

Aide à la Décision

Jeux (à deux joueurs)

Révision 1

© Hervé Barbot, 2005-2011

- Pourquoi ?
 - Tâches bien structurées
 - Abstraction
 - Mesure de performance
 - Pas toujours besoin de grandes quantités d'informations
 - Défi intellectuel
- Limites de l'étude
 - Jeux à 2 joueurs

(C) Hervé Barbot, 2005-2011

2

Abstraction

- Espace d'états
 - Les situations de jeu
 - Un état initial :
 - Le début de partie
 - La position courante
 - Fonction « successeur »
 - Les règles du jeu
 - Connaissance parfaite (pas d'incertitude sur le résultat des actions)
 - Test solution
 - Fin de partie : gagné, perdu, nul
 - Fonction de score

(C) Hervé Barbot, 2005-2011

3

Mais, par rapport aux
« recherches off-line dans un espace d'états »...

- Adversaire
 - Incertitude dans l'expansion d'un « arbre de recherche »
 - L'ordinateur ne contrôle pas tout
- Nécessité de faire face à l'imprévu
- Complexité : les solutions exhaustives sont vite trop complexes pour être envisagées
 - Ex. facteur de branchement pour un jeu d'échec de l'ordre de 35...
- Malgré l'incertitude, on cherche pas à pas à choisir le prochain « meilleur » coup à jouer

(C) Hervé Barbot, 2005-2011

4

Quoi qu'il en soit ...

▪ Jeu de plateau

- État initial
- Transitions (règles de jeu, déplacement des pièces)
 - Donc une certaine « exploration » est possible
- État gagnant / perdant / nul
 - Test solution
- Possibilité de savoir si une situation est « meilleure » ou non qu'une autre
 - « évaluation » / « heuristique »
 - Mais dépend de nombreux paramètres, et notamment du jeu de l'adversaire !

(C) Hervé Barbot, 2005-2011

5

▪ Adversaire imprévisible

donc prendre en compte toutes les situations...

▪ Mais, par exemple aux échecs :

- Facteur de branchement = 35
- Nombre moyen de coups par joueur = 50
- Nombre total d'états = 35^{100}
- Nombre d'états dans l'espace d'analyse = 10^{120}

▪ Une exploration off-line est-elle envisageable ?

(C) Hervé Barbot, 2005-2011

6

Composants de base

▪ Générateur de mouvements

- La fonction « successeur »

▪ Le test de terminaison

- Décision sur l'état courant
 - Gain, perte, nul, rien

▪ Fonction d'évaluation (gain / utilité)

- Qualité d'un état donné indépendamment des coups passés ou futurs
- Évaluation intrinsèque d'une situation de jeu

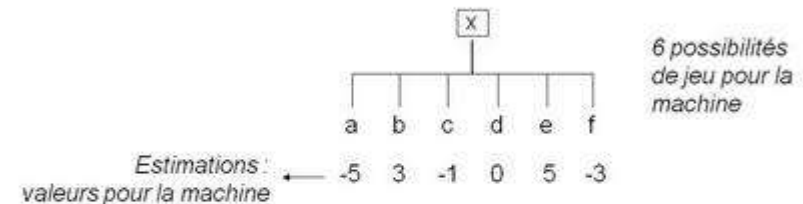
▪ Stratégie de contrôle

- Choix entre différentes options de jeu
 - Stratégie de jeu la plus prometteuse dans l'objectif de vaincre

(C) Hervé Barbot, 2005-2011

7

Approche intuitive : analyse à 1 niveau



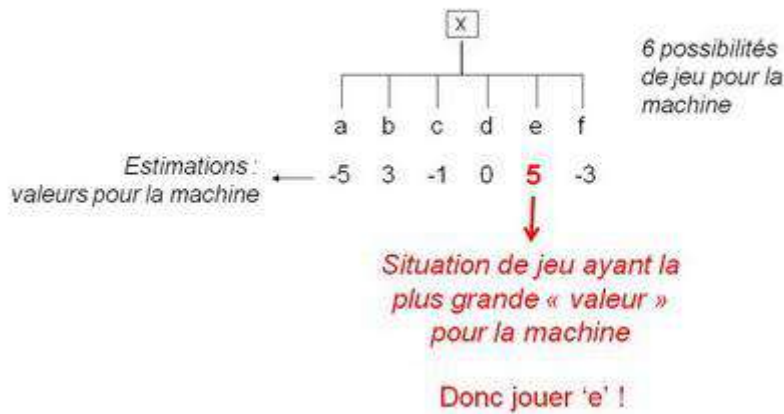
→ Quelle branche la machine doit-elle jouer ?

position laissée à la machine qui doit jouer

(C) Hervé Barbot, 2005-2011

9

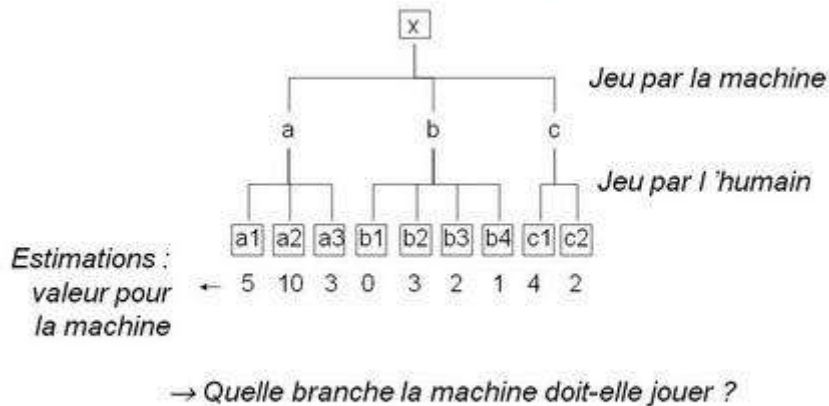
Approche intuitive : analyse à 1 niveau



(C) Hervé Barbot, 2005-2011

10

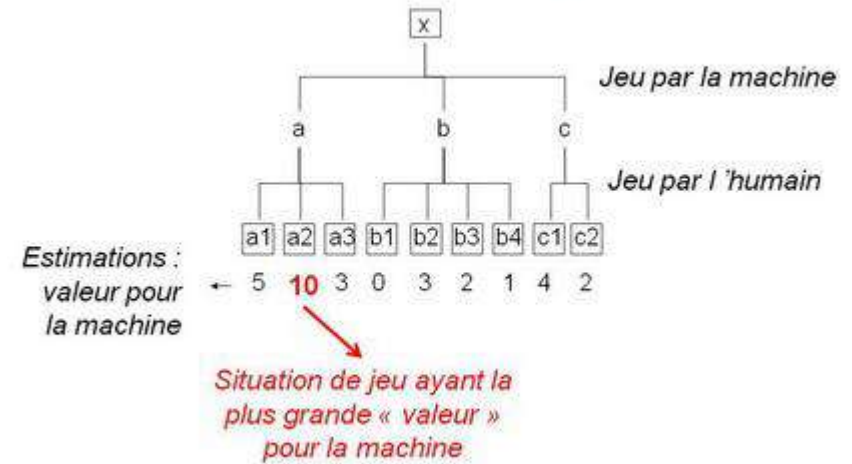
Analyse à 2 niveaux



(C) Hervé Barbot, 2005-2010

11

Analyse à 2 niveaux

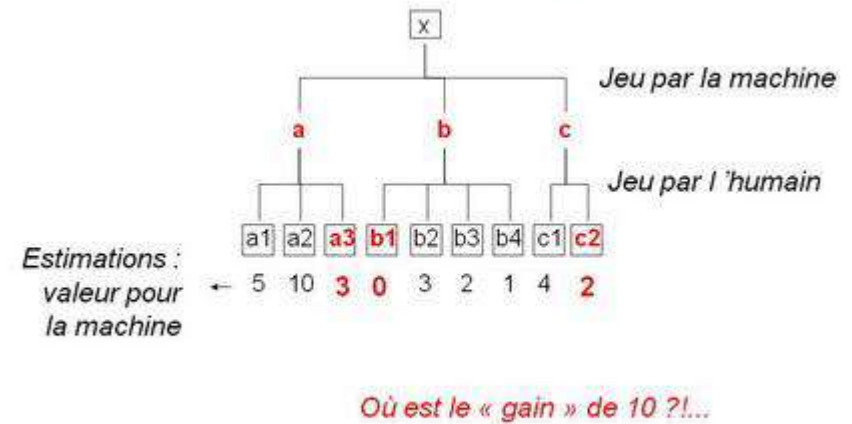


Mais...

(C) Hervé Barbot, 2005-2010

12

Analyse à 2 niveaux



(C) Hervé Barbot, 2005-2010

15

Fonction d'évaluation

- $e(s) = +\infty$ si gain pour « Max »
 $e(s) = -\infty$ si gain pour « Min »
 $e(s) = 0$ si égalité
- Exemple tic-tac-toe :
 - $e(s) =$
nombre de ligne / colonnes / diagonales 'ouvertes'
pour « Max »
-
nombre de lignes / colonnes / diagonales 'ouvertes'
pour « Min »

(C) Hervé Barbot, 2005-2010

16

- Principe (évident) de base :
ce qui est bon pour « Max » ne l'est pas pour
« Min »
et vice-versa !
- La fonction d'évaluation doit être
 - statique,
 - capable de mesurer la « qualité » de n'importe
quelle situation de jeu par rapport à un joueur,
 - sans nécessiter une exploration complète de
l'arbre jusqu'à une situation « terminale » (jeu
terminé).

(C) Hervé Barbot, 2005-2011

17

Hypothèses

- Les objectifs des deux joueurs sont
contradictoires
- La fonction d'évaluation de la qualité d'une
situation symétrique par rapport à 0

(C) Hervé Barbot, 2005-2011

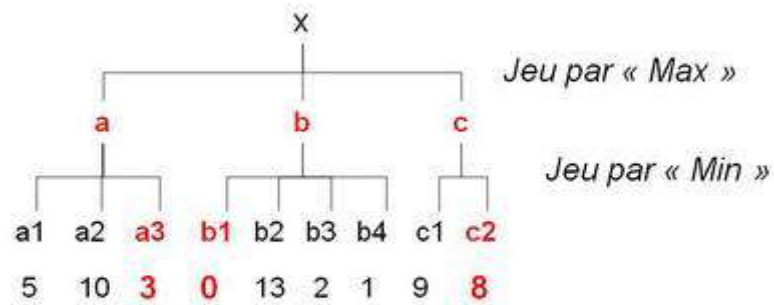
18

MinMax

- Hypothèse pour « Max » :
« Min » joue aussi bien que lui !
 - Et donc toute position que « Max » juge bonne pour
lui sera jugée par « Min » comme « mauvaise »
 - Et vice-versa !... Mais ça je l'ai déjà dit.
- Principe de base :
Supposons un arbre d'exploration fini,
les valeurs sont « remontées » du bas (des
feuilles) vers le haut (les successeurs directs de
la position courante) pour permettre à « Max »
de faire le meilleur choix

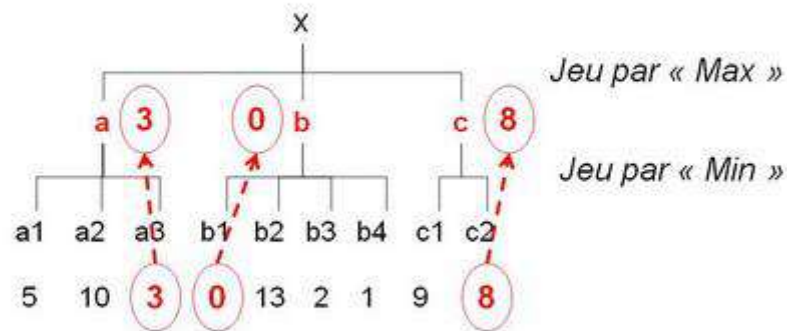
(C) Hervé Barbot, 2005-2011

19



(C) Hervé Barbot, 2005-2010

20



(C) Hervé Barbot, 2005-2010

22

Généralisation de l'algorithme MinMax

- Nœuds « Max » et nœuds « Min »
- Remonter la valeur minimale des successeurs d'un nœud « Min »
- Remonter la valeur maximale des successeurs d'un nœud « Max »

(C) Hervé Barbot, 2005-2010

23

```
function MinMax ( nœud )
```

```
Le nœud est à la profondeur maximale d'analyse
  => return e ( nœud )
```

```
Le nœud est « Max »
  => return MAX [ MinMax(x) ]
           x ∈ Γ(nœud)
```

```
Le nœud est « Min »
  => return MIN [ MinMax(x) ]
           x ∈ Γ(nœud)
```

(C) Hervé Barbot, 2005-2011

24

- Questions évidentes :

- Qu'est-ce que la « profondeur » ?
- Les branches d'exploration sont-elles toutes de même « profondeur » ?
 - Une branche peut amener plus rapidement qu'une autre à la fin de la partie...
 - Donc il faut pouvoir s'arrêter n'importe où ?...

(C) Hervé Barbot, 2005-2010

25

```
Function MinMax ( nœud )
Returns a performance value
```

```
Partie terminée en 'nœud'
  => return performance ( nœud )
```

```
→ Partie terminée en 'nœud'
^ 'nœud' est « Max »
  => return MAX [ MinMax(x) ]
             x ∈ Γ(nœud)
```

```
→ Partie terminée en 'nœud'
^ 'nœud' est « Min »
  => return MIN [ MinMax(x) ]
             x ∈ Γ(nœud)
```

(C) Hervé Barbot, 2005-2011

26

- Complétude

- Oui si l'arbre de jeu est fini

- Optimalité

- Oui si l'adversaire est aussi optimal

- Complexité en temps

- $\mathcal{O}(b^m)$

b = facteur de branchement → 35

m = horizon de la recherche → 100

aux échecs :

Réaliste ?

- Complexité en espace

- $\mathcal{O}(b \cdot m)$

(C) Hervé Barbot, 2005-2010

28

- Solution 1 : profondeur limitée

- Solution 2 : élagage alpha-béta

(C) Hervé Barbot, 2005-2011

29

MinMax en profondeur limitée

- Explorer l'arbre sur une profondeur limitée
- Calculer les valeurs des feuilles
- Les remonter selon l'alternance Min/Max

- Nécessite une bonne fonction d'évaluation
 - D'autant plus qu'on peut être encore loin de la fin de la partie...
 - Ex. pondération de la valeur des pièces
 - Mais cela peut être beaucoup plus compliqué
 - Par exemple valeurs et pondérations différentes selon le moment de la partie...

(C) Hervé Barbot, 2005-2011

30

```
Function MinMax ( nœud , profondeur )
Returns a performance value
```

```
partie terminée en 'nœud'
```

```
⇒ return performance ( nœud )
```

```
profondeur = 0 en 'nœud'
```

```
⇒ return performance ( nœud )
```

```
↪ Partie terminée en 'nœud'
```

```
∧ 'nœud' est « Max »
```

```
⇒ return MAX [ MinMax(x,profondeur-1) ]
               x ∈ Γ(nœud)
```

```
↪ Partie terminée en 'nœud'
```

```
∧ 'nœud' est « Min »
```

```
⇒ return MIN [ MinMax(x,profondeur-1) ]
               x ∈ Γ(nœud)
```

(C) Hervé Barbot, 2005-2011

32

Elagage

- Ne pas analyser certaines branches de l'arbre de recherche lorsqu'elles n'apporteront aucun bénéfice supérieur à, i.e. lorsqu'elles n'auront pas une meilleure « qualité » que ce que l'on a déjà analysé.

- Détection de « seuils » au-delà desquels il est inutile d'explorer
 - Calcul, mise à jour, propagation du « seuil »

(C) Hervé Barbot, 2005-2011

34

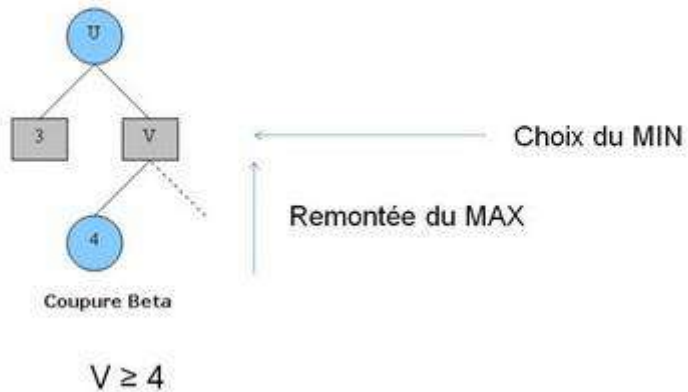
MinMax et élagage $\alpha - \beta$



(C) Hervé Barbot, 2005-2011

37

MinMax et élagage $\alpha - \beta$



(C) Hervé Barbot, 2005-2011

38

▪ Alpha

- Approximation vers le bas de la vraie valeur d'un nœud
- Initialisation = $-\infty$

▪ Beta

- Approximation vers le haut de la vraie valeur d'un nœud
- Initialisation = $+\infty$

(C) Hervé Barbot, 2005-2011

35

▪ Nœud « Max » : α -valeur

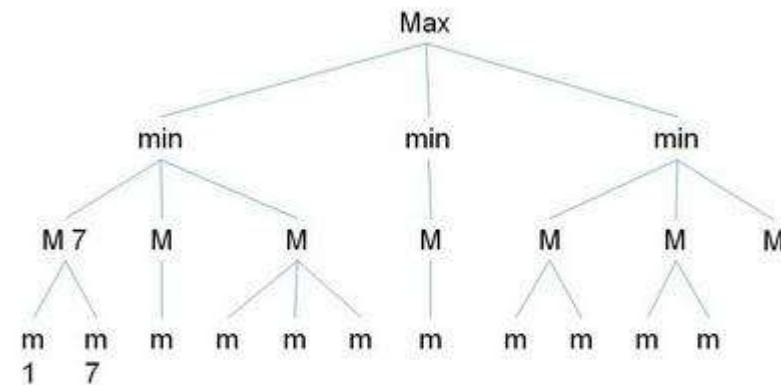
- Valeur de son « meilleur » successeur trouvé jusqu'à présent
- init = $-\infty$

▪ Nœud « Min » : β -valeur

- Valeur de son « plus mauvais » successeur trouvé jusqu'à présent
- init = $+\infty$

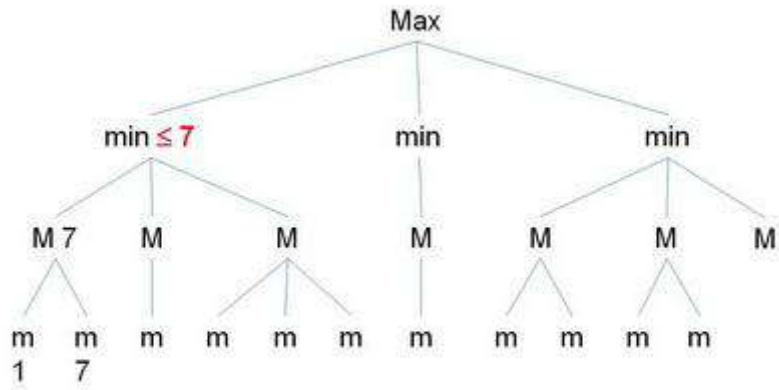
(C) Hervé Barbot, 2005-2010

39



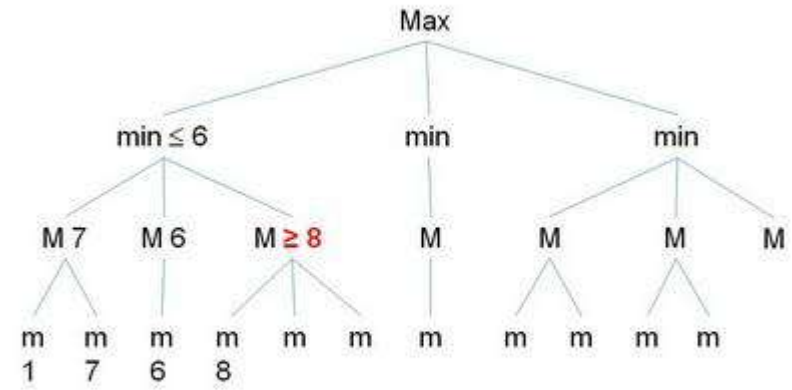
(C) Hervé Barbot, 2005-2010

41



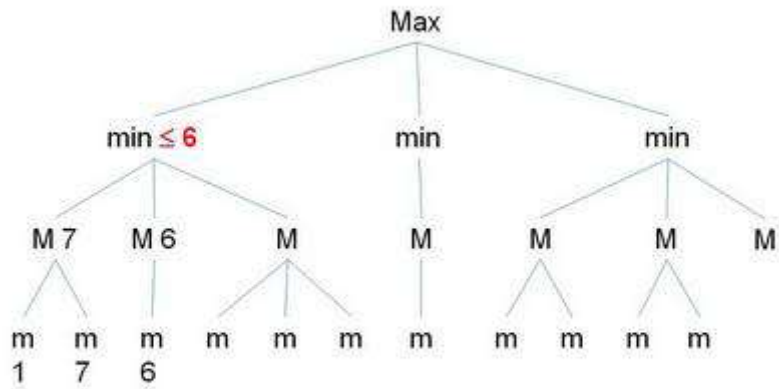
(C) Hervé Barbot, 2005-2010

42



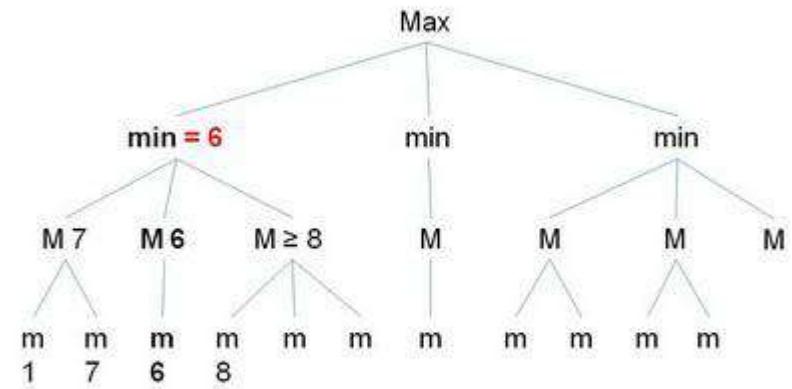
(C) Hervé Barbot, 2005-2010

44



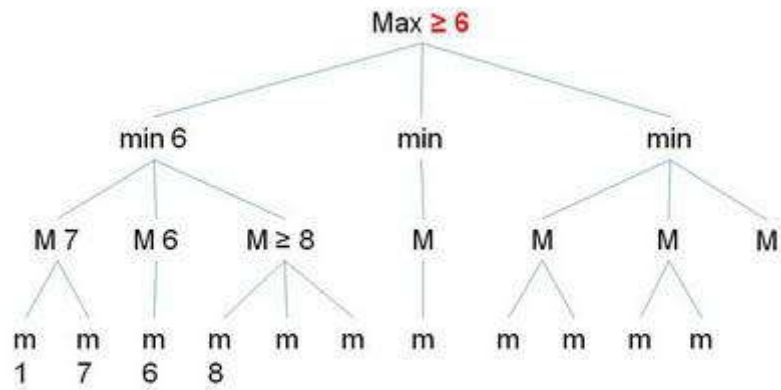
(C) Hervé Barbot, 2005-2010

43



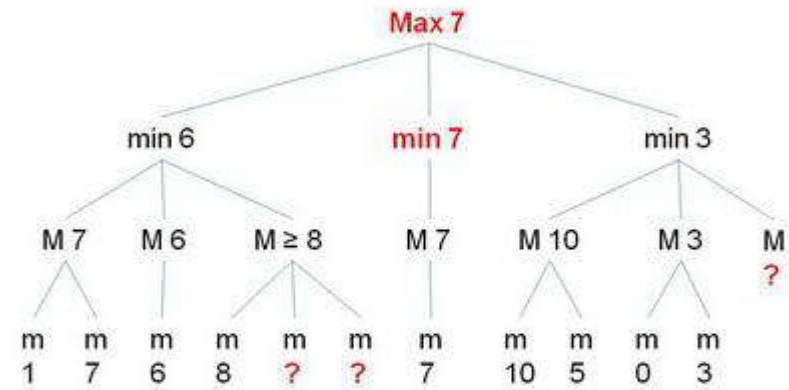
(C) Hervé Barbot, 2005-2010

45



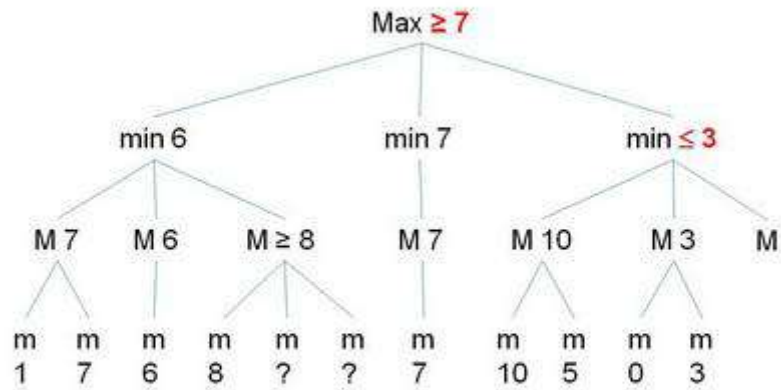
(C) Hervé Barbot, 2005-2010

46



(C) Hervé Barbot, 2005-2010

48



(C) Hervé Barbot, 2005-2010

47

```

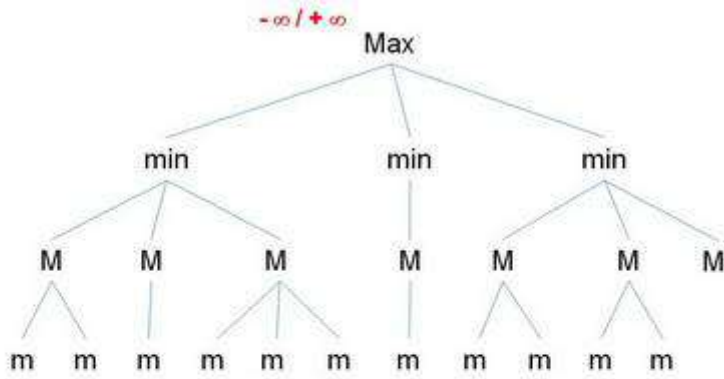
fonction ALPHABETA(P, alpha, beta) /* alpha est toujours inférieur à beta */
si P est une feuille alors
    retourner la valeur de P
sinon
    si P est un nœud Min alors
        pour tout fils Pi de P faire
            Val = ALPHABETA(Pi, alpha, beta)
            beta = Min(beta, Val)
            si alpha >= beta alors /* coupure alpha */
                retourner beta
        finpour
        retourner beta
    sinon
        pour tout fils Pi de P faire
            Val = ALPHABETA(Pi, alpha, beta)
            alpha = Max(alpha, Val)
            si alpha >= beta alors /* coupure beta */
                retourner alpha
        finpour
        retourner alpha
    finsi
finsi
...

```

Vu sur wikipedia...

(C) Hervé Barbot, 2005-2010

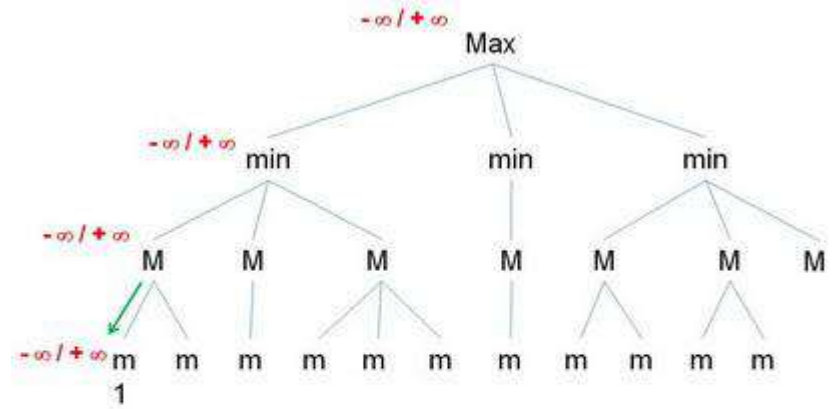
49



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

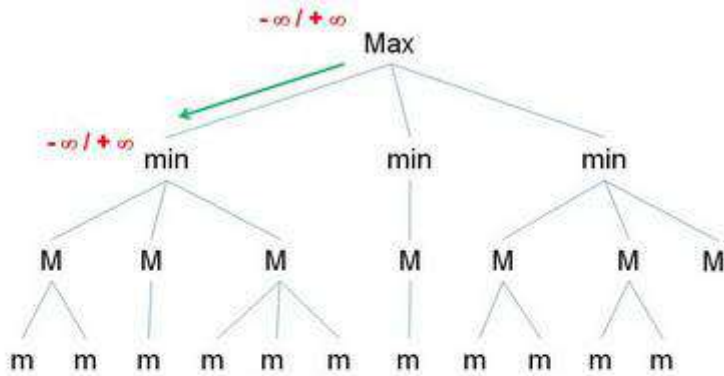
50



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

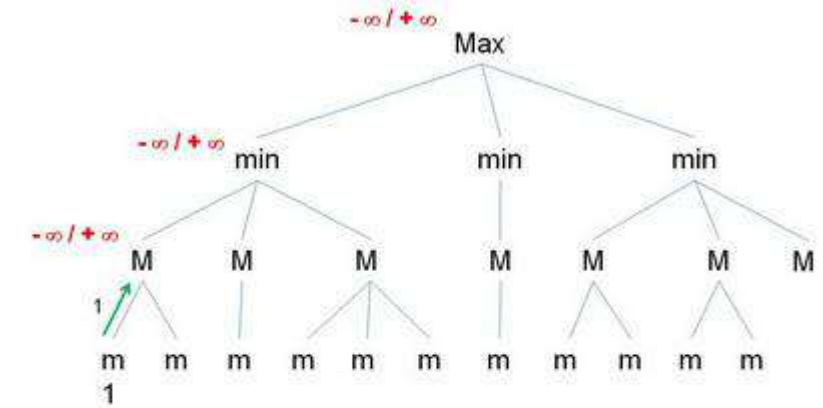
53



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

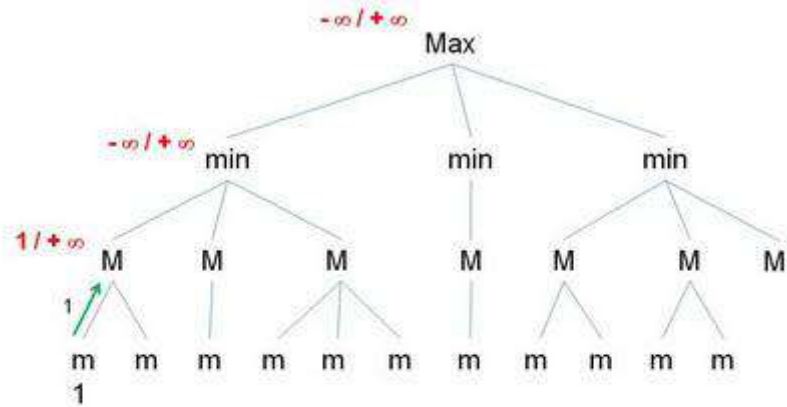
51



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

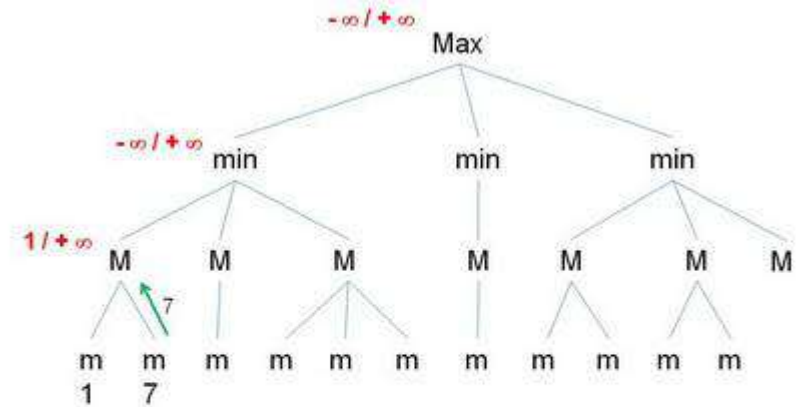
54



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

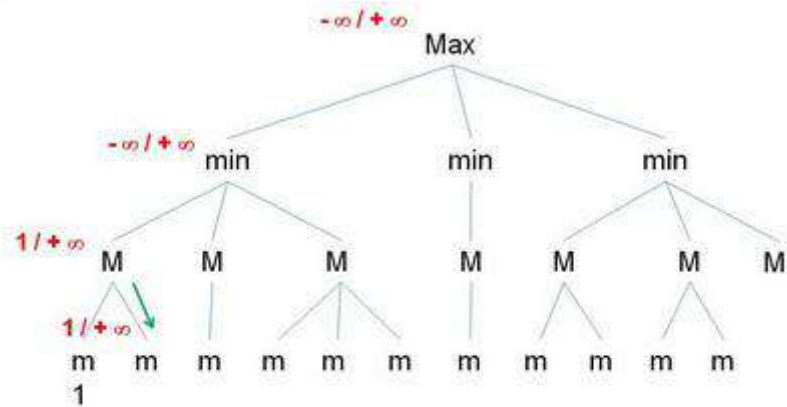
55



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

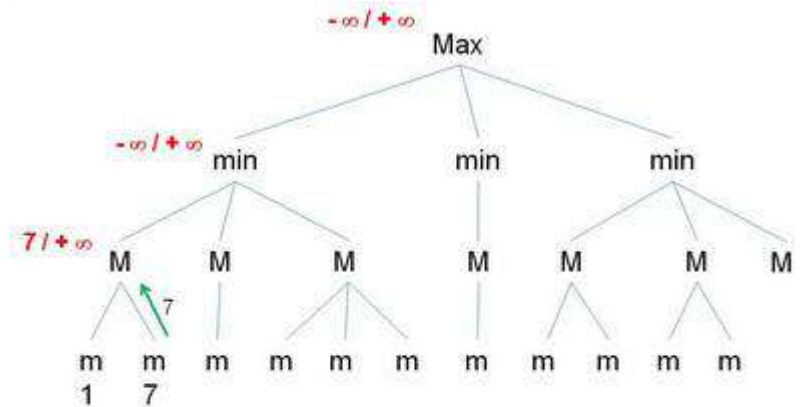
57



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

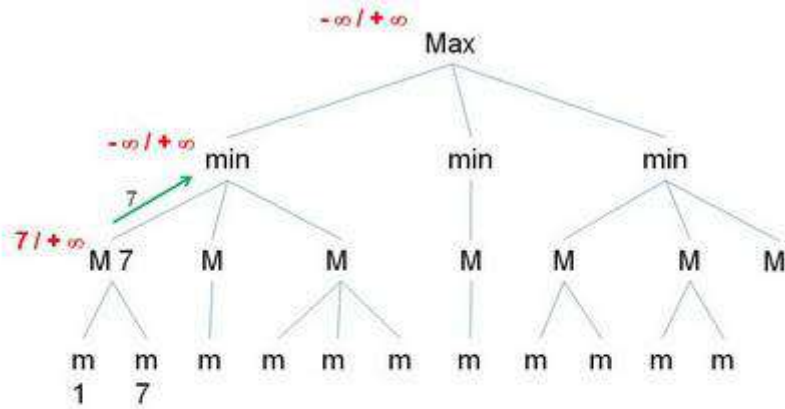
56



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

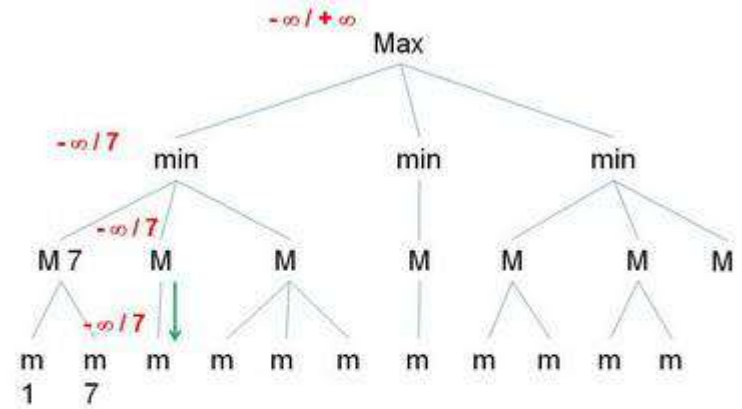
58



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

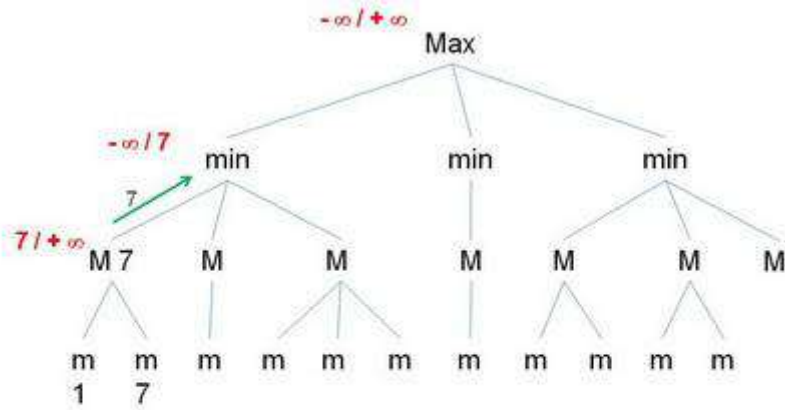
59



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

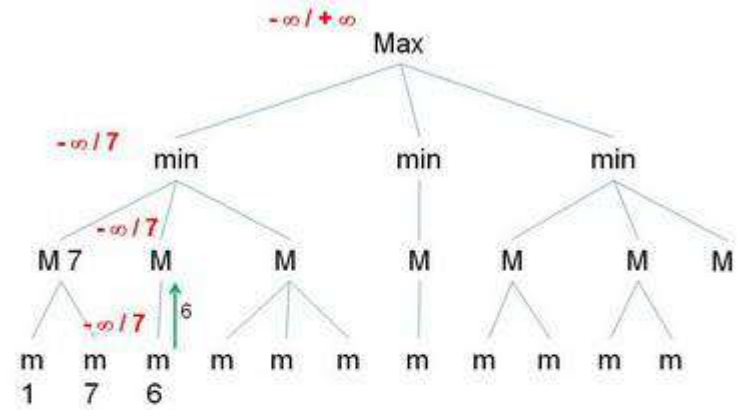
62



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

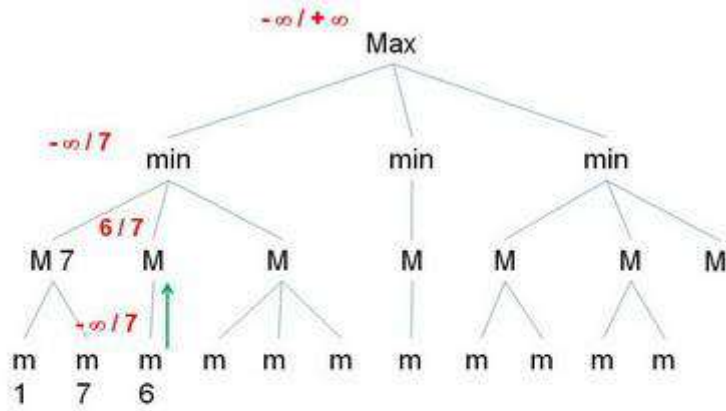
60



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

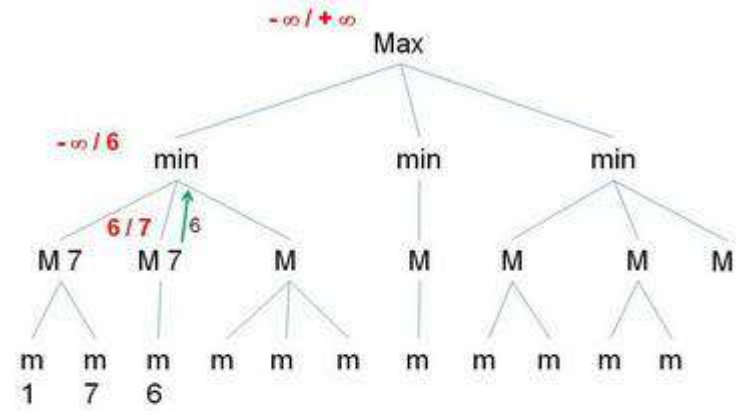
63



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

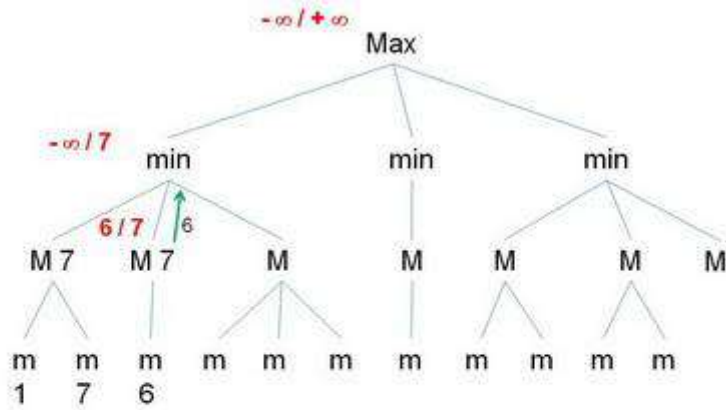
64



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

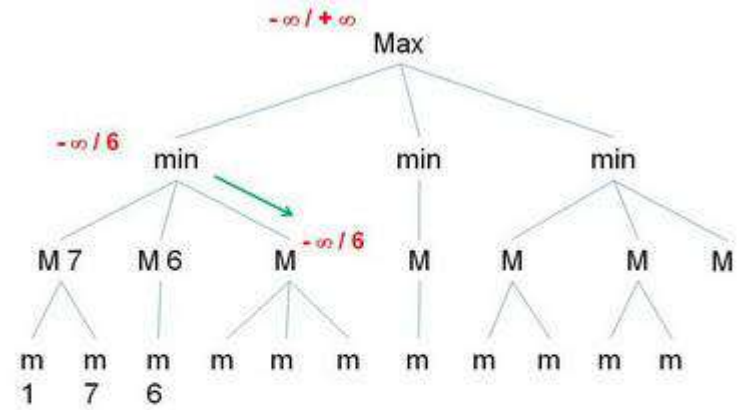
66



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

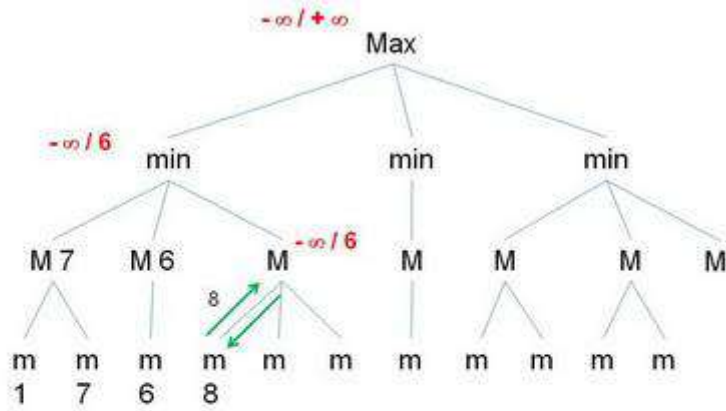
65



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

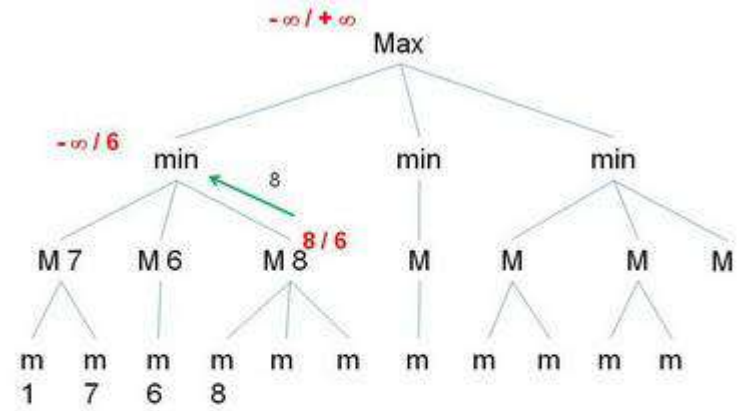
67



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

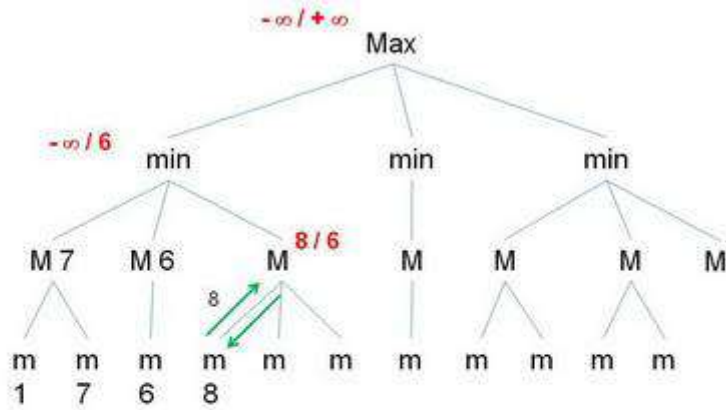
68



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

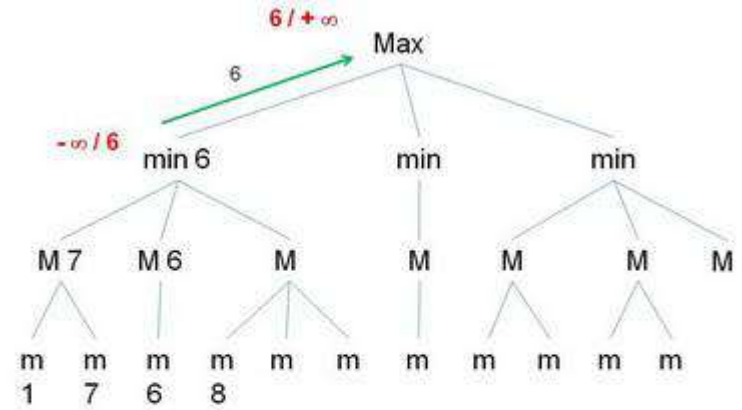
70



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

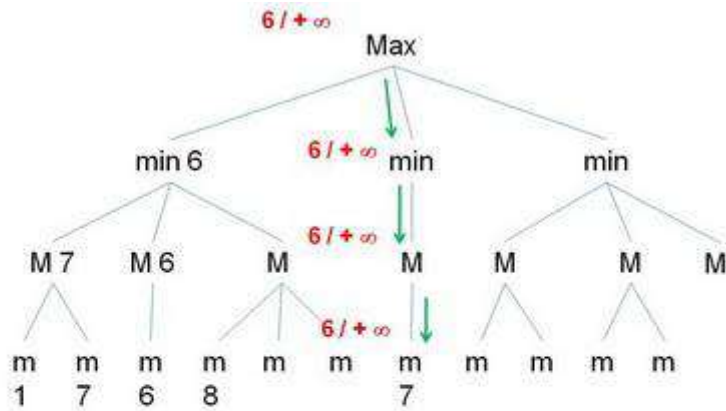
69



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

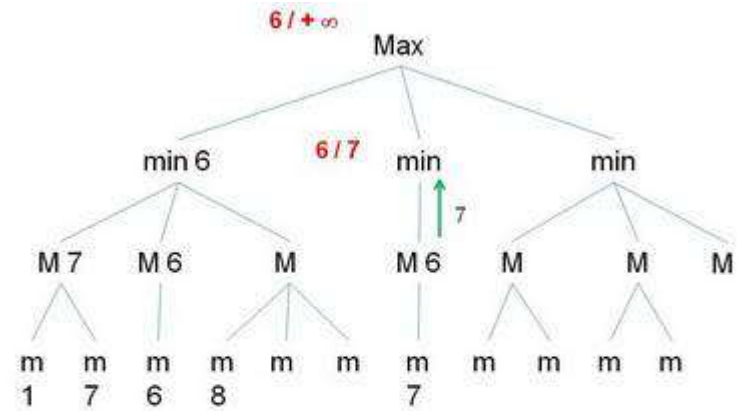
72



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

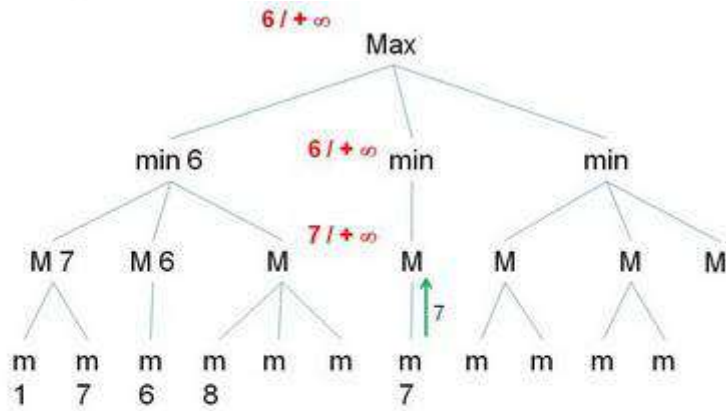
73



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

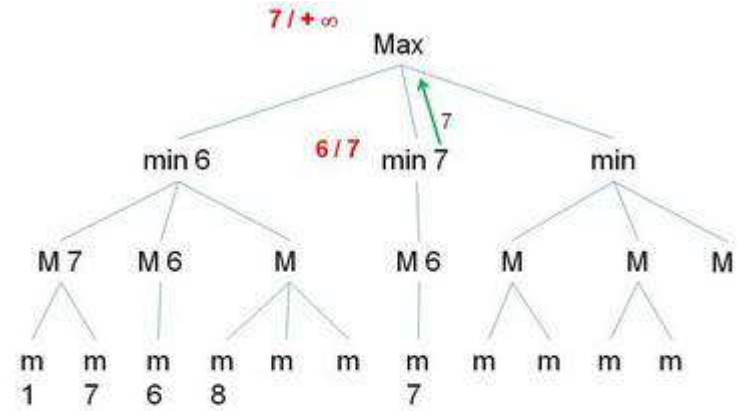
77



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

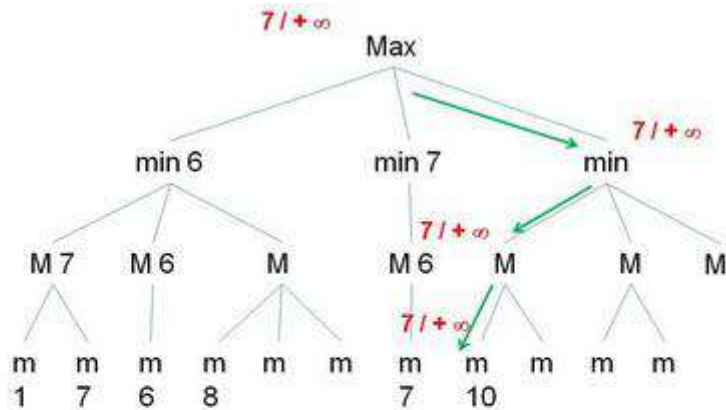
75



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α
Min	Min (β , éval)	Retourner β

(C) Hervé Barbot, 2005-2010

79



Noeud	Pour chaque descendant	Coupe si $\alpha \geq \beta$
Max	Max (α , éval)	Retourner α .
Min	Min (β , éval)	Retourner β .

et caetera !

(C) Hervé Barbot, 2005-2010

80

Nombres conspirants

- Objectif : décider si un sous-arbre est intéressant ou pas
- Instrument de mesure = « nombre conspirant »
= nombre minimum de feuilles qui doivent changer leurs valeurs en approfondissant la recherche pour que la valeur MinMax du sous arbre change.
- Seuil de conspirant
= nombre minimum des nombres conspirants au dessus duquel il est considéré comme improbable que la valeur du sous arbre change
- Règles de calcul pour les différents noeuds
 - Feuilles, noeuds intermédiaires, noeuds min/max

(C) Hervé Barbot, 2005-2010

82

Scout / Tactique d'espionnage

- Estimer sans évaluer
 - N'évaluer que si bénéfice attendu
- Technique de la « fenêtre Zéro »
 - Tirée de alpha-beta
 - Tricherie : les valeurs beta transmises sont faussées
 - Les coupures interviennent rapidement

(C) Hervé Barbot, 2005-2011

83

Elagage et résultat

- L'élagage n'affecte pas le résultat final
...ouff !!!
- L'efficacité de l'élagage est fonction de l'ordre d'apparition / prise en compte des positions « successeurs »
- Au pire, on fait du MinMax « normal »

(C) Hervé Barbot, 2005-2010

84