V Recherche dans les arbres de jeux

Damien Olivier

Damien.Olivier@univ-lehavre.fr

Faculte des Sciences et Techniques du Havre

Plan

- 1. Quels jeux ?
- 2. Caractéristiques
- 3. Moteur de jeu
- 4. Exemple, Jeu de Grundy
- 5. MIN/MAX
- 6. α/β

V.1 Quels jeux ?

- Jeux faisant intervenir deux adversaires ;
- Information parfaite $\Rightarrow$ cela exclu les jeux dans lesquels le hasard intervient même partiellement ;
- Jeux dans lesquels il y a un gagnant, un perdant ou match nul ;

Tous les jeux ne sont donc pas adaptés à une étude par l'IA.

V.1 Quels jeux ?

	déterministe	avec hasard
information parfaite	échecs, dames go, othello . . .	backgammon monopoly
information imparfaite		bridge, poker, scrabble

V.2 Caractéristiques

- Présence d'un adversaire ;
  - Cela introduit de l'incertain ;
  - Le programme ne contrôle pas tous les mouvements ;
- Les programmes doivent faire face à l'imprévu ;
  - Les mouvements de l'adversaire ne sont pas toujours prévisibles ;
- Espace de recherche souvent très vaste
  - Il n'est pas possible d'explorer tout l'espace de recherche ;
  - Ex : les echecs, nombre total d'états 35^{100} ...
- A chaque pas il faut choisir le prochain coup ;
- Les coups sont évalués suivant leurs coûts.

Commentaire

- Nombre de choix par coup : 35 (facteur de branchement) ;
- Nombre moyen de coups par joueur : 50
- Nombre total d'états 35^{100}
- Nombre de nœuds de l'arbre : 10^{40}

De la recherche

- Jouer à un jeu c'est rechercher le mouvement qui permet de gagner ;
- La fonction heuristique doit permettre de savoir s'il y a gain ;
- Très voisin des systèmes de production, on a aux échecs par exemple :
  - Une BDG qui contient une représentation de toutes les pièces sur l'échiquier ;
  - Des règles de production qui modélisent les coups autorisés ;
  - L'application des règles à la BDG initiale et ses suivantes produit un graphe ou un arbre de jeu.

V.3 Moteur de jeu

- Générateur de mouvements
  - Génère les mouvements valides à partir de l'état courant ;
- Test de terminaison
  - Décide si l'état courant est un état fi nal (victoire, défaite, nul) ou un état intermédiaire ;
- Fonction d'évaluation statique
  - Evaluation d'un état indépendamment des états futurs ou passés ;
- Stratégie de contrôle
  - Permet d'effectuer un choix.

V.4 Exemple : Jeu de Grundy

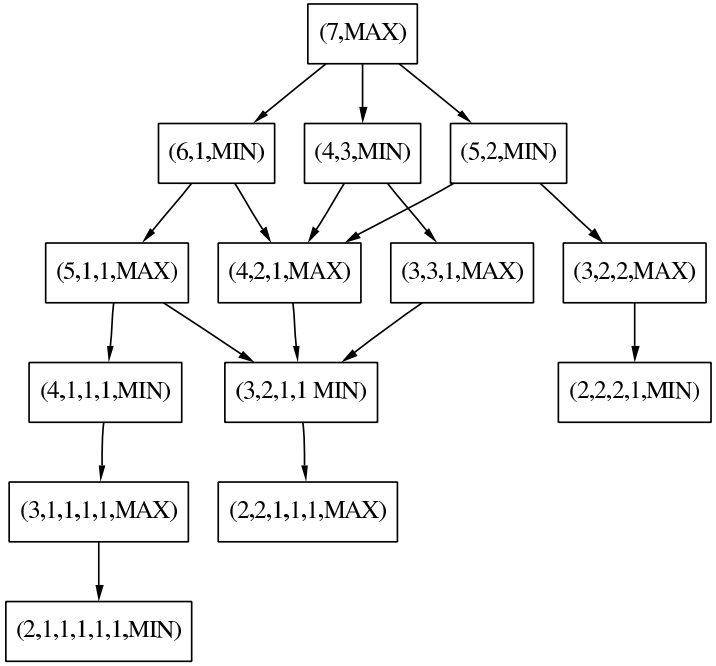
- Deux joueurs ;
- Un tas unique de pièce en Euro ;
- Le 1^{er} joueur divise le tas original en 2 tas inégaux ;
- Chaque joueur fait la même chose avec un seul tas ;
- Le jeu continue jusqu'à ce que chaque tas ne posséde qu'une ou deux pièces ;
- Le perdant est celui qui ne peut plus jouer.

V.4 Exemple : Jeu de Grundy

Une partie ?

- 2 joueurs MIN et MAX ;
- 7 pièces ;
- MAX joue le premier ;
- BDG
  - Séquence non ordonnée qui représente le nombre de pièces dans les tas ;
  - Paramètre identifi ant le joueur qui doit jouer.

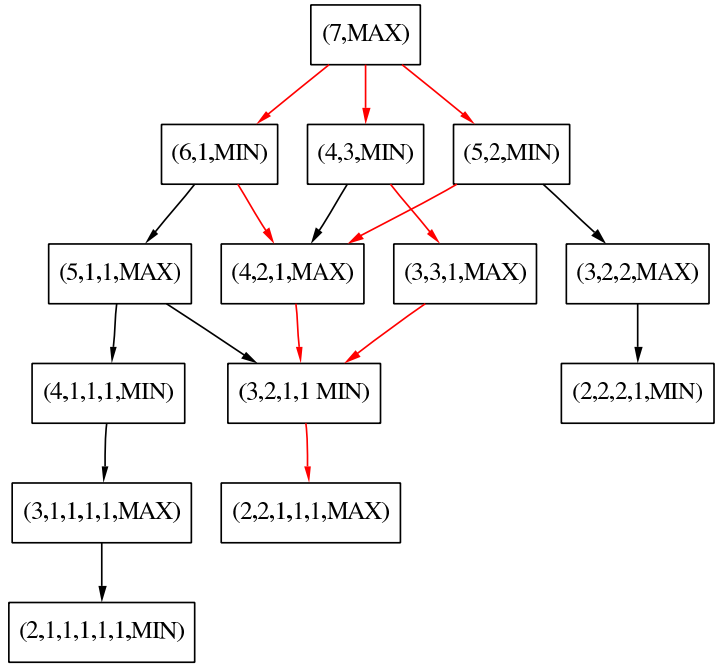
V.4 Exemple : Jeu de Grundy



- Toutes les feuilles représentent des situations perdantes ;
- Ce graphe démontre que peu importe ce que fait MAX, MIN peut toujours gagner ;

V.4 Exemple : Jeu de Grundy

Stratégie gagnante pour MIN = graphe ET/OU



V.5 Algorithme MIN/MAX

Convention :

- MAX joue en premier ;
- Les coups alternent ;
- Les nœuds situés à une profondeur paire = nœuds où c'est à MAX de jouer $\Rightarrow$ Nœuds MAX ;
- Nœuds MIN ;
- Premier sommet du graphe = profondeur 0.

V.5 Algorithme MIN/MAX

On peut utiliser des méthodes de recherche en largeur d'abord, en profondeur d'abord ou heuristiques, mais les conditions d'arrêts doivent être modifi ées :

- Limite de temps de calcul ;
- Limite d'espace mémoire ;
- Limite de profondeur ;
- . . .

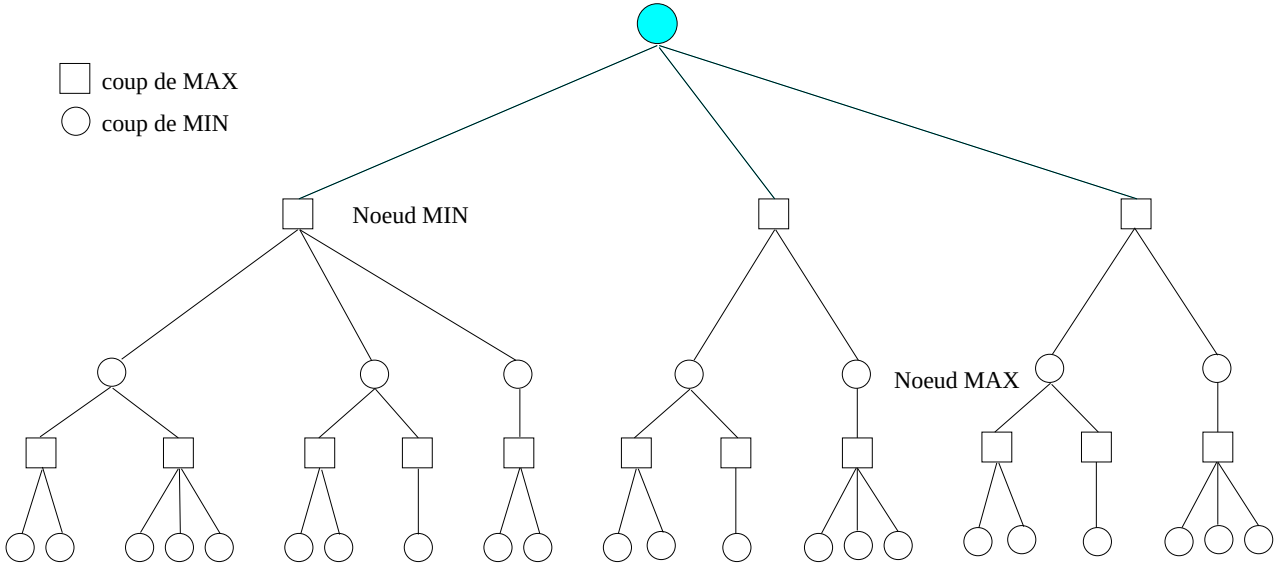
V.5 Algorithme MIN/MAX

Fonction d'évaluation Une fois la recherche terminée, il faut estimer le meilleur coup de départ

- Fonction d'évaluation statique $f(n)$
  - S'applique aux feuilles du graphe ;
  - Détermine la qualité d'une position associée à une feuille ;
  - $f(n) > 0$ position favorable à MAX ;
  - $f(n) < 0$ position favorable à MIN.

V.5 Algorithme MIN/MAX

Arbre de jeu



On choisit le noeud ayant la plus grande valeur

Pour chacun de ces noeuds
on donne le min des e des fils = E

Pour chacun de ces noeuds
on donne le max des E des fils = e

Pour chacun de ces noeuds
on donne le min des e des fils = E

Pour chacun de ces noeuds
on calcule e

Application au Tic-Tac-Toe

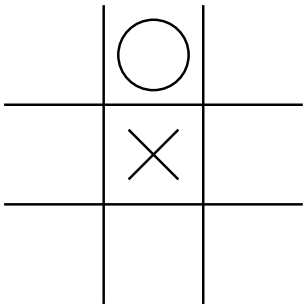
- MAX marque des X ;
- MIN marque des O ;
- On effectue une recherche en largeur d'abord jusqu'au niveau 2 ;
- $e(p)$ fonction d'évaluation statique ;

$$e(p) = nb(R, C, D, MAX) - nb(R, C, D, MIN)$$

- $e(p) = \infty$ Si p est une victoire pour MAX

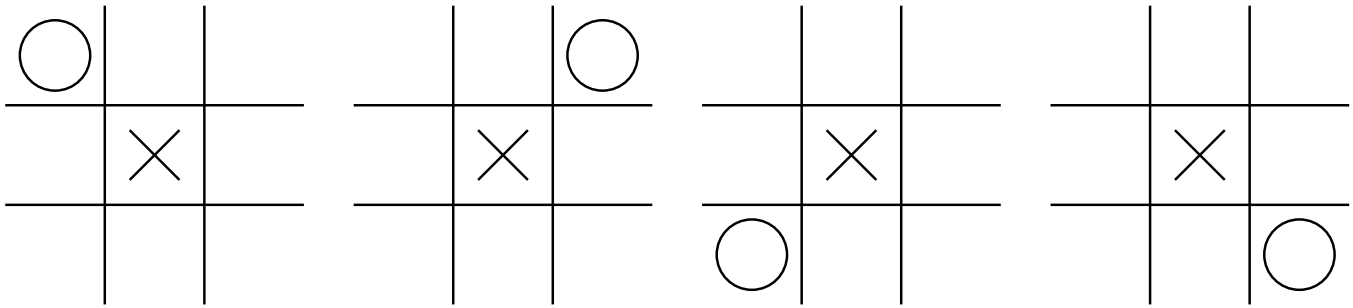
  $e(p) = -\infty$ Si p est une victoire pour MIN

Application au Tic-Tac-Toe

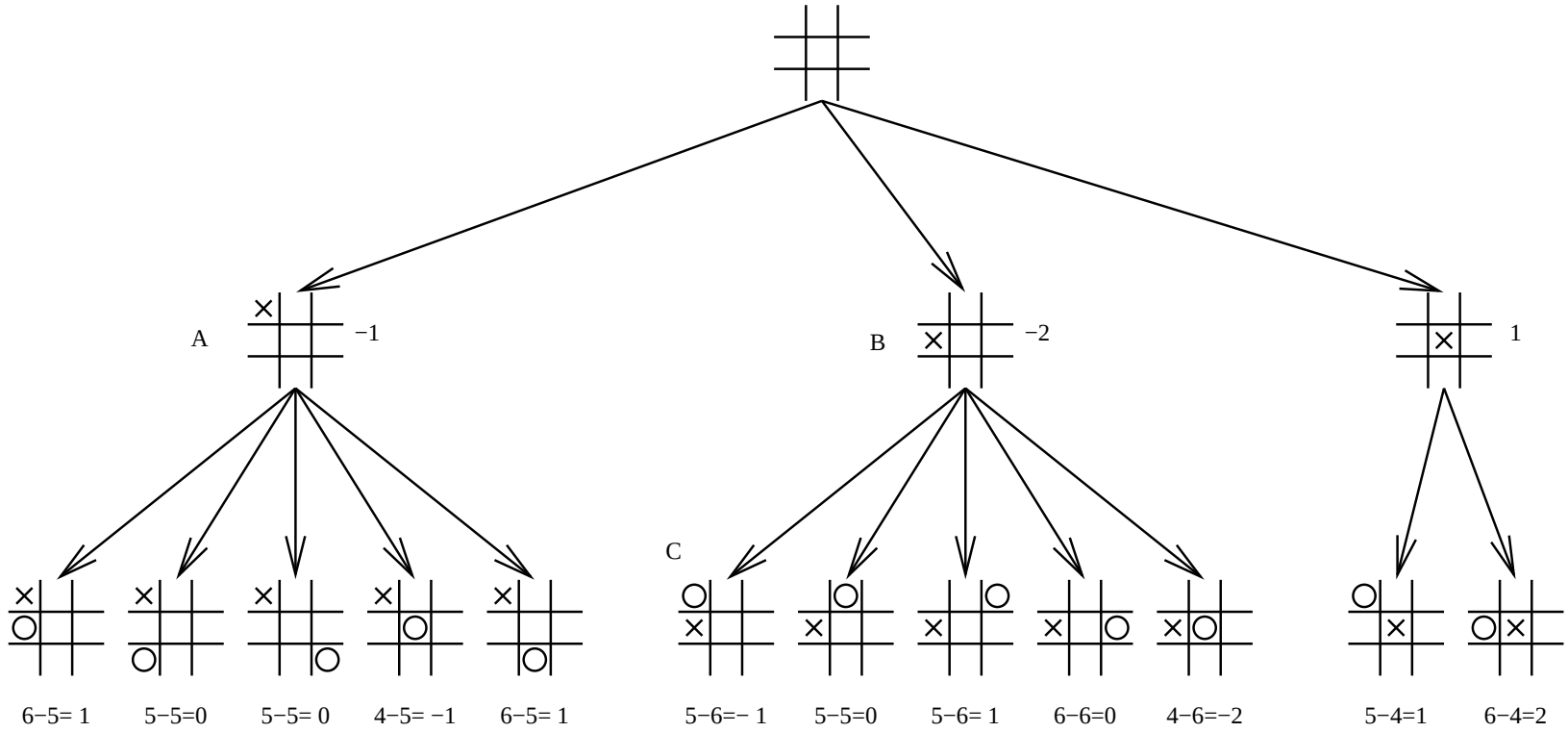


$$e(p) = 6 - 4 = 2$$

Utilisation des symétries



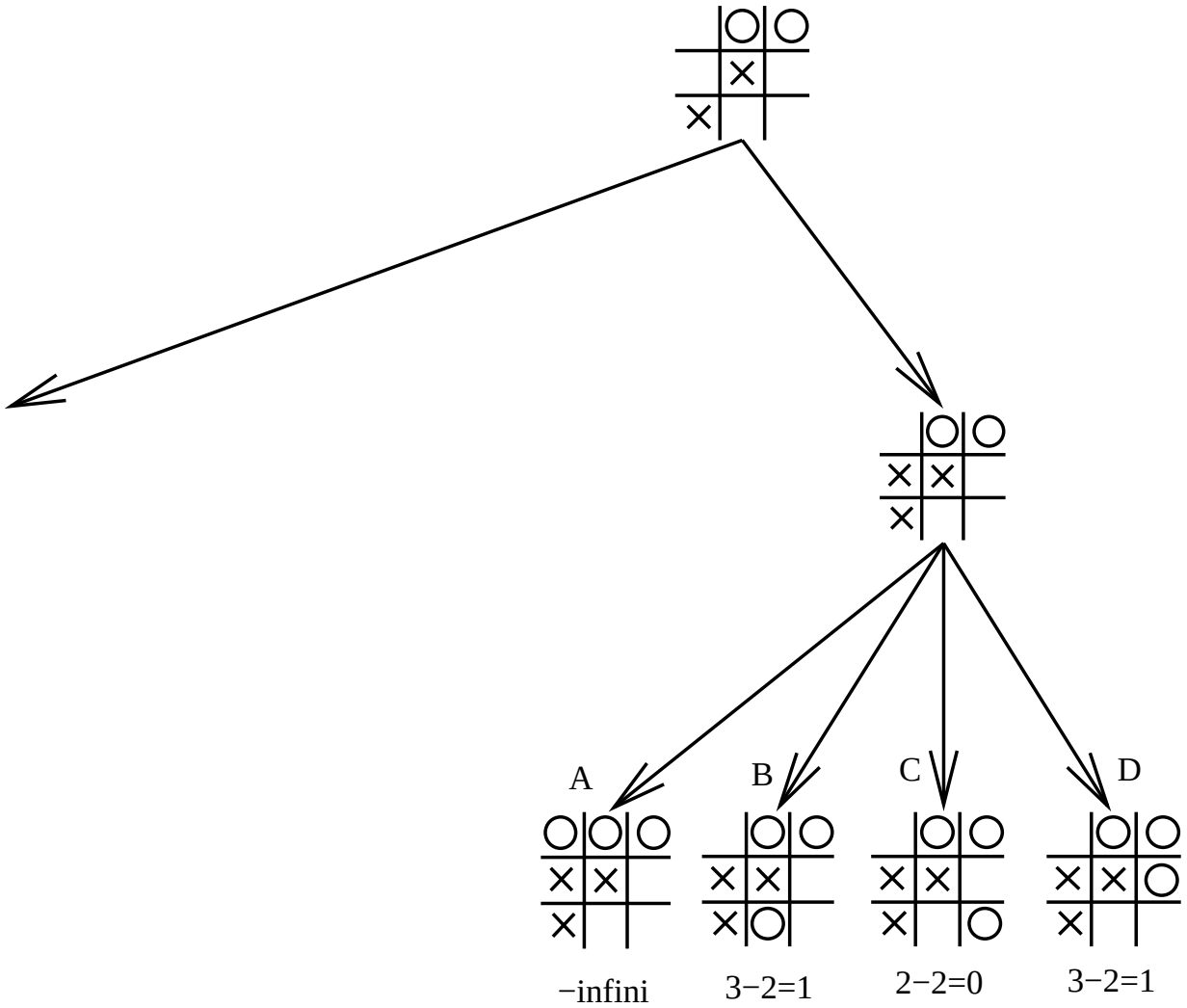
Arbre de jeu



V.6 La coupure Alpha-Béta

- MIN/MAX Evaluation des positions après génération de l'arbre
- Idée : Evaluation aux feuilles et propagation aux ancètres lorsque l'arbre est généré ;
- $\Rightarrow$ Une feuille est évaluée dès qu'elle est produite.

V.6 La coupure Alpha-Béta

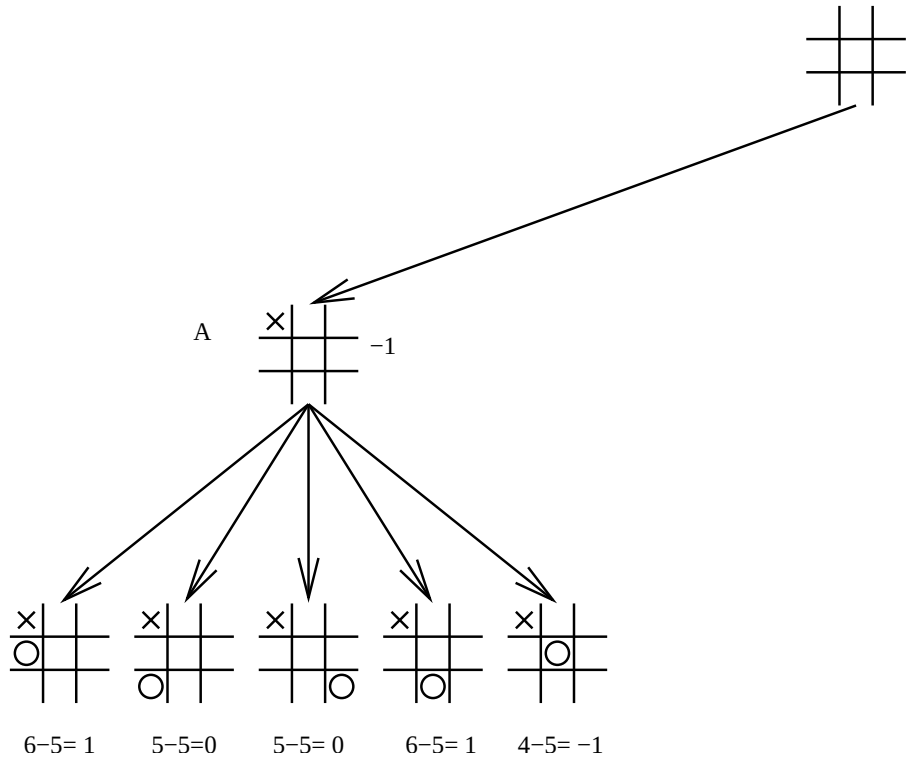


V.6 La coupure Alpha-Béta

- On attribue au nœud père $-\infty$;
- Il n'est pas nécessaire de produire B, C, D ;
- Le nombre d'états non développés peut-être plus important encore si on effectue une recherche en profondeur.

V.6 La coupure Alpha-Béta

Le nœud A et tous ses successeurs ont été engendrés, mais le nœud B ne l'a pas encore été. Les valeurs sont propagées.



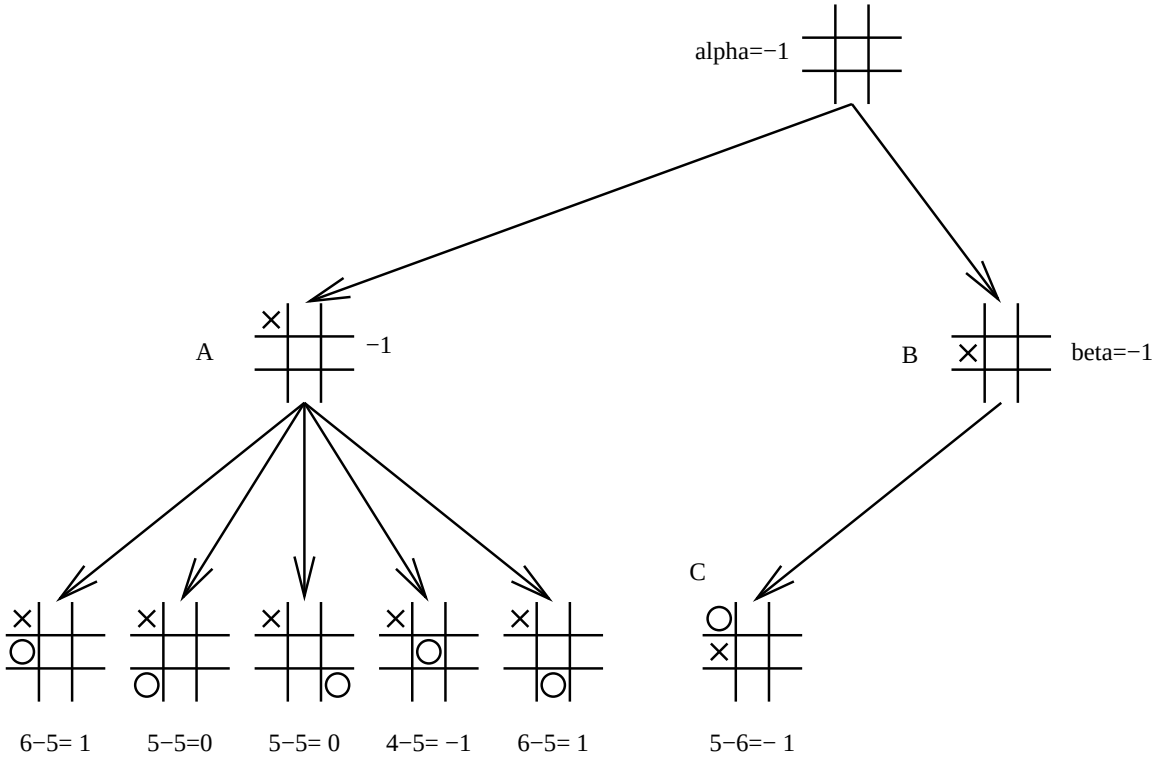
V.6 La coupure Alpha-Béta

- La valeur propagée au nœud de départ est minorée par -1 ;
- La valeur de ce nœud de départ peut être supérieure à -1 mais jamais inférieure ;
- Ce minorant c'est la valeur alpha du nœud de départ.

V.6 La coupure Alpha-Béta

On continue la recherche . . .

- B et son premier successeur C sont engendrés ;
- C = -1 ;



V.6 La coupure Alpha-Béta

- La valeur fi nale du nœud B peut être inférieure à -1 mais jamais supérieure ;
- -1 est un majorant, c'est la valeur beta ;
- Au niveau de l'exemple, la valeur β du nœud B ne dépassera la valeur α du nœud de départ $\Rightarrow$ recherche sur B interrompue.

V.6 La coupure Alpha-Béta

Règles

- Les valeurs α des nœuds MAX ne peuvent jamais diminuer (y compris le nœud de départ) ;
- Les valeurs β des nœuds MIN ne peuvent jamais augmenter.

V.6 La coupure Alpha-Béta

Règles

La recherche peut être interrompue au-dessous de tout nœud

Coupure α - MIN ayant une valeur β inférieure ou égale à la valeur de ses ancètres MAX. La valeur propagée vers le haut pour ce nœud MIN est sa valeur β.

Coupure β - MAX ayant une valeur α supérieure ou égale à la valeur de ses ancètres MIN. La valeur propagée vers le haut pour ce nœud MIN est sa valeur α.

Calcul des valeurs α et β

- La valeur α d'un nœud MAX prend la valeur fi nale la plus élévée propagée vers le haut à partir de ses successeurs au moment du calcul ;
- La valeur β d'un nœud MIN prend la valeur fi nale la plus faible propagée vers le haut à partir de ses successeurs au moment du calcul ;

Commentaire

Pour effectuer des coupes α/β il est indispensable de construire une partie de l'arbre de recherche à sa profondeur maximale. C'est pôurquoi on utilise une recherche en profondeur d'abord. Le nombre de feuilles de profondeur D qui sont engendrés par une recherche α/β est à peu près le même que le nombre de feuilles engendrées à une profondeur D/2 sans α/β.