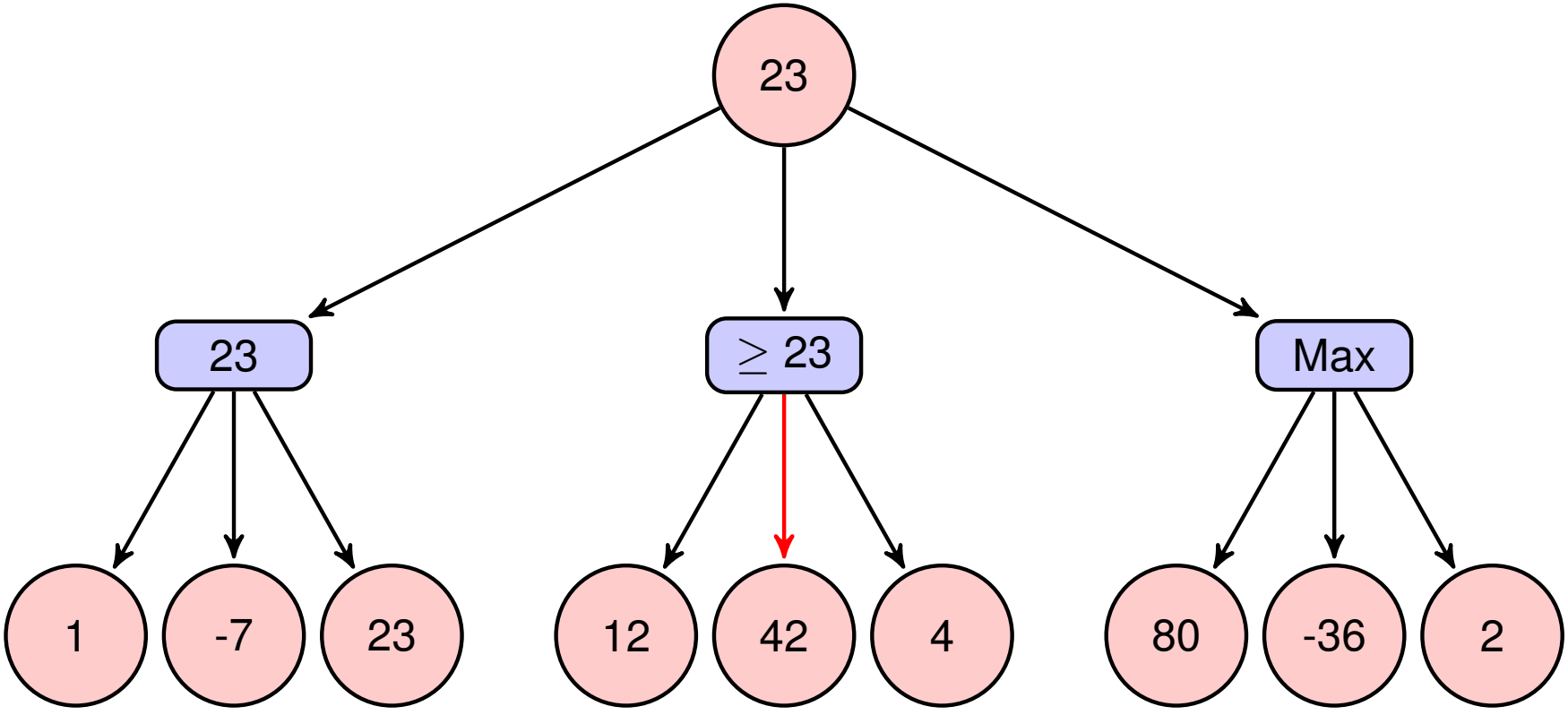
Exemple



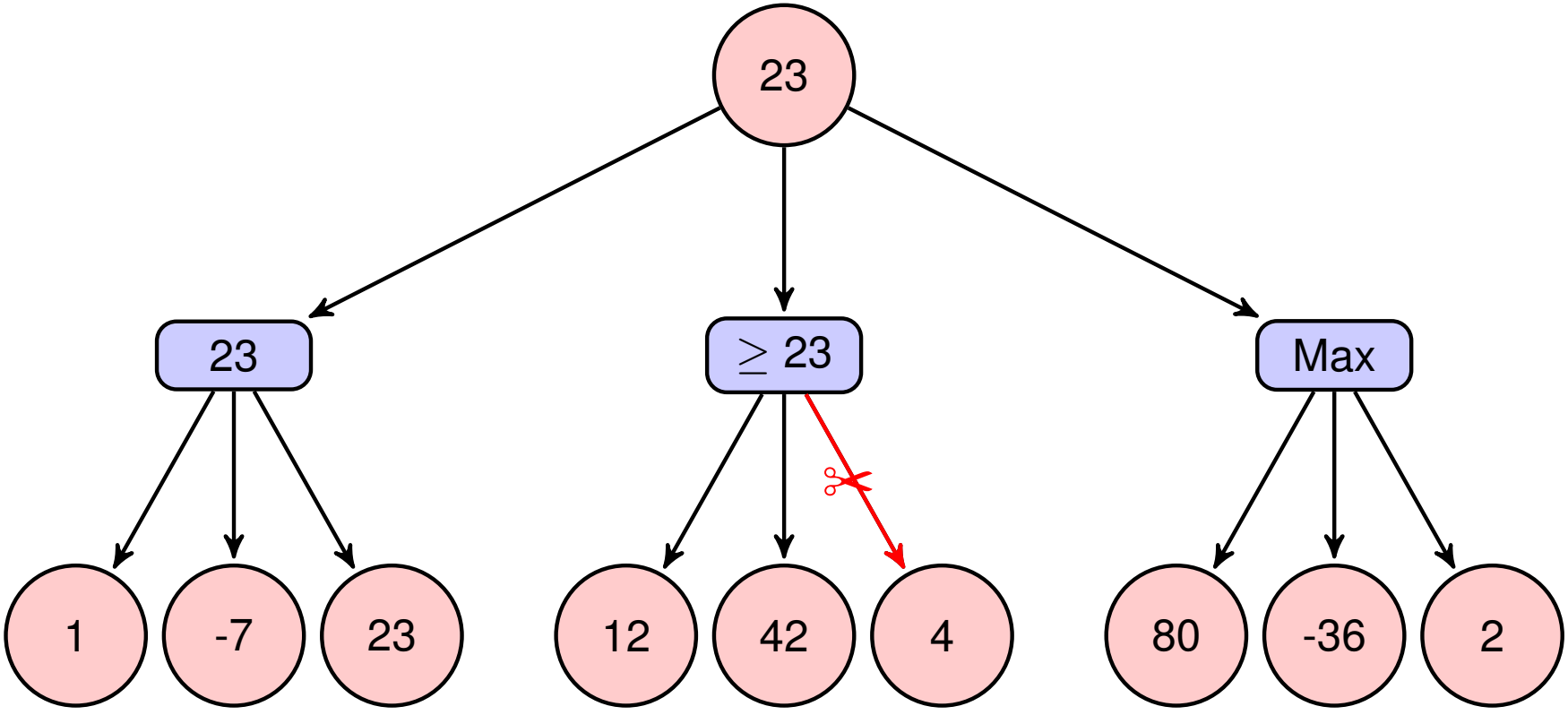
Exemple



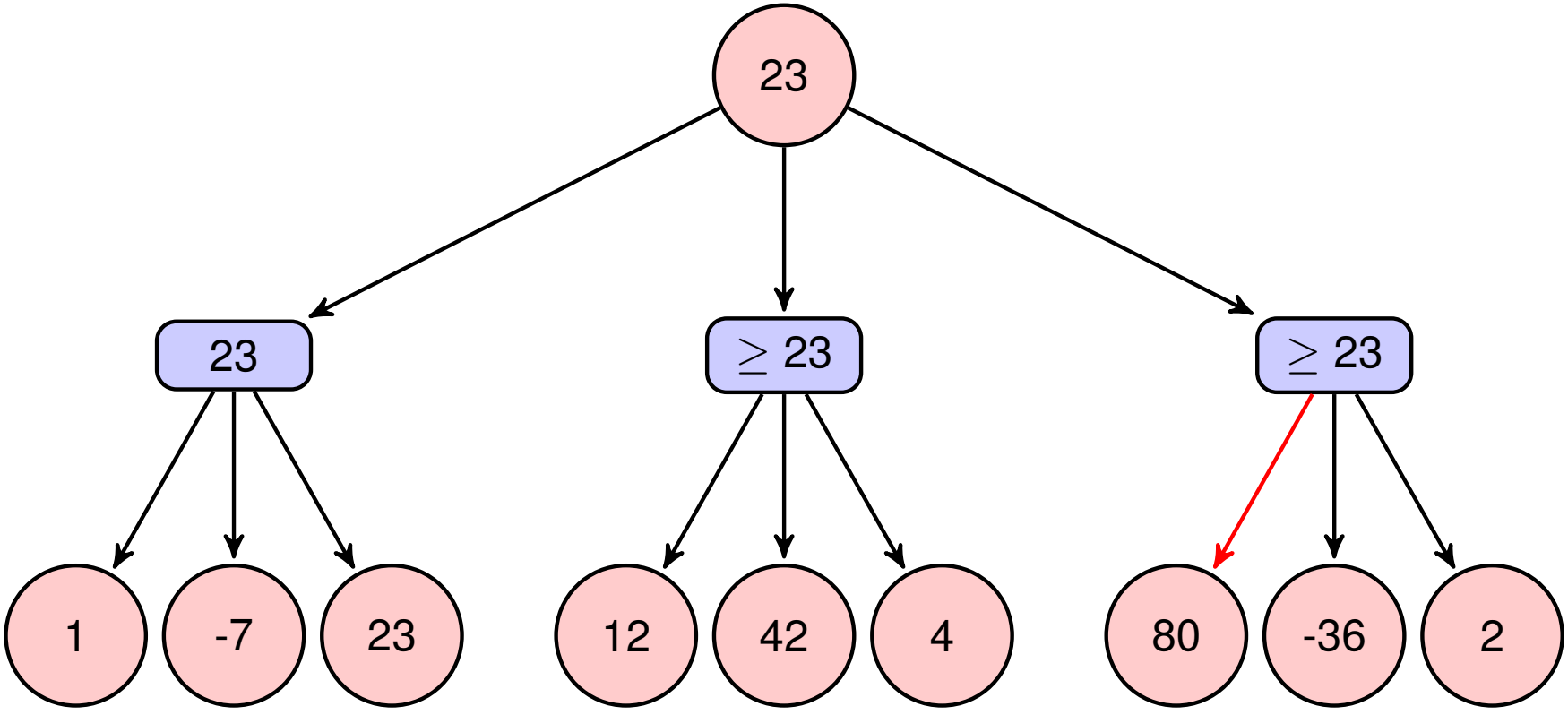
Exemple



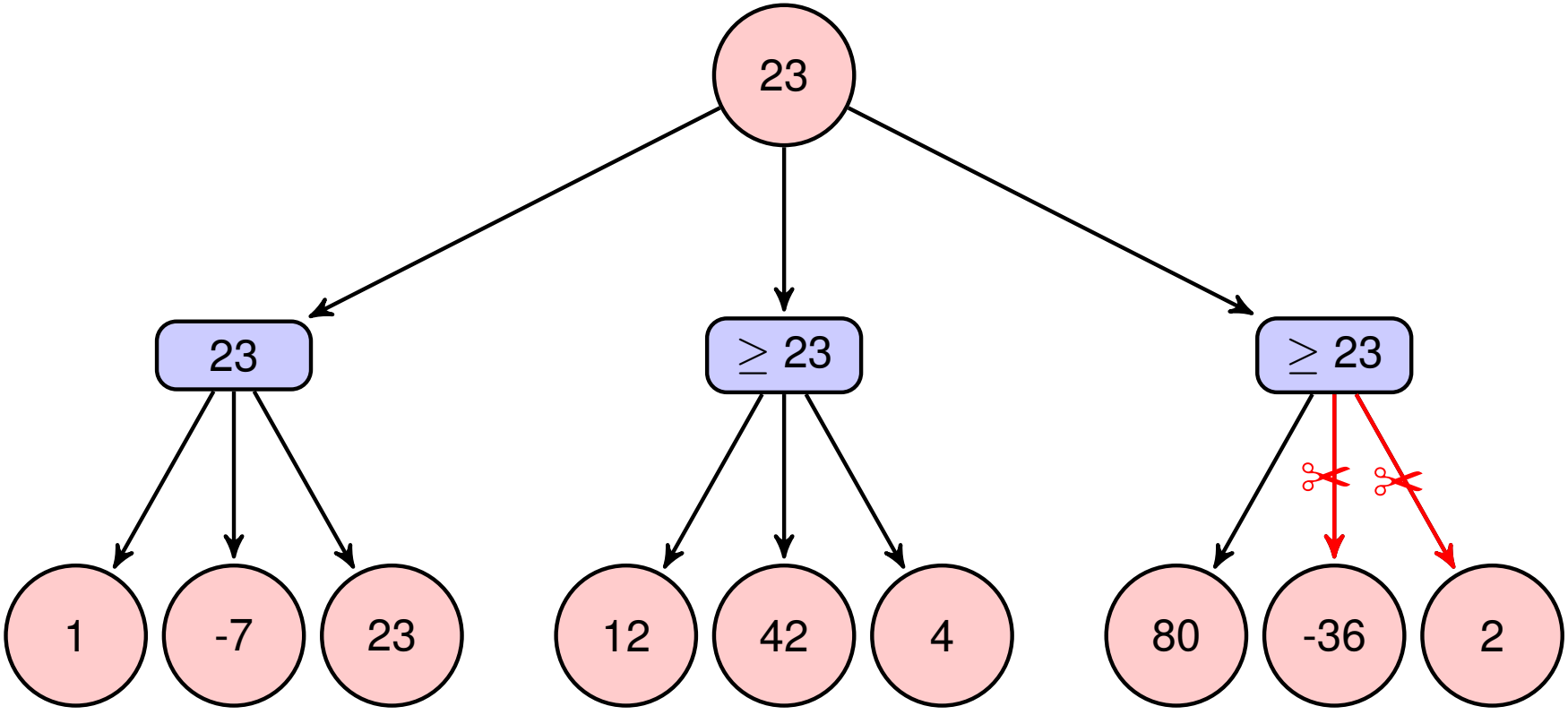
Exemple



Exemple



Exemple



Élagage par fenêtre alpha-beta

En appliquant la règle d'élagage

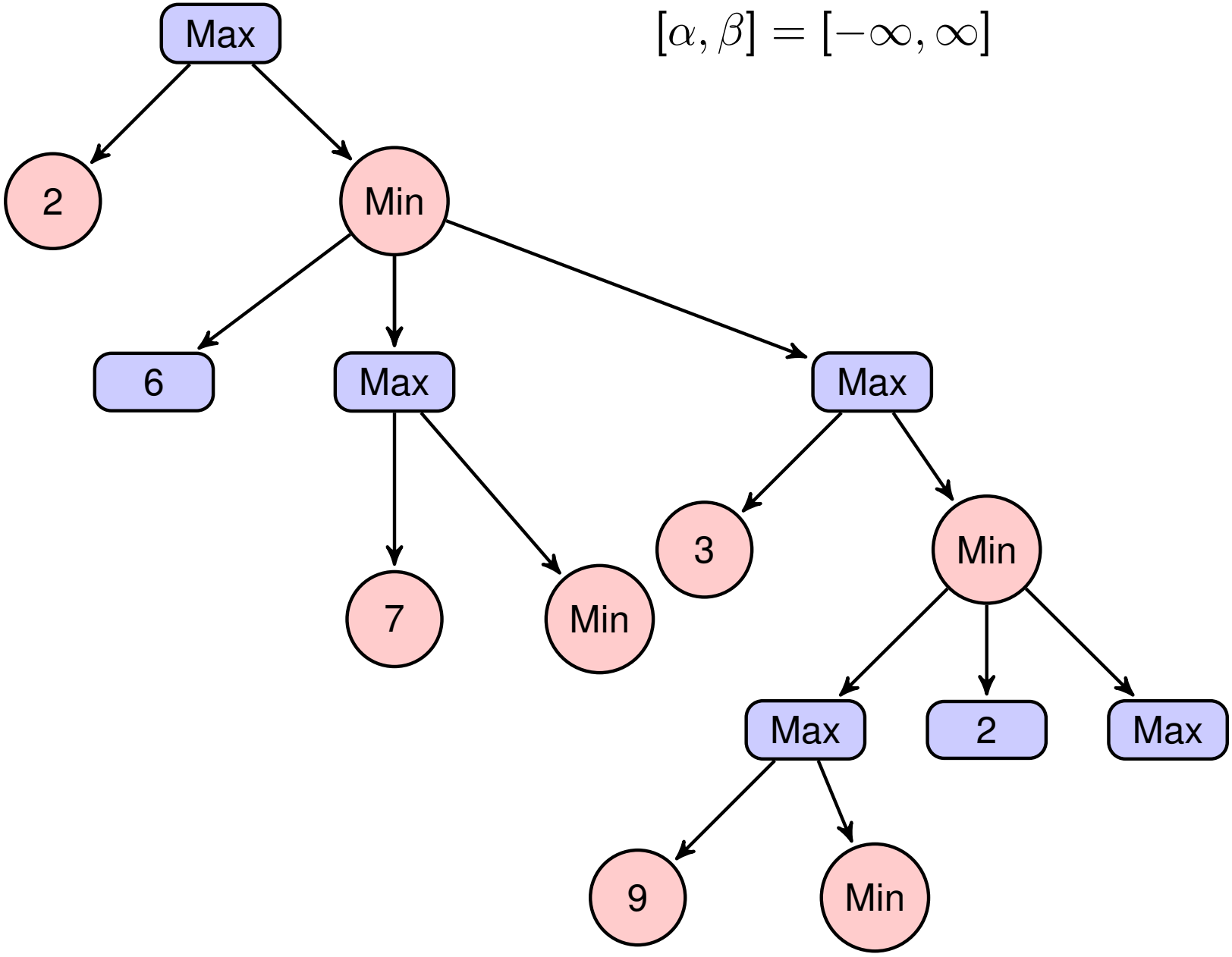
beta : par borne sup pour les noeuds MAX

alpha : par borne inf pour les noeuds MIN

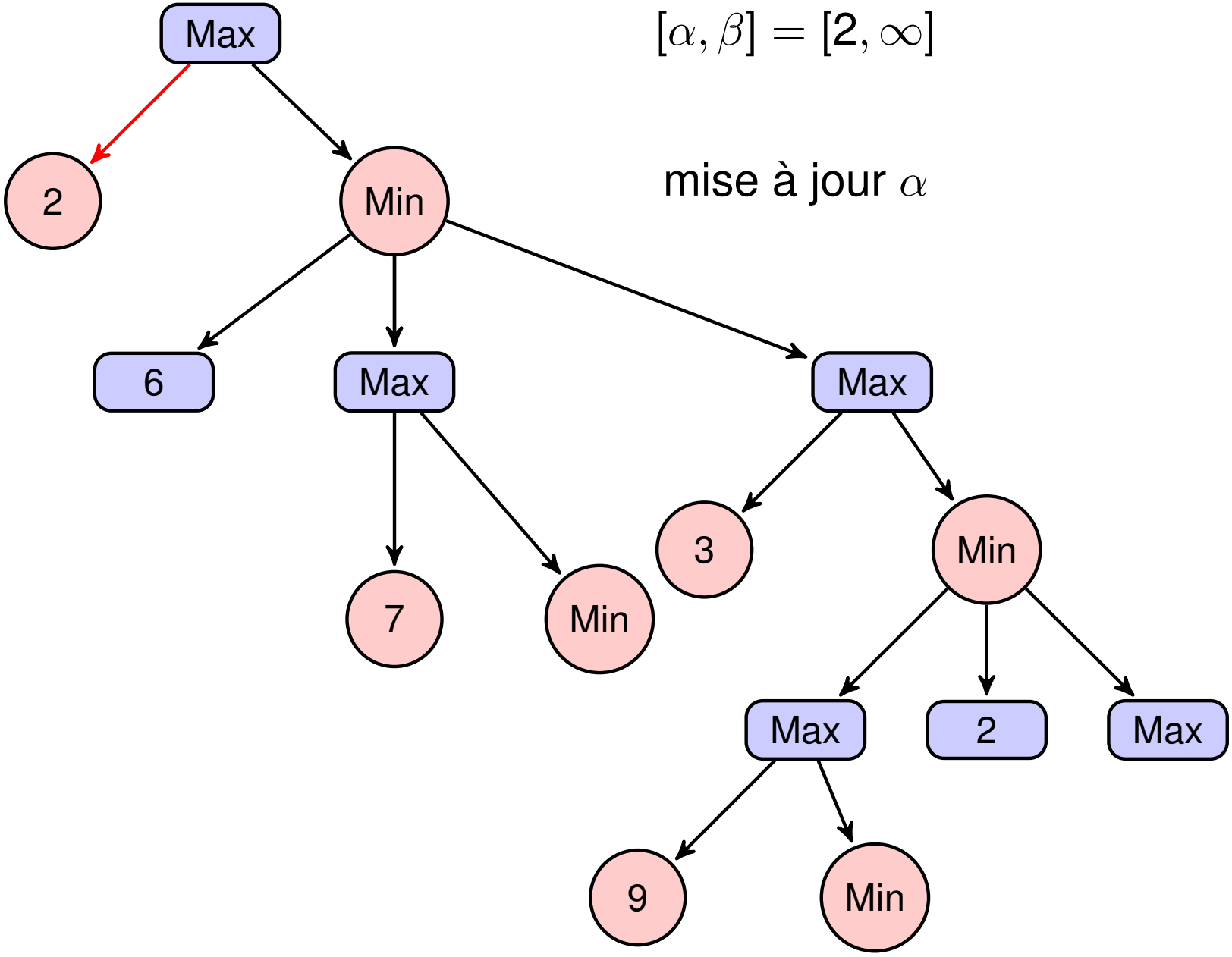
On maintient un intervalle $[\alpha, \beta]$ dans lequel on va limiter l'exploration :

- si la valeur d'un MIN est $\geq \beta \rightsquigarrow$ coupe des frêres
- si la valeur d'un MAX est $\leq \alpha \rightsquigarrow$ coupe des frêres

Exemple


Max
$[\alpha, \beta] = [-\infty, \infty]$
2
Min
6
Max
Max
3
Min
7
Min
Max
2
Max
9
Min

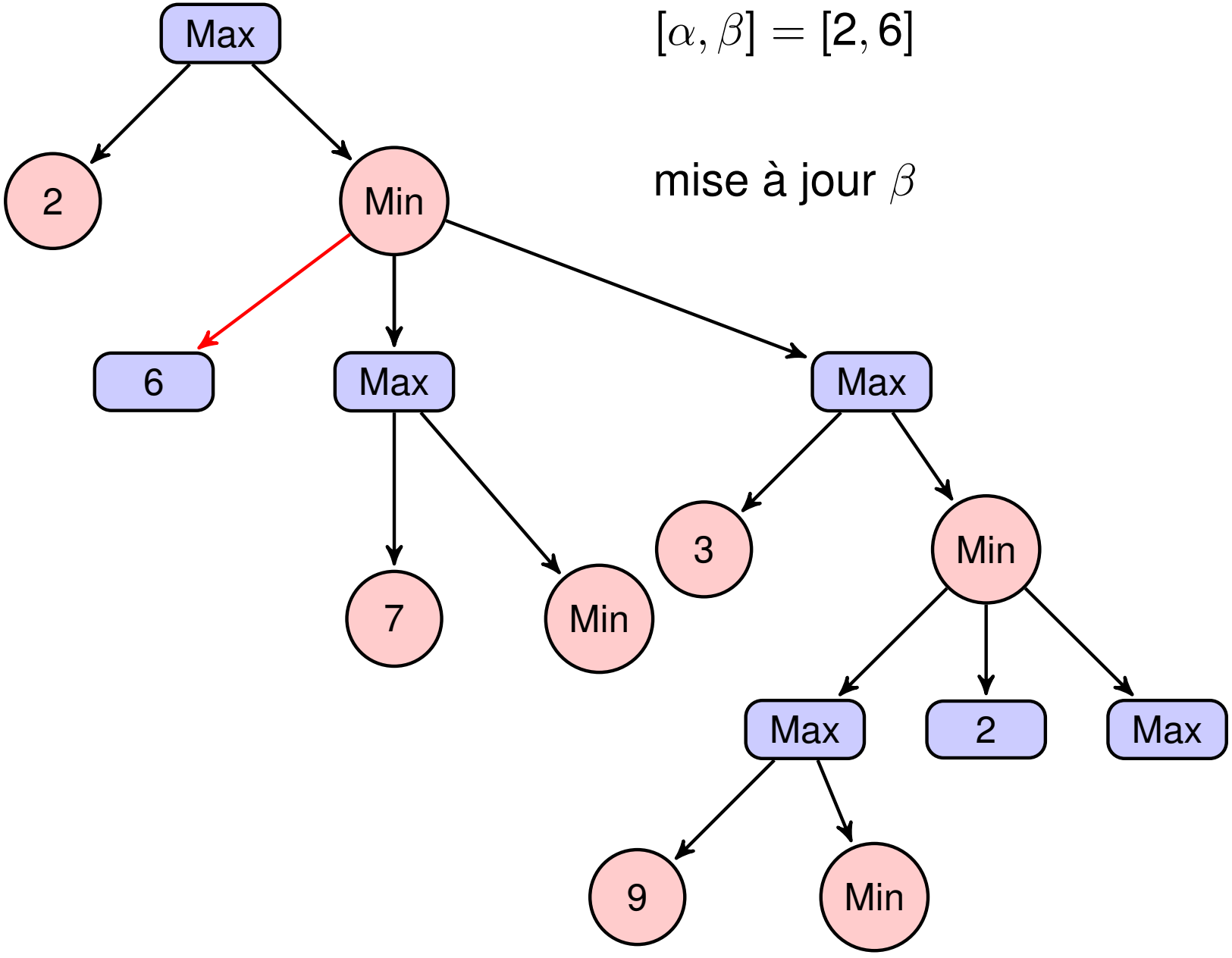
Exemple



$[\alpha, \beta] = [2, \infty]$

mise à jour α

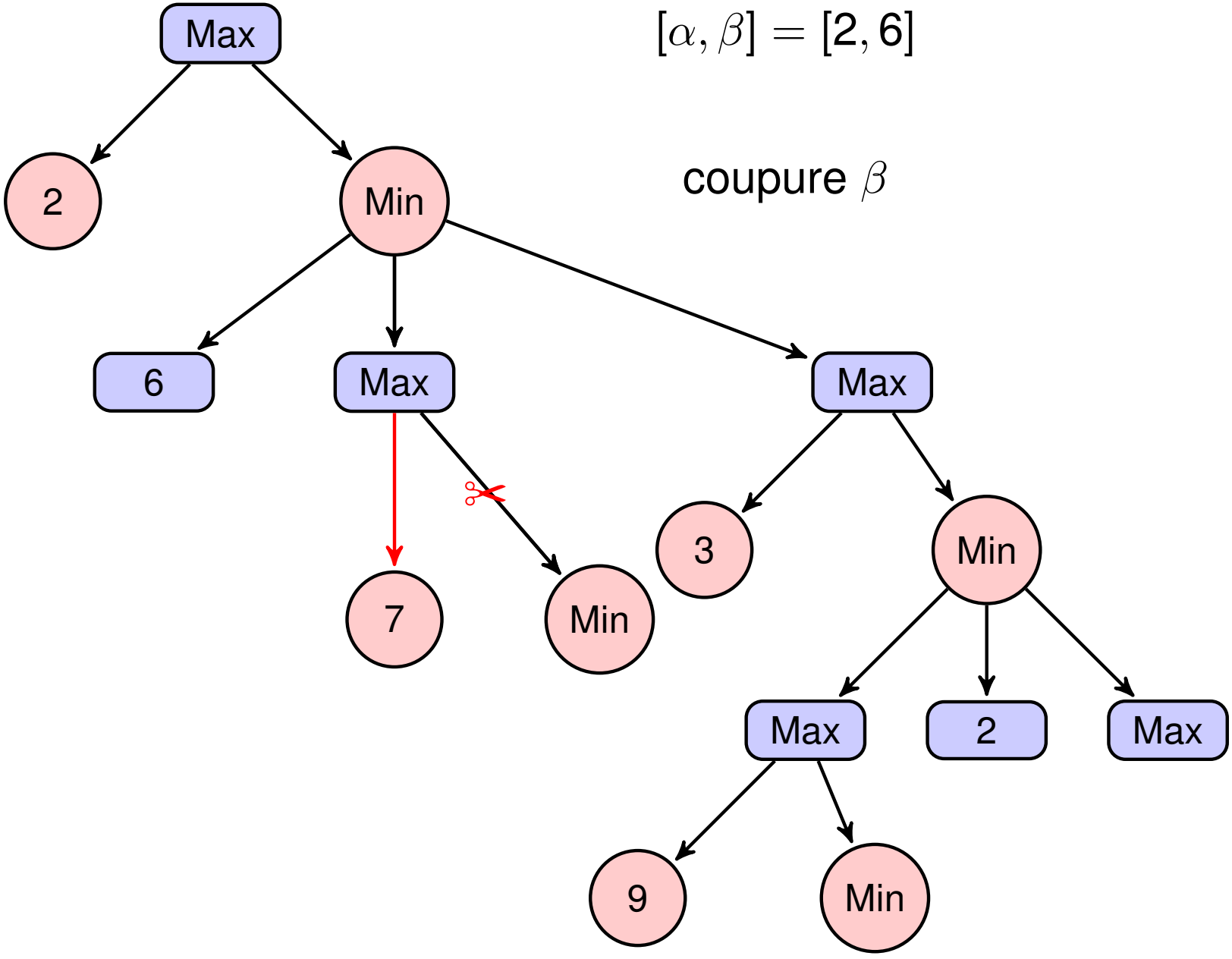
Exemple



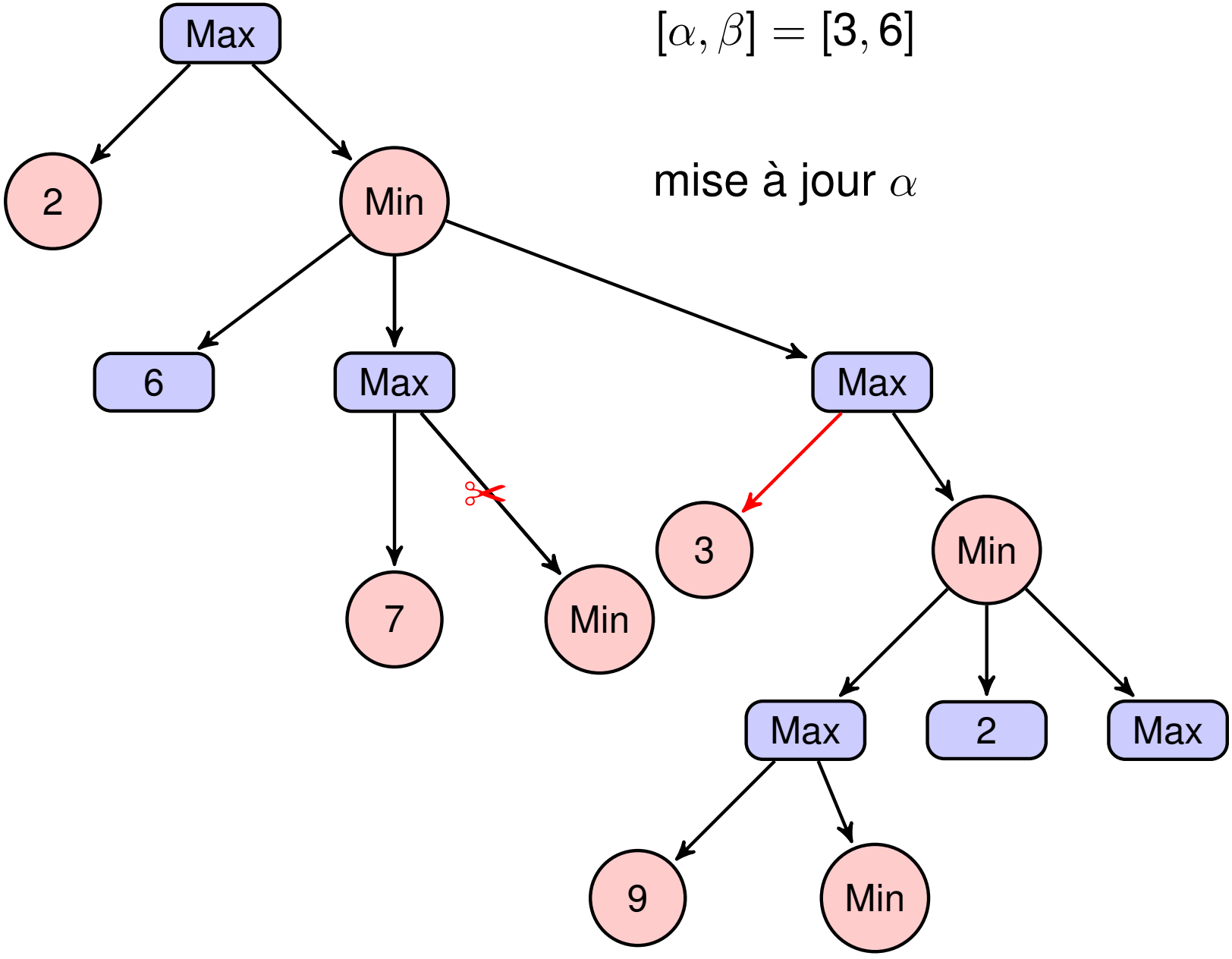
$[\alpha, \beta] = [2, 6]$

mise à jour β

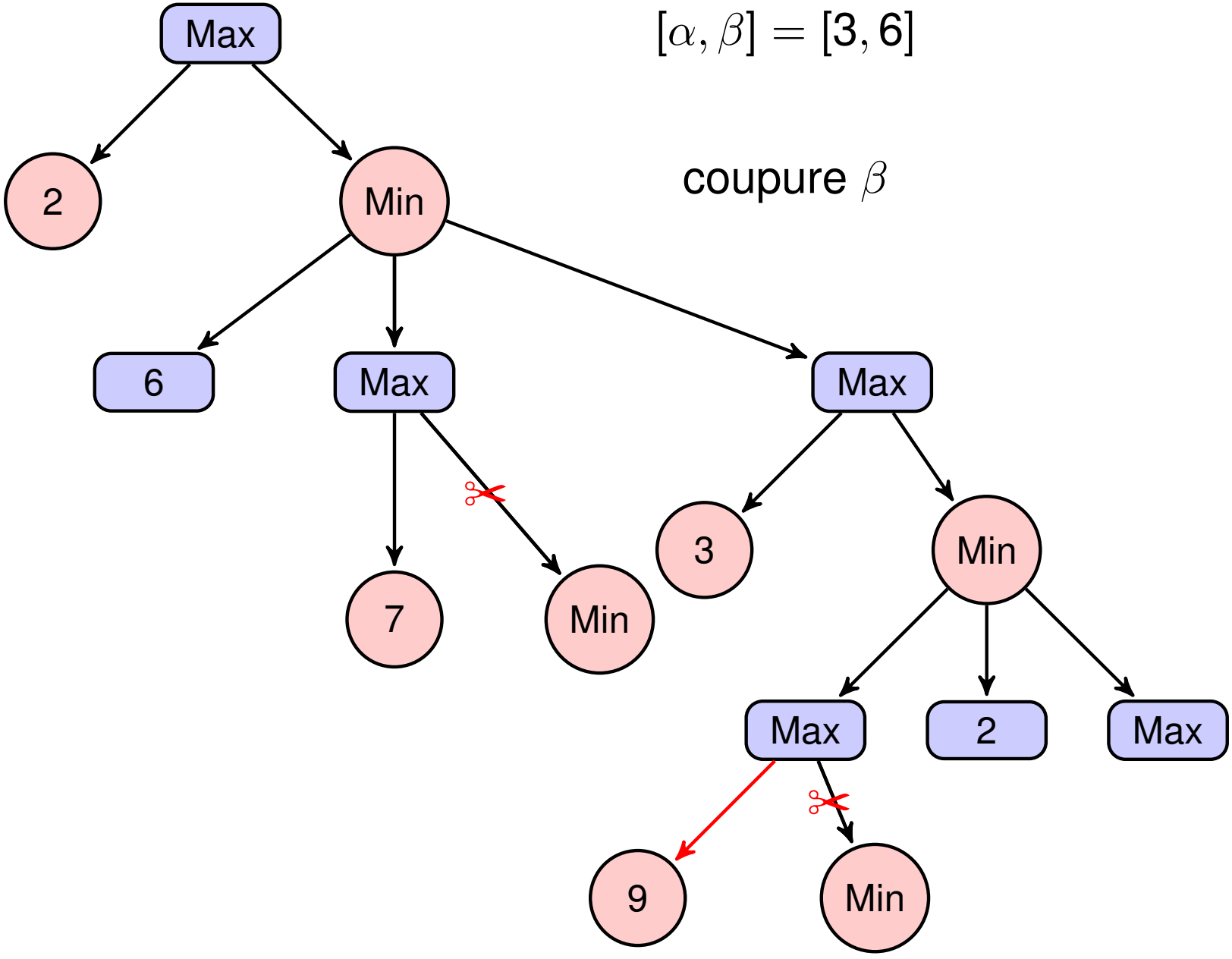
Exemple


Max
$[\alpha, \beta] = [2, 6]$
2
Min
coupure β
6
Max
Max
7
Min
3
Min
Max
2
Max
9
Min

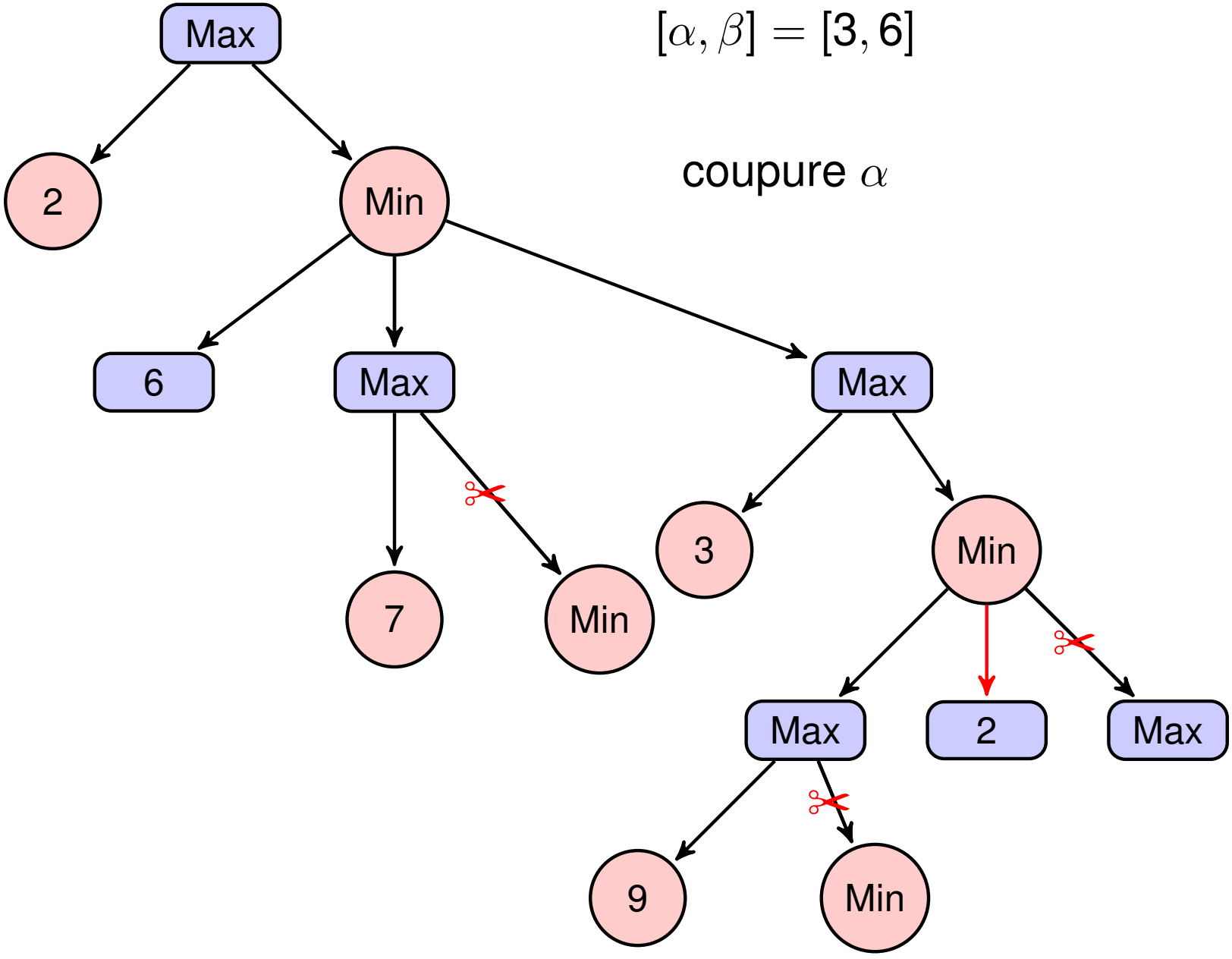
Exemple


Max
$[\alpha, \beta] = [3, 6]$
2
Min
mise à jour α
6
Max
Max
7
Min
3
Min
Max
2
Max
9
Min

Exemple


Max
$[\alpha, \beta] = [3, 6]$
2
Min
coupure β
6
Max
Max
7
Min
3
Min
Max
2
Max
9
Min

Exemple


Max
$[\alpha, \beta] = [3, 6]$
2
Min
coupure α
6
Max
Max
7
Min
3
Min
Max
2
Max
9
Min

Amélioration

Pour augmenter le nombre de coupes :

- il faut trouver le noeud hors intervalle le plus tôt possible
- Idéalement : trier les noeuds par score
  - décroissant pour les fils d'un MAX
  - croissant pour les fils d'un MIN

  $\rightsquigarrow$ mais le score n'est pas connu
- possibilité d'introduire une fonction d'estimation de score heuristique rapide pour trier les fils