

# 30 – Prohledávání herního stromu

algoritmus Minimax, alfa-beta prořezávání

- Hry v **extenzivní formě** – hry vyjádřené stromem (piškvorky, šachy, dáma,...)
- Dvouhráčové **zero-sum hry** (tzn. hráč získá všechno, co protivník ztratí)
- **Utilitní funkce** 2 hráčů nahradíme jednou funkcí  $u: T \rightarrow R$ 
  - o Někáká heuristika (v případě šachů naivní příklad  $u = \#bílých - \#černých$  figurek – pokud je  $u > 0$ , bílý vyhrává, pokud je  $u < 0$ , černý vyhrává)
  - o rychlá, spolehlivá (opravdu koreluje s kvalitou stavu)
- Hráči:
  - o **MAX** ( $\Delta$ ) – snaží se maximalizovat utilitní fci
  - o **MIN** ( $\nabla$ ) – snaží se minimalizovat utilitní fci
- **Perfektní hra** – hráč volí v každém tahu ideální (bezchybnou) akci
  - o Většina her nemožná hrát perfektně (kombinatorická exploze)
    - Využíváme proto heuristiky
  - o Prakticky za použití heuristiky prohledáváme herní strom do malé hloubky

## MiniMax algoritmus

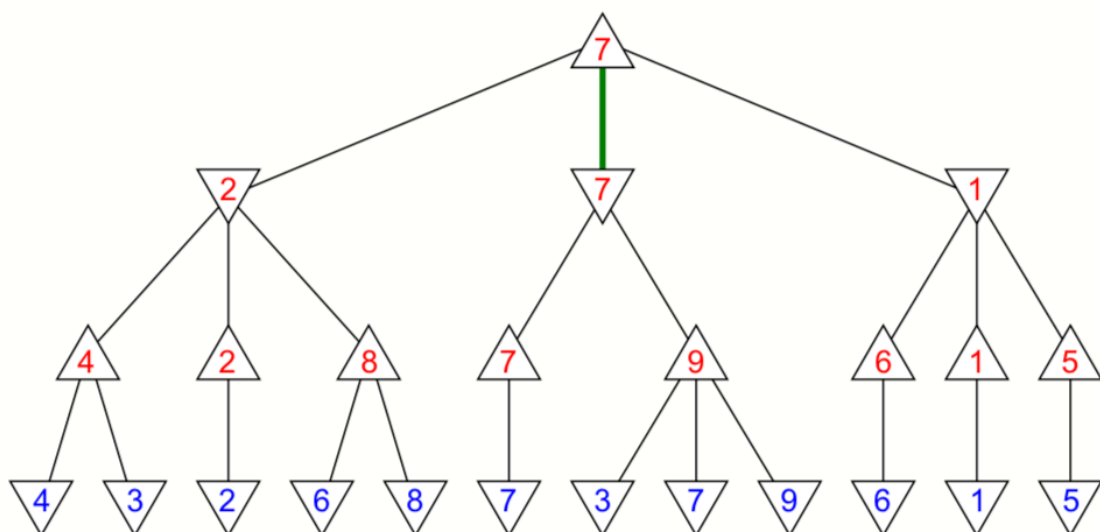
- Algoritmus který umožňuje zvolit ideální akci
- Předpokládá, že soupeř hraje ideální hru

### Průběh algoritmu (rekurzivní, do hloubky d)

1. Začni v počátečním uzlu  $x_0$  typu MAX
2. Rekurzivně volej potomky vrcholu až do hloubky d
  - Buď narazíš na terminál, tak se zastav a vrať terminál. Jinak se zastav v hloubce d kde místo zanořování spočítej utilitní fci.
  - Po návratu hodnot potomků vyber buď MIN nebo MAX podle typu vrcholu.
3. Po návratu do  $x_0$  vidíš, odkud přišla nejvyšší hodnota a tím směrem se vydej.

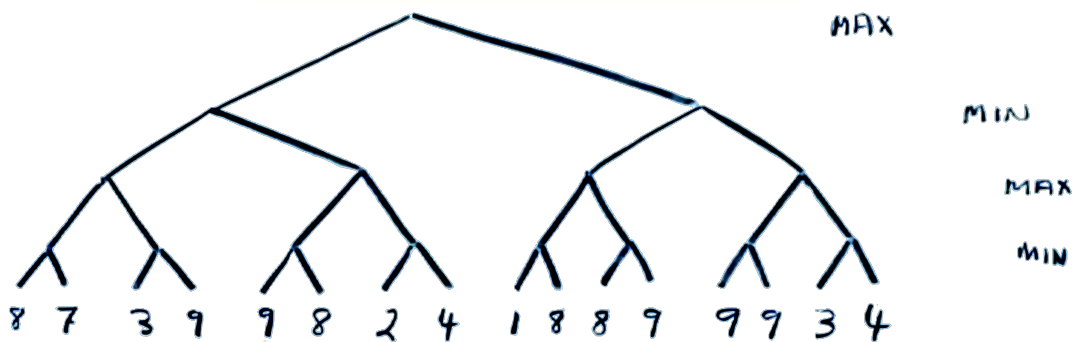
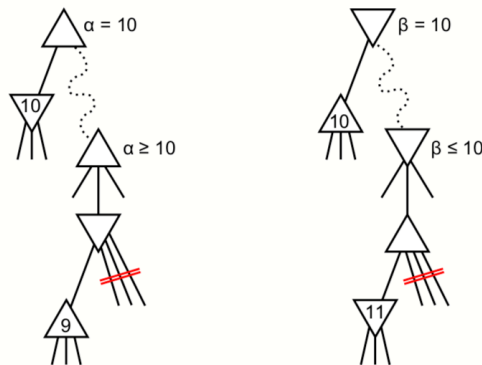
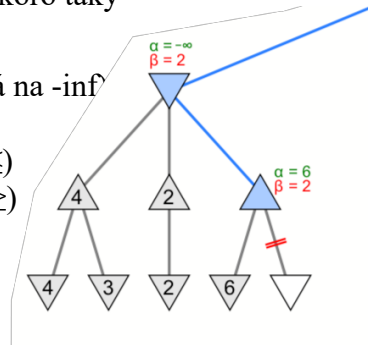
$\text{MINIMAX}(s) =$

$$\begin{cases} \text{UTILITY}(s) & \text{if } \text{TERMINAL-TEST}(s) \\ \max_{a \in \text{Actions}(s)} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \min_{a \in \text{Actions}(s)} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MIN} \end{cases}$$



## Alfa-beta prořezávání

- Dobrý návod: <https://www.youtube.com/watch?v=l-hh51ncgDI>
- Příklady: <http://people.cs.pitt.edu/~litman/courses/cs2710/lectures/pruningReview.pdf>
- Úspěchy např. Deep Blue porazil Kasparova, AlphaGo (monte carlo tree search)
- MiniMax funguje dobře jen pro malá  $d$  – množství možných stavů roste exponenciálně
- Většina her nemožná hrát perfektně (kombinatorická exploze)
- Alfa-beta prořezávání je jen efektivnější verze MinMaxu
  - o Prakticky za použití heuristiky prohledáváme herní strom do malé hloubky
- MinMax má složitost  $O(b^d)$  kde  $d$  je hloubka a  $b$  branching factor
  - o V ideálním pomoci alfa/beta pruning  $O\left(b^{\frac{d}{2}}\right)$ , reálně ale skoro taky
- $\alpha, \beta$  jsou parametry algoritmu
- V každém vrcholu si navíc uchováváme hodnotu  $\alpha$  (inicializovaná na  $-\infty$ ) a  $\beta$  (inicializovaná na  $\infty$ )
  - o  $\alpha$  – největší hodnota utility, kterou má MAX hráč jistou ( $\leq$ )
  - o  $\beta$  – nejmenší hodnota utility, kterou má MIN hráč jistou ( $\geq$ )
- Hodnoty aktualizujeme při průchodu grafem.
- Pokud  $\beta \leq \alpha$ :
  - o Break. Neprohledáváme další potomky.



- Řešení: Nebudou se prozkoumávat 4, 7, 8, 10, 15 a 16tý list

