

Cvičení 7

SEMESTRÁLNÍ ÚLOHY – PROGRAMOVÁ ŘEŠENÍ (zadání úloh)

Poznámka: Úlohy lze naprogramovat a odladit:

- a) v Pythonu
- b) v Prologu
- c) v Javě (pro ty, kteří se na programování ve výše uvedených jazycích necítí)

1) Hra NIM

Je dáno N hromádek zápalek, ze kterých střídavě odebírají dva hráči tak, že mohou odebrat libovolný počet zápalek z jedné hromádky, případně odebrat stejný počet zápalek ze dvou hromádek. Vyhrává ten, kdo odebral poslední zápalku. Napište program, který hledá optimální strategii pro jednoho hráče.

2) Přelévání kapaliny se zdrojem a kanálem

Je dáno N nádob, jejich objem, počáteční naplnění a koncové naplnění. Vytvořte program, který najde nejkratší posloupnost přelití vedoucí z počátečního stavu do koncového stavu a pro přelévání je k dispozici nevyčerpatelný zdroj kapaliny a nepřeplnitelný kanál.

3) Lloydova 15

Ve čtverci $N \times N$ je $N \times N - 1$ kostiček a jedno volné pole. Vytvořte program, který složí kostičky tak, aby byly uspořádané. Uspořádání vhodně zvolte.

4) Vážení

Je dáno N identických předmětů. Nejvýše jeden z těchto předmětů je vadný, což se pozná podle jeho menší, resp. větší, hmotnosti (nevíte, která varianta platí). Napište program, který pro dané N zjistí nejmenší počet vážení na rovnoramenných vahách tak, aby se zjistilo, který předmět je vadný a zda je těžší nebo lehčí než ostatní. Program také vrátí doporučený postup vážení.

5) Zebra

Napište program, který je schopen řešit úlohy typu Zebra (např. jsou čtyři domy, v každém bydlí jeden člověk mající nějaké zaměstnání a auto a jsou dány podmínky typu "doktor jezdí v mercedesu" apod.; najděte příslušné rozložení obyvatel domu).

6) Mastermind (Logik)

Napište program, který řeší hru Mastermind (počet políček a barev je vstupním parametrem) sofistikovaným způsobem, tj. ne prohledáváním všech alternativ. K dispozici by měly být dvě varianty programu; jedna umožní zadávat ohodnocení zvenku (ručně), druhá ohodnocuje automaticky podle zadaného vzoru.

7) Ortogonální latinské čtverce

Latinský čtverec řádu n obsahuje čísla 1 až n , každé n krát, rozmístěné tak, že v žádném sloupci ani řádku se žádné číslo nevyskytuje dvakrát. Ortogonální latinský čtverec obsahuje dvojice čísel $\langle i, j \rangle$, $1 \leq i, j \leq n$, každou jedenkrát, tak, že v první i druhé složce tvoří latinský čtverec. Naprogramujte generování ortogonálního latinského čtverce daného řádu tak, aby se co nejdříve vyloučily nevyhovující možnosti (tj. ne jednoduchý backtracking, ale například dopředná kontrola).

8) **Obecné hledání optimální (herní) strategie barvením grafu**

Je dán graf, ve kterém vrcholy odpovídají proměnným s doménami (obory hodnot) a hrany jsou označeny jménem podmínky mezi proměnnými. Obarvíte graf, tj. přiřadíte každé proměnné hodnotu z její domény tak, aby byly splněny všechny podmínky. Navrhněte vlastní reprezentaci grafu a vytvořte generátor, který připraví graf např. pro řešení problému N-dam. Realizujte lokálním prohledáváním s Tabu seznamem: Nejprve se "náhodně" obarví všechny vrcholy. Pokud existuje nějaký vrchol, který je v konfliktu se svým okolím (není splněna podmínka na hraně), program najde pro tento vrchol jinou barvu tak, aby se konflikt zmenšil. Pro odstranění lokálního minima použijte techniku Tabu seznamu.

9) **Letiště**

Je dána množina typu letadel a pro každý typ letadla je dán seznam stojánek, u kterých může stát. Na každé stojánce může v daný čas stát pouze jedno letadlo, s výjimkou stojánky "volná plocha", na které může stát libovolně mnoho letadel (existují ale letadla, která nemohou stát na volné ploše – např. Concorde). Dále je dán letový plán, tj. seznam letadel s určením jejich typu, doby příletu a odletu. Napište program, který rozvrhne letadla na jednotlivé stojánky tak, aby co nejvíce letadel bylo obhospodařeno stojánkami (tj. ne na volné ploše). Letiště je definováno množinou všech stojánek.

10) **Alokace pultu (odbavovací přepážky) na letišti**

Odbavovací přepážky na letišti jsou popsány jako množina pultů, které jsou po několika sdruženy do "ostrovů". Dále je dán letový řád, kde je u každého letu zadán čas odbavení (interval od-do) a počet pultů, které jsou potřeba. Napište program, který provede rozmístění pultů pro lety tak, že všechny pulty daného letu se nacházejí ve stejném ostrově.

11) **Nemocnice**

Je dána množina sester, každá sestra má přiřazenu množinu činností, které může vykonávat (kvalifikaci). Dále je dán rozvrh požadavků na činnosti v daném časovém období. Napište program, který vytvoří rozvrh pro konkrétní sestry tak, aby požadavky byly naplněny a přitom žádná sestra nesloužila déle jak tři směny za sebou.

12) **Výrobní podnik**

Je dána množina strojů a množina produktů. Každému produktu je přiřazena množina strojů a doba, za kterou daný stroj produkt vyrobí. Dále je dána množina preferencí říkajících, jaký produkt musí být vyroben před jiným produktem. Napište program, který naplánuje výrobu tak, aby trvala co nejkratší dobu.

13) **Reklamy**

Je dána sada reklam a u každé reklamy je zadán interval, ve kterém musí být vysílána. Spojitě za sebou smí být promítnuto jen M reklam a mezi reklamními bloky musí být minimálně N volných časových jednotek. Napište program, který provede rozvržení reklam.

14) **Obchodní cestující**

Je dán graf, ve kterém vrcholy představují města a hrany jsou ohodnoceny časem nutným pro cestu mezi dvěma městy. Každému městu/vrcholu je přiřazena dvojice: délka jednání/obsluhy a interval, ve kterém musí jednání/obsluha proběhnout. Vytvořte program, který najde cestu obchodního cestujícího začínající a končící v daném městě tak, že se zúčastní všech jednání.

15) Večeře

Dnes večer přijdu domů a budu mít chuť na Kung-pao. Znalostní systém mi potvrdí, zda je to možné (na základě stavu mé ledničky a spíže). Na rozdíl od mechanického porovnání seznamu obsahu ledničky s receptem, znalostní systém mi potvrdí, že je to možné i v případě, že bude chybět nějaká nepodstatná záležitost nebo pokud bude možné nahradit požadovanou surovinu něčím jiným, event. poradí, kde je možné chybějící surovinu obstarat (která prodejna v blízkosti má ještě otevřeno).

16) Účet v bance

Navrhněte jednoduchý znalostní systém, který Vám pomůže vybrat filiálku banky, která Vám nabídne založení spořicího účtu s nejvýhodnějšími podmínkami. Vzhledem k jednoduchosti řešení omezte realizaci na výběr mezi třemi různými bankami (výběr bank volte podle situace v ČR) a podmínky vedení vkladů (znalosti uložené v bázi znalostí) zvolte tak, aby výběr možností nebyl úplně triviální, a Váš výběr doplňte vlastními podmínkami (výše úroku nemusí být rozhodující, důležité jsou např. možnosti výběru peněz, podmínky vedení účtu, resp. bankovní poplatky apod.)

17) Řešení lineárních podmínek eliminací

Je dána množina proměnných a množina lineárních rovností a nerovností nad těmito proměnnými. Napište program, který řeší podmínky postupnou eliminací proměnných (podobně jako při symbolickém řešení soustav rovnic a nerovnic pomocí Gaussovy a Fourierovy eliminace).

18) Přepisovací pravidla

Je dána sada přepisovacích pravidel tvarů $Vzor \Rightarrow Prepis$ | $Straz$ a $Vzor \Rightarrow Pridej$ | $Straz$, kde $Vzor$ a $Prepis$ jsou seznamy termů a $Straz$ je prologovský cíl, který může případně chybět (př.: $[X \leq Y, Y \leq Z] \Rightarrow [X \leq Z]$ nebo $[X \leq X] \Leftrightarrow []$). Interpretace pravidel je následující: je-li vzor nalezen ve vstupním seznamu predikátů a je-li splněna $Straz$, potom

a) v případě \Leftrightarrow je vzor nahrazen seznamem $Prepis$

b) v případě \Rightarrow je přidán seznam $Pridej$.

Napište program, který převede seznam termů na redukovaný seznam termů aplikací všech možných přepisovacích pravidel.

19) Analýza výrazů s operátory

Vstupem programu je seznam definic operátorů ve tvaru **op(Priorita, Asoc, Navez)** a seznam tokenů (atomy, čísla a závorky). Analyzujte posloupnost tokenů podle operátorů a převed'te ji do struktury prologovského termu.

Např.:

```
?-analyzuj([op(4,yfx,plus),op(2,yfx,times)], [a,plus,b,times,c], X).
```

vrací: **X = plus(a,times(b,c))** .

20) Splnitelnost logického výrazu

Je dán logický výraz se spojky **and**, **or**, **not**, **xor**, **impl**, **equiv** (definujte pomocí operátorů) obsahující logické konstanty **true**, **false** a proměnné (reprezentované atomy). Vytvořte program, který rozhodne, zda je zadaný výraz pravdivý, resp. splnitelný, případně vydá nutné ohodnocení proměnných tak, aby byl výraz splněn.

Např.: **?-sat((x or not x) and y, R)** . vrací **R=[y/true]**