Omezující podmínky pro rozvrhování

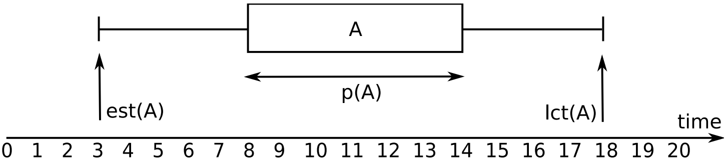
14. března 2023

Rozvrhování jako CSP: čas

Aktivita A: entita vyžadující prostor (zdroje) a čas

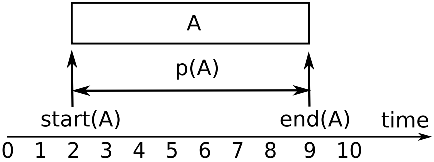
Proměnné a jejich domény pro časové přiřazení aktivity

- start(A): startovní čas aktivity
  - aktivita nemůže začít před svým termínem dostupnosti
  - est(A) = min(start(A)), earliest start time/nejdřívější startovní čas
- end(A): čas skončení aktivity
  - aktivita musí skončit před svým deadline
  - lct(A) = max(end(A)), latest completion time/nejpozdější koncový čas
- p(A): doba provádění aktivity
  - start(A) = {est(A), . . . , (lct(A)-p(A))}
  - end(A) = {(est(A)+p(A)), . . . , lct(A)}

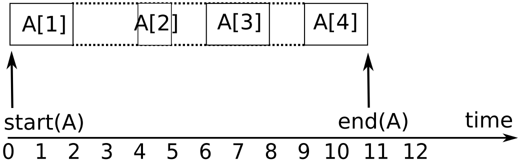


Rozvrhování jako CSP: základní omezení I.

- Nepřerušitelná aktivita: žádné přerušení během jejího zpracování
  - start(A) + p(A) = end(A)



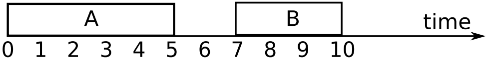
- Přerušitelná aktivita: může být přerušena během svého zpracování
  - start(A) + p(A) $\leq$ end(A)



p(A) = p(A[1]) + p(A[2]) + p(A[3]) + p(A[4])

Rozvrhování jako CSP: základní omezení II.

- Seřazení A<<B aktivity A,B
  (také: precedenční omezení mezi aktivitami A,B)
    - end(A) $\leq$ start(B)



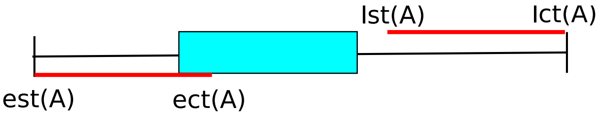
- Disjunktivní omezení: nepřekrývání aktivit A a B
    - nepřerušitelné aktivity
    - A<<B or B<<A
    - end(A) $\leq$ start(B) or end(B) $\leq$ start(A)
    - viz dále unární zdroje

Rozvrhování jako CSP: zdroje

Doménové proměnné pro zdroje:

- cap(A): požadovaná kapacita zdroje aktivitou A
  - unární/disjunktivní zdroje
    - v daném čase může být zpracována maximálně jedna aktivita
  - kumulativní zdroje
    - několik aktivit se může zpracovávat paralelně, ovšem za předpokladu, že není překročena kapacita zdroje
  - produkovatelné/spotřebovatelné zdroje
    - aktivita konzumuje (snižuje) nebo produkuje (navyšuje) aktuální množství zdroje
    - na zdroji musí zbýt nějaká minimální volná kapacita a maximální kapacita zdroje nemůže být překročena
    - příklad: reservoár
- resource(A): alternativní zdroje pro A
  - pro zpracování A vybereme jeden z alternativních zdrojů
  - jednotlivé hodnoty z domény resource(A) odkazují na konkrétní zdroje

Značení

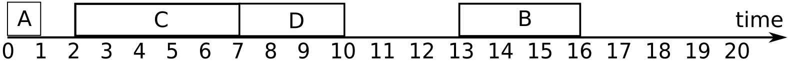


- est(A) nejdřívější startovní čas aktivity A (earliest start time)
- ect(A) nejdřívější koncový čas aktivity A (earliest completion time)
- lst(A) nejpozdější startovní čas aktivity A (latest start time)
- lct(A) nejpozdější koncový čas aktivity A (latest completion time)

- Ω je množina aktivit
- $p(\Omega) = \sum_{A \in \Omega} p(A)$
- $est(\Omega) = \min\{est(A) \mid A \in \Omega\}$
- $lct(\Omega) = \max\{lct(A) \mid A \in \Omega\}$

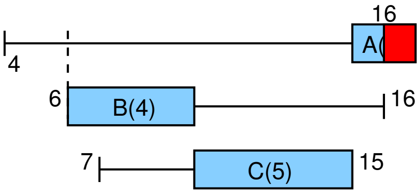
Unární zdroje

- Aktivivity se nemohou překrývat
  - v daném čase běží maximálně jedna aktivita, proto se těmto zdrojům říká unární
  - Grahamova klasifikace: jeden stroj
- Předpokládáme, že aktivity jsou nepřerušitelné
  - nepřerušitelná aktivita zabírá zdroj od svého startu až do ukončení
- Jednoduchý model s disjunktními omezeními
  - A<<B ∨ B<<A
    end(A)≤start(B) ∨ end(B)≤start(A)
  - těmto zdrojům se proto někdy říká disjunktní

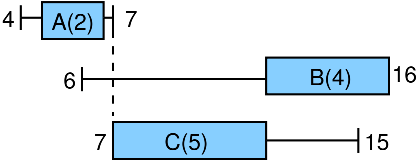


Hledání hran *(edge finding)*

- Baptiste & Le Pape (1996)
- Co se stane, pokud nebude aktivita A zpracována jako první?

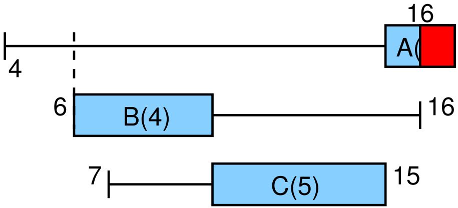


- Pro A,B,C není dost času, a tedy aktivita A musí být první



Hledání hran: příklad s odvozovacími pravidly

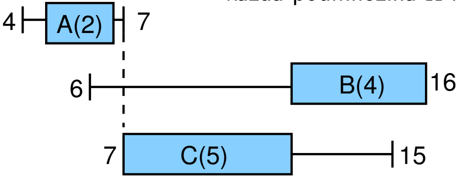
- $lct(\Omega \cup \{A\}) - est(\Omega) < p(\Omega \cup \{A\}) \Rightarrow A << \Omega$



$\Omega = \{B, C\}$

- $A << \Omega \Rightarrow end(A) \leq \min\{lct(\Omega') - p(\Omega') \,|\, \Omega' \subseteq \Omega\}$

každá podmnožina Ω musí stihnout své zpracování

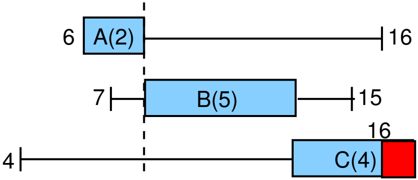


Hledání hran: všechna odvozovací pravidla

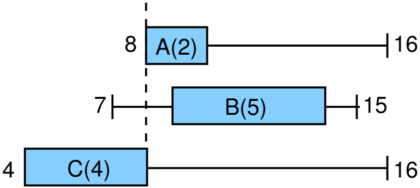
- Zopakujme tedy odvozovací pravidla: pro $end(A)$
  (co se stane, pokud nebude aktivita A zpracována jako první?)
  - $lct(\Omega \cup \{A\}) - est(\Omega) < p(\Omega \cup \{A\})$
    $\Rightarrow A << \Omega$
  - $A << \Omega \Rightarrow$
    $end(A) \leq \min\{lct(\Omega') - p(\Omega') \mid \Omega' \subseteq \Omega\}$
- Symetrická odvozovací pravidla: pro $start(A)$
  (co se stane, pokud nebude aktivita A zpracována jako poslední?)
  - $lct(\Omega) - est(\Omega \cup \{A\}) < p(\Omega \cup \{A\})$
    $\Rightarrow \Omega << A$
  - $\Omega << A \Rightarrow$
    $start(A) \geq \max\{est(\Omega') + p(\Omega') \mid \Omega' \subseteq \Omega\}$
- V praxi:
  - celkem existuje $n \cdot 2^n$ párů A,Ω (příliš mnoho!)
  - Carlier & Pinson 1994, Vilím & Barták & Čepek 2004
    algoritmus s časovou složitostí $O(n \log n)$
    (je nutné kontrolovat pouze některé páry A,Ω)

Ne-první/ne-poslední *(not-first/not-last)*

- Torres & Lopez 2000
- Co se stane, pokud aktivita A bude zpracována jako první?

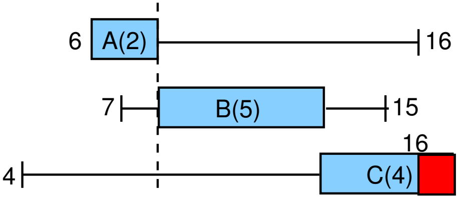


- Pro B a C není dost času, a tedy aktivita A nemůže být první



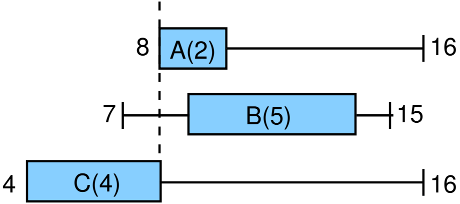
Ne-první/ne-poslední: př. s odvozovacími pravidly

- $p(\Omega \cup \{A\}) > lct(\Omega) - est(A) \Rightarrow \neg A << \Omega$



$\Omega = \{B, C\}$

- $\neg A << \Omega \Rightarrow start(A) \geq \min\{ect(B)|B \in \Omega\}$



Odvozovací pravidla pro ne-první/ne-poslední

- Zopakujme Ne-první pravidla:
  (co se stane, pokud aktivita A bude zpracována jako první?)
  - $lct(\Omega) - est(A) < p(\Omega \cup \{A\})$
    $\Rightarrow \neg A << \Omega$
  - $\neg A << \Omega$
    $\Rightarrow start(A) \geq \min\{ect(B) | B \in \Omega\}$

- Ne-poslední (symetrická) pravidla:
  (co se stane, pokud aktivita A bude zpracována jako poslední?)
  - $lct(A) - est(\Omega) < p(\Omega \cup \{A\})$
    $\Rightarrow \neg\Omega << A$
  - $\neg\Omega << A \Rightarrow$
    $end(A) \leq \max\{lst(B) | B \in \Omega\}$

- V praxi:
  - Vilím 2004: algoritmus s časovou složitostí $O(n \log n)$

Alternativní zdroje

Jak modelovat alternativní zdroje pro danou aktivitu?

Pro každý zdroj uděláme duplikát aktivity

- duplikát se účastní příslušných zdrojových podmínek,
  ale neomezuje další aktivity na daném zdroji
  - „neúspěch" u duplikátu znamená odstranění zdroje z domény proměnné resource(A) příslušné aktivity
  - odstranění zdroje z domény proměnné resource(A) znamená „smazání" odpovídajícího duplikátu
- původní aktivita se účastní precedenčních podmínek
  (např. v rámci multi-operační úlohy)
- omezení časů u duplikátu se propaguje do originálu aktivity a naopak

Alternativní zdroje: odvozovací pravidla

Nechť A_u reprezentuje duplikát aktivity A na zdroji u$\in$resource(A), pak probíhají následující propagace:

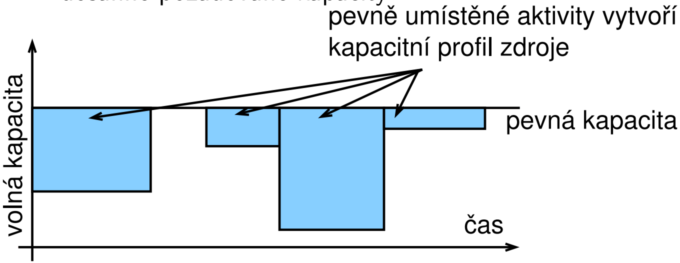
- u$\in$resource(A) $\Rightarrow$ start(A) $\leq$ start(A_u)
- u$\in$resource(A) $\Rightarrow$ end(A_u) $\leq$ end(A)

- start(A) $\geq$ min{start(A_u): u $\in$ resource(A)}
- end(A) $\leq$ max{end(A_u): u $\in$ resource(A)}

- neúspěch pro A_u $\Rightarrow$ resource(A)$\setminus${u}

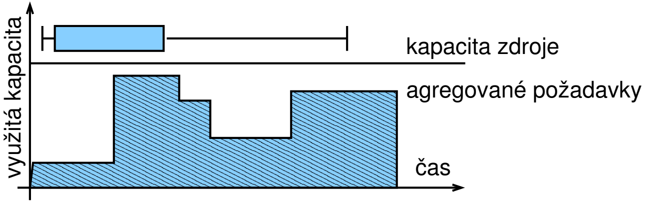
Kumulativní zdroje

- Každá aktivita využívá jistou kapacitu zdroje cap(A)
- Aktivity mohou být zpracovány paralelně, pokud není překročena kapacita zdroje
- Kapacita zdroje může být v čase proměnná
  - takové zdroje lze modelovat pomocí v čase neměnné kapacity, od které se odečte kapacita pevně umístěných aktivit, čímž se v každém čase dosáhne požadované kapacity

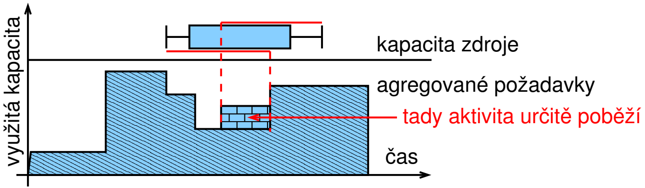


Agregované požadavky

- Baptiste et al. 2001
- Kdy je dostatečná kapacita pro zpracování aktivity?



- Jak se konstruuje graf agregovaných požadavků?



Podmínka tabulky *(timetable constraint)*

- Uvažujeme diskrétní čas
- Jak zajistit, že v žádném čase není překročena maximální kapacita?

$$\forall t \sum_{start(A_i) \leq t \leq end(A_i)} cap(A_i) \leq MaxCapacity$$

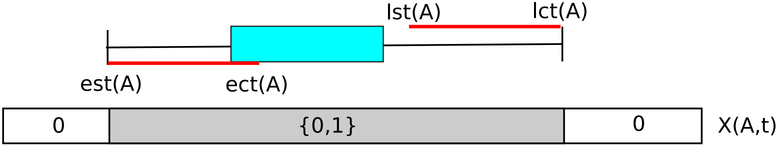
- Tabulka *(timetable)* pro aktivitu A je množina boolovských proměnných $X(A,t)$ udávajících, zda A běží v čase t

$$\forall t \sum_{A_i} X(A_i,t) cap(A_i) \leq MaxCapacity \qquad (*)$$

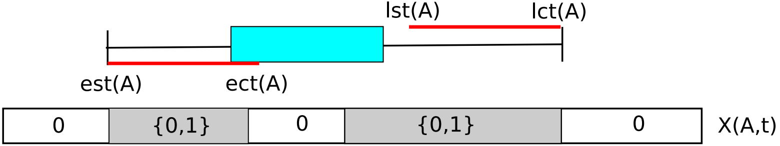
$$\forall t, i \; start(A_i) \leq t \leq end(A_i) \Leftrightarrow X(A_i,t)$$

Podmínka tabulky: př. s odvozovacími pravidly

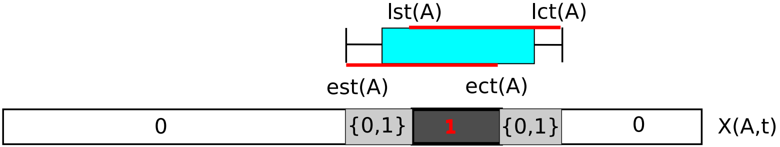
Počáteční stav



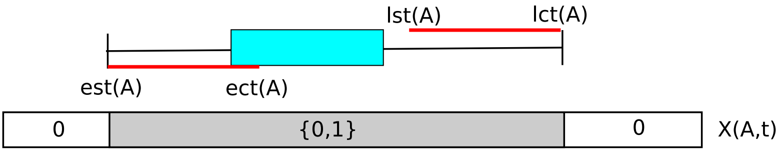
Některé pozice zakázány vzhledem ke kapacitě zdroje



Nový stav



Podmínka tabulky: odvozovací pravidla



Jak realizovat filtraci přes omezení

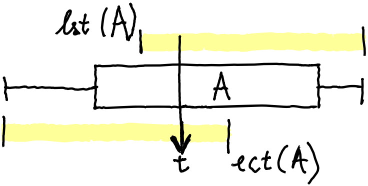
$$\forall t, i \;\; start(A_i) \leq t < end(A_i) \Leftrightarrow X(A_i, t) \;\; ?$$

Problém: t je zároveň index a také proměnná

- start(A) $\geq$ min{t | 1 $\in$ X(A,t)}
- end(A) $\leq$ 1+max{t | 1 $\in$ X(A,t)}

- X(A,t)=0 $\wedge$ t < ect(A) $\Rightarrow$ start(A)>t
- X(A,t)=0 $\wedge$ lst(A)$\leq$t $\Rightarrow$ end(A)$\leq$t

- lst(A)$\leq$t $\wedge$ t < ect(A) $\Rightarrow$ X(A,t)=1



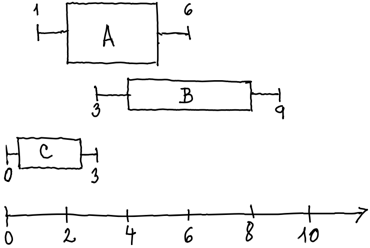
Cvičení: podmínka tabulky

Máme zadány zdroj s kapacitou 2 a aktivity

j	cap(j)	est(j)	lct(j)	p(j)
A	2	1	6	3
B	1	3	9	4
C	1	0	3	2

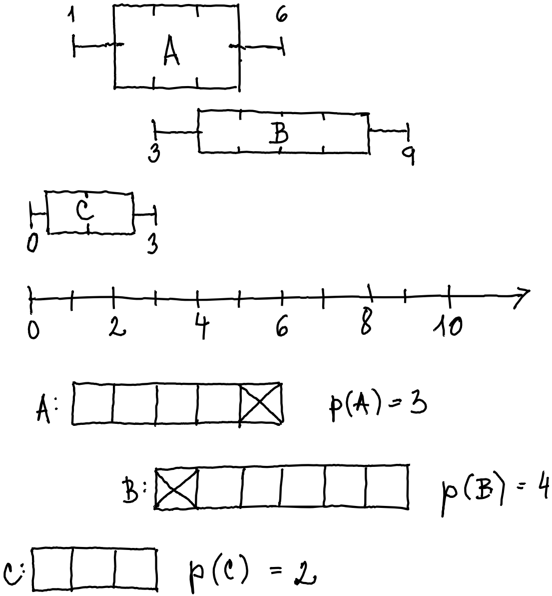
1. Jak jsou inicializovány proměnné X(j,t)?
2. Jak se jejich hodnoty mění při použití odvozovacích pravidel podmínky tabulky?
3. Jak by mohly vypadat výsledné rozvrhy po aplikaci pravidel?

Cvičení: podmínka tabulky

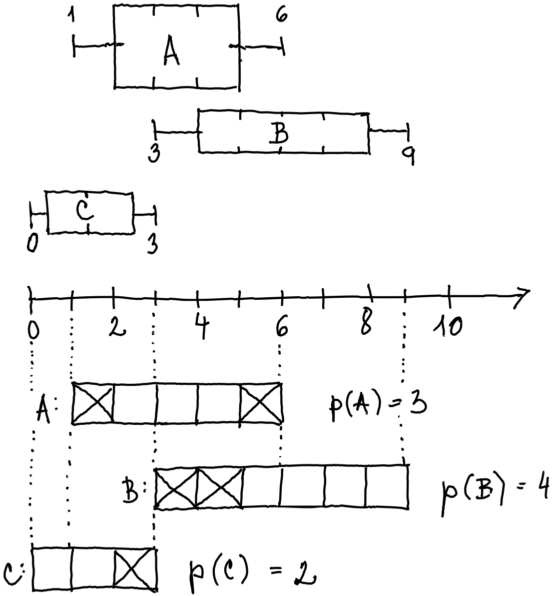


1. Jak jsou inicializovány proměnné X(j,t)?
   - $X(A,1)$ až $X(A,5)$ jsou $\{0,1\}$, $X(B,3)$ až $X(B,8)$ jsou $\{0,1\}$, $X(C,0)$ až $X(C,2)$ jsou $\{0,1\}$, ostatní proměnné nulové
2. Jak se jejich hodnoty mění při použití odvozovacích pravidel podmínky tabulky?
   1. dle $(*)$ B může začít nejdříve v čase 4 kvůli A, tj. $X(B,3) = 0$ a A musí být před B, tj. A nejpozději skončí v čase 5 a $X(A,5) = 0$
   2. dále z $(*)$ $X(C,2) = 0$, C začne v čase 0
      $X(A,1) = 0$ a A začne v čase 2 a také
      B musí začít až v čase 5 a $X(B,4) = 0$
      a máme jediné řešení

Cvičení: podmínka tabulky



Cvičení: podmínka tabulky



Unární a kumulativní zdroje: shrnutí a možná rozšíření

Disjunktivní omezení

- známe: unarní zdroje, nepřerušitelné aktivity
- rozšíření: přerušitelné aktivity, kumulativní zdroje

Hledání hran

- známe: unarní zdroje, nepřerušitelné aktivity
- rozšíření: přerušitelné aktivity, kumulativní zdroje

Ne-první/ne-poslední

- známe: unarní zdroje, nepřerušitelné aktivity
- rozšíření: kumulativní zdroje

Podmínka tabulky

- známe: kumulativní zdroje, nepřerušitelné aktivity
- rozšíření: přerušitelné aktivity