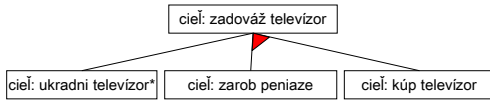


## Riešenie problémov rozkladom



A-ALEBO graf

\*hanba!

1

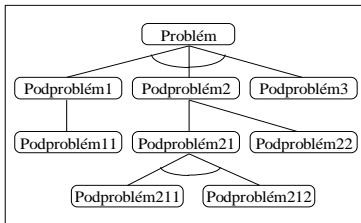
## Riešenie problémov rozkladom

### Rozložiteľný problém

- reprezentácia problémov a ich riešení:
  - rozklad problému na podproblémy
  - priestor rozkladov: konjunkcia podproblémov

2

## Riešenie problémov rozkladom – A / ALEBO graf



- hrany ALEBO - spájajú uzol s alternatívami riešenia.
- hrany A - spájajú uzol s uzlami, ktoré spolu predstavujú riešenie

3

## Riešenie problémov rozkladom – A / ALEBO graf

- Problém je v podstate daný
  - začiatočným stavom a
  - množinou rozkladajúcich operátorov.
- Riešenie v A/ALEBO grafe
  - jeho podgraf a nie cesta

4

## Riešenie problémov rozkladom

```

function RIEŠENIE-ROZKLADOM(problém)
    returns graf_riešenia alebo neúspech
    if problém je elementárny then
        if problém je riešiteľný then
            return VYTVOR-GRAF-RIEŠENIA(problém)
        else return neúspech
    loop do
        rozlož problém na podproblémy
        if RIEŠENIE-ROZKLADOM je pre všetky podproblémy je úspešné then
            return VYTVOR-GRAF-RIEŠENIA(problém,
                množina grafov riešenia pre podproblémy)
    end
    
```

5

## Proces značkovania uzlov v grafe priestoru rozkladov

- Uzly značujeme ako "vyriešené" takto:
  - každý koncový uzol, predstavujúci elementárny problém sa označí za "vyriešený";
  - každý nie koncový uzol, z ktorého vychádzajú hrany ALEBO a aspoň jeden z jeho nasledovníkov je označený ako "vyriešený" sa označí za "vyriešený";
  - každý nie koncový uzol, z ktorého vychádzajú hrany A a všetky jeho nasledovníky sú označené ako "vyriešený" sa označí za "vyriešený".

6

## Proces značkovania uzlov v grafe priestoru rozkladov

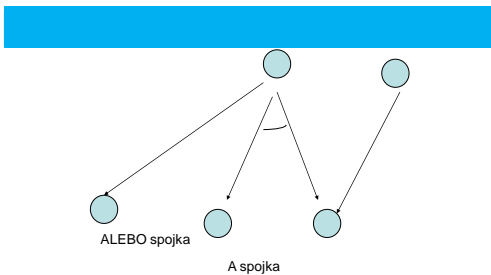
- Uzly sa značkujú ako **"neriešiteľné"** takto:
  - úlohy hrán A a ALEBO sa vymenia.
  - Tiež každý uzol, ktorý nereprezentuje elementárny problém a nemá žiadnych nasledovníkov sa označí za "neriešiteľný".
- Cieľový test**
  - testovanie, či je začiatkový stav "vyriešený" alebo "neriešiteľný".

7

## A/O\* algoritmus

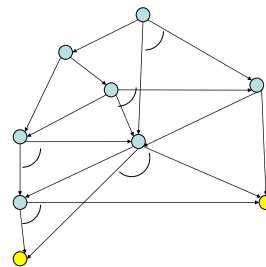
- dátová štruktúra
  - graf
  - označované spojenia (hrany alebo dvojice hrán) (orientované značky; nie ako A\*)
  - ceny  $q()$  sa udržiavajú na uzloch
  - označenie vyriešené
- obmedzíme sa na acyklické grafy

8



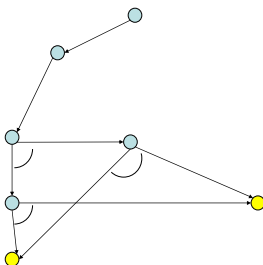
9

## A/ALEBO graf



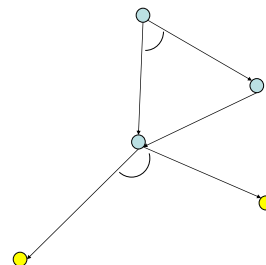
10

## podgraf riešenia



11

## iný podgraf riešenia



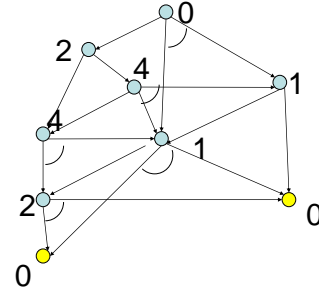
12

### podgraf riešenia $G'$ A/ALEBO grafu $G$

- z uzla  $n$  do koncových (vyriešených) uzlov  $T$ :
- ak  $n$  je v  $T$ ,  $G'$  je len uzol  $n$ .
- inak  $n$  má jednu spojku do množiny uzlov  $n_1, n_2, \dots, n_k$ .
  - pre každý uzol  $n_i$  jestvuje graf riešenia z neho do  $T$ .
  - $G'$  je  $n$ , tá spojka, uzly  $n_1, n_2, \dots, n_k$
  - a k tomu grafy riešenia z každého uzla  $n_i$ .

13

### heuristické hodnoty: odhad ceny do množiny vyriešených



14

### obmedzenie monotónnosti

- $h(n) \leq c + h(n_1) + h(n_2) + \dots + h(n_k)$

kde  $c$  je cena spojky medzi  $n$  a množinou  $n_1, n_2, \dots, n_k$ .

toto obmedzenie zaručuje, že  $h(n) \leq h^*(n)$ .

15

### hodnoty ceny ( $q(n)$ )

- ak  $n$  nemá nasledovníky, tak  $q(n) = h(n)$

- inak počítajúc hodnoty odspodu,
  - $q(n) = \text{cena spojky} + \text{súčet odhadnutí } q(\text{nasledovníkov})$

z odhadnutí  $q(n)$  uzla  $n$  vyber najmenšie a označ smer spojenia.

ak v označenom smere sú všetky nasledovníky *vyriešené* tak označ *vyriešený*.

16

### základná myšlienka A/O\*

- najprv sa prechádza graf zhora nadol tak, že sa vyberajú najlepšie čiastočné podgrafy riešenia.
- rozvinie sa jeden listový uzol tohto grafu
- potom sa prechádza zdola nahor, upravujú sa odhadnutia, podľa toho smerovania a značuje sa *vyriešené*.

17

### AO\* algoritmus

1. vytvor  $G = \langle s \rangle$ ;  $q(s) = h(s)$   
ak  $s \in \text{TERM}$  tak označ  $s$  *vyriešený*
2. kým nie je  $s$  *vyriešený*:
  1. vypočítaj  $G'$  podgraf čiastočného riešenia sledovaním označených spojok v  $G$  z  $s$ .
  2. vyber  $n$  v  $G'$ ,  $n$  nie je v  $\text{TERM}$ ,  $n$  je list.
  3. rozviň  $n$ , pridaj jeho nasledovníky do  $G$  a pre každý nasledovník, ktorý nie je ešte v  $G$ , priradiť  $q(\text{nasledovník}) = h(\text{nasledovník})$ .  
Označ *vyriešené* všetky nasledovníky v  $\text{TERM}$ . (ak nie sú nasledovníky tak nastav  $q(n) := \infty$ ).

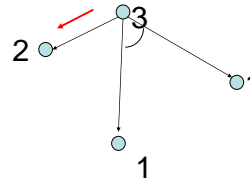
18

## AO\* algoritmus

4. priradiť  $S := \{n\}$ .
5. pokiaľ nie je  $S = \emptyset$  :
  1. odstrániť uzol  $m$  z množiny  $S$  taký, že žiadny z jeho potomkov (nielen priamych nasledovníkov) v grafe  $G$  nie je aj v množine  $S$ . ak taký uzol neexistuje v  $S$ , odstrániť z  $S$  ľubovoľný uzol.
  2. vypočítať cenu každej spojky vychádzajúcej z  $m$ . oprav cenu uzla  $m$ :  
 $q(m) = \min [c + q(n_1) + \dots + q(n_k)]$ .  
 označ zvolenú spojku.  
 ak všetky nasledovníky spojené práve označenou spojkou sú *vyriešené* tak označ  $m$  *vyriešený*.
  3. ak bol práve označený vyriešený alebo sa práve menilo ohodnotenie  $q(m)$  tak pridať do  $S$  všetkých "spojených" predchodcov uzla  $m$ .
  4. koniec
6. koniec

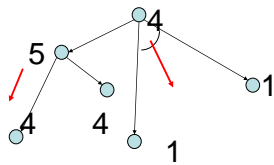
19

## stopa algoritmu



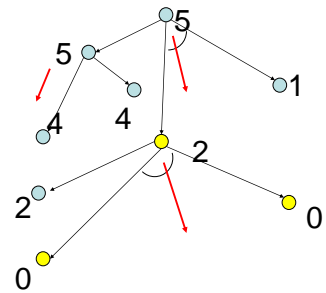
20

## stopa algoritmu



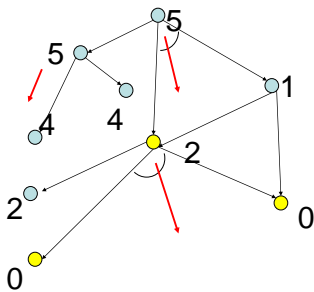
21

## stopa algoritmu



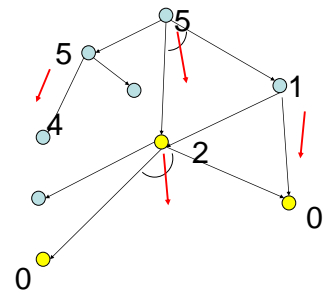
22

## stopa algoritmu



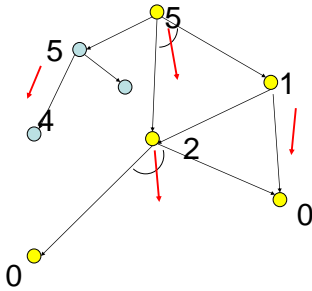
23

## stopa algoritmu



24

## stopa algoritmu



25

## optimálnosť of A/O\*?

- ak  $h(n) \leq c + h(n_1) + \dots + h(n_k)$   
pre všetky uzly  $n$  a spojky  
tak cena riešenia je optimálna.

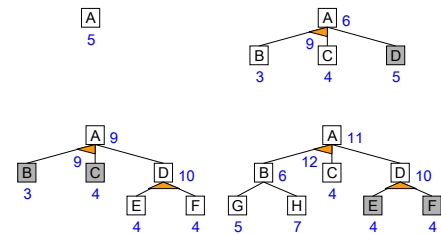
26

## poznámky

- Acykličnosť grafu znamená, že množina  $S$  sa nakoniec vyprázdni.
- Ako vybrať ne-elementárny uzol v grafe čiastočného riešenia na rozvítie?  
– najlepšia voľba je uzol s najvyšším ohodnotením  $h^*$  (aj tak sa bude musieť rozvinúť)

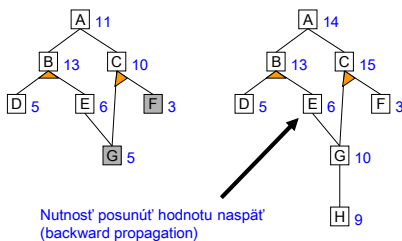
27

## Riešenie problémov rozkladom : AO\*



28

## Riešenie problémov rozkladom: AO\*



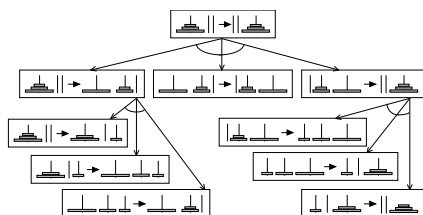
29

## Problém Hanojskej veže

- Problém Hanojskej veže možno opísať takto:
- **stavy**: umiestnenie troch diskov na troch stojanoch tak, že disk s menším priemerom leží vždy na disku s väčším priemerom;
- **začiatkový stav**: všetky disky na prvom (ľavom) stojane tak, že disk s menším priemerom leží vždy na disku s väčším priemerom;
- **operátory**: presun disku na iný stojan, pričom nesmie nikdy vzniknúť situácia, aby na nie-ktorom stojane ležal väčší disk na menšom;
- **cieľový test**: všetky disky na poslednom (pravom) stojane tak, že disk s menším priemerom leží vždy na disku s väčším priemerom;
- **cena cesty**: každá aplikácia operátora má cenu 1.

30

## Problém Hanojskej veže



31