

UIR - Témata semestrálních prací pro rok 2015

2015VM02

Ve vhodně zvoleném programovacím jazyce realizujte demonstrační program pro obecnou úlohu obchodního cestujícího, tj. úlohu nalezení nejkratší, resp. nejméně nákladné, trasy pro „objetí“ n náhodně generovaných míst ($10 \leq n \leq 25$) metodou simulovaného žihání (simulated annealing) nebo zakázaného prohledávání (tabu search).

2015VM04

V libovolném programovacím jazyce zpracujte programový systém pro simulaci hry se čtyřmi řadami (hromádkami) knoflíků (příp. zápalek). Pravidla: Do hry vstupují dva hráči (v našem případě uživatel a počítač), na stole jsou 4 řady knoflíků (4 hromádky zápalek) obsahující 7, 5, 3 a 1 knoflík (zápalek). Hráči se střídají v tazích. Hráč, který táhne, odebere libovolný počet knoflíků (zápalek), avšak vždy jen z jedné z řad (hromádek), např. stav 7, 5, 3, 1 můžeme odebráním knoflíků (zápalek) převést na stav 7, 1, 3, 1, nikoli však na 6, 4, 3, 1. Hráč, který provede poslední možný povolený tah, vyhrává.

Implementujte inteligentní hrací strategii. Systém zobecněte pro libovolné počty knoflíků (zápalek) v libovolném počtu řad (hromádek), eventuálně pro změněnou cílovou podmínku. Dále ověřte řešitelnost úlohy pro vyšší počty knoflíků (zápalek) a určete její algoritmickou a paměťovou složitost.

Inspirace a částečně i návod na řešení: http://www.alga.cz/users/papacek/sp_games.htm

2015OR06

Naprogramujte deskovou hru Dáma. Implementujte algoritmus MinMax s alfa-beta prořezáváním.

2015OR10

Navrhněte a realizujte vhodný algoritmus, který bude automaticky generovat křížovky. Vstupem programu bude 1) velikost křížovky 2) tajenka 3) slovník přípustných slov ze kterých je možné křížovku sestavit (např. <http://scrabble.hrejsi.cz/pravidla/slovník.htm>). Výstupem bude textový soubor obsahující jak křížovku s tajenkou tak křížovku bez tajenky.

2015OR13

Připravte program, který řeší problém rozvrhu na školách tj. který učitel, kdy, jakou třídu a jaký předmět učí. Pro řešení problému použijte některý z evolučních algoritmů. Pro svoji úlohu použijte reálná data – např. ze střední školy, kterou jste navštěvovali. Počítejte s tím, že třídu je možné dělit (výuka jazyků, tělocvik, povinně volitelné semináře).

2015OR14

Napište program, který bude umět náhodně rozmíchat a poté zase složit známý hlavolam „Rubikova kostka“. Umožněte také ruční zadání počátečního (rozmíchaného) stavu úlohy. Návod na složení hlavolamu zapište do textového souboru. Umožněte vizualizaci v grafickém režimu nebo jako ASCII art (uživatel si může zvolit, zda vizualizaci chce zapnout nebo vypnout).

2015OR15

Vytvořte genealogický expertní systém. Vstupem bude informace databáze základních informací o osobě [id, jméno, matka, otec, pohlaví, partner]. Program bude umět odpovídat na otázky: Kdo je můj pokrevní příbuzný? Kolik má moje prarabata sestřenic? Jaký je vztah mezi Petrem a Pavlem? ... a další podobné.

2015OR16

Napište program, který bude řešit obecné aritmetické hříčky typu $[S,E,N,D] + [M,O,R,E] = [M,O,N,E,Y]$ nebo $[C,R,O,S,S] + [R,O,A,D,S] = [D,A,N,G,E,R]$. Smyslem hádanky je určit (jednoznačně nebo víceznačně – podle typu úlohy), jaké číslice jsou reprezentovány jednotlivými písmeny. Umožněte (minimálně) aritmetické operace sčítání a odčítání. Cítíte-li se na složitější úlohu, realizujte i násobení (za vyšší počet bodů). Dělení nerealizujte.

2015OR17

Vytvořte znalostní systém, který bude umět převádět čísla zapsaná v desítkové soustavě do češtiny a do vyjádření v římských číslicích (a naopak). V případě římským číslic pracujte do čísla 4000. U češtiny a desítkové soustavy pracujte do čísla 1000000000000.

2015OR18

Napište program pro automatické generování labyrintů různých velikostí. Výstup umožněte do textového souboru (cesta = mezera, zeď = znak volitelný jako parametr) a do souboru vhodného grafického formátu (PNG, GIF, PCX, EPS nebo PS). Zároveň bude program umět předvést řešení úlohy – nalezení cesty od vchodu do labyrintu k východu z labyrintu resp. zevnitř labyrintu ven.

2015KT02

Vytvořte program, který řeší následující úlohu: Posel přinesl opatovi zprávu, že na cestě k opatství jsou poutníci a budou hledat ubytování. Rozhodl, že je mají ubytovat ve čtvercové budově o dvou podlažích. Na každém podlaží je osm místností - na každé straně budovy jsou tři místnosti (schodiště je uprostřed budovy). Na každé straně budovy musí být ubytováno 11 osob a na horním podlaží musí být dvakrát více lidí než na spodním. Navíc musí být každá místnost musí být obsazena a v žádné nesmí více než 3 osoby. Poté co mniši vyřešili malý problém s ubytováním přišli poutníci a ukázalo se, že jich je o 3 více než bylo původně ohlášeno. Mnichům se je ale i tak podařilo ubytovat, aniž by porušili opatova nařízení. Kolik přišlo celkem poutníků a jak byli ubytováni?

2015KT03

Sestrojte gramatiku, lexikon a programový systém, který bude přepočítávat jednoduché příklady (sčítání, odčítání) zadané v číslovkami v přirozeném jazyce. Systém rozpozná syntakticky špatně zadaný příklad vypočítá výsledek a odpoví v přirozeném jazyce. Příklady na násobení a dělení rozpozná jako správně syntakticky zadané, ale odpoví, že je neumí řešit. Neuvažujte závorky.

2015KT06

Řešte následující problém: čtyři lidé mají projít tunelem v nejkratším čase, prvnímu cesta trvá 5 minut, druhému 10 minut, třetímu 20 minut a čtvrtému 25 minut. K cestě tunelem je třeba baterka, skupina má pouze jednu baterku, současně mohou jít maximálně dva lidé. Dvojice se vždy pohybuje tempem slabšího člena. Při zpracování řešte následující otázky:

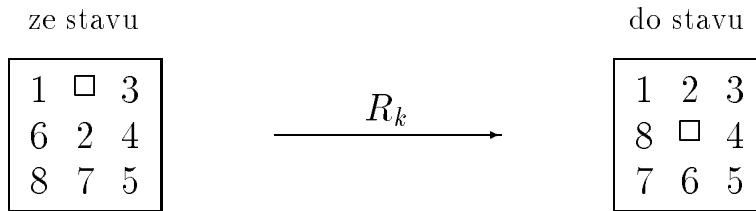
- a) pokud úlohu chápeme jako prohledávání stavového prostoru, co jsou stavy a co jsou akce evokující přechody mezi stavy,
- b) úlohu zobecněte pro libovolnou velikost skupiny a libovolné časy průchodu, vytvořte program řešící tuto úlohu,
- c) odhadněte paměťovou a časovou složitost při řešení metodou slepého prohledávání do hloubky resp. do šířky, navrhněte omezující heuristiky,
- d) stanovte jednoduchou horní mez pro větvící faktor.

2015KT07

Sestrojte systém pro simulaci hry se sirkami. Pravidla: Do hry vstupují dva hráči (v našem případě uživatel a počítač), na stole jsou 4 hromádky sirek (7, 5, 3 a 1 sirka). Hráči se střídají v tazích. Hráč, který táhne, odebere libovolný počet sirek z jedné z hromádek. (Např.: 7, 5, 3, 1 může přejít na 7, 1, 3, 1 nikoli však na 6, 4, 3, 1). Hráč, který provede poslední možný povolený tah, vyhrává.

Implementujte inteligentní strategii. Systém zobecněte pro libovolné počty sirek v libovolném počtu hromádek. Ověřte řešitelnost a její složitost pro vyšší počty sirek.

104. V programovacím jazyce C zpracujte jednoduchý výukový program, který plně demonstuje nalezení řešení následující úlohy: Jde o převedení hlavolamu "8" (při respektování pořadí aplikace produkčních pravidel uvedeného na přednášce)



metodou prohledávání grafu s využitím heuristické funkce

$$\begin{aligned} \hat{f}(n_i) &= \hat{g}(n_i) + \hat{h}(n_i), \\ \hat{g}(n_i) &= d(n_i) \quad (\text{délka cesty z } n_0 \text{ do } n_i), \\ \hat{h}(n_i) &= P(n_i) + 3Q(n_i), \quad \text{kde} \end{aligned}$$

- $P(n_i)$ je součet vzdáleností každého kamene hlavolamu od svého cílového místa (v možných posuvech),
- $Q(n_i)$ je míra porušení pořadí kamenů zahrnutá tak, že
 - přičítáme hodnotu 2 za každý kámen nenacházející se ve středu pole a jenž není následován správným kamenem,
 - za kámen ve středu pole přičítáme 1.

105. V programovacím jazyce C zpracujte jednoduchý demonstrační program, který provede a vhodným způsobem zobrazí vyhodnocení algoritmické a paměťové složitosti jednotlivých řešení úlohy "Misionáři a kanibalové" (popis úlohy viz níže) získaných metodami

- a) navracení (backtracking),
- b) prohledávání do hloubky,
- c) prohledávání do šířky,
- d) hledání v grafu s využitím heuristické funkce

$$\hat{f}(n_i) = \hat{g}(n_i) + \hat{h}(n_i),$$

kde $\hat{g}(n_i) = d(n_i)$ (hloubka uzlu),
 $\hat{h}(n_i) =$ počet misionářů zbylých na prvním břehu.

Popis úlohy: Tři misionáři a tři kanibalové se chtějí dostat z jednoho břehu řeky na druhý. Jediným prostředkem, který by to umožnil, je malá loďka, která unese maximálně dva lidi. Všech šest poutníků umí s loďkou zacházet.

Úkolem je nalézt takové řešení přepravy poutníků na druhý břeh řeky, aby na žádné straně řeky nanastala situace, že tam bude více kanibalů než misionářů, neboť menšina misionářů může být kanibaly snědena. Je tedy třeba určit posloupnost jízd a obsazení loďky, která vyhoví výše uvedené podmínce.

106. V programovacím jazyce C zpracujte jednoduchý výukový program, který plně demonstuje nalezení správného řešení (včetně všech ostatních) následující úlohy:

Máte k dispozici dva kameninové neprůhledné džbány – jeden o objemu a litrů a druhý o objemu b litrů (při naplnění po okraj), přičemž nechtě $b < a$. Vaším úkolem je naplnit větší (tj. a -litrový) džbán naléváním, přeléváním a vyléváním přesně c litry vody ($c < a$ a současně $c \neq b$); například máte čtyř- a třílitrový džbán a postupným naléváním, přeléváním či vyléváním máte větší (čtyřlitrový) naplnit dvěma litry vody.

Program zpracujte zcela obecně, tj. tak, aby bylo možné zadávat libovolné hodnoty a , b i c , avšak nezapomeňte testovat správnost zadání vstupních hodnot, aby úloha byla řešitelná. V případě řešitelnosti vypište či vykreslete graf řešení úlohy a vypište všechna nalezená řešení úlohy v pořadí od nejlepšího po nejhorší.

107. V programovacím jazyce C zpracujte jednoduchý výukový program, který plně demonstuje nalezení správného řešení (včetně všech ostatních) následující úlohy:

Máte k dispozici dva neprůhledné kovové barely – jeden o objemu a litrů a druhý o objemu b litrů (při 100% naplnění), přičemž nechtě $b < a$. Vaším úkolem je naplnit větší (tj. a -litrový) barel napouštěním, přepouštěním a vypouštěním přesně c litry nafty ($c < a$ a současně $c \neq b$); například máte sto- a pětasedmdesátilitrový barel a postupným napouštěním, přepouštěním, příp. vypouštěním máte větší (stolitrový) naplnit padesáti litry nafty.

Program zpracujte zcela obecně, tj. tak, aby bylo možné zadávat libovolné hodnoty a , b i c , avšak nezapomeňte testovat správnost zadání vstupních hodnot, aby úloha byla řešitelná. V případě řešitelnosti vypište či vykreslete graf řešení úlohy a vypište všechna nalezená řešení úlohy v pořadí od nejlepšího po nejhorší.

108. Ve vhodně zvoleném programovacím jazyce realizujte demonstrační (výukový) program pro úlohu nalezení všech možných rozmístění dam na šachovnici libovolné velikosti ($n \leq 16$) takových, aby se žádné dvě dámy se ve smyslu pravidel šachu (dámy) neohrožovaly, metodou "simulovaného žíhání" (simulated annealing).

109. Ve vhodně zvoleném programovacím jazyce realizujte demonstrační (výukový) program pro úlohu nalezení všech možných rozmístění dam na šachovnici libovolné velikosti ($n \leq 16$) takových, aby se žádné dvě dámy se ve smyslu pravidel šachu (dámy) neohrožovaly, metodou zakázaného prohledávání (tabu search).

110. Ve vhodně zvoleném programovacím jazyce realizujte demonstrační (výukový) program pro obecnou úlohu obchodního cestujícího (tj. úlohu nalezení nejkratší, resp. nejméně nákladné trasy pro "objetí" n náhodně generovaných míst, $10 \leq n \leq 25$) aplikací genetických algoritmů.
111. Ve vhodně zvoleném programovacím jazyce realizujte demonstrační (výukový) program pro obecnou úlohu obchodního cestujícího (tj. úlohu nalezení nejkratší, resp. nejméně nákladné trasy pro "objetí" n náhodně generovaných míst, $10 \leq n \leq 25$) metodou "simulovaného žíhání".
112. Ve vhodně zvoleném programovacím jazyce realizujte demonstrační (výukový) program pro obecnou úlohu obchodního cestujícího (tj. úlohu nalezení nejkratší, resp. nejméně nákladné trasy pro "objetí" n náhodně generovaných míst, $10 \leq n \leq 25$) metodou zakázaného prohledávání ("tabu search").
113. Ve vhodně zvoleném programovacím jazyce programově realizujte úlohu detekce primitiv v rastrové reprezentaci následujících číslic (předpokládejte rozměr reprezentujícího rastru 15 x 21 obrazových bodů)

1 2 3 4 5 6 7 8 9 0

metodou porovnávání se vzory primitiv (template matching). Druhy i velikost reprezentace primitiv volte podle vlastního uvážení tak, abyste jimi postihli všechny tvarové charakteristiky rozpoznávaných číslic a přitom dosáhli uspokojivé algoritmické složitosti vyhledávání. V závěru programového chodu zřetězte nalezená (detekovaná) primitiva do jednoznačných strukturních popisů (obrazů) – zřetězení proveďte algoritmicky i paměťově nejméně náročným způsobem (řetězení lze provádět i v průběhu detekce jednotlivých primitiv); výstupem Vašeho programu pak budiž řetězec symbolů představující vytvořený strukturní popis (obraz) zpracovávané číslice.

Textovou reprezentaci rastrových snímků všech výše vyobrazených číslic najdete v souborech `nula.trs` až `devět.trs` uložených v obvyklých adresářích.

114. V libovolném programovacím jazyce zpracovajte jednoduché programové vybavení pro klasifikaci barevných snímků ve formátu "true-color", uložených na serveru **Lucifer**. Programové vybavení musí umožnit klasifikaci snímků bez nutnosti jejich převodu na šedotónové s tím, že programově realizujete následující kroky:
- (a) zobrazení snímku na monitoru bez přílišné ztráty informace obsažené ve snímku;
 - (b) binarizaci snímku prahováním každé "barvy" do alespoň čtyř jasových oblastí (využitím tří prahových úrovní jasu);
 - (c) vytvoření efektivního příznakového popisu scény umožňujícího
 - paměťově nenáročné uložení obrazu scény do knihovny vzorových obrazů,
 - rozpoznání testovacího snímku scény porovnáním se vzorovými obrazy;
 - (d) vytvoření knihovny vzorových obrazů ve fázi trénování;
 - (e) efektivní porovnávání obrazu testovací scény se vzorovými obrazy uloženými v knihovně vzorových obrazů – nezbytné pro vlastní klasifikaci;
 - (f) výpis výsledku rozpoznávání (klasifikace) spolu se zdůvodněním, na základě čeho k rozhodnutí o zařazení do dané klasifikační třídy došlo, příp. jaké míry shody bylo dosaženo.

Pro demonstraci funkce zpracovaného programového vybavení využijte dodaných snímků (adresář **UIR\OBR**), případně vlastních pořízených snímků s tím, že jako testovací (rozpoznávaný) snímek využijete některý z připravených testovacích snímků označených v názvu snímku číslicí 2 (např. **derese2.rs**) nebo (v případě vlastních snímků) speciálně připravený testovací snímek, tj. snímek např. opakovaně sejmete pod jiným úhlem či jinak natočený, uměle rozostříte, tvarově poněkud zdeformujete apod.

115. V libovolném programovacím jazyce zpracujte jednoduché programové vybavení pro klasifikaci barevných snímků ve formátu "true-color", uložených na serveru **Lucifer**. Programové vybavení musí umožnit:

- (a) zobrazení snímku na monitoru bez přílišné ztráty informace obsažené ve snímku;
- (b) převedení snímku do šedotónové stupnice podle transformačního vztahu
$$Y = 0.30 * R + 0.59 * G + 0.11 * B$$
a jeho zobrazení na monitoru;
- (c) binarizaci snímku prahováním alespoň do čtyř jasových oblastí (využitím tří prahových úrovní jasu);
- (d) vytvoření efektivního příznakového popisu scény umožňujícího
 - paměťově nenáročné uložení obrazu scény do knihovny vzorových obrazů,
 - rozpoznání testovacího snímku scény porovnáním se vzorovými obrazy;
- (e) vytvoření knihovny vzorových obrazů ve fázi trénování;
- (f) efektivní porovnávání obrazu testovací scény se vzorovými obrazy uloženými v knihovně vzorových obrazů – nezbytné pro vlastní klasifikaci;
- (g) výpis výsledku rozpoznávání (klasifikace) spolu se zdůvodněním, na základě čeho k rozhodnutí o zařazení do dané klasifikační třídy došlo, příp. jaké míry shody bylo dosaženo.

Pro demonstraci funkce zpracovaného programového vybavení využijte dodaných snímků (adresář **UIR\OBR**), případně vlastních pořízených snímků s tím, že jako testovací (rozpoznávaný) snímek využijete některý z připravených testovacích snímků označených v názvu snímku číslicí 2 (např. **derese2.rs**) nebo (v případě vlastních snímků) speciálně připravený testovací snímek, tj. snímek např. opakovaně sejmete pod jiným úhlem či jinak natočený, uměle rozostříte, tvarově poněkud zdeformujete apod.

116. V libovolném programovacím jazyce zpracujte jednoduché programové vybavení pro klasifikaci barevných snímků ve formátu "true-color", uložených na serveru Lucifer. Programové vybavení musí umožnit:

- (a) zobrazení snímku na monitoru bez přílišné ztráty informace obsažené ve snímku;
- (b) převedení snímku do šedotónové stupnice podle transformačního vztahu
$$Y = 0.30 * R + 0.59 * G + 0.11 * B$$
a jeho zobrazení na monitoru;
- (c) proporcionální úpravu jasových úrovní šedotónového snímku i jednotlivých barevných složek jednotlivých obrazových bodů transformací snímku do celého rozsahu jasových úrovní (0 .. 255)
 - a) metodou "roztažení" histogramu jasových úrovní,
 - b) metodou přibližného vyrovnání (equalizace) histogramu.
- (d) zobrazení transformovaných snímků (šedotónového i barevného) na monitoru
- (e) binarizaci šedotónového snímku prahováním alespoň do čtyř jasových oblastí (využitím tří prahových úrovní jasu);
- (f) vytvoření efektivního příznakového popisu scény umožňujícího
 - paměťově nenáročné uložení obrazu scény do knihovny vzorových obrazů,
 - rozpoznání testovacího snímku scény porovnáním se vzorovými obrazy;
- (g) vytvoření knihovny vzorových obrazů ve fázi trénování;
- (h) efektivní porovnávání obrazu testovací scény se vzorovými obrazy uloženými v knihovně vzorových obrazů – nezbytné pro vlastní klasifikaci;
- (i) výpis výsledku rozpoznávání (klasifikace) spolu se zdůvodněním, na základě čeho k rozhodnutí o zařazení do dané klasifikační třídy došlo, příp. jaké míry shody bylo dosaženo.

Pro demonstraci funkce zpracovaného programového vybavení využijte dodaných snímků (adresář UIR\OBR), případně vlastních pořízených snímků s tím, že jako testovací (rozpoznávaný) snímek využijete některý z připravených testovacích snímků označených v názvu snímku číslicí 2 (např. `derese2.rs`) nebo (v případě vlastních snímků) speciálně připravený testovací snímek, tj. snímek např. opakovaně sejmete pod jiným úhlem či jinak natočený, uměle rozostříte, tvarově poněkud zdeformujete apod.

117. Ve vhodně zvoleném programovacím jazyku zpracujte jednoduchý programový systém, který přeloží libovolnou formuli výrokového počtu zapsanou běžným způsobem, tj. s využitím běžných logických spojek (použijte symboliku: \sim nebo $!$ pro negaci, \wedge nebo $\&$ pro konjunkci, \vee nebo $|$ pro disjunkci, \rightarrow pro implikaci a \leftrightarrow nebo $==$ pro ekvivalenci), do podoby pravdivostní tabulky, s jejíž pomocí prokážete, zda zapsaná logická formule je logicky pravdivá či nikoli.

Poznámka: Nezapomeňte na patřičnou lexikální a syntaktickou analýzu formule odpovídající Vám nadefinovanému způsobu zadávání formule!

118. V programovacím jazyce C nebo C++ zpracujte v podobě procedury programovou realizaci obecné rezoluční metody pro dokazování logické pravdivosti formulí predikátové logiky 1. řádu.

a) Jako výchozí podmínky předpokládejte, že formule je již převedena do klauzulární formy (zapsána jako množina klauzulí) a uložena jako textový řetězec (formu reprezentace vhodně zvolte) v poli, jehož symbolické jméno bude vstupním parametrem procedury.

b) Algoritmus hledání rezolventy (rezoluci jednotlivých klauzulí) volte podle uvážení.

c) Výstupem procedury bude informace, zda formule je či není logicky pravdivá.

119. Ve vhodném programovacím jazyku zpracujte jednoduchý programový systém, který

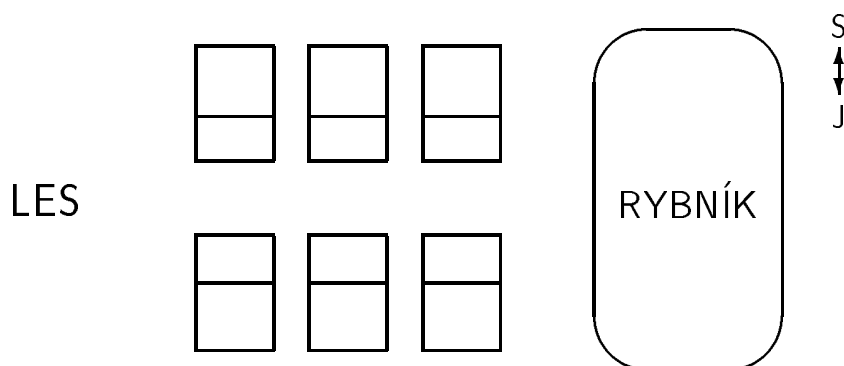
a) převede (transformuje) libovolnou formuli predikátové logiky 1. řádu zapsanou jako textový řetězec na množinu klauzulí,

b) s využitím procedury zpracované ve výše uvedené zápočtové úloze zjistí pravdivost závěru (teorému) formulemi zapsané úlohy,

c) srozumitelnou formou vypíše výsledek rezolučního procesu (logickou pravdivost nebo nesplnitelnost prověřovaného teorému).

120. V Prologu запиšte a vyřešte následující úlohu:

U rybníka trávilo dovolenou šest rodin, které přijely osobními automobily. Měly různobarevné stany, které byly postaveny podle obrázku:



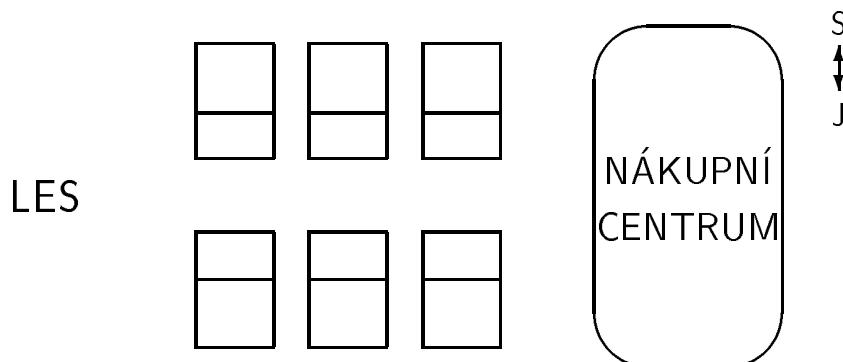
Dále víme, že:

- manželé, kteří přijeli trabantem, měli stan u rybníka s vchodem na sever;
- Jan s manželkou (není to Dana) měli červený stan, přijeli škodovkou a měli stan u lesa s vchodem k jihu;
- Josef a Milada Vančurovi měli stan s vchodem k jihu;
- modrý stan byl u lesa;
- žena, která byla v šedém stanu uprostřed s vchodem k jihu, není manželkou Oskara ani Pavla;
- Michal s Janou přijeli na dovolenou wartburgem;
- Květa byla v hnědém stanu, paní Stříbrná v zeleném (nejmenovala se Pavla) a paní Veselá ve žlutém stanu;
- Pavel s manželkou přijeli fiatem; neměli stan proti Novákovým, kteří měli žigulíka;
- manželé Oskar a Pavla se nejmenují Novákoví, Eva se jmenuje Kovářová;
- s vchodem na sever byl stan Dany Matulové proti stanu Kovářových;
- Oskar byl ve stanu u rybníka s vchodem na sever;
- Stříbrných neměli stan těsně u rybníka.

Podle uvedených údajů a obrázku určete, která rodina měla dacií a jak se jmenoval Petr.

121. V Prologu запиšte a vyřešte následující úlohu:

Šest mladých rodin si postavilo v nové ulici u nákupního centra šest stejných domků, které je možno rozlišit pouze barvou fasády a umístěním v ulici. Každá z rodin má navíc jiné auto. Rozmístění domů budiž takové, jak je znázorněno na obrázku, menší obdélníček necht' značí předzahrádku s vchodem do domku:



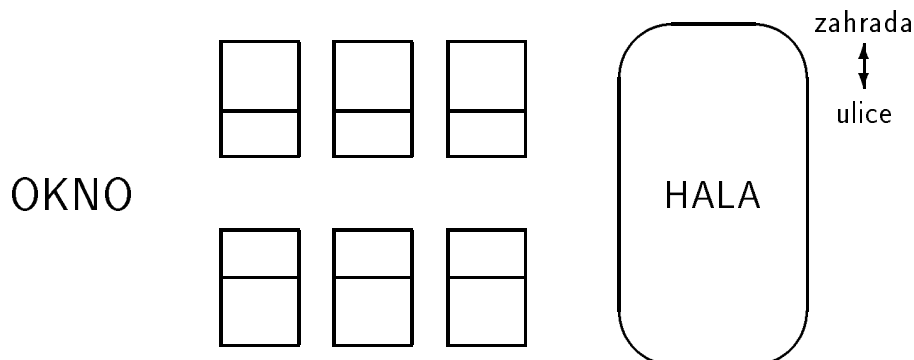
Dále víme, že:

- manželé, kteří mají VW Polo, mají domek u nákupního centra s vchodem na sever;
- Jan s manželkou (není to Dana) mají domek s červenou fasádou, Škodu Fabii a jejich domek stojí u lesa s vchodem k jihu;
- Josef a Milada Vančurovi mají domek s vchodem k jihu;
- domek s modrou fasádou stojí u lesa;
- žena, která bydlí v šedém domku uprostřed s vchodem k jihu, není manželkou Oskara ani Pavla;
- Michal s Janou si koupili nového renaulta;
- Květa bydlí v domku s hnědou omítkou, paní Stříbrná v zeleném domku (nejmenuje se Pavla) a paní Veselá bydlí v domku se žlutou fasádou;
- Pavel s manželkou mají fiata a nebydlí proti Novákovým, kteří mají starého žigulíka;
- manželé Oskar a Pavla se nejmenují Novákoví, Eva se jmenuje Kovářová;
- s vchodem na sever je domek Dany Matulové, a to proti domku Kovářových;
- Oskar bydlí v domku u nákupního centra s vchodem na sever;
- Stříbrných nemají domek u nákupního centra.

Podle uvedených údajů a obrázku určete, která rodina má Audi a jak se jmenoval Petr.

122. V Prologu запиšte a vyřešte následující úlohu:

Šest mladých rodin si pronajalo v nové ubytovně pro novomanžele šest malých bytů v chodbě vedle haly; byty jsou rozlišeny barvou dveří a umístěním na chodbě. Každá z rodin má navíc jiný počet nebo pohlaví dětí. Rozmístění bytů budiž takové, jak je znázorněno na obrázku, menší obdélníček nechtě značí předsíňku s vchodem:



Dále víme, že:

- manželé, kteří mají šestiletého syna, mají byt vedle haly s vchodem směrem k zahradě;
- Jan s manželkou (není to Dana) mají byt s červenými dveřmi, dvouletou dcerku a jejich byt je na konci chodby u okna s vchodem směrem k ulici;
- Josef a Milada Vančurovi mají byt s vchodem k ulici;
- byt s modrými dveřmi je u okna;
- žena, která bydlí v bytě s šedými dveřmi uprostřed, s vchodem k ulici, není manželkou Oskara ani Pavla;
- Michal s Janou mají dvojčata;
- Květa bydlí v bytě s hnědými dveřmi, paní Stříbrná v bytě se zelenými dveřmi (nejmenuje se Pavla) a paní Veselá bydlí v bytě se žlutými dveřmi;
- Pavel s manželkou mají dva syny (ne dvojčata) a nebydlí proti Novákovým, kteří mají tři děti;
- manželé Oskar a Pavla se nejmenují Novákoví, Eva se jmenuje Kovářová;
- s vchodem směrem k zahradě je byt Dany Matulové, a to proti bytu Kovářových;
- Oskar bydlí v bytě u haly, s vchodem směrem k zahradě;
- Stříbrných nemají byt u haly.

Podle uvedených údajů určete, která rodina je bezdětná a jak se jmenoval Petr.

123. V Prologu запиšte a vyřešte úlohu "Misionáři a kanibalové" (viz níže uvedený popis úlohy), avšak pro případ obecného počtu misionářů a kanibalů, eventuálně vyšší počet člunů (aktuální údaje zadávejte z klávesnice), které však – stejně jako v uvedeném zadání – mohou převézt nejvýše dva pasažéry.

Popis úlohy: Tři misionáři a tři kanibalové se chtějí dostat z jednoho břehu řeky na druhý. Jediným prostředkem, který by to umožnil, je malá loďka, která unese maximálně dva lidi. Všech šest poutníků umí s loďkou zacházet.

Úkolem je nalézt takové řešení přepravy poutníků na druhý břeh řeky, aby na žádné straně řeky nanastala situace, že tam bude více kanibalů než misionářů, neboť menšina misionářů může být kanibaly snědena. Je tedy třeba určit posloupnost jízd a obsazení loďky, která vyhoví výše uvedené podmínce.

124. V programovacím jazyce C, případně v Prologu, zpracujte program, který

- a) v jednoduchém aritmetickém výrazu typu

$$\begin{array}{r}
 \text{číslo1} \\
 \text{operátor} \text{ číslo2} \\
 \hline
 \text{číslo3}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{Př. :} \qquad 4378 \\
 + \quad 1235 \\
 \hline
 5623
 \end{array}
 ,$$

v němž veškeré údaje jsou zadány v textové formě (řetězce znaků) a operátor je prvkem množiny $\{ +, -, * \}$, odhalí případnou chybu a nabídne všechny možnosti opravy;

- b) pro danou kryptoaritmetickou úlohu typu

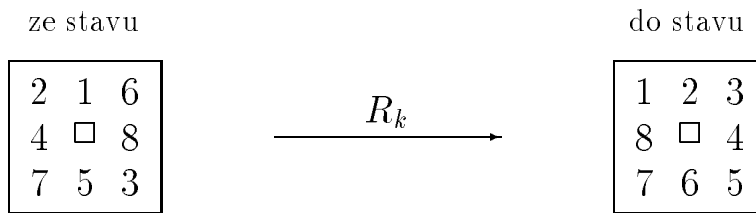
$$\begin{array}{r}
 \text{ADF} \\
 \text{operátor} \text{ CBA} \\
 \hline
 \text{EABC}
 \end{array}
 \qquad
 \text{operátor} \in \{ +, - \}$$

nalezne všechna možná řešení úlohy (všechna možná dosazení číslic za symboly A, ..., F, pro něž bude výraz splněn).

125. Ve vhodném programovacím jazyce zpracujte jednoduchý výukový program, kterým plně demonstujete nesplnitelnost (nemožnost nalezení řešení) tzv. Eulerovy úlohy (sedm mostů města Královce)

- a) klasickým grafovým algoritmem (tahem v grafu),
 b) procházením stromu řešení úlohy.

126. Ve vhodně zvoleném programovacím jazyce zpracujte jednoduchý výukový program, který plně demonstruje nalezení řešení následující úlohy: Jde o převedení hlavolamu "8" (při respektování pořadí aplikace produkčních pravidel uvedeného na přednášce)



metodou prohledávání grafu s využitím heuristické funkce

$$\begin{aligned} \hat{f}(n_i) &= \hat{g}(n_i) + \hat{h}(n_i), \\ \hat{g}(n_i) &= d(n_i) \quad (\text{délka cesty z } n_0 \text{ do } n_i), \\ \hat{h}(n_i) &= P(n_i) + 3Q(n_i), \quad \text{kde} \end{aligned}$$

- $P(n_i)$ je součet vzdáleností každého kamene hlavolamu od svého cílového místa (v možných posuvech),
- $Q(n_i)$ je míra porušení pořadí kamenů zahrnutá tak, že
 - přičítáme hodnotu 2 za každý kámen nenacházející se ve středu pole a jenž není následován správným kamenem,
 - za kámen ve středu pole přičítáme 1.