

# **Prehl'adávania stavového priestoru**

**RNDr. Eva Rakovská, PhD.**

**Zdroj: P. Návrat, Umelá inteligencia,  
2007 (2015)**

# *Procedúry hľadania:*

## *Kritériá efektívnosti algoritmu hľadania:*

- *Úplnosť*=ak existuje riešenie problému, algoritmus ho vždy nájde
- *časová zložitosť*
- *pamäťová zložitosť*
- *Prípustnosť*= ak existuje aspoň jedno optimálne riešenie problému algoritmus ho vždy nájde (nemusí nájsť všetky)

## *Procedúry neinformovaného hľadania:*

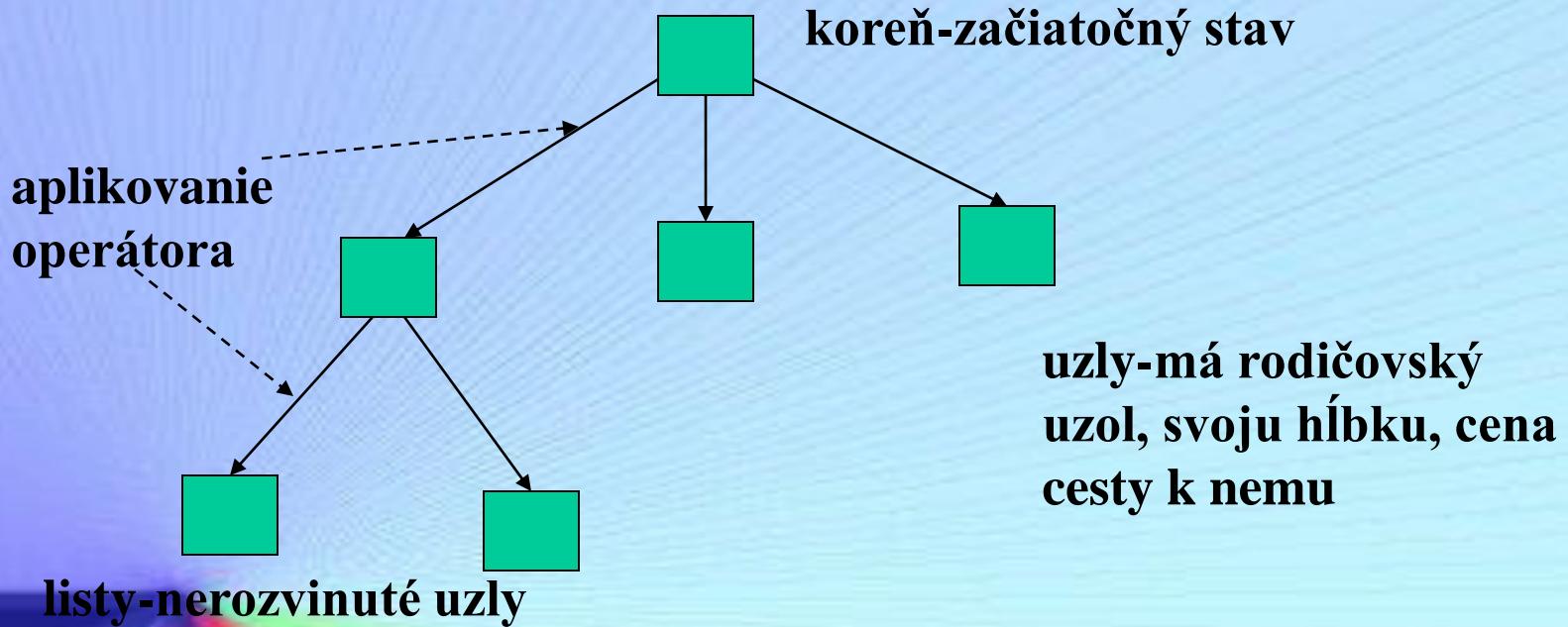
- a) základná procedúra slepého prehľadávania
- b) prehľadávanie do šírky
- c) prehľadávanie do hĺbky
- d) obojsmerné prehľadávanie
- e) nedeterministické prehľadávanie

# Základná procedúra:

- *hl'adanie v stavovom priestore  $\Rightarrow$  aplikovanie operátorov na stavy  $\Rightarrow$  rozvíjanie stavu:*
- **čiastočne** (*aplikujeme len jeden operátor*)
- **úplne** (*aplikujeme v jednom kroku všetky možné operátory*)
- **vytváranie stromu hl'adania (koreň, listy, uzly-údajové štruktúry; majú hĺbku a rodičov;)**

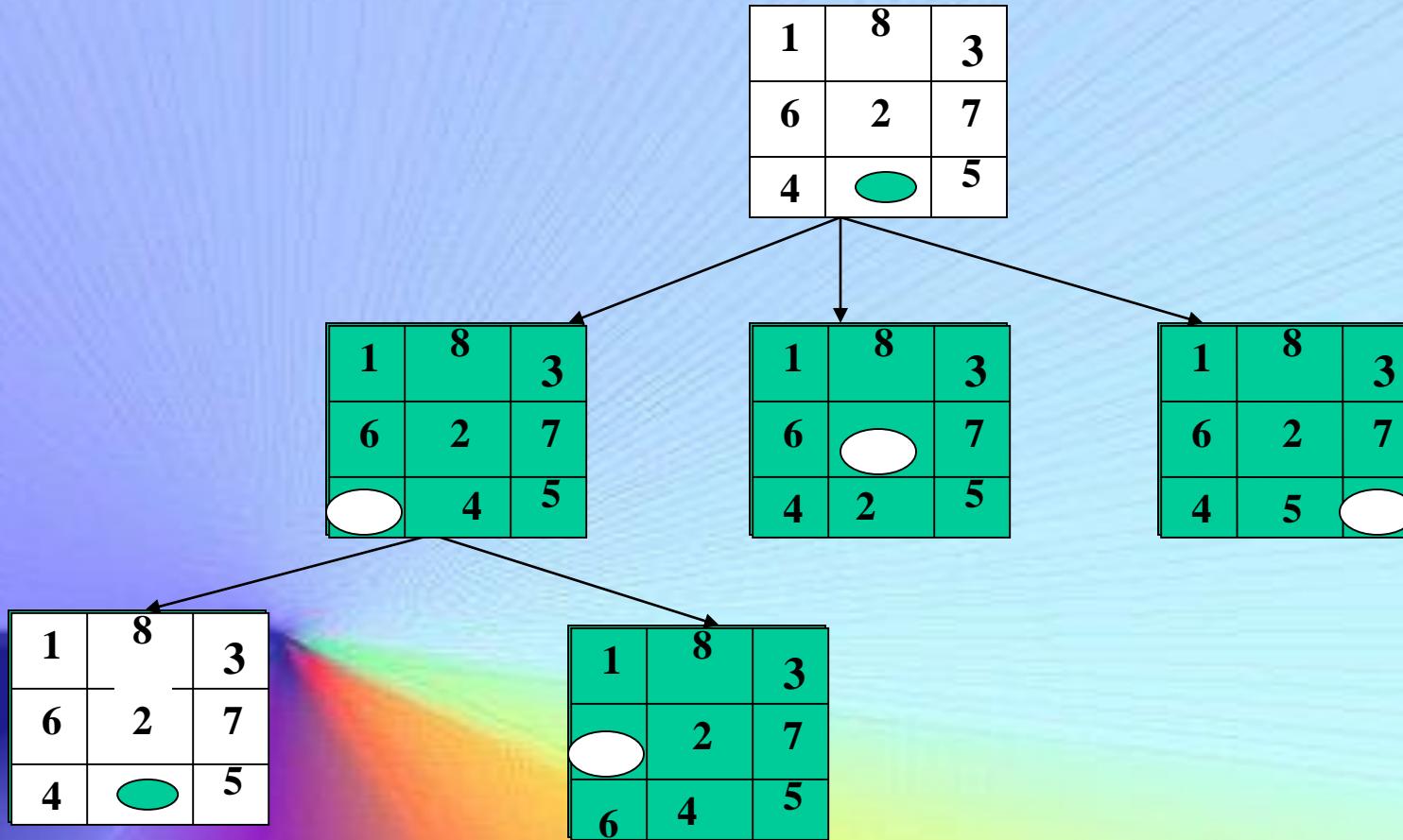
**POZOR:** rozdiel medzi reprezentáciou stavového priestoru grafom a stromom prehľadávania! Stavový priestor je často konečný, ale operátory nás môžu dovest' k nekonečnému množstvu ciest (procedúra).

# Základné pojmy



*Strom má v pamäti svoju reprezentáciu pomocou algoritmu prehľadávania. Je to len prehľadaná časť stavového priestoru. Uzly sú reprezentované štruktúrou-majú „rodičov“ a uzly nie. 2 uzly môžu reprezentovať ten istý stav.*

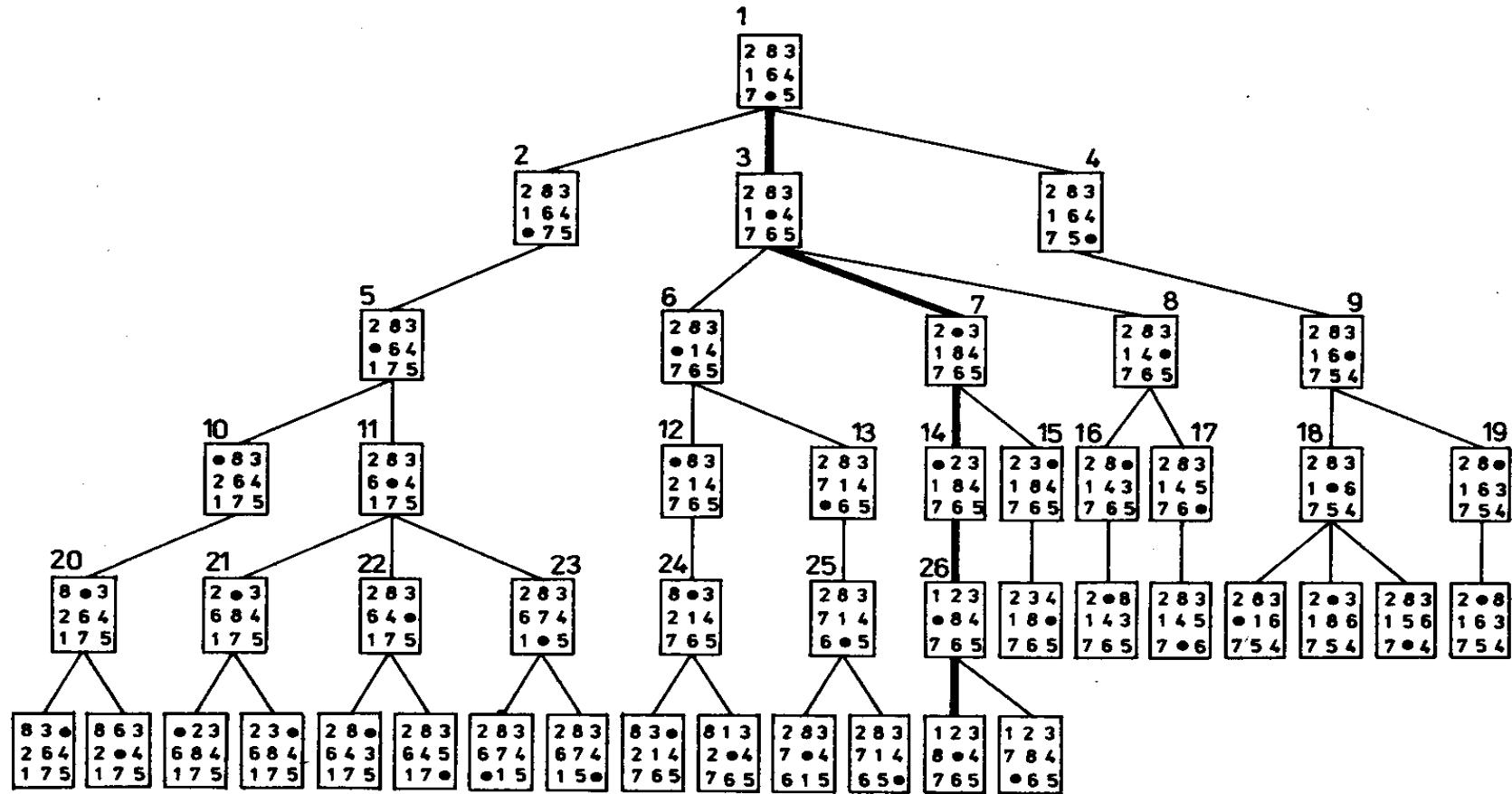
# *Ako možno vytvárať nekonečné množstvo uzlov?*



## *A. Prehl'adávanie do šírky.*

- *stavy sú usporiadané v štruktúre stromu*
- *stavy „rozvíjame“ operátorom*
- *nutnosť odstránenia cyklov*
- *generovanie stromu z  $s_0$  a kontrola cielového stavu*
- *základný princíp rozvijania uzlov: najprv rozvíjame stavy v hĺbke  $d$  a potom  $d+1$*
- *v procedúre musíme ukladať uzly v takom poradí ako ich rozvíjame operátorom (ukladáme ich do pamäte do frontu)*
- *používa sa princíp „zarad’ na koniec frontu“*
- *nájde najplytšie riešenie*
  - » vid’ obrázok (vysvetlený na prednáške)

# A. Prehľadávanie do šírky.



Obr. 20. Strom riešenia generovaný procedúrou prehľadávania do šírky

# *A. Prehľadávanie do šírky - efektivita*

- *Kritériá:*
  1. *požiadavky na pamäť-vel'ké*
  2. *časová zložitosť-vel'ká*
  3. *Prípustnosť-nie*
  4. *úplnosť -áno*
  5. *systematická stratégia -áno*

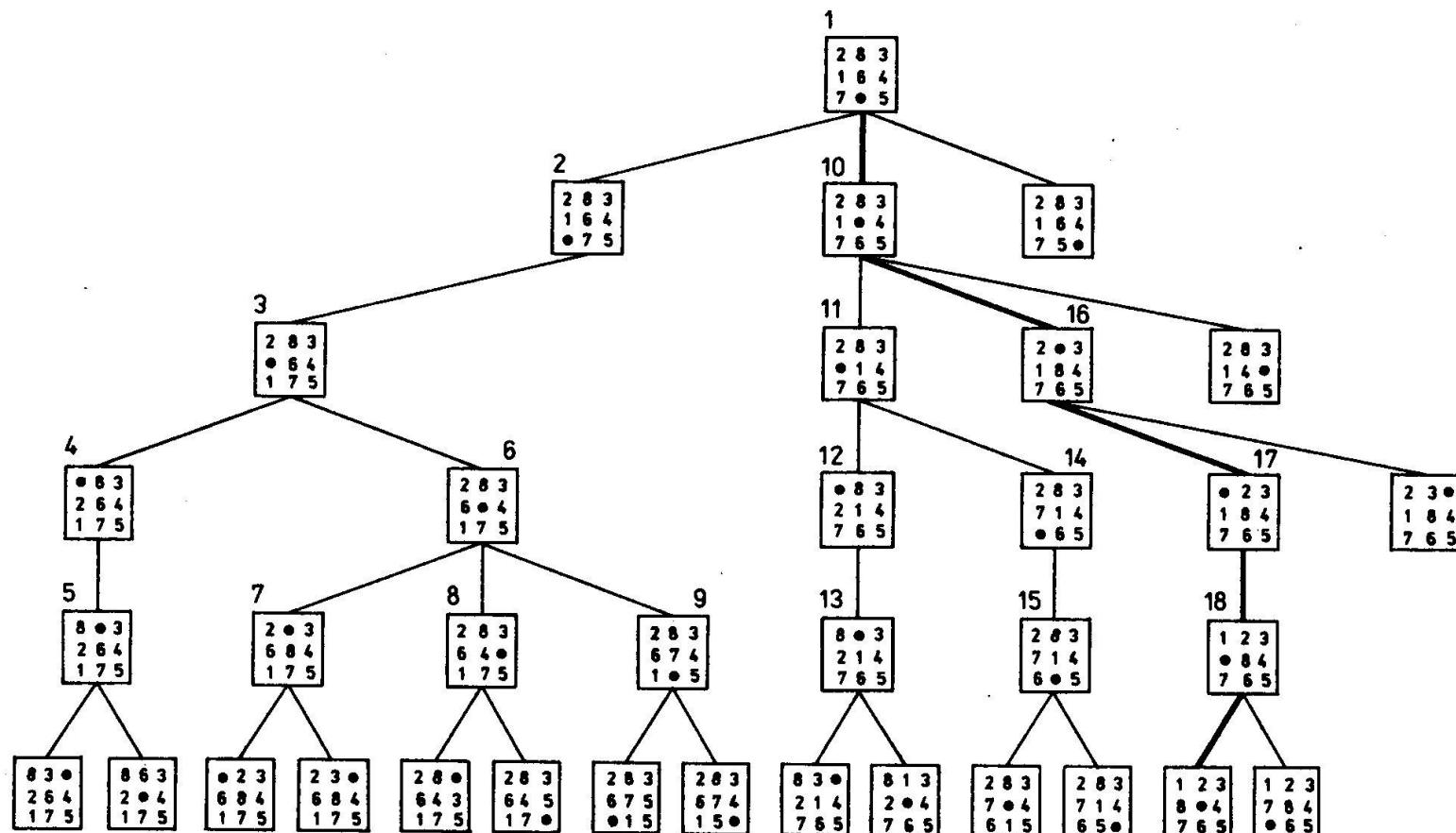
*A je systematická, je úplná , má vel'kú výpočtovú zložitosť (exponenciálna explózia), nie je prípustná*

*Modifikácia: stratégia rovnomernej ceny (optim. Úlohy-už ide o informovanú stratégiu) používa hodnotenie (oceňovanie) ciest medzi jednotlivými stavmi – logistické úlohy*

## *B. Prehl'adávanie do hĺbky.*

- *rozvíjanie uzlov smerom „dolu po ceste“, kým je to možné*
- *rozvíja sa uzol, ktorý bol posledný vygenerovaný*
- *organizácia ukladania do pamäte : zásobník (na vrch sa dáva posledne vygenerovaný a rozvíja sa ten na vrchu)*
- *nemáme však zaručenú konečnosť ani nájdenie riešenia (môžeme ist' po ceste na ktorej neleží riešenie)*
- *nutnosť obmedziť hĺbku, po ktorú hľadáme riešenia*
- *ohraničené hľadanie do hĺbky*
- *hľadanie do hĺbky s návratom (viď vysvetlenie prednáška)*
- *stratégia cyklicky sa prehlbujúceho hľadania*

# B. Prehľadávanie do hĺbky.



Obr. 19. Strom riešenia generovaný procedúrou prehľadávania do hĺbky

## *B. Prehl'adávanie do hĺbky- efektivita.*

- *pamäťová zložitosť je lineárna*
- *časová zložitosť je lepšia než u A*
- *nezaručuje nájdenie najoptimálnejšieho riešenia*
- *nezaručuje nájdenie riešenia! (nie je úplné)*
  - porovnanie efektivity: vid' prednáška
- **C. Obojsmerné hľadanie (komb. A+B)**
  - *riešením takýchto úloh je nájst' cestu tj. poznáme počiatočný stav a koncový stav*
  - *nutná podmienka použitia metódy: k operátorom musí existovať inverzny operátor*
  - *hľadáme objsmerne (z s0 dopredu a z st dozadu)*

# **Heuristické stratégie prehľadávania stavového priestoru**

- **Kvantitatívne heuristiky**
- **Kvalitatívne heuristiky**

# Kvantitatívne heuristiky.

- „najlepší, najsľubnejší stav“ podľa vlastností stavu, jeho kvality
- na ohodnotenie stavu (uzla) sa používa **vyhodnocujúca funkcia  $f(s)$** , ktorá všeobecne závisí od popisu konc. stavu, informácií dôležitých pre riešenie a od našich poznatkov
- jej definičným oborom je množina uzlov a po aplikácii vracia číslo ako ohodnotenie
- základný algoritmus (**stratégia na zoradenie nerozvinutých uzlov**) nekladie obmedzenia na typ vyhodnocujúcej funkcie a hľadá sa na rozvinutie stav s najnižším ohodnotením  $f(n)$
- ak máme dve cesty do jedného stavu (pozor graf nie strom!) vyberáme najlepšie ohodnotený, druhému nevenujeme pozornosť
- vytvárame tzv. **traverzný strom  $T$  implicitného grafu**  
**traverzný strom  $T$  – predstavuje problém, ktorý riešime pomocou algoritmu**

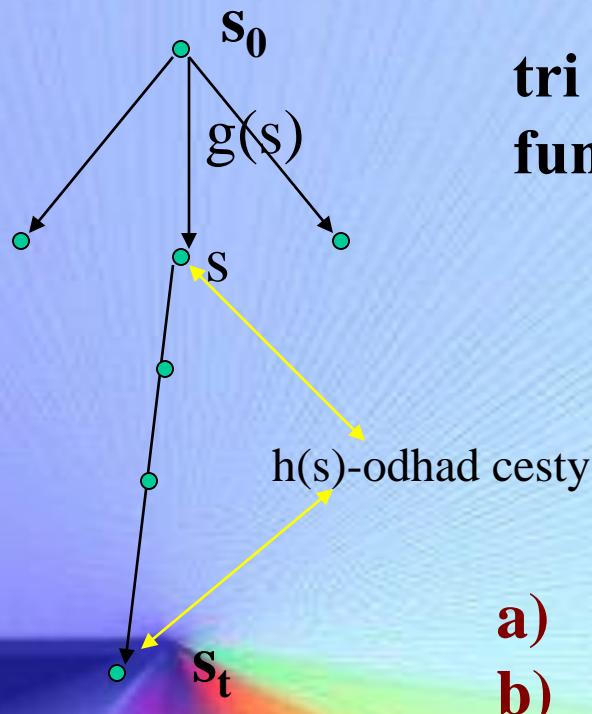
# Informované hľadanie (kvant.heuristiky).

- **heuristika** – „informačná“ funkcia, ktorá nám ohodnocuje cestu medzi vrcholmi nenájde vždy najlepší uzol, nešlo by o hľadanie, ale o kroky „naisto“, teda algoritmické riešenie bez hľadania
- **Heuristika** – vyjadruje odhad, vyberá sa uzol, ktorý sa zdá byť najlepší
- vezmieme funkciu  $f(s)$ , ktorej  $D(f)$  je množina stavov  $S$ , nech istým spôsobom ohodnocuje vrcholy (napr. podľa hĺbky, ceny ap.)
- potom algoritmus **hladania „najskôr najlepšieho“** hľadá najlepšie riešenie daného problému

## A\* a jeho modifikácie:

- ak funkciu  $f(s)$  položíme  $f(s)=g(s)+h(s)$ , kde
  - $g(s)$  je funkcia cesty, ktorá vedie od  $s_0$  k uzlu  $s$ , ktorý chceme rozvinúť
  - $h(s)$  je funkcia heuristická, robí odhad ceny cesty z uzla  $s$  do uzla cielového  $s_t$

# Informované hľadanie (kvantitatívne heuristiky).



tri prístupy podľa spôsobu výberu funkcií  $g(s)$ ,  $h(s)$ :

- a) algoritmus rovnomernej ceny
- b) algoritmus lačného hľadania
- c) algoritmus A\*

# Kvantitatívne heuristiky.

a) ak  $h(s) = 0$  potom

$f(s)=g(s)$  a ak  $g(s)$  položíme rovné napr.

jednotkovej cene hrán,  
resp. definujeme ako hĺbku príslušného uzla  
potom dostávame  
hl'adanie do šírky  
**stratégiu rovnomernej ceny**

b) ak použijem prípad, že  $h(s)$  položím ako odhad najlacnejšej cesty z  $s$  do konc. uzla  $s_t$  a chcem, aby som vybral uzol s najbližšie“ vrcholu  $s_t$ , potom od  $s_0$  vyberiem vrchol  $s$ , ktorý je „najbližšie“ k  $s_t$  a teda nehľadím na minimalizáciu  $g(s)$

- z grafu „odhryznem“ čo najväčšie sústo, aby som sa čo najrýchlejšie dostať k cieľu,
- teda  $h(s_t)=0$

# Kvantitatívne heuristiky A\*.

## c) vhodný na optimalizačné úlohy

- $f(s)=g(s)+h(s)$
- **f(s) je definovaná ako súčet ceny s minimálnou cenou uzla s od začiatočného uzla a ceny analogickej cesty z s do konc.uzla** tj. f(s) je ohodnotením ceny cesty s najnižšou cenou za predpokladu, že cesta prechádza cez uzol s
- takto definovaná f(s) umožňuje rozlíšiť bočné uzly od uzlov ležiacich na optimálnej ceste a stratégia negeneruje zbytočné bočné uzly
- problém je nájsť vhodnú heuristiku tzv. **prípustnú heuristiku**

# Kvantitatívne heuristiky A\*.

- prípustná heuristika je také  $h$ , ktoré nikdy nenadhodnotí cenu dosiahnutia cieľa z daného uzla (vždy posúdi celkovú cenu riešenia  $f(s)$  ako nižšiu nanajvýš rovnú reálnej)
- aby sme sa nedostali k a) ani k b) je nutné postaviť  $g(s)$  a  $h(s)$  vyvážene a teda použiť váhu  $w \in \langle 0,1 \rangle$  a definovať  $f(s)=w*g(s)+(1-w)*h(s)$
- ak sa s  $w \rightarrow 1$  potom A\* nadobúda charakter hľadania do šírky
- ak  $w \rightarrow 0$  potom  $h$  má väčší význam a dostaneme strategiu vhodnú pre problémy, ktoré majú veľa rovnako dostupných riešení
- niektoré heuristiky znížia rozsah hľadania, ale nemusia nájsť optimálne riešenia
- empiricky i teoreticky je dokázané, že pri optimalizácii algoritmus A\* dosahuje požadovanú rýchlosť

# Kvalitatívne heuristiky - systém GPS.

- Simon, Newell – programovanie ľudskej intuície, kvalitatívna heuristika – nedá sa vyjadriť kvantitatívne
- 1959 – systém GPS –General Problem Solver, ktorý mal dokázať riešiť všetky typy úloh
- prílišné očakávania: nedokáže hrať šach
- prenášanie „kvality“ z človeka do programu tým, že človek dokáže intuitívne definovať pojmy potrebné na ďalšiu prácu počítača
- **Čo musíme mať? Ako postupovať?**
- práca v stavovom priestore s operátormi, ktoré však vyjadrujú „možnosť konat“ a preto musia mať tri základné komponenty: transformačnú funkciu, podmienky aplikovateľnosti, zoznam diferencií

# Kvalitatívne heuristiky - systém GPS.

- **základ GPS:** analýza prostriedkov a cieľov
- **princíp:** vytváranie tzv. diferencií stavov a ich postupná redukcia, kým neprídeme k cieľu (k nulovej differencii)
- **diferencia:** vlastnosť, ktorá odlišuje dva stavy – nie je merateľná, ide o kvalitatívnu differenciu
- **Čo musíme mať?**
  1. definované stavy (človek rozozná stavy)
  2. *operátory – možnosti konat'*
  3. uvedomenie si problému ( $s_t, f_t$ ) (problém nastáva ak je situácia iná ako by mala byť')
  4. *diferencie medzi stavmi (rozdiely) dif  $\subseteq S \times S$*
  5. *tabuľka diferencií a operátorov, ktorá určuje priority ako odstrániť diferencie*
  6. *lineárne usporiadanie diferencií, aby som vedel ktorú differenciu prioritne redukovať'*

# Kvalitatívne heuristiky - systém GPS.

- nejednoznačnosti v GPS:
  - akú diferenciu vyberieme ako prvú? Odstránenie: vytvorenie lusp(intuícia)
  - aký operátor vyberieme? Na výber môže byť viac „kandidátov“. Pomoc: princíp maximálneho zachovania už dosiahnutého (intuícia)

**Príklad:** robotický systém STRIPS (Stanford)

**Literatúra:** Kelemen, Základy UI, str.159-178