

# IV130 Přínosy a rizika inteligentních systémů

**Určitost prostředí, nejistota, pravděpodobnost, teorie her**

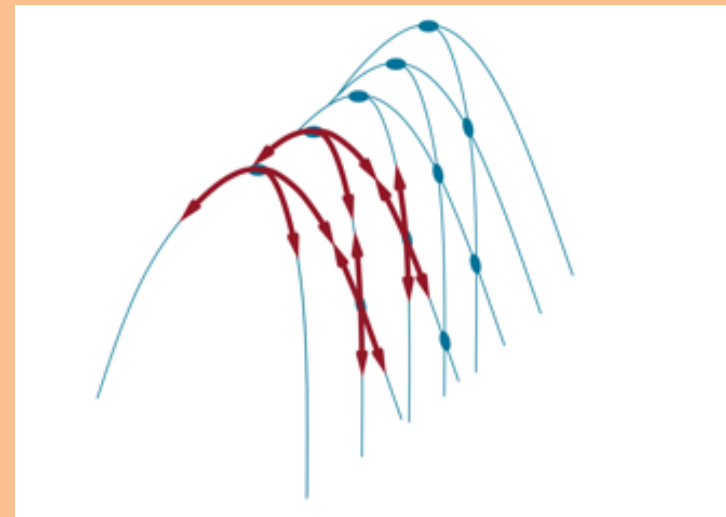
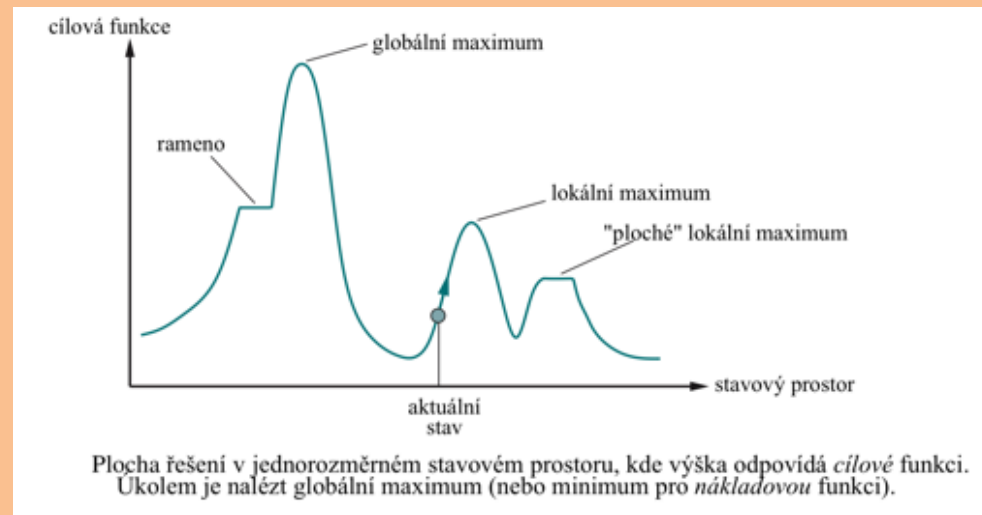
31. března 2023

# Vyhledávání ve složitých prostředích

- Lokální hledání a optimalizace
- Hledání šplháním do kopce
- Stochastické šplhání, šplhání s náhodným restartem, simulované žíhání, čidla jsou jediným zdrojem informací o prostředí, šplhání s místním reflektorem (kombinace výsledků z několika paralelních běhů), evoluční algoritmy
- Nedeterministická hledání
- Hledání v částečně pozorovatelném prostředí
- Hledání v neznámém prostředí a on-line aktéři zpracovávající vjemy okamžitě (a rizika slepých uliček)

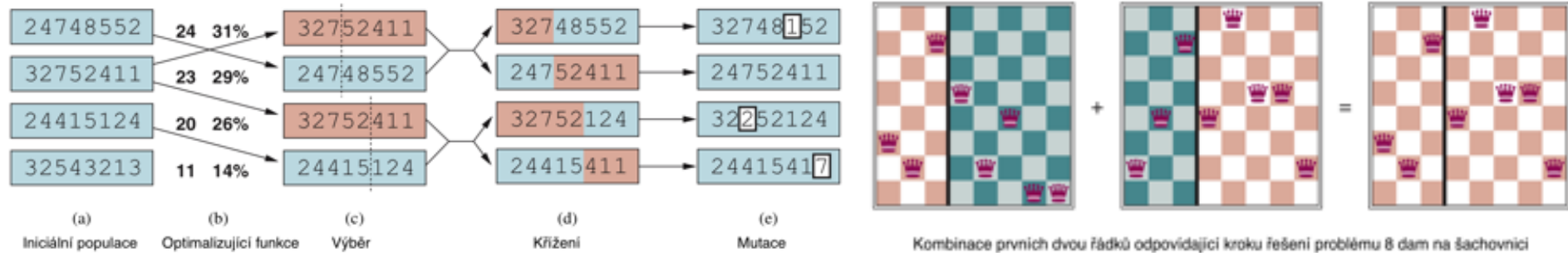
# Lokální hledání a optimalizace

- Hledání řešení lze často popsat jako optimalizační problém – hledání
  - ❖ maxima nějaké cílové funkce nebo
  - ❖ minima nákladové funkce
- Pohyb ve směru rostoucí hodnoty cílové funkce (nebo poklesu gradientu u nákladové) zpravidla dostačuje u hledání lokálního minima/maxima („šplhání do kopce“/“sestup do údolí“)
- Globální minimum/maximum může vyžadovat nelokální prohledávání
  - Ramena (hřebeny u vícerozměrných) mohou vést k zabloudění – hodnoty na hřebenu nemusejí být v sousedství
    - *stochastické hledání* generováním více sousedních bodů
    - *návraty s náhodnou volbou* dalších pokusů
    - *simulované žíhání* – náhodné přeskládání následníků („zatřesení“ stavovým prostorem a náhodné hledání jiného minima/maxima).



# Evoluční algoritmy

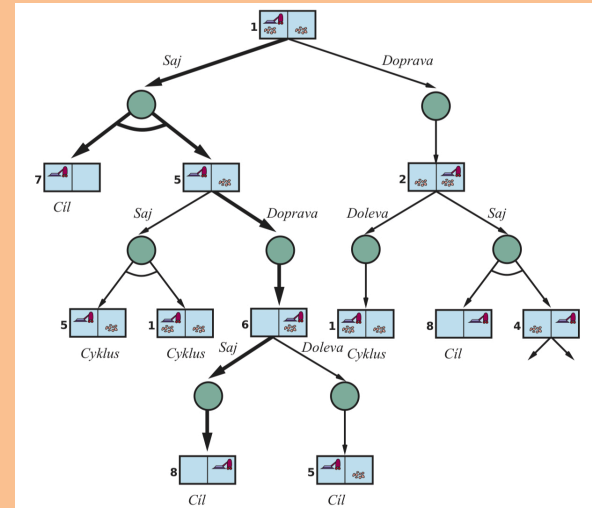
- Stochastické vyhledávání motivované metaforou přírodního výběru v biologii:
- ❖ v populaci stavů hledáme kombinace lokálně nejlepších stavů k vytvoření následníka (stavu) vytvořených *rekombinací* stavů



- Evoluční strategie zadává způsob výběru dalších potomků
- Proces selekce určuje způsob výběru rodičovských stavů (např. s nejvyššími hodnotami cílové funkce nebo náhodný výběr s následným určením nejlepších potomků)
- Mutační míra definující frekvenci náhodných mutací s výběrem nejlepších potomků
- Stavby mohou mít podobu schémat definující podřetězce a náhodné změny se odehrávají ve zbylých částech (podřetězce stanoví „znaky“ společné generovaným mutacím)
- Přidáním učení lze omezovat velikost stavového prostoru, který je procházen náhodně, atd.

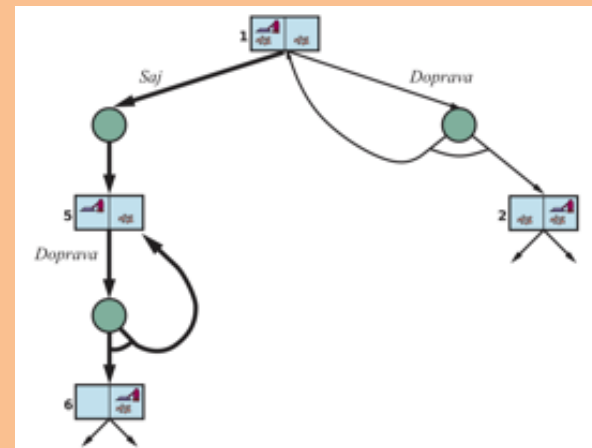
# Prohledávání s nedeterministickými akcemi

- Akce nemusí mít jednoznačně definovaný výsledek
- Příklad: vysavač, který funguje s nějakou chybovostí:
  - ❖ na poli s nečistotou někdy na pokyn „vysát“ nevysaje
  - ❖ na poli bez nečistoty občas vyprázdní svůj koš
- Stavový diagram odpovídá AND-OR stromu, kde AND uzly určují možné potenciální výsledky, které musejí být při prohledávání zkoumány současně a řešení odpovídá větvicímu se stromu
- Část cest možných cest stavovým prostorem je cyklická



První dvě úrovně stavového AND-OR stromu pro chybující vysavač. Tučně je vyznačen strom odpovídající jednomu možnému řešení.

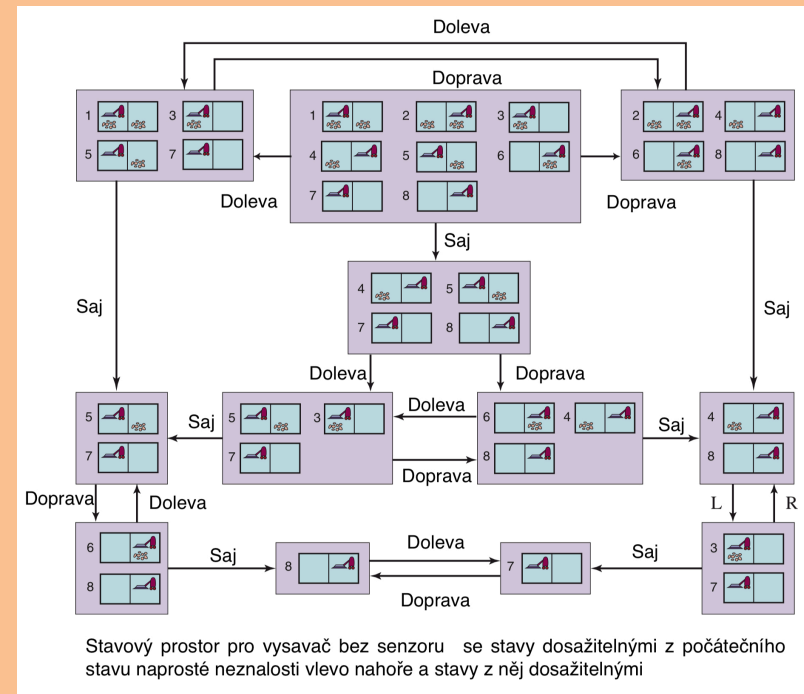
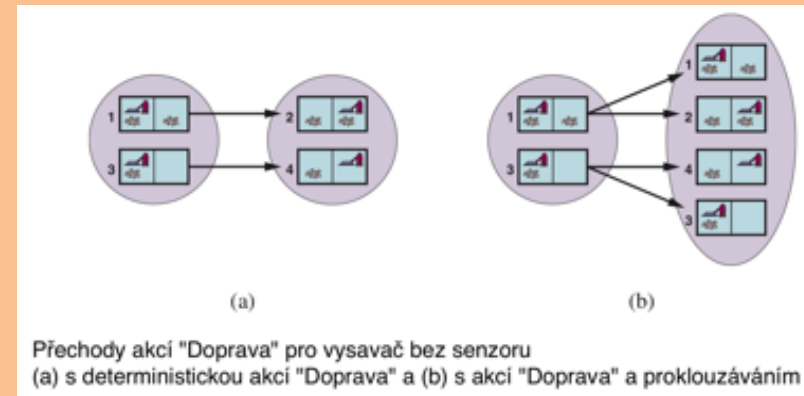
- Příklad: vysavač, který klouže a někdy neprovede posun požadovaným směrem
- Stavový diagram odpovídá AND-OR stromu, v němž jsou všechna řešení cyklická



Část stromu hledání pro klouzající vysavač, kde všechna řešení jsou cyklická

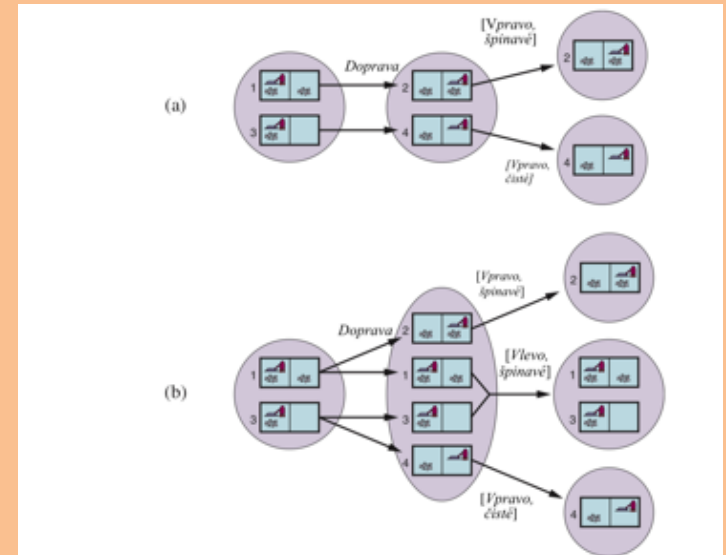
# Hledání v prostředí bez viditelnosti

- Aktér, který nemá možnost vidět prostředí (nemá sensorický vstup)
- ❖ Stavys jsou tvořeny podmnožinami možných stavů (přesvědčení o světě na základě možných konfigurací)
- ❖ Počáteční stav může být množinou všech možných stavů nebo podmnožinou s nějakým apriorním omezením
- ❖ Akce odpovídají přechodům mezi množinami generovaným jednotlivými stavy v nich
- ❖ Přejchodové stavys pro deterministické nebo nedeterministické akce tvořeny množinami sjednocení všech možných výsledků akcí
- ❖ Cílový stav definován jako *možný* nebo *nutný* úspěch
- ❖ Cenu akcí lze snadno definovat jen pro akce stejné ceny
- Odpovídající stromy hledání mohou být exponenciálně velké (díky kardinalitě množin ve stavech)
- Inkrementální konstrukce může vést k modifikaci prohledávacích algoritmů, která je zvládnutelnější
- Relevantní modifikace zahrnují také stavys s chybným/vadným senzorem

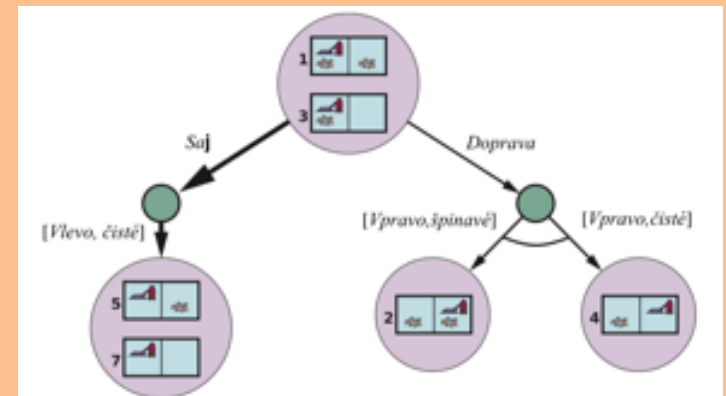


# Hledání v prostředí s částečnou viditelností

- Aktér mající omezenou viditelnost okolí (pouze částečný sensorický vstup)
- Stavys jsou tvořeny podmnožinami možných stavů s dodatečnou informací ze senzoru, která je omezuje (přesvědčení o světě na základě možných konfigurací zúžené na základě sensorického vstupu)
- Příklad: problém 8 dam není řešitelný bez sensorického vstupu, stačí však u každé figury jen viditelnost čtverce vlevo nahoře
- ❖ Akce odpovídají přechodům mezi množinami generovaným jednotlivými stavys v nich spolu s predikcí odpovídající stavům, které jsou výsledkem akce
- ❖ Přechodové stavys pro deterministické nebo nedeterministické akce tvořeny množinami sjednocení všech možných výsledků akcí aktualizované pomocí stavu vnímaného senzorem
- Příklad: Vysavač se senzorem vnímající pouze stav políčka, na němž vysavač stojí: *[poloha, čistota]*



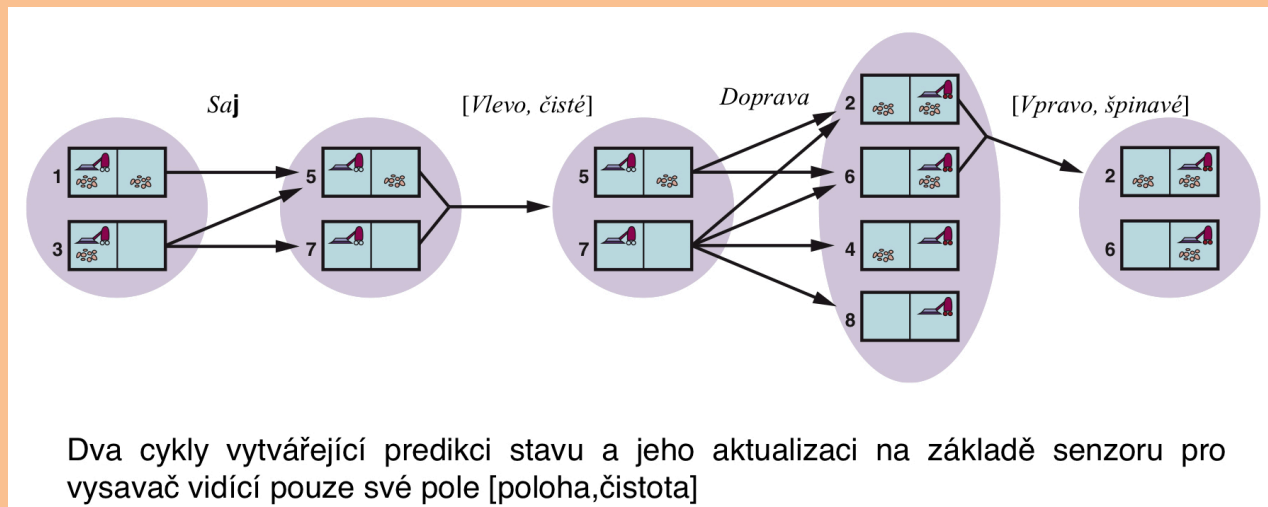
Dva příklady přechodů pro vysavač vidící pouze svoje pole [polohu a stav]:  
(a) v deterministickém případě a akci "Doprava"  
(b) v případě s prokluzováním a akci "Doprava"



První úroveň AND-OR stromu hledání pro vysavač se senzorem jen pro vlastní pole se sáním jako první akcí řešení

# Hledání v prostředí s částečnou viditelností

- ❖ Akce odpovídají přechodům mezi množinami generovaným jednotlivými stavy v nich spolu s predikcí odpovídající stavům, které jsou výsledkem akce
- ❖ Aktér si vytváří stav možné konfigurace prostředí (přesvědčení o světě) pomocí predikce výsledku provedené akce a jeho zúžení na základě následného senzorkého vstupu
- Příklad: Vysavač se senzorem vnímající pouze stav políčka, na němž vysavač stojí:  $[poloha, čistota]$ , v daném prostředí se navíc kdykoli může objevit smetí na kterémkoli políčku, pokud je aktér zrovna nečistí:





# Reálné hledání v neznámém prostředí

- Hledání reálných aktérů (robotů ve fyzickém světě) sestává ze sensorického vstupu hodnotícího stav prostředí a následného výpočtu akce, která má být provedena (a opakování této dvojice)
- Aktér typu robota si zpravidla vytváří mapu prostředí, kterou dále doplňuje a/nebo aktualizuje
- On-line hledání bez předběžné znalosti prostoru prohledávání
- Zpravidla doprovázeno možností zabloudění do slepé uličky – představuje problém ve stavových prosorech, kde prováděné akce nejsou reverzibilní
- Hledání na bázi stochastických postupů závisí na riziku vydání se do slepé uličky
- Prohledávání je zpravidla spojeno s heuristickými odhady dalšího pokračování a průběžné aktualizace těchto heuristik na základě známé mapy stavového prostoru

# Problémy řešené jako plnění omezení

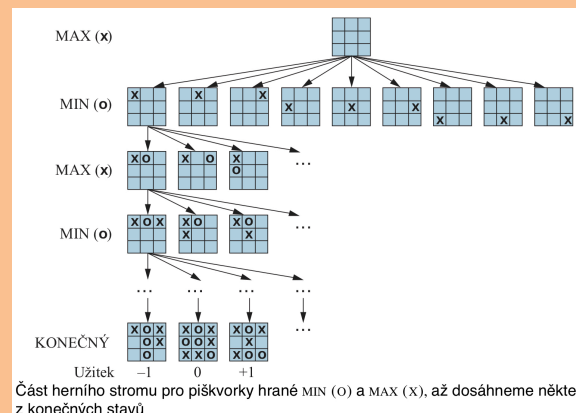
- *Znalost struktury stavů* (namísto atomických stavů) může pomoci použít efektivnější algoritmus namísto prohledávání celeého stavového prostoru
- Jedním ze způsobů otevření stavů je **formulace omezení a řešení problému plnění těchto omezení** (CSP – constraint satisfaction problem)
- Omezení dána pomocí relací mezi proměnnými
  - Příklad: problém barvení mapy barvami (Omezení barev kolem Česka: {ČR#SR,ČR#R,ČR#N,ČR#P,SR#R,R#N,N#P,P#SR})
  - Příklad: Plánování práce v továrnách (návaznost dílů, potřeba nástrojů a dělníků)
  - Příklad: Plánování rozvrhu hodin (učitelé i studenti v různých předmětech a vyloučení kolizí)
- Možný obecný přístup k řešení: hledání s navracením (backtracking), volba řešených proměnných v uzlu, typ odvození provedené v uzlu, hledání řešení pomocí prohledávání do hloubky s navracením a vyzkoušením jiné varianty
- Oproti „neinformovanému“ hledání do hloubky jsou možné heuristiky výběru proměnných a prováděných odvození výrazně zrychlující hledání (plus další speciální metody řešení)
- Postup řešení problému 8 dam na šachovnici formuloval jako hledání s navracením již C. F. Gauss v roce 1850 („Tatonniren“ od francouzského šátrání kolem sebe ve tmě)

# Kompetitivní prostředí, teorie her

- Větší počet aktérů sledujících potenciálně navzájem konfliktní cíle (hledání řešení v problému s *protivníky*)
- Multi-aktérové prostředí v případech:
  - ❖ Velký počet aktérů a sledování agregovaných veličin, bez nutnosti sledovat jednotlivé aktéry ... ekonomie
  - ❖ Konkurenční aktéři mohou mít charakter faktoru zavádějícího nedeterminističnost do prostředí – ale obecně budou mít vlastní strategie konání a budou se nás snažit porazit
  - ❖ Aktéři mohou svou činnost protivníků dělat na půdoryse, který odpovídající hrám hraným protivníky a studovatelným také aparátem teorie her
- Terminologie občas užívaná ve hrách: „tah“ je synonymem pro akci, „pozice“ je synonymem pro stav

# Dva hráči, hra s nulovým součtem

- Základní typ hry je deterministická hra s dokonalými informacemi o stavech, s nulovým součtem (přínos pro jednoho hráče odpovídá ztrátě druhého hráče)
- Počáteční stav je výchozí postavení hry
- Akce jsou platné tahy podle pravidel
- Přejítodový dán výsledky tahu v daném herním poli
- Konečné stavy, kde hra končí
- Funkce užitku / cílová funkce
- Herní strom je strom prohledávání s akcemi a odpovídajícími změnami stavu



# Minimax hledání a alfa-beta

- Prohledávání stromu řešení do hloubky s návraty

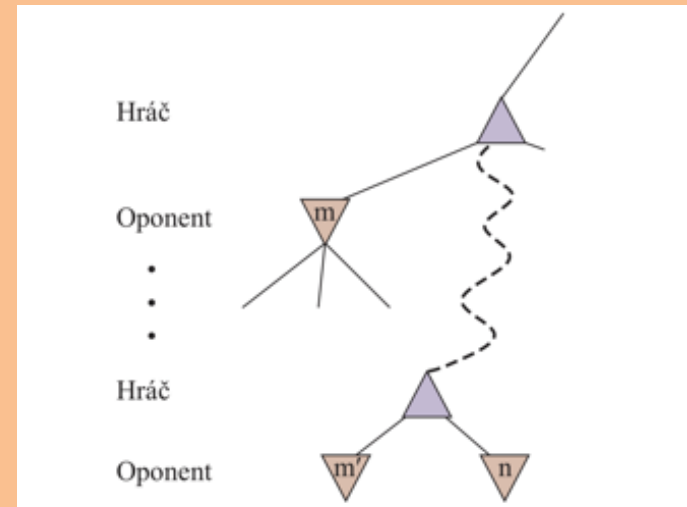
- Hledá se nejlepší výsledek pro daného hráče (max krok) a minimální pro protivníka (min krok)

- Problémem je exponenciální časová složitost

- Alfa-beta ořezávání jako vylepšením minimaxu ořezáváním některých alternativ a výběrem optimálních uzlů; účinnost ořezávání lze zvýšit výběrem vhodné heuristiky

- Navrženo již C. Shannonem v roce 1950 (Programming a Computer for Playing Chess): *Strategie A* zkoumání všech tahů do určité hloubky (široká, ale mělká) a *Strategie B*, která pomíjí tahy, které vypadají špatně, prohledává do hloubky úzkou část stromu.

- Strategie B se nyní ukazuje lepší v řadě her (pro Go byla užívána dříve než pro šachy), závisí na vhodné heuristice výběru slibné cesty (zpravidla kombinace heuristiky, včetně náhodně generovaných, a prohledávání s návraty – důležité je neuříznout vyhrávající variantu).



Oběcny případ alfa-beta ořezávání.  
Je-li pro Hráče  $m$  nebo  $m'$  lepší než  $n$ , pak se ve hře nikdy nedostaneme k  $n$ .

# Částečná viditelnost ve hrách

- Hry typu dětských her „lodě“ bez celkové znalosti herní plochy
- Varianty her typu šachy, kde např. není znám typ kamene (Kriegspiel – šachovnici vidí plně jen rozhodčí, který hlásí i nepřípustnost tahů, atd., hráči vidí jen své vlastní kameny)
- Tvorba stavů odpovídajících přesvědčení o světě (hrací desce), plus koncepty typu „pravděpodobnostní mat“ či „náhodný mat“
- Strategie zahrnující také minimalizaci informací, které protivník může mít o poloze některých kamenů
  
- Karetní hry se stochastickou částečnou informací (protivníkovy karty jsou skryty, lze však pracovat s pravděpodobnostmi (minimax hledání s očekávanými stavy s odvozenou pravděpodobností)
- Libratus pro poker pracuje na úrovni jedné hry s cca 25 miliony hodin CPU (nestejně pravděpodobnosti založené mimo jiné na licitování); úroveň již přečí lidské hráče
  
- Člověk dosud lepší v Kriegspiel, v bridži; vyrovnaná je již situace ve videohrách (StarCraft, Dota 2)