



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Cerca amb adversari ¹

Albert Sanchis
Alfons Juan
Jorge Civera

DSIC

Departament de Sistemes
Informàtics i Computació

¹Per a una correcta visualització, es requereix l'Acrobat Reader v. 7.0 o superior

Objectius formatius

- ▶ Conèixer la cerca amb adversari bàsica.
- ▶ Aplicar l'algorisme *minimax* i poda *alfa-beta*.

Índex

1	Cerca amb adversari	3
2	Algorisme minimax i poda alfa-beta	5

1 Cerca amb adversari

La cerca amb adversari bàsica consisteix a triar jugada en jocs:

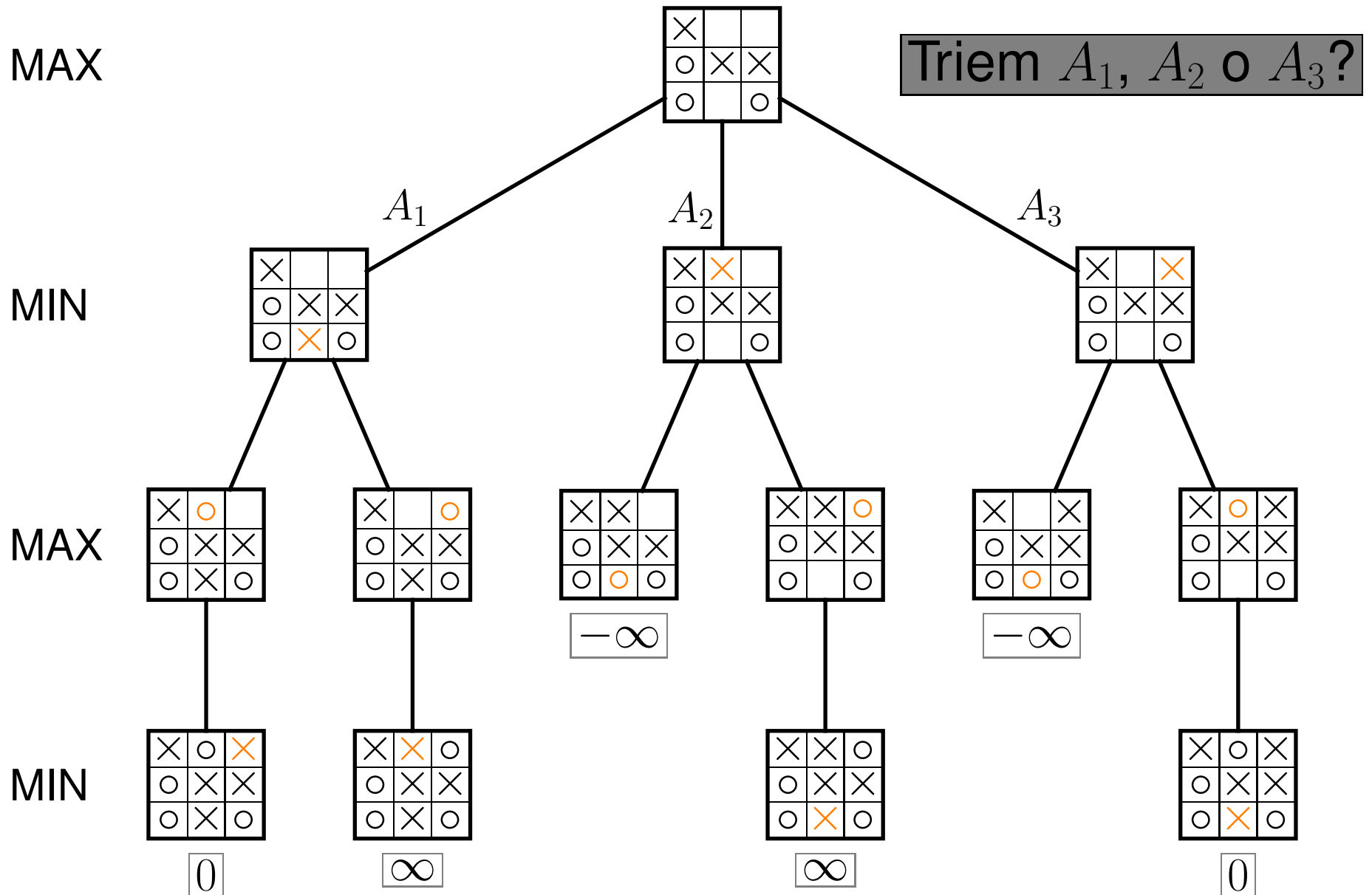
- ▶ **deterministes**, i.e. la sort no intervé
- ▶ **de 2 jugadors**, MAX (el sistema) i MIN (l'adversari)
- ▶ **per torns**, comença MAX i ha de triar jugada
- ▶ **info. perfecta**, coneguem estats i regles del joc (i.e. escacs)
- ▶ **suma zero**, utilitats MAX/MIN al final del joc oposades

Elements bàsics:

- ▶ **Estat inicial** s_0 : des d'on MAX ha de triar jugada.
- ▶ **Accions**(s): jugades legals des de l'estat s .
- ▶ **Terminal**(s): indica si s és estat terminal del joc o no.
- ▶ **Utilitat**(s): utilitat per a MAX de l'estat terminal s .

Objectiu: triar jugada (que porte a un estat) de màxima utilitat

Exemple: triar jugada en el tres en ratlla



2 Algorisme minimax i poda alfa-beta

Valor, decisió i algorisme minimax:

- ▶ **Valor minimax** d'un estat/node: utilitat (MAX) del node terminal al qual arribem si ambdós jugadors juguen òptimament
- ▶ **Decisió minimax**: triar la jugada/acció de major valor minimax
- ▶ **Algorisme minimax**: càlcul de la decisió minimax mitjançant cerca amb adversari per profunditat (limitada)

Algorisme minimax bàsic

```
mm( $n, p, max$ )           // node, profunditat,  $max$ ="juga max?"  
  si  $n$  és terminal retorna utilitat de  $n$   
  si  $p = 0$            retorna valor heurístic de  $n$   
  // si  $max$  torna el màxim de valors minimax dels fills  
  si  $max$   $v = -\infty; \forall s \in succ(n): v = \max(v, mm(s, p - 1, FALSE))$   
  // si no torna el mínim de valors minimax dels fills  
  si no  $v = \infty; \forall s \in succ(n): v = \min(v, mm(s, p - 1, TRUE))$   
  retorna  $v$ 
```

Exemple resultat amb minimax

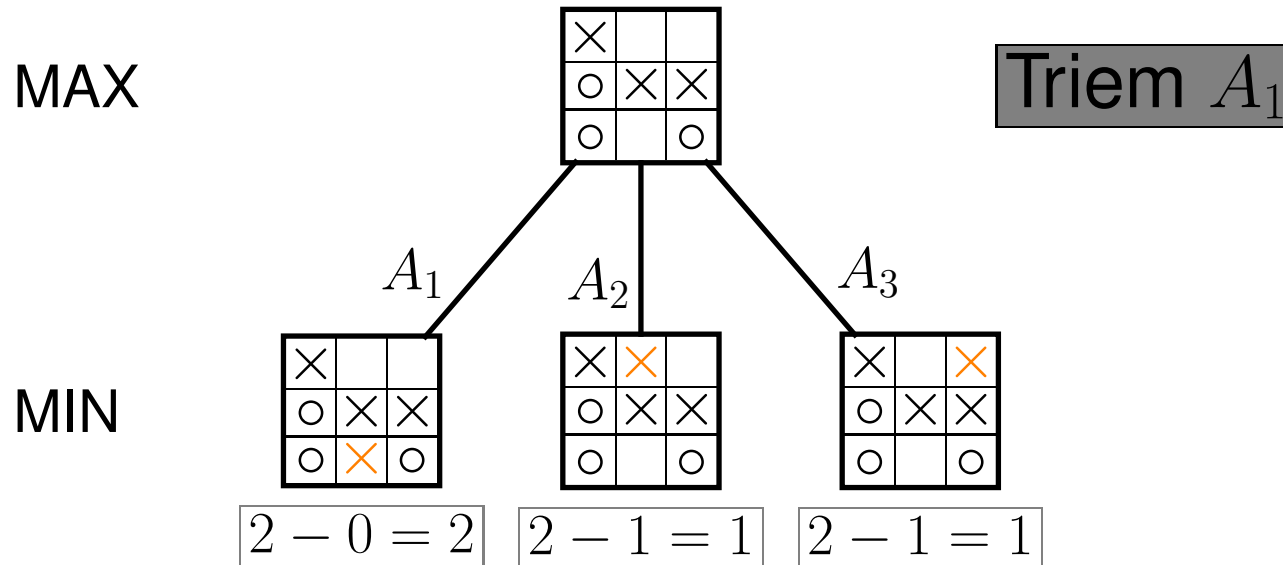
Algorisme minimax i poda alfa-beta

```
mm( $n, p, max$ ) // node, profunditat,  $max$ ="juga max?"  
si  $n$  és terminal retorna utilitat de  $n$   
si  $p = 0$  retorna valor heurístic de  $n$   
si  $max$   $v = -\infty$ ;  $\forall s \in \text{succ}(n)$ :  $v = \max(v, \text{mm}(s, p - 1, \text{FALS}))$   
si no  $v = \infty$ ;  $\forall s \in \text{succ}(n)$ :  $v = \min(v, \text{mm}(s, p - 1, \text{CERT}))$   
retorna  $v$ 
```

```
 $\alpha$ - $\beta$ ( $n, p, \alpha, \beta, max$ )  
si  $n$  és terminal retorna utilitat de  $n$   
si  $p = 0$  retorna valor heurístic de  $n$   
si  $max$   $v = -\infty$   
     $\forall s \in \text{succ}(n)$   
         $v = \max(v, \alpha\text{-}\beta(s, p - 1, \alpha, \beta, \text{FALS}))$   
         $\alpha = \max(\alpha, v)$ ; si  $\beta \leq \alpha$ : trenca  $\forall$  // tall  $\beta$   
si no  $v = \infty$   
     $\forall s \in \text{succ}(n)$   
         $v = \min(v, \alpha\text{-}\beta(s, p - 1, \alpha, \beta, \text{CERT}))$   
         $\beta = \min(\beta, v)$ ; si  $\beta \leq \alpha$ : trenca  $\forall$  // tall  $\alpha$   
retorna  $v$ 
```


Exemple resol't amb poda alfa-beta

Exemple resol't amb $p = 1$ i heurística



Funció heurística:

$$h(n) = \text{obertes}(\text{MAX}) - \text{obertes}(\text{MIN})$$

on

$\text{obertes}(j) = \text{"\# de files, columnes i diagonals obertes per a } j\text{"}$

Conclusions

- ▶ Hem vist en què consisteix la cerca amb adversari bàsica.
- ▶ Hem vist com aplicar l'algorisme *minimax* i poda *alfa-beta*.
- ▶ Consulteu [1, Cap. 5] per a més detalls.

Referències

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.