Příjmení a jméno: Datum: 20. prosince 2012
Login studenta:

Zápočtová písemka z předmětu „Automatizované řešení úloh s omezeními“

Doba trvání: **90 minut** Max. zisk: **40 bodů**

Příklad [1] (8 bodů): Uvažujme následující problém:

- Hlavní sestra přiděluje směny na 3 vánoční dny.
- Na každý den musí přiřadit jednu sestru sálovou a jednu pro péči o pacienty na pokojích.
- Žádná sestra nebude o Vánocích sloužit vícekrát.
- K dispozici má Annu, Danu, Janu, Marii, Petru a Zoru.
- Petra a Zora nemají dostatečnou kvalifikaci na sálovou sestru.
- Annu a Danu si chirurgové vyžádali, aby určitě sloužily na sále.
- Marie má slíbené volno na štědrý den, Anna na 1. svátek vánoční.
- Kamarádky Jana a Marie budou sloužit spolu.
- Anna se nesnese s Petrou, takže spolu sloužit nebudou.

Popište pomocí sítě omezení problém, který se hlavní sestra snaží vyřešit. Nakreslete graf (nebo hypergraf) omezení odpovídající této síti.

Řešení: Síť omezení je dána množinou proměnných, množinou domén a množinou omezení (v případě relačních se jedná o relace, kde prvky relací jsou povolené kombinace).

Proměnné a příslušné domény je možné zvolit různě, nabízí se proměnná na každou směnu a domény obsahují příslušné sestry nebo naopak proměnná pro každou sestru a v doménách by byly směny. Pro první případ by řešení vypadalo takto:

$X = \{S_1, S_2, S_3, P_1, P_2, P_3\}$

$D = \{D_{S_1}, D_{S_2}, D_{S_3}, D_{P_1}, D_{P_2}, D_{P_3}\}$

Proměnné S jsou pro směny na sále, P na pokoji, index určuje den.

Unární omezení (setra nemůže sloužit v některý den, musí/nesmí dělat určitý druh směny apod.) je možno promítnout do domény nebo udělat jako unární relaci (doména směny by obsahovala všechny sestry a v omezeních by byla relace s rozsahem na tuto směnu, která by obsahovala jen povolenou podmnožinu sester). Zde zakomponujeme unární omezení do domén. Tedy (sestry označujeme počátečním písmenem): $D_{S_1} = \{A, D, J\}$, $D_{S_2} = \{D, J, M\}$, $D_{S_3} = \{A, D, J, M\}$, $D_{P_1} = \{J, P, Z\}$, $D_{P_2} = \{J, M, P, Z\}$, $D_{P_3} = \{J, M, P, Z\}$

Dále musíme popsat množinu omezení C. V ní budou třeba binární omezení, omezující, která sestra se kterou můžou a nesmí sloužit apod.

$R_{S_1,P_1} = R_{S_2,P_2} = R_{S_3,P_3} = \{(J,M),(M,J),(A,D),(D,A),(A,Z),(Z,A),(D,P),(P,D),(D,Z),(Z,D),(P,Z),(Z,P)\}$

J a M mají sloužit spolu, proto se v kombinaci s jinou sestrou nevyskytují, ostatní mají povolené všechny vzájemné kombinace mimo A s P. Třeba kombinace (A,D) vlastně také není povolená (obě mají sloužit na sále, čili nemůžou sloužit ve stejný den). Ale toto už je odvozené omezení, které by vzniklo spojováním omezení (spojením binárního s unárním resp. s doménou). Stejně tak nemusíme uvažovat různá binární omezení pro různé dny - povolené dvojice zůstavají stejné a že se v některý den díky omezení určité sestry nemůže určitá dvojice vyskytnout by se odvodilo skládáním omezení.

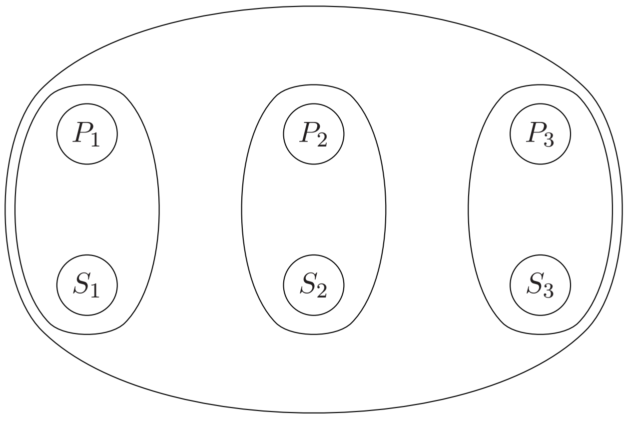
Poslední omezení, které se vyskytuje, je typu all-different. Pokud jej chceme zachytit relací, mohlo by to být např. takto

$R_{S_1,S_2,S_3,P_1,P_2,P_3} = \{(x_1,x_2,x_3,x_4,x_5,x_6) \mid x_i \in \{A,D,J,M,P,Z\}, x_i \neq x_j \text{ pro } i \neq j\}$ neboli $R_{S_1,S_2,S_3,P_1,P_2,P_3}$ obsahuje všechny možné permutace množiny $\{A,D,J,M,P,Z\}$. Tímto omezením je zajištěno, že každá sestra bude mít právě jednu službu. Ale dalo by se to zajistit i binárními omezeními - na každou možnou dvojici proměnných byla binární relace a jejími prvky by byly všechny možné dvojice sester kromě dvojic $(A,A),(D,D),(J,J),(M,M),(P,P),(Z,Z)$.

Výsledná síť omezení je tedy $\mathcal{R} = (X,D,C)$, kde $X = \{S_1,S_2,S_3,P_1,P_2,P_3\}$, $D = \{D_{S_1},D_{S_2},D_{S_3},D_{P_1},D_{P_2},D_{P_3}\}$ a $C = \{R_{S_1,P_1},R_{S_2,P_2},R_{S_3,P_3},R_{S_1,S_2,S_3,P_1,P_2,P_3}\}$ a jednotlivé prvky těchto množin jsou popsány výše.

Graf omezení v tomto případě by měl 6 vrcholů pojmenovaných S_1,S_2,S_3,P_1,P_2,P_3 a díky omezení all-different by byl graf úplný – hranami by byly spojeny vzájemně všechny vrcholy, protože každý je vůči každému dalšímu nějak omezen.

Hypergraf by situaci vystihoval lépe – za každé omezení je v něm oblast obsahující vrcholy odpovídající proměnným z rozsahu onoho omezení. V našem případě pro 3 binární a jedno omezení na všechny proměnné by vypadal takto:



Příklad [2] (4 body): Uveďte 2 základní metody, kterými je možné vylepšit zpětnou fázi backtrackingu. Ke každé stručně (1-2 věty) napište myšlenku, na které je založena.

Řešení:

- Backjumping – při dead-endu se nevrací jen o jednu úroveň výše (nezruší jen poslední přiřazenou hodnotu jedné proměnné), ale skočí o více úrovní na základě analýzy příčin dead-endu (zruší více posledních přiřazených hodnot proměnných)

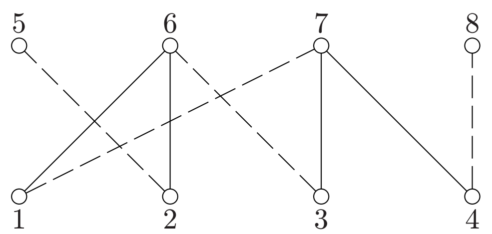
- Zaznamenání příčin dead-endu ve formě nových omezení, aby ze stejných příčin nenastal dead-end v budoucnu znovu. Používají se pro tyto metody názvy jako constraint learning, no-good recording apod.

Příklad [3] (4 body): Vysvětlete pojem cenová síť (cost network) - čím je zadaná, k čemu se používá.

Řešení: Cenová síť je rozšířením sítě omezení pro potřeby optimalizačních úloh. Používá se tedy pro reprezentaci konkrétních optimalizačních úloh. Je zadaná čveřicí $C = (X, D, C_h, C_s)$, kde (X, D, C_h) je síť omezení (tedy množina proměnných, množina domén proměnných a množina tvrdých omezení) a C_s je množina cenových komponent (neboli měkkých omezení).

Příklad [4] (15 bodů): Uvažujme bipartitní grafy (vrcholy je možné rozdělit do dvou disjunktních množin nazývaných partity tak, že žádná hrana nevede mezi vrcholy stejné množiny). Předpokládejme navíc, že partity jsou stejně velké. Problém úplného párování na takových grafech hledá podmnožinu množiny hran tak, aby každý vrchol grafu byl krajním vrcholem právě jedné z hran v této podmnožině.

Příklad bipartitního grafu, čárkované hrany tvoří jedno z možných úplných párování, partity jsou $\{1, 2, 3, 4\}, \{5, 6, 7, 8\}$:



a) Popište, jak se obecně jakákoliv instance problému úplného párování dá reprezentovat prostřednictvím problému SAT. Tedy popište způsob, jak k zadanému grafu vytvořit formuli a jak z nalezeného ohodnocení splňujícího tuto formuli zpětně získat párování.

b) Vytvořte formuli podle vašeho postupu pro výše uvedený konkrétní bipartitní graf.

c) Napište obsah CNF souboru (tedy v DIMACS CNF formátu), který odpovídá formuli z předchozího bodu.

Řešení:

a) Je možné zvolit různé reprezentace. Jednou z nich je zavést proměnnou za každou hranu existující v grafu. Potom pro každý vrchol bude podformule říkající, že alespoň jedna proměnná reprezentující hranu vedoucí do tohoto vrcholu musí být true a pro každou dvojici proměnných reprezentujících hrany vedoucí do tohoto vrcholu musí být jedna false.

Párování z nalezeného ohodnocení získáme snadno – patří do něj ty hrany, jejichž odpovídající proměnná bude ohodnocena true.

b) Pro konkrétní případ si proměnné označíme x_{ij} pro hranu z vrcholu i do j. Podformule na každém řádku odpovídá jednomu vrcholu (v pořadí odpovídajícím očíslování vrcholů).

$$
\begin{aligned}
&(x_{16} \vee x_{17}) \wedge (\neg x_{16} \vee \neg x_{17}) \wedge \\
&(x_{25} \vee x_{26}) \wedge (\neg x_{25} \vee \neg x_{26}) \wedge \\
&(x_{36} \vee x_{37}) \wedge (\neg x_{36} \vee \neg x_{37}) \wedge \\
&(x_{47} \vee x_{48}) \wedge (\neg x_{47} \vee \neg x_{48}) \wedge \\
&x_{25} \wedge \\
&(x_{16} \vee x_{26} \vee x_{36}) \wedge (\neg x_{16} \vee \neg x_{26}) \wedge (\neg x_{26} \vee \neg x_{36}) \wedge (\neg x_{16} \vee \neg x_{36}) \wedge \\
&(x_{17} \vee x_{37} \vee x_{47}) \wedge (\neg x_{17} \vee \neg x_{37}) \wedge (\neg x_{37} \vee \neg x_{47}) \wedge (\neg x_{17} \vee \neg x_{47}) \wedge \\
&x_{48}
\end{aligned}
$$

c) CNF soubor by vypadal třeba takto (proměnné jsou očíslovány 1-8 v pořadí $x_{16}, x_{17}, x_{25}, x_{26}, x_{36}, x_{37}, x_{47}, x_{48}$:

```
p cnf 8 18
1 2 0
-1 -2 0
3 4 0
-3 -4 0
5 6 0
-5 -6 0
7 8 0
-7 -8 0
3 0
1 4 5 0
-1 -4 0
-4 -5 0
-1 -5 0
2 6 7 0
-2 -6 0
-6 -7 0
-2 -7 0
8 0
```

Příklad [5] (5 bodů): Specifikujte problém "Horn SAT". Popište některý z algoritmů řešících tento problém v polynomiálním čase.

Řešení: Horn SAT je speciální případ problému SAT, kde formule musí mít speciální tvar. Na vstupu je tedy hornova formule a otázka je, jestli je tato formule splnitelná (tedy, zda existuje ohodnocení, při kterém je formule pravdivá, neboli, zda má formule model).

Hornovu formuli můžeme definovat např. tak, že jde o konjunkci implikací, kde každá implikace má na levé straně konjunkci proměnných a symbolů $\perp$ a $\top$ a na pravé straně jednu proměnnou nebo jeden symbol $\perp$ nebo $\top$.

Algoritmus např.:

- Označ všechny výskyty $\top$ ve formuli.
- Dokud existuje implikace s označenými všemi proměnnými a symboly na levé straně a neoznačenou pravou stranou, označ tuto pravou stranu (a pokud jde o proměnnou, tak i všechny její výskyty na levých stranách implikací).
- Pokud je označen nějaký výskyt $\perp$, vrať nesplnitelná, jinak vrať splnitelná.

Příklad [6] (4 bodů): K čemu je možné využít algoritmus nazvaný "List Scheduling"? Popište stručně, jak tento algoritmus pracuje.

Řešení: Jde o algoritmus pro online přidělování úloh na dostupné stroje.

Úlohy jsou uspořádány v seznamu (sekvenci). Kdykoliv je nějaký stroj volný, naplánuje na něj první úlohu ze seznamu, která je k dispozici (tedy není ještě naplánována, aktuální čas je větší než release čas úlohy a všechny předcházející úlohy v precedenčním grafu jsou skončeny).
