

Zápočtové příklady z Prologu

Jan Hric

10. dubna 2012

Zápočtové příklady z prologu: v.15 9.04.2012 - ... v.14 10.04.2011 - ... v.13 27.05.2010 - ... v.12 10.06.2009 v.11 30.11.2006 - ... v.10 16.11.2004 - ... v.9 21.11.2003 - ... v.8 25.11.2002 - 28.11.2002 v.7 22.11. - 7.12.2000, v.6, 11.11'99 - 26.11. v.5, 11.11.1998 v.4, 7.,10.11.1997 v.3, 1.11.1996 v.2, 13.11.1995 v.1, 25.11.1991

Podmienky na zápočet: odovzdať okomentovaný program s (primerane veľkými) testovacími dátami a dokumentáciu. V programe sú zapínateľné ladiace/trasovacie tlačie, ktoré zviditeľňujú chovanie programu a umožňujú ho skontrolovať. Dokumentácia obsahuje meno autora, zadanie, meno použitej implementácie, popis riešenia, program a datových štruktúr, popis I/O (volania a výstupu) s príkladmi, podrobnejšie viz samostatný súbor na mojej WWW stránke (<http://kti.ms.mff.cuni.cz/hric/> - > výuka - > dokumentácia). Program má (obvykle) obsahovať hlavný predikát (popísaný v dokumentácii) pracujúci "dávko" a testovacie dáta. Používajte štandardný Prolog (napr. SWI, Eclipse, Sicstus alebo BinProlog).

Ak nie je v zadaní podrobnejšie určená (alebo odporúčaná) reprezentácia dát, je jej vymyslenie súčasťou úlohy (ale napr. grafy rozoberieme na cvičení). Vyberte si jednu z variánt a), b), ... jednotlivých príkladov, ale počítajte s tým, že pri veľkom záujme môžete dostať inú variantu. Niektoré príklady alebo ich varianty sú označené ako (príliš) ľahké (pomocou (1)) a nie sú vhodné ako zápočtové. Ľahke/zjednodusene varianty slúžia (aj) ako navod na prvú verziu programu a ukazujú, ako sa dospelo k "spravne" tazkej variante.

Tento výčet nie je úplný. Vlastnej iniciatíve sa medze nekladú: ak máte inú rozumne ťažkú/ľahkú úlohu, rozmyslite si jej (presné) zadanie a navrhnete mi ju. (Taketo úlohy sem doplnam.) Primerane tazka úloha by mala obsahovať:

- netrivialne datove štruktúry (nielen zoznamy alebo matice) špecifické pre problemovú domenu, buď vstupne (interpret, spracovanie jazykov) alebo vnútorne
- netrivialny algoritmus (na ktorom sa niečo naučíte), ktorý nie je prepisom vzorcov z učebnic
- primeranu parametrizáciu z reality, nema to byť (len) matematicka abstrakcia problemu. Realizovane

pomocou predavania konfiguracie alebo kontextu ako parametru.

- overitelne a zdokumentovatelne vysledky; nielen najde sa *nejaky* vystup alebo pouzije sa *nejaka* heuristika.

Vstupy predpokladajte v štrukturovanej podobe (!; syntakticka analiza zadarmo), symbolickej (nie retazce). a obvykle davkovej. Vystupy vyrobte primerane informacne, obvykle znakové (ich graficka "krasa" nie je to podstatne). V niektorých kompaktných alebo vypočtově narocných úlohách je vhodné naprogramovať ladiace tlačie, ktoré zdokumentujú priebežné chovanie vašeho algoritmu (napr. u rozvrhovacích problémov alebo hier a hlavolamov). Úloha by mala byť kompaktná, vhodné nie sú knižnice mnohých (nepreviazaných) procedúr (napr. podpora pre prácu s kombináciami, permutáciami a variáciami). U jednoduchých úloh je možné sa dohodnúť na naprogramovaní konkrétneho netrivialného algoritmu.

Čísla udávajú mnou odhadnutú zložitosť:

(0),(1) - príliš ľahké na zápočet (Alebo nechcem, aby ste úlohu riešili v danej podobe. Často je to varianta, ktorá je "málo" parametrická)

(2) ľahké - (4) ťažké

Ak nie je uvedené nič, je úloha vhodná ako zápočtová.

Možná realizácia testovacích dát a demo predikátu:

demo:-

```
data(X),my_main(X,V),write(in(X)-out(V)),fail.
```

Programy by mali umožniť trasovanie (na žiadosť), tj. vypisovanie nejakých medzivýsledkov, ktoré charakterizujú postup výpočtu.

ZADANIA PRÍKLADOV:

KAPITOLA 1: Datové štruktúry.

121. Analýza podgrúp.

a) Pre danú (konečnú) grupu nájsť všetky jej pod-grupy. Pre každú podgrupu určiť množinu generátorov a ďalej určiť ich vzájomné inklúzie.

b) pre iné algebraické štruktúry

b1) napr. teleso

122. Grupový izomorfizmus.

Pre dve dané grupy zistiť, či sú izomorfné. Vydať

a) (1) jeden izomorfizmus

b) všetky izomorfizmy (v datovej štruktúre), backtrackingom, ale s inteligentným dopočítavaním

b1) nájsť odpovedajúce si generátory a vydať len ich korešpondencie

b2) nájsť rády prvkov a skúšať izomorfizmus len pre prvky rovnakého rádu

123. Generovanie konečných štruktúr. (2002)

Je daná báza (konečná, výčtom), nad ktorou máte generovať štruktúru. Sú dané niektoré "obvyklé" požiadavky: asociativita operácia, symetria, existencia (lavej, pravej) jednotky, distributivita ...

a) Máte vygenerovať všetky štruktúry, ktoré splňujú požiadavky a zistiť ich počet.

b) vygenerovať a spočítať len neizomorfné varianty.

KAPITOLA 2: Grafy.

205. Ofarbenie grafu.

a) (0) Minimálne, prehľadávaním.

b) (1, cvicenia) Heuristické, [1]. Dostatočne zložitá heuristika.

c) (1) Pomocou nezávislých množín.

V grafe sa nájde nezávislá množina, jej vrcholy sa ofarbujú jednou farbou a potom sa ofarbujú (indukovaný) podgraf na zostávajúcich vrcholoch.

222. (1) Hranová súvislosť.

Pre daný graf G určite jeho hranovú súvislosť.

230. (1) Dosažiteľnosť v hypergrafe.

Orientovaný hypergraf je zadaný dvoma množinami: množinou vrcholov V a množinou hyperhrán H . Každá hyperhrana h je aspoň dvojprvková podmnožina V , v ktorej je určený jeden výstupný vrchol, ostatné vrcholy hyperhrany sú vstupné.

Dosažiteľnosť vrcholu z danej počiatkovej množiny M vrcholov je definovaná takto: Každý vrchol z množiny M je dosažiteľný z M . Ak v_1 je výstupný vrchol hyperhrany h_1 a všetky vstupné vrcholy h_1 sú dosažiteľné z M , potom je v_1 dosažiteľný z M ; hovoríme, že hranu h_1 sme použili. Minimálny počet použitých hrán, pomocou ktorých je v_1 dosažiteľný z M , je vzdialenosť vrcholu v_1 od M .

Úloha: Vstup je orientovaný hypergraf G a počiatková množina vrcholov M . Nájdite všetky vrcholy G , ktoré sú dosažiteľné z M a vydajte nejakú množinu dosvedčujúcich hrán. (Pôvodne: Určite horný odhad vzdialenosti vrcholov od M .) Výsledok vráťte ako datovú štruktúru. (Hľadáme len horné odhady, pretože hrana sa môže započítať viackrát.) (Použite napr. variantu Dijkstrovho algoritmu)

231. (1) Vzdialenosť v hypergrafe.

Ohodnotený orientovaný hypergraf je zadaný dvoma množinami: množinou vrcholov V a množinou hy-

perhrán H . Každá hyperhrana h je aspoň dvojprvková podmnožina V , v ktorej je určený jeden výstupný vrchol (ostatné vrcholy hyperhrany sú vstupné), a je ohodnotená cenou, t.j. nezáporným reálnym číslom.

Cena dosiahnutia vrcholu z danej počiatkovej množiny M vrcholov (ďalej len cena vrcholu) je definovaná takto: Každý vrchol z množiny M má cenu 0 (vzhľadom k M). Ak v_1 je výstupný vrchol hyperhrany h_1 s cenou c_{h_1} a všetky vstupné vrcholy v_i hrany h_1 majú postupne ceny c_i , potom cena v_1 cez hranu h_1 je súčet cien c_i a ceny c_{h_1} . Cena vrcholu v_1 je minimum ceny cez všetky použiteľné hrany, v ktorých je v_1 výstupný vrchol. Hrana je použiteľná, ak všetky vstupné vrcholy majú stanovenú cenu.

Úloha: Vstup je ohodnotený orientovaný hypergraf G , počiatková množina vrcholov M a cieľový vrchol V . Zistite, či je vrchol V dosažiteľný a ak je, určite jeho cenu (resp. horný odhad) a vráťte zoznam použitých hrán (v nejakej datovej štruktúre) (použite napr. variantu Dijkstrovho algoritmu)

232. Presná vzdialenosť v ohodnotenom hypergrafe.

Zadanie hypergrafu ako v príklade 231, ale nechceme horný odhad, ale presnú hodnotu ceny vrcholu, t.j. hrany použité viackrát (pre medzivrcholy) sa počítajú len raz. Návod: zobecnite počítanie z vzdialenosti jedného vrcholu na vzdialenosť množiny vrcholov.

232b. (1) Presná vzdialenosť v hypergrafe.

To isté, ale hrany nemajú ohodnotenie a zaujíma nás len ich minimálny počet.

240. Izomorfizmus orientovaných hypergrafov.

Vydať všetky izomorfizmy hypergrafov. Hľadať rozkladom, t.j. vrcholu môžu byť v odpovedajúcich si triedach rozkladu, ak majú rovnaký počet vstupných aj výstupných hyperhrán rovnakého stupňa. Rekurzívne zjemňovanie, keď stupeň (celkový počet vrcholov) je nahradený počtom prvkov v jednotlivých triedach.

240b. dtto, ale iný vhodný algoritmus (nie tupý backtracking)

241. Izomorfizmus neorientovaných hypergrafov.

To isté, ale pre neorientované hypergrafy.

242. Izomorfizmus rozkladom pre iné štruktúry

Berú sa do úvahy konkrétne vlastnosti štruktúry, napr. závisí na poradí vrcholov (u bin. op. v grupe), je viac druhov operácií (...)

a) grupy

b) stromy (1)

250. Toky v sitích.

Je dan orientovaný graf a jeho dva vrcholy zdroj a stok. Úlohou je nájsť maximálny tok medzi zdrojom a stokom. a) (1) Libovolný alg. b) (1) Jako zlepšující cestu použít tu s největší propustností b1) Uspořádat globálně vrcholy (např. podle vzdálenosti od stoku) a další

vrchol vyberať najbliž celi. b2) Usporiadať vrcholy lokálne (napr. podle propustnosti hran) a procházet vrcholy cyklicky. c) Jako zlepšující cestu použít nějakou nejkratší (vytvorit podsít pouze pro hrany z nejkratších cest, postupne zvetšovat delku. - Dinic) d) Goldbergov alg. e) Alg. tří indů: (ad Dinic) Naplnění vrstvené sítě od vrcholu s nejmenší rezervou.

260. Generovanie konečných struktur.

(aj inde?) Pre daných n prvkov (nosič) generujete všetky algebraické štruktúry daných vlastností, napr. grupoidy, grupy, (čiastočné) usporiadania, grafy ...

a) s vylúčením izomorfných

b) (1) bez vylúčovania izomorfných

c) Navrhnuť jazyk na popisovanie vlastností relácií/operácií, pomocou ktorého by sa zadávali požiadavky na štruktúry. Príklady vlastností: asociativita, symetria, reflexivita, (slabá) antisymetria, (jednostranná) distributivita, existencia (jednostrannej) jedničky, idempotencia ...

(d) preformulovať do omeďzujúcich podmienok)

261. Testovanie konečných štruktúr.

(automatické dokazovanie) Pre konečné štruktúry vygenerované v príkl. 260c) testovať, či majú ďalšie požadované vlastnosti. Tie sú zadané analogickým jazykom.

262. Grupy izomorfizmov.

a) grafu

b) nejakých konečných štruktúr (viz 260)

Pre (každú) vygenerovanú štruktúru určiť a analyzovať jej grupu izomorfizmov. Analýza: generátory grupy, rozložiteľnosť ...

270. Hľadanie Hamiltonovskej kružnice, viz 360 (2011)

a) (1) naivne, tj. jeden pokus

b) heuristicky, viz 360

c) reštart, skúšam viac pokusov (s roznyimi vstupnými/úvodnými dátami)

271. Hľadanie Hamiltonovskej cesty (2011)

a), b), c) ako 270, ale koncové vrcholy nie sú spojené

272. Hľadanie obchodného cestujúceho pomocou Hamiltonovskej kružnice (2011)

a) Nechávam najlacnejšie hrany v grafe, kým nachádzam kružnicu. Implementačne: rozne heuristiky

273. Hľadanie obchodného cestujúceho zdola (2011)

a) V grafe hľadám hamiltonovskú kružnicu, ak nemožem pokračovať, pridám najlacnejšiu hranu z vrcholu a reštartujem. Rozne implementácie.

b) Štartujem v grafe s vrcholmi stupňa aspoň 2.

274. Obchodný cestujúci, 3-opt. (2011)

a) Na začiatku náhodná kružnica. Krok: z aktuálnej kružnice sa odstránia 3 hrany a ich koncové vrcholy sa

znova správne spoja. Ak je kružnica lepšia, pokračuje sa z nej. Vhodné koncové podmienky.

a2) var: Odobrané hrany sa spoja najlepším možným spôsobom.

b) Ďalšie heuristiky: b1) reštart, b2) simulované žihanie, b3) 2-opt. b4) k-opt.

c) lokálne vylepšovanie

275. Obchodný cestujúci v špec. grafoch (2011)

a) klastrované grafy: úloha ako b), grafy sú tvorené klastrami, ktoré navzájom oddelené.

a2) heuristika: odstránia sa dlhé hrany (než vhodný práh) a spájajú sa pomocou TSP komponenty. Komponenty sa riešia presne, ak sú dosť malé, alebo rekurzívne. (V komponentách rozne prahy, "outliers" - vrcholy vzdialené od zvyšku komponenty)

b) bez trojuholníkovej nerovnosti: navrhnuť použiteľnú(-é) heuristiku(-y)

276. (?) Generátor testovacích grafov (pre TSP, HAM)

a) Navrhnuť vlastný jazyk pre popis generátora

277. Hľadanie dlhej kružnice (2011)

a) zväčšovaním, beam-search (súčasne si pamätám k najlepších riešení)

280. Hľadanie častých podgrafov (2011)

Myslí sa ohodnotených: a) vrcholovo, b) hranovo, c) oboje

Motivácia: Hľadanie častých chemických podskupín

281. Hľadanie častých podstromov (2011)

Motivácia: XML-data

282. Hľadanie častých podsekvencií. (2011)

Motivácia: bioinformatika. Záleží/nezáleží na poradí: u - v,x - z znamená, že v,x je za u a pred y, ale môžu byť v ľub. vzájomnom poradí

b) vnorenia DAGu

KAPITOLA 3: Hříčky.

314. Oslík v záhradke.

Počiatočný stav:

```

xxxxxxxxxxxxx
x+---+---+---+x
x+---+---+---+
x| 0 | +-+
x+---+---+---+
x+---+---+---+x
xxxxxxxxxxxxx

```

resp.

```

xxxxxxx
x11223x
x4456--
x4457--
x88990x
xxxxxxx

```

Úlohou je nájsť postupnosť ťahov, ktoré dostanú veľkú kostku označenú O (oslíka) úplne doprava doprostred, aby sa dala vysunúť z plánu. Počas práce sa všetky časti musia pohybovať vo vyhradenom mieste 5 x 4.

315. Polohové hlavolamy (2004)

A) Polohové. Zahrňuje Oslíka (314), Lloydovu 15 (311), Rubikovu kocku (312), ?n-cube, tri kruhy v rovine. Hlavolamy majú pevný počet miest a medzi nimi sa posúvajú "kostičky", obecné rôzneho tvaru a polohy. (Prázdne miesto je tiež kostička.) Zadáva sa popis hlavolamu (tvar, povolené (alebo zakázané) ťahy a ich podmienky), počiatkové a koncové pozície.

z) Varianta s interface: pretože váš program by mal byť trochu obecný a obsahovať nejaký interface, tak primeraná varianta je navrhnúť program tak, aby bol schopný riešiť dva rôzne hlavolamy, ktoré sa popíšu cez spoločný interface (poč. pozícia, koncové pozície - podmienka, povolené ťahy). Realizujte d).

a) dosažiteľnosť (množin) pozícií. (Vhodné je Iteratívne prehlbovanie.)

b) dosažiteľnosť pozícií na daný počet ťahov.

c) heuristické prehľadávanie. (napr. v kombinácii s beamsearch)

d) prehľadávanie s dopomocou: užívateľom sú dané vhodné! priebežné medzipozície (neúplne popísané). Ich možné využitie: Dosiahnutú pozíciu pridám do množiny počiatkov. Prvý! prechod cez pozíciu (typ pozície) skraca cestu (zmenšuje cenu). Hladovo zablokujem ďalšie riešenia (neúplné prehľadávanie), prípadne sa k nim vrátim.

B) Skladacie: 1) Pentomino (332) - navrhnúť s heuristikou, viz aj 337. (2/ Povrchy 3D kociek 5x5x5 z nepravidelných blokov 5x5.) 3) Skladanie štvorcov 2D s okrajovými podmienkami (náväznosti), aj 3b) obojstranne. 4) Dtto, nepravidelná sieť (posuny o 1/3). 5) Šesťuholníková sieť. 0) Globálne podmienky pre jednotlivé hlavolamy - typicky vytvorenie obrazca, napr. vytvorenie dlhej (spojitej) cesty.

x1) zobecnit aj na pentomino, viz aj 337.

C) Rozvíjacie hlavolamy. Požiadavka je odstraňovať symetrie, tj. izomorfizmy. 1) skladanie 3D povrchu 4D-kocky. 2) 3D hady, štandardne z 27 kostičiek; aj iné veľkosti. Počet možností poskladania daného hada. Rozvinutie do plochy.

D) Permutačné hlavolamy. e) nájsť (krátke) makroťahy - premiestnia len málo kostičiek. 1) Rubikova kocka. 2) n-Cube 3) tri kruhy. 3) dva kruhy 4) Valce, aj abstraktné. Plné, tj. posúvacie. S voľným miestom, tj. uschovanou kostičkou.

E) 3D skladačky/rozoberačky. Aká je vnútorná štruktúra neviditeľných častí hlavolamu, aby šiel rozobrať,

ale nerozpadol sa. 1) Maltézsky kríž 336, varianty.

F) Sokoban.

G) Dve Rybičky.

316. Drôtové hlavolamy. (2004)

So šnúrkou. Na základe 2D premietnutia drôtového hlavolamu rieši vytiahnutie a navlečenie šnúrky.

b) Hlavolam zjednoduší, vyrieši a výsledné riešenie upraví pre pôvodný hlavolam. Úprava systematicky (napr. si uschovám delenie hrany) alebo (lokálnym) dohľadám.

317. Rush Hours (2012)

Hlavolam. Obdĺžniková doska je delena na stvorce a obsahuje neprekryvajúce sa dominové kocky 1x2 a 1x3 (reprezentujúce auta). Tie sa smú po plane pohybovať len v smere dlhšej strany. Cieľom je previesť urcenu kostičku z jednej polohy na plane do inej, typicky od steny ku stene. Na zaverenej polohe ostatných kostičiek nezaleži.

b) Previesť na minimalny pocet tahov.

330. Výroba krížovky.

a) Je zadaný slovník, tajnička (a prípadne jej poloha), tvar krížovky (prípadne neúplný), úlohou je zostaviť krížovku.

b) (Kompilácia krížovky) dtto, štruktúra krížovky je presne určená. Úlohou je „skompilovať“ tvar krížovky na prologovský cieľ, ktorý obsahuje len volania slovníka a testy rovnosti písmen na zhodných miestach.

Druhy krížoviek:

A) normálna, s oddeľovacími čiarami

B) s oddeľovacími políčkami

C) slabiková

D) s diagonálne delenými políčkami

E) ... a iné

332. Pentomino.

Daná plocha sa má presne pokryť danými obrazcami.

a) (1) Typicky je plocha delená na štvorce a pokrýva sa (dvanástimi) obrazcami z piatich spojených štvorcov. Obrazce sa môžu otáčať a prevracať.

b) dtto, ale tvary obrazcov sú súčasťou zadania

c) dtto, trojuholníková štruktúra plochy

d) dtto, šesťuholníková štruktúra plochy

e) priestorové pentomino 3D

333. Tangram

a) (1) Ako pentomino, ale obrazce sú štvorce a ich diagonálne oddelené poloviny. Pre pevné tvary obrazcov.

b) Tvary obrazcov sú súčasťou zadania.

c) Obrazce a tvary sú aj obdĺžniky 2x1 a ich diagonálne oddelené poloviny.

334. Ethernity

a) Úloha ako pentomino. Plocha a obrazce sú dané trojuholníkmi a ich polovicami.

334-II. Ethernity II

a) Plocha (štvorcová) je delená na štvorce (16x16) a na okraji je ofarbená (šedou). Voľne položitelné štvorce sú ofarbené na stranách a úlohou je poskladať ich, aby susediace farby súhlasili. Okrajové štvorce majú 5 farieb, vnútorné 17. (Jeden štvorček, na (8,8), je určený napevno)

b) Navrhniť heuristiky. b1) Beam-search b2) lokálny update

335. Maltézsky kríž.

a) Pre daných 6 častí zistí, či môžu priestorovo tvoriť maltézsky kríž.

b) dtto, ale jednoznačne, až na symetriu

c) dtto (a) + zistí, či sa dá maltézsky kríž rozložiť na jednotlivé časti.

Časti sú 2x2x6 až 2x2x10, v ktorých chýbajú celé kocky. Pohľad nižšie. Časť 6 je schovaná za 5.

1122

1122

1332

55533555

55544555

1442

1122

1122

336. Sudoku

Doplniť na miesta tabuľky čísla 1 až 9, resp. obecne od 1 do $m * n$, tak aby v každom riadku, stĺpci a oddelenej skupine boli všetky čísla v danom rozsahu, a to práve raz.

a) klasické sudoku 9 * 9 - ľahké

b) štvorcové obecných rozmerov $m * n$ v obidvoch smeroch. V zvislom smere sú riadky rozdelené na skupiny po m riadkoch, vo vodorovnom smere sú skupiny n stĺpcov. Každá skupina, stĺpec a riadok obsahuje $m * n$ polí a musí obsahovať všetky čísla od 1 do $m * n$.

Variety prehľadávania.

1) Statické usporiadanie premenných. Poradie premenných k ohodnoteniu sa určí staticky dopredu (heuristicky). Heuristika uprednostňuje miesta, kde môže byť menej hodnôt a miesta, ktoré majú viac už určených susedných hodnôt.

2) Statické, so zrefazením: viz 1) ale vybrané miesto musí byť susedné aspoň k jednomu predtým vybranému miestu. Okrem prvého miesta.

3) Statické usporiadanie s odhadmi kolízií: ako minule, ale pri výpočte počtu možných hodnôt na určitom mieste sa odhadne, s akou pravdepodobnosťou bude dané číslo inde (pri rovnomernom rozložení). Viz príklad 337.

4) Dynamicky: Vyberá sa premenná, ktorá má aktuálne najmenej možných hodnôt na výber. Pre vybranú premennú sa do susedov (cez nerovnosť) pribežne propagujú zakázané hodnoty. (Použitelné na farbenie grafu a podobné úlohy, s omedzenými doménami a/alebo predvybranými hodnotami premenných.) Výber premennej - najviac omedzená, výber hodnoty - najmenej omedzujúca.

Výber podľa heuristiky (aj pre iné úlohy).

Výber premennej: prvotné kritérium: najviac omedzená premenná (má najmenej hodnôt k výberu), druhotné kritérium: najviac omedzujúca premenná (súvisí s najviac susednými premennými), ...

Výber hodnoty: najmenej omedzujúca hodnota.

(Aplikácia na izomorfizmus grafov: staticky prvotne: vrchol má najviac už vybraných susedných vrcholov, druhotne: vrchol má najviac nevybraných susedných vrcholov, hodnota: ?; dynamicky: trieda ekvivalencie (podľa susednosti s už určenými vrcholmi) má najmenej vrcholov, druhotne: niektorá - minimálna - susedná trieda má málo vrcholov)

c) Heuristické prehľadávanie - s odstraňovaním konfliktov, neúplné. (Použiteľný aj beamsearch.) Nahradzujeme (jednu) najviac konfliktnú hodnotu najmenej konfliktnou. (Prípadne lokálne dohľadávanie.) Vhodné pre podšpecifikované úlohy (tj. ktoré majú veľa riešení)

Konflikty dvoch druhov: i) neprípustné riešenie - dve rovnaké premenné, ii) neúplné riešenie - chýbajúca hodnota.

Heuristika pre sudoku: 1) v stĺpcoch mám permutácie, udržiavam ich transpozíciami, 2) na miestach mám množiny hodnôt, pri vzniku neúplného riešenia pridám hodnotu(-y) do bloku, riadku, stĺpca.

337. Sudoku, statický výber (a podobné).

a) ad 336 b3). Výpis štatistik - dôvodov výberu.

b) hľadá sa lokálne (hladovo) najlepšie usporiadanie pri danom výhlade - usporiadanie lexikograficky. Varianta: globálne najlepšie (vylučuje symetrie a prehľadáva až do zrušenia symetrie).

338. Úlohy typu CSP.

Spôsoby riešenia: viz sudoku 336.

a) Pre iné (diskrétné) úlohy tohoto druhu (CSP - Constraint Satisfaction Problem). Farbenie grafu, splniteľnosť SAT (, dámy na šachovnici: $n=1000$), krížovka, latinské a magické štvorce??, zebra, rozvrhovanie (lietadiel - dynamicky, s pretečením a preplánovaním, seštričky - pevný rozvrh, integrované obvody na plochu) a plánovanie (stavby, s omedzeniami, pájanie plošných spojov, ukladanie súčiastok). TSP - obchodný cestujúci.

b) Obecné prostredie s interface, pre podobné úlohy.

c) Algoritmus odstraňovania konfliktov. Heuristický.

340. NIM.

Z daných n kôpok zápaliek odoberajú dvaja súper. Úlohou je nájsť a odohrať optimálnu stratégiu pre nejakú variantu hry. Varianty hry sa líšia ukončením a povolenými ťahmi:

- 1) kto nemá ťah, prehral (obvyklá varianta)
- 2) kto tiahol posledný, prehral

Povolené ťahy (vždy musíme vziať aspoň jednu zápalku):

- a) smie sa vziať ľubovoľný počet zápaliek z jednej kôpky
- b) smie sa vziať najviac n zápaliek z jednej kôpky
- c) smie sa vziať práve i zápaliek z jednej kôpky, možnosti pre i sú dané výčtom
- d) $a + b$ smie sa vziať zhodný počet záp. z dvoch kôpok
- e) iné druhy ťahov, zadané externým programom "Stav - > Stav", resp. "Stav" - > "Množina_následníkov"

Nájsť optimálnu stratégiu znamená nájsť (všetky) prehrávajúce pozície: ak som na ťahu, tak vždy prehrám.

i) Pre program "všetky pozície" znamená všetky menšie než zadaný počiatočný stav.

ii) Program má nájsť výčtom n "najmenších" prehrávajúcich pozícií.

iii) Tabelačia pre pozície s najviac N (vstup) zápalkami na najviac K kôpkach (inteligentné hľadanie bez opakovania, výstup bez symetrií).

iv-a) pre vyhrajúce pozície vydat všetky ťahy, ktoré vedú k vyhre najrychlejšim spôsobom a počet nutných ťahov k víťazstvu.

iv-b) pre vyhrajúce pozície vydat všetky možné ťahy a ich charakteristikou: vyhrajúci/prehrávajúca počet ťahov k víťazstvu.

iv-c) pre prehrávajúce pozície vydat všetky ťahy a ku každému (najväčši) počet ťahov k prehre.

Napr. pre $b1$ s jednou kôpkou sú preh. pozície $k \cdot (n+1)$ pre prirodzené k .

Varianty:

I) povolené ťahy sú dané výčtom, ak niekto nemôže hrať, je to remíza

II) ťah môže odoberať zápalky z viac kôpok.

Upozornenie: Vybraná zápočtová úloha sa zostaví z poskytnutých možností (skoro) ľubovoľne, ale musí byť odstoatočne parametrická.

341. Generické hľadanie optimálnej stratégie.

a) Po doplnení procedúr možný_ťah, vyhrajúca_pozícia, prehrávajúca_pozícia ... s vhodnými parametrami hraje optimálnu hru za jedného z hráčov.

b) dtto, ale tabelačia prehrávajúcich pozícií

342. Mastermind alias logik.

Logik je hra, keď počítač hádá rozloženie farebných kolíčkov v N daných zdierkach. Možných farieb je K a

1) môžu, 2) nemôžu sa opakovať. Odpoveď na každý tip sú dve čísla: počet (farebne) správne uhádnutých kolíčkov na správnych miestach a počet správne uhádnutých kolíčkov na nesprávnych miestach.

a) (1) Program háda. (Každá ďalšia voľba je možné riešenie vzhľadom k minulým výsledkom)

b) (1) Program háda aj vyhodnocuje. Časť, ktorá vyhodnocuje, je inteligentný podvodník: vyberie si takú (správnu) odpoveď, ktorá poskytuje najväčší počet možných riešení. Vydávať nejakú skontrolovateľnú štatistiku o riešeniach.

b1) (4?) dtto (b), ale vyberie si odpoveď, ktorá zaručuje najdlhšie hádanie súpera. (Uvedomte si, že sa to nemusí zhodovať s odpoveďou podľa b)

351. Zakódované obrázky (2001).

a) Na štvorcovej sieti je čiernobiely obrázok. Pre každý riadok a stĺpec obrázku je daný popis, ktorý obsahuje počty súvisiacich čiernych pixelov v príslušnom riadku alebo spĺpci. Jednotlivé skupiny čiernych bodov sú oddelené aspoň jedným bielym. Úlohou je z daného popisu zrekonštruovať všetky vyhovujúce obrázky.

a1) Navyiac sa skupiny nesmú dotýkať rohom. (podľa hry Lode)

b) Pre šesťuholníkovú sieť

c) Pre trojuholníkovú sieť

d) Pre 3D sieť.

352. Zakódované obrázky s trojuholníkmi (2006)

Varianta k 351: Okrem čísel označujúcich plné štvorce sú zadané aj diagonálne oddelené polovice štvorcov (vrátane orientácie - tj. napr. horný pravý trojuholník.) Tieto trojuholníky musia tiež súhlasíť v oboch smeroch.

353. Farebné zakódované obrázky (2006)

varianta k 351: Obrázky môžu byť viacfarebné, v riadkoch a stĺpcoch sú počty susedných zhodne ofarbených bodov, v poradí zľava, resp. zhora. Medzi roznyimi farbami v obrázku nemusí byť voľné políčko, ale môže byť.

354. Iné (japonské) hlavolamy - z wikipedie (2011)

zz) aj nejaponské, zx) Nikoli: http://en.wikipedia.org/wiki/List_of_Nikoli_puzzle_types#Nikoli

a) Kakuro - číselná krížovka, b) Killer sudoku - kombinácia Sudoku a Kakuro, c) ...

354a. Kakuro geneticky (2011)

Fitness je rozdiel, hľadanie je genetickým alg.

b) iné vhodné hlavolamy geneticky

355. (Obecný riešič hlavolamov) - tj. omedzení

356. Splniteľnosť CNF.

Riešič CNF - konjunktívnej normálnej formy. Je zadaná výroková formula v CNF (var. 2: alebo obecná formula) a program hľadá ohodnotenie, pri ktorom je formula pravdivá. Heuristiky: viz 336.

a) jedno ľubovoľné b) s maximálnym počtom jednotiek

Heuristiky: jednoznačne určené premenné (len pozitívne alebo len negované výskyty; klauzula dĺžky 1), podľa počtu výskytov premennej, dĺžky klauzuli (kratšie su viac omeďujúce), kombinovane ... , (podľa najproblematickejšej klauzule - a v nej najlepšiu premennu, aj pravdepodobnostne), (podľa historie - najviac problematicka premenná) ...

d) genetické riešenie, fitness je počet nesplnených klauzulí

357. Riešenie QBF - Quantified Boolean Formula (2011)

Zadanie je v CNF, ale výrokové (boolovské) premenné sú v prefixe kvantifikované (univerzálne alebo existenčne).

a) riešič

a2) heuristický, a3) "diskrepančný"

b) zápis a overenie riešenia, (zápis kompaktný a inkrementálny)

c) zápis a overenie neexistencie riešenia

358. Generátor CNF/QBF (2011)

a) CNF, b) QBF

generuje: c) ťažké formule, d1) náhodné, d2) čiastočne náhodné, d3) štrukturované, d4) štrukturované so zmenou, e) rekurzívne štrukturované, f) riešiteľné ...

Ideálne: vypisovať v Dimacs, resp. QDimacs formáte

360. Hamiltonovská kružnica (2009)

Naprogramujte heuristické hľadanie Hamiltonovskej kružnice. (Default heuristik: a: h1, h4, h3) Základná metóda je predĺžovanie cesty (otočením): ak z posledného vrcholu v_k nenájde dostupný nepokrytý vrchol u , potom hľadáme v_i , t.j. v_i, v_k je hrana a v_{i+1}, u je hrana. Nová cesta je $v_0, \dots, v_i, v_k, \dots, v_i + 1, u$.

a) Heuristiky sú zamerané na výber v_i (h1: maximalizácia i), výber u (h2: minimálny deg. u , h3: min. adeg u - minimálny aktuálny stupeň u (t.j. len vzhľadom k nepokrytým vrcholom)), výber u (h4: max. vzdialenosť(v_0, u))

b) kontrolovať vynútené spojenia: ak vrchol y má posledných 2 neobsadených susedov, potom s nimi musí byť spojený (resp. jeho susedia s ním). b2) Heuristika na pripojenie najdlhšieho takto vynúteného retiazku. b3) vynútené predĺžovanie od v_0 (k v_{-1}).

c) Záchytné metódy: ak nedokážem pokračovať (alebo uzatvoriť kružnicu), skrátim cestu o v_0 . (napr. daný počet krát)

var. d) lokálne prehľadávanie (ala WalkSAT): V každom kroku je dostupná (virtuálna) ham. kružnica, prípadne s neplatnými hranami. Križením odstraňujem neplatné hrany: z $a-b^*-c-d^*-a$ na $a-c^*-b-d^*-a$. Ideálne vzniknú 2 platné hrany. Heuristiky na výber križenia: podľa stupňa vrcholov, podľa počtu pokusov (t.j. penalizujem hrany alebo vrcholy) d2) +beam search: z n

priebežných riešení vygenerujem (nie nutne všetky) varianty a z nich vyberiem n najlepších, a iterujem.

Testovanie: 20 sten, Petersenov graf (nemá HK), Petersenov graf s 1 vnútornou spojku (má HK).

361. Heuristiky pre iné NPÚ problémy (2009)

(Farbenie grafu), (SAT pre CNF) ... Riešenie niektorej NPÚ úlohy pomocou (primerane zložitých) heuristických metód.

Heuristiky s primárnym, sekundárnym, ... kritériom, s podmienkou, s randomizáciou

362. Generovanie "ťažkých" dát pre testovanie 360,361.

a) Napr. pre 360:HK pridávam hrany náhodne do kružnice, kým heur. algoritmus prestane nachádzať HK. Pridávam hrany len do určitého stupňa vrcholov, napr. 3, prípadne do určitej vzdialenosti. Nepridávam križenie.

Kryptografia: 370+

370. Analýza zamennej šifry (2012)

Pomocou frekvencnej tabuľky (písmen, dvojíc, trojíc, stvoríc) desifrovať obecnú zamenú šifru. Obecná zamená šifra znamená, že písmena v plaintexte a ciphertextu su zmenené obecnou permutáciou. Pre test kvality permutácie použiť (napr./doporučené) chi-kvadrat test a možné permutácie prehľadávať lokálnym zlepšovaním.

371. Analýza transpozicnej šifry (2012)

a) jednoduchá transpozícia

b) dvojita transpozícia

c) ?? šifrovacia mriežka (otočná), c2) šifrovacia mriežka preklapacia, c3) š.m. otočná a preklapacia

Pomocou frekv. analýzy (dvojíc) najst pravdepodobný plaintext a následne kľuč (transpozíciu, resp. š. mriežku). Nemalo by ísť o riešenie prechádzaním všetkých možností (prípadne len pre kontrolu).

varianty transpozície (kľuč zoradiť stĺpce podľa abecedy): a2) s doplnením-zarovnaním, a3) bez zarovnaní, a4) schodovité vyplňovanie a5) kľuč s opakovaním písmen, jednoznačne písmena v kľuči citat zvisle, opakované vodorovne

KAPITOLA 4: Hry.

Mierne varovanie: Šachové úlohy sú pomerne pracné.

401. Zohrávka šachovej koncovky K + V kontra K

402. Zohrávka šachovej koncovky K + D kontra K

403. Zohrávka šachovej koncovky K + 2S kontra K

404. Zohrávka šachovej koncovky K + V kontra K +

P

405. Riešenie šachových úloh.

a) Dvojťažky

b) n-ťažky

c) pomocný mat, exoúlohy ...

406. Generovanie šachových úloh.

Generovanie „zaujímavých“ dvojťažiek, n-ťažiek, ...
Zaujímavosť, kvalita, možnosti sú dané materiálom, neexistenciou vedľajších riešení ...

410. Radca bridžových výnosov.

Na základe pravidiel z literatúry, zadanej ruky vynášajúceho hráča a prípadne licitácie poradí najlepší výnos (prípadne niekoľko výnosov, prípadne podá zdôvodnenie)

411. Bridž - dopočítavanie kariet

a) Pre dané informácie o stole, vlastnej ruke a doterajšej zohrávke sa pokúsi dopočítať karty ostatných hráčov. Výsledok sú intervaly možného počtu kariet.

b) dtto, aj informácie z licitácie, prípadne už v predspracovanej podobe (u ľavého obrancu piky sú 5-list, kára sú dlhšie než srdcia, ...)

412. Bridžový licitátor. (4)

Na základe kariet v jednej ruke a predchádzajúcej licitácie vydá ďalšiu hlášku. (príklad interface)

Skontrolovateľné, tj. na základe pravidiel o "sile" ruky spocita vhodnú hlášku.

b) Systém, do ktorého by sa dali licitačné pravidlá dodávať samostatne.

415. Mariášový licitátor. (2002)

Ako 412 Bridžový licitátor, pre mariáš.

420. Pasiáns.

Hraje samostatne nejakú variantu pasiánsu, vypisuje povolené možnosti a zvolené ťahy.

440. (1) Dáma.

a) Vygenerovať z danej pozície dámy všetky platné ťahy.

b) Vygenerovať všetky možné skoky ľubovoľnej dĺžky.

c) (1) zjednodušené (a) - nie sú tam dámy.

d) (1) zjednodušené (b) - nie sú tam dámy.

450. Piškvorky.

Počítač hrá piškvorky. Analýzou pozície pomocou vzorov (a) alebo prehľadáním na niekoľko ťahov dopredu (b) počítač určí najlepší ťah. Nad touto časťou je hracie prostredie, ktoré umožní hrať proti počítaču alebo dvom počítačom proti sebe. Typické použitie vzorov:

a) V "databáze" je niekoľko vzorov a k nim odpovedajúce (najlepšie) ťahy a ich hodnota. Vyberá sa ťah s najlepšou hodnotou medzi všetkými vzormi alebo s najlepšou sumou hodnôt.

b) Vzor má svoju cenu a jeho výskyt v pozícii je takto ocenený. Cena (koncovej) pozície je cenou všetkých obsiahnutých vzorov. Najlepší ťah sa zistí prehľadávaním stromu metódou minimax vzhľadom k zistenej cene koncových pozícií.

451. Miny (2004)

Hm. Windowsovska NT-hra.

a) Dabelsky miner. Podsunie minu, ak tam moze byt.

b) Dokonaly pyrotechnik. Urci bezpecne policka.

c) Pravdepodobnostny pyrotechnik. Na zaklade znalosti počtu zostavajucich min urci pravdepodobnosti vyskytu.

D) Sestuholnikova siet

E) Trojuholnikova siet.

452. Sokoban (2004)

Hra. Riesenie pozicii.

b) varianty: ?sestuholnikovy, trojuholnikovy

453. Mars explorer (2004)

Robot v prekazkach alebo bludisku hlada cestu. Robot sa nedokaze obracat na mieste, ale zatacky maju dany minimalny polomer.

a) vie, tj. moze cuvati.

b) nemoze cuvati. Ale ma zadany minimalny bezpecny odstup od prekazok. (alebo tvar)

454. Stavba zeleznice (2004)

V hornatom terene (z vyskovo zadanych polygonov) staviamе trat o danom povolenom stupani a polomere zatacok. Co najlacnejsie.

460. Analýza pozície go.

a) Pre danú koncovú pozíciu go systém určí (približne) živosť skupína následne územie a spočíta výsledok.

b) Pre danú (skoro koncovú) pozíciu hry systém určí miesta, kde je ešte možné tvoriť územie (alebo ktoré sú otvorené pre boj).

c) Pre danú pozíciu ko systém spočíta druh ko: (zjednodušené) kolkokrát ho ktorý hráč musí vyhrať.

d) ...

470 ... úlohy z iných hier

Typicky koncovky sú primerane jednoduché.

471 Planet Wars (2011)

<http://ai-contest.com> Google contest 2010. a) Server: Odohranie ťahov b) (jednoduchý) Heuristický hráč

KAPITOLA 5: Interpretý a spracovanie "zdrojových" textov.

501. Analýza prologovských štruktúr.

a) Klauzule v danom vstupnom programe sa spracujú tak, že sa spočíta, kolkokrát bol ktorý atom (s četnosťou) použitý ako funkčný symbol v datovej štruktúre, ako predikátový symbol v hlave (volanie) a v tele (cieľ). Zoznam sa utriedi a vypíše.

b) dtto, ale jemnejšie rozdelenie - funkčné symboly sa rozlíšia podľa použitia v jednotlivých procedúrach. Inšpirácia z krížových referencií.

Program sa číta zo vstupného súboru (alebo je predaný cez parameter ako zoznam klauzulí).

502. Graf závislosti.

a)(1) Ak klauzula procedúry P obsahuje volanie proc. Q, tak P závisí na Q. Zistíte graf závislosti.

- b) (a) + nepriame závislosti, cyklické závislosti
- c) (a) + závislosti cez vstavané metapredikáty (argumenty setof, bagof, call, not ...)

503. Analýza výrazu s operátormi. (Syntaktická analýza)

Je daná postupnosť tokenov a zoznam infixných (binárných) operátorov s precedenciami. Tokeny sú atomy, čísla a zátvorky. Úlohou je analyzovať postupnosť a previesť ju do štruktúry termu alebo prefixného zápisu. Možná metóda: nájde sa prvý operátor s najväčšou precedenciou a vstupná postupnosť sa rozdelí na spracovanie častí pre a za operátorom. Časti výrazov v zátvorkách sa preskakujú. Spracovanie môže skončiť chybou "neanalyzovateľný výraz".

- a) (1) bez rozlišovania asociativity - vstup musí byť dostatočne uzátvorkovaný
- b) (1) so zadanou asociativitou
- c) s rôznymi druhmi zátvoriek
- d) "zátvorky" majú vnútorné symboly ("kľúčové slová"), tj. ohraničujú viac podvýrazov než jeden. Každá zátvorka začína a končí symbolom (pre jednoduchosť), takže nie sú nutné priority a asociativita. Zátvorka je napr. if - then - else - endif, a pod. K zátvorke je daná výstupná reprezentácia, tj. meno funkčného symbolu. (tzv. distfixné operátory)
- e) d+b, distfixné a binárne op., s prioritami a asociativitou

pr.?- analyzuj([op(plus,4), op(krat,2)], ['(,x, plus, x,)', krat, 4], X).

X = [krat, plus, x, x, 4]

X = [krat/2, plus/2, x, x, 4]

X = [krat, [plus, x, x], 4]

X = krat(plus(x,x),4)

... a varianty

Podkapitola: Hradlové siete.

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (v danom poradí). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete. Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe. Výsledok jedného hradla sa posiela na všetky vstupy daného mena, môže ich byť niekoľko.

Príklad reprezentácie siete pre výpočet $x*y+z+x$: [h(h(1),plus,[h(2),z,x]), h(h(2),krat,[x,y])]

, kde h(1) a h(2) sú mená hradiel, x,y,z sú mená vstupných bodov, hradlá majú rôzny počet vstupov.

512. Alokácia registrov

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (s daným poradím). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete.

Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe.

Vstup programu je hradlová sieť. Úloha je

a) (1) usporiadať hradlá tak, aby každé hradlo bolo až za hradlami, ktoré počítajú jeho vstupy.

b) priradiť vstupným bodom siete a výsledkom hradiel "registre CPU". Register sa môže použiť pre výsledok ďalšieho hradla, ak hodnotu z neho už nepotrebujete na vstupe žiadneho hradla. Výstupný register hradla je vždy odlišný od všetkých vstupných registrov. Počet registrov nie je omezený, ale snažte sa ich použiť čo najmenší počet (tj. vyhnite sa triviálnemu riešeniu, kedy výsledku každého hradla je priradený iný register).

513. (1) Prevod hradlovej siete na polynóm.

a) (1) Je dana acyklicka hradlova sieť, kde hradla majú najviac dva vstupy a jeden výstup, ktorý sa môže vetviť. Hradla su troch druhov podľa funkcie ktoru realizuju: Plus, Krat, Unarne minus. Napiste program, ktorý pre konkrétne hradlo v sieti vypocita polynom, ktorý vyjadruje závislosť výstupu zvoleného hradla na vstupe siete.

b) Varianta za a) má nevýhodu, že spoločné podvýrazy (podsiete) sa vo výslednom výraze objavia viackrát. Výstup v tejto variante bude postupnosť výrazov a ich priradení, tak aby sa aj viackrát použité hradlá spočítali len raz. Výsledok je najkratšia postupnosť priradení, ktorá spočíta výraz tak, že žiadne hradlo sa nepočíta viackrát.

514. (1) Prevod hradlovej siete na logický výraz.

a) (1) Je dana acyklicka hradlova sieť, kde hradla majú najviac dva vstupy a jeden výstup, ktorý sa môže vetviť. Hradla su troch druhov podľa funkcie ktoru realizuju: And, Or, unarne Not. Napiste program, ktorý pre konkrétne hradlo v sieti vypocita formulu, ktorá vyjadruje závislosť výstupu zvoleného hradla na vstupe siete.

b) analogicky k 513b pre logický výraz.

515. (1) Hľadanie ekvivalencií v hradlovej sieti.

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (s daným poradím). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete. Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe.

Hradlá v sieti sú ekvivalentné, ak majú zhodnú operáciu a ekvivalentné alebo zhodné vstupy. Je daná hradlová sieť. Nájdite v nej všetky ekvivalentné hradlá a vyrobte novú sieť, v ktorej všetky navzájom ekvivalentné hradlá nahradíte jedným z nich.

Návod: Hradlá najprv usporiadajte a potom prechádzajte sieť od najnižšej vrstvy.

516. (1) Prevod aritmetického výrazu na hradlovú sieť

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (s daným poradím). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete. Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe.

Vstup programu je daný výraz v tvare prologovského termu s číslami, identifikátormi a operáciami +, *, -, /, sin a cos. Vytvorte k danému výrazu hradlovú sieť, ktorá bude počítat' zhodný výraz zhodným postupom (tj. neoptimalizujte).

Máte k dispozícii hradlá s danými šiestimi operáciami. Hradlá s operáciami + a * môžu mať ľubovoľný počet argumentov a ich vstupom nesmie byť výsledok rovnakej operácie.

516b. (1) Prevod postupnosti výrazov na hr. sieť.

Podobné minulej úlohe. Prevádzate postupnosť priradení, raz spočítaný výsledok sa môže použiť niekoľko krát.

Motivácia: Jeden výraz má nevýhodu, že každý jeho podvýraz sa dá použiť len jedenkrát (implicitne na mieste výskytu)

517. (1) Prevod logického výrazu na hradlovú sieť

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (s daným poradím). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete. Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe.

Vstup programu je daný logický výraz v tvare prologovského termu s identifikátormi a operáciami and, or, not a imp. Vytvorte k danému výrazu hradlovú sieť, ktorá bude počítat' zhodný výraz zhodným postupom (tj. neoptimalizujte).

Máte k dispozícii hradlá s danými štyrmi operáciami. Hradlá s operáciami and a or môžu mať ľubovoľný počet argumentov a ich vstupom nesmie byť výsledok rovnakej operácie.

518. Prevod vyrazov do stvoric. (2004)

Kompilacia postupnosti priradzovacích príkazov (Pascal, C, C++, Java) do stvoric (tj. hradiel).

a) optimalizácia spoločných podvyrazov

b) bez optimalizácie

x) varianty

520. Interpret dataflow počítača.

Model dataflow počítača je (cyklická) hradlová sieť. Po hranách siete sa posielajú "pakety" - jednotky informácií a spracovania. Jednotlivé hradlá pracujú asynchrone, aktivujú sa, ak majú pakety na vstupoch. Hradlá obvykle požadujú pakety na všetkých vstupoch,

aby sa mohli aktivovať. Okrem „datových“ operácií sú potrebné aj „riadiace“ operácie (a riadiace hrany a pakety), napr. if - 2 vstupy a 2 výstupy - jeden vstup je výsledok podmienky a podľa neho sa druhý vstup pošle na jeden z výstupov.

Asynchrónnosť znamená, že na spracovanie na určitom vstupe určitého hradla môže čakať niekoľko paketov na spracovanie (v danom poradí).

Jeden vygenerovaný paket sa obvykle posielajú na jeden (určený) vstup. Môžu existovať deliace hradlá (posielajú jeden svoj vstup na všetky svoje výstupy), alebo delenie môže byť implicitné (ako u hr. sietí)

Úloha: naprogramovať interpret dataflow počítača a odladiť jednoduchý program, napr. faktoriál, alebo súčet vstupných paketov.

521. Petriho siete (2001)

Návrh reprezentácie, simulácia. Analýza siete: konečnosť (?), dosažiteľnosť stavu, vykonateľnosť danej množiny/postupnosti prechodov, ...

522. (Konečne) automaty (2004)

(Vhodne pre studentov 3. ročníka, ktorí už vedia, o čom sa hovorí.)

a) Simulator automatov. Reprezentácia by mala umožniť aj skladanie automatov.

b) Prevod regularneho vyrazu na konecny automat a naspat: prevod konecneho automatu na regularny vyraz.

c) prevod NKA+lambda na KA, c1) simulator NKA a KA c2) redukcia KA a prevod na normalny tvar.

530. Optimalizácia (Pascalského) príkazu.

(a la 515. Hľadanie ekv. v hradlovej sieti)

Pascalský priradzovací príkaz sa prevedie na postupnosť priradzovacích príkazov tak, aby sa spoločné podmienky vyhodnocovali iba jedenkrát a uschovali v pomocných premenných. Výraz sa zadá v tvare vhodnej prologovskej reprezentácie.

a) len pre aritmetické operácie

b) vstavané funkcie, ...

531. Prevod siete na (pascalský) výraz.

a) (1) elementárne operácie

b) selektory polí a záznamov, ...

c) (4?) funkcie ...

532. (1) Konštrukcia logického výrazu.

a) (1) Pre danú logickú funkciu o n premenných zostrojíte ekvivalentný logický výraz (zo spojiek and, or, not).

a1) z daných spojiek (nand alebo nor stačí)

b) dtto, minimálny

c) dtto (b), ale funkcia nemusí byť daná úplne, niektoré hodnoty môžu byť nevýznamné a je možné si ich zvoliť, aby výsledok bol čo najmenší.

533. (1) Analýza logickej siete.

Pre danú logickú sieť s and a or hradlami a s jedným výstupom hľadám hodnoty vstupov, pre ktoré je výstup pravda. Program vydá výsledok v datovej štruktúre (nie backtrackingom)

- a) všetky možnosti
- b) minimálne možnosti
- c) sieť obsahuje aj not, ... (alias prehľadávanie AND-OR grafu)

540. L-systémy

Generovanie a interpretácia L-systémov na základe popisu. [2]

541. Riešič podmienok.

a) (4) Sú zadané podmienky v tvare rovností a nerovností na premenné a čísla. Premenné majú konečný obor hodnôt. Riešič hľadá možné riešenia pre dané podmienky tak, že propaguje známe informácie o jednotlivých premenných cez podmienky na ostatné premenné. Až nedokáže propagovať ďalšiu informáciu, zvolí si premennú a podľa nej vykoná rozbor prípadov.

b) v podmienkach môžu byť aritmetické operácie

c) Premenné v podmienkach sú logické - majú 2 možné hodnoty.

Použitie: riešenie algebrogramov, 8 dām na šachovnici, zostavovanie krížovky

542. Sústavy omedzení v konečných doménach

Omedzenia môžu byť troch druhov a sú dané v tvare termov:

1) obor($\{premenná_i\}$, $\{obor_hodnôt_i\}$) kde premenná je atom reprezentujúci meno premennej a obor_hodnôt je množina elementárnych hodnôt, ktoré môže daná premenná nadobudnúť. Každá premenná má práve jeden obor hodnôt, ktorý môže byť aj jednoprvkový.

2) zhodne($\{množina_premenných_i\}$) kde všetky premenné musia nadobudnúť tú istú hodnotu. Premenná sa vyskytuje najviac v jednom omedzení zhodne/1.

3) rozne($\{množina_premenných_i\}$) kde premenné musia mať navzájom rôzne hodnoty. Vstup je zoznam omedzení a zoznam hodnôt niekoľkých premenných. Ak je premenná so známou hodnotou v omedzení zhodne/1, môžete jej hodnotu propagovať na ostatné premenné v omedzení. Ak je premenná so známou hodnotou v omedzení rozne/1, môžete jej hodnotu vylúčiť z oboru hodnôt ostatných premenných. Ak premennej ostane jednoprvkový obor hodnôt, premenná nadobudne túto hodnotu.

Výstup je buď informácia, že pri daných omedzeniach a hodnotách neexistuje riešenie (napr. ak jednej premennej sú v dvoch závislostiach priradené rôzne hodnoty) alebo zoznam premenných so známymi hodnotami

543. Sústavy aritmetických závislostí.

Závislosti sú dané v tvare termov: $zavisle(\{zoznam_premenných_i\}, \{popis_závislostí_i\})$ kde popis závislostí je množina trojíc: určujúce_premenné, určené_premenné, meno_predikátu s týmto významom: Ak sú známe hodnoty všetkých určujúcich premenných v niektorej trojici, potom je možné zavolať daný predikát s dvoma argumentami. Prvý argument je vstupný a obsahuje hodnoty určujúcich premenných, druhý argument je výstupný a po návrate z volania bude obsahovať spočítané hodnoty všetkých určených premenných. Predpokladajte, že potrebné predikáty sú pripravené mimo váš program. Vstup je zoznam závislostí a zoznam hodnôt niekoľkých premenných.

Známe hodnoty premenných sa použijú pre výpočet podľa závislostí. Výstup je buď informácia, že pri daných omedzeniach a hodnotách neexistuje riešenie (napr. ak jednej premennej sú v dvoch závislostiach priradené rôzne hodnoty) alebo zoznam premenných so známymi hodnotami

544. Sústavy aritmetických omedzení.

Omedzenia môžu byť dvoch druhov a sú dané v tvare termov: 1) obor($\{premenná_i\}$, $\{obor_hodnôt_i\}$) kde premenná je atom reprezentujúci meno premennej a obor_hodnôt je množina elementárnych hodnôt, ktoré môže daná premenná nadobudnúť. Každá premenná má práve jeden obor hodnôt, ktorý môže byť aj jednoprvkový.

2) $zavisle(\{zoznam_premenných_i\}, \{predikát_redukcia_oborov_i\})$ Predikát redukcie oborov má dva argumenty a použije sa vždy, ak sa zmení obor hodnôt aspoň jednej premennej. Prvý argument je vstupný a pri volaní musí obsahovať obory hodnôt premenných v poradí podľa zoznamu (1), druhý je výstupný a po návrate obsahuje nové zredukované obory hodnôt. Zredukovaný obor hodnôt môže byť aj prázdny.

Predpokladajte, že potrebné predikáty redukcie oborov sú pripravené mimo váš program a vy ich len používate. Vstup je zoznam omedzení a zoznam hodnôt niekoľkých premenných. Známe a neskôr aj zmenené hodnoty premenných sa použijú pre redukciu oborov hodnôt. Ak premennej ostane jednoprvkový obor hodnôt, premenná nadobudne túto hodnotu. Výstup je buď informácia, že pri daných omedzeniach a hodnotách neexistuje riešenie (napr. ak premenná má prázdny obor hodnôt) alebo zoznam premenných so známymi hodnotami a nový zoznam omedzení so zredukovanými obormi.

545. Generovanie grafu z podmienok.

a) Je daná postupnosť konečných omedzení vo forme zoznamu podmienok na rovnosť dvoch premenných, nerovnosť dvoch premenných alebo definícia premennej a jej domény. Úlohou je vygenerovať graf, kde vrcholy

odpovedajú premenným a sú ohodnotené množinou prípustných hodnôt a hrany sú ohodnotené množinou prípustných dvojitých hodnôt (v nejakom tvare).

b) na vytvorený graf sa vykoná statická redukcia: hodnoty vrcholov, ktoré nemajú žiadne odpovedajúce ohodnotenie hrany a susedného vrcholu, sa vypustia. Podobne hodnoty hrán, ktoré nemajú odpovedajúcu hodnotu vo vrchole. Vykonáva sa, kým je čo vypúšťať.

2a) obory hodnôt sú celé alebo prirodzené čísla a podmienky môžu byť na rovnosť alebo nerovnosť dvoch výrazov. Výrazy obsahujú základné celočíselné operácie.

2b) analogicky

550. Overenie typov v Prologu.

Pre daný program a dané typy (a la Haskell) overiť, že program je typovo správny.

560. Prepisovací systém.

Je daná množina prepisovacích pravidiel (s premennými) tvaru ľavá_strana --> pravá_strana a term t (bez premenných). pr. pravidla: krat(plus(x1,1),x2) --> plus(x2,krat(x1,x2))

a) Nájdite v t všetky podtermy, na ktoré sa dá použiť nejaké prepisovacie pravidlo. Vydajte pravidlo (jeho meno alebo číslo), pozíciu v terme (napr. "3.2" znamená v treťom podterme druhý podterm: xx v plus(a, b, krat(c, xx))) a substitúciu.

b) Aplikujte všetky pravidlá, ktoré je možné použiť na t a novo vzniknuté termy. Úlohu si (kvôli konečnosti) vhodne omezte, napr. aplikuje sa len určitý, parametrom zadaný, počet krát.

561. Hľadanie kritických párov

Kritický pár je dvojica termov, ktoré môžu vzniknúť z jedného termu prepísaním rôznymi pravidlami.

Kritický pár v prepisovacom systéme vznikne, ak nejaký podterm na ľavej strane pravidla, ktorý nie je premenná, je možné prepísať podľa iného pravidla. Kritický pár tvorí pôvodná pravá strana a prepísaná ľavá strana.

a) Nájdite všetky kritické páry.

b) (4) (Zúplňovanie systému prepisovacích pravidiel)

Podkapitola: Doménové špecifické jazyky.

570. Kreslenie 2D obrázkov. (2003)

Navrhnuť jazyk a interpret pre "kreslenie" 2D obrázkov. (a la interpret mninových výrazov z cvičení) Elementárne objekty sú kruhy, štvorce, trojuholníky, (čiar) ... Elementárne operácie sú transformácie otočenia, roztiahnutia/zúženia, (perspektivy) ... a ďalej prienik, zjednotenie (s daným prekrytím farieb, s parametrickým zmiešavaním (u farieb)) ... Vlastný výstup stačí ako matica.

a) čiernobiele b) šedé c) farebné

571. Testovanie vlastností programov.

A la Quickcheck.

572. Generovanie programu pre plotter (2004)

a) Prevod zloziteho obrazku na postupnosť „jednoduchych“ prikazov. Napr. srafovanie danej oblasti; co najmenej preruseni (zdvihnuti pera)

b) Optimalne.

573. Generovanie programu pre CNC stroj (2004)

Vyroba daneho tvaru (s urcitou danou presnostou) pomocou nastroja jednoducheho tvaru (valec, gula). Nastroj sa otaca v 5 osach. (3D plotter)

KAPITOLA 6. ... a ďalšie.

601. (1) Editačná vzdialenosť.

Editačná vzdialenosť dvoch reťazcov S a T je minimálny počet operácií nižšie vymenovaných druhov, ktoré musíme vykonať, aby sa S zmenil na T.

Obvyklé operácie sú INSert písmena, DELete písmena, REPlace jedného písmena iným, EXChange vedľa seba stojacích písmen. ďalšia možná operácia je SWAp, kedy sa namiesto (postupnosti) písmen ABC napíše CBA

a) (1) Pre dva reťazce určiť ich editačnú vzdialenosť, ak sú povolené op. INS, DEL, REP, EXC. Vydáť dosvedčujúcu editačnú postupnosť.

b) (1) Určiť editačnú vzdialenosť, ak sú povolené operácie dané parametrom.

c/ K danému slovu a počtu povolených chýb vygenerovať prechodový automat /vyhľadávací stroj/, ktorý /pri interpretácii/ pre nejaké vstupné slovo určí minimálny počet chýb /tzv. Hammingovu vzdialenosť/ a prípadne editačnú postupnosť.

(Na túto úlohu /a,b/ sa dá použiť dynamické programovanie.)

d) konfigurovatelna editacna vzdialenost. (2004) Kazda operacia ma svoju cenu (i nekonecno ;-), zacatie diery a vsuvky ma svoju cenu (motivacia z porovnania DNA), konkretne dvojice pismen pre nahradenie maju svoju cenu (motivacia z genetiky a preklepov).

ulohy: d1) minimalna vzdialenost dvoch retazcov, d2) vycet v poradí rastucej vzdialenosti, d3) ...

602. Univerzálny riešič dynamického programovania.

a) Zadá sa vzťah a kritérium, program si určí poradie a spočíta to. Ide vlastne o tabeláciu výpočtov, poradie môže byť dané implicitne, tj. podvýrazy sa vyhodnocujú podľa potreby.

b) určí sa aj poradie.

603. Letovanie plošných spojov.

... hm. Presné spočítanie dĺžky dynamickým programovaním pri jednom pravoľavom prechode (varianty: viac prechodov, lokálne cyklenie).

a) jednou hlavou / pájkou

b) viac hlavami, (ktoré do seba nemajú nabúrať)

(604.) (Iné úlohy dynamického programovania)

Minimálne

605. Kompresie (2004)

Nejaký algoritmus kompresii. Vhodne su asi slovníkové algoritmy, LZ77 posuvne okno, varianty LZR, LZSS LZB, LZH; LZ78 rastuci slovník, varianty LZW, LZC, LZT, LZMW, LZJ, LZFG)

610. Rozvrhovanie. (viz stránka: R. Barták)

... Je daná ... uniformné/neuniformné zdroje, optimálne podľa kritéria / nejaký rozvrh, štartovacie a koncové časy činností vs bez nich, tvrdé vs. penalizačné časy, ...

Konkrétne úlohy: a) Letisko, b) alokace pultov (s ostrovmi), c) rozvrh sestričiek, d) výrobný podnik, e) reklamy, f) osadzovanie súčiastok, letovanie (t.j. viac obchodných cestujúcich) ...

611. Rozvrhovanie s roznorodými zdrojmi.

a) Máme daný výčet strojov vo výrobnej hale. Stroje sú roznych druhov. ďalej je daný zoznam výrobkov. U každého výrobku je popis jeho výrobnej postupnosti, t.j. v ktorom čase (relatívne od začiatku jeho výroby) a ako dlho potrebuje jednotlivé druhy strojov (a v akom počte). Výroba každého výrobku je neprerušiteľná operácia. Naplánujte výrobu, aby trvala čo najkratší čas.

b) Popis výrobku je prerušiteľný na miestach špecifikovaných vo výrobnej postupnosti. Naplánujte najkratšiu výrobu.

c) Na výrobkoch je dané čiastočné usporiadanie, ktoré určuje poradie, v akom sa musí začať ich výroba.

d) a) + Výroba je cyklická, chcem vyrobiť daný počet výrobkov v jednom cykle. Koniec a začiatok cyklu sa môže prípadne prekrývať. Snahou je naplánovať čo najkratší cyklus.

612. Vlcek (2006- prevzate?)

a)(1) pre nosnosť obluka, počet vagonov na obluku a danu hmotnosť vagonov určiť ich poradie, aby vlak mohol most prejsť.

b) Most ma viac oblukov

c) mam k dispozícii prázdné vagony s danou hmotnosťou

d) vagony su rozne dlhe

620. Heuristické hľadanie pre obchodného cestujúceho.

Navrhnuť rôzne heuristiky. Umožniť priebežné sledovanie.

630. Genetický algoritmus pre obchodného cestujúceho.

Hm.

631. Prostredie pre geneticke algoritmy.

Navrhnuť nejakú reprezentáciu genomu, naprogramovať na nej základné geneticke operácie (mutáciu, kríženie, ...) a vyvinuť prostredie, ktoré bude zabezpečovať celkový chod algoritmu podľa daných parametrov. Uka-

zat chovanie na jednoduchom (umelom) príklade. Kriterium (fitness) vstupuje ako proceduralny parameter.

a) Genom je zoznam. Kriterium (fitness) je napr. počet určitých znakov, zhoda s daným retazcom, počet určitých dvojnakových (resp. krátkych) postupností ...

b) Genom je matica. Kriterium je zhoda s danou maticou ...

c) Podpora populácii. Algoritmus nepracuje s jednou populáciou, ale s niekoľkými (prípadne s roznoými parametrami). Prostredie obsahuje operácie pre "komunikáciu" populácii.

d) Navrhnuť špeciálnu mutáciu (napr. hillclimbing) alebo špeciálne kríženie pre nejakú špeciálnu úlohu (alebo špeciálne dátá).

632. Simulacia "Broucci" Hm.

633. Prostredie pre simulované žihanie. 2006 Laské?

Prostredie pre optimalizačné úlohy, v ktorom sa hľadá riešenie pomocou simulovaného žihania. Zahnuť: Aa) znižovanie teploty: podľa krokov, podľa zlepšujúcich krokov, podľa kvality riešenia (napr. vzhľadom k cieľu alebo k odchýlke (epsilon) od iných riešení. Ab) reštart Ac) reštart z okolia dobrého riešenia (nahodne, podľa parametrov)

B) Prípadne: beamsearch - paralelné spracovanie daného počtu riešení

640. Chemické názvoslovie

Zo vzorca vyrobiť český názov.

a) pre nesubstituované anorganické kyseliny a soli, a vybrané skupiny (amonna, kyanová ...)

b) aj kyslicníky (, kyanovodík a jeho derivaty)

c) aj pre substituované kyseliny. Jeden alebo viac kyslíkov je nahradených inými skupinami /napr. sírou, amidovou skup., ...: amidosíran draselný KSO_3NH_2 /. Tiež spracovať: $CaMg(CO_3)_2$ uhličitan vapanato-horečnatý (dolomit).

d) Pre časť organickej chémie.

641. Chemické vzorce.

K názvu chemickej zlúčeniny vyrobiť vzorec.

b) pre časť organickej chémie.

Podkapitola: Spracovanie prirodzeného jazyka

650. Morfológia. (2003)

Navrhnuť systém, ktorý umožni popis jazykových vzorov pre sklonovanie a časovanie. Systém zahrna dátové štruktúry pre popis vzorov a interpretacny program. Vzory maju byt schopne popisat nielen koncovky, ale aj zmeny v kmene (kun $j-i$ kone, bra't $j-i$ beru). Z rovnakých dát roznoú interpretáciou sa vykona analýza aj generovanie.

a) Analýza. K danému základnému tvaru so vzorom a danému tvaru slova určiť, či tvar je odvodený. V tejto úlohe v princípe dovoľujeme "pregenerovanie", pretože

sa predpoklada, ze budeme zadavat len spravne tvary slova.

b) Generovanie (synteza). K danemu zakladnemu tvaru a popisu vzoru vygenerovat vsetky (alebo jeden urcity) tvary slova. Presne, bez nespravnych variant.

Data mozu pre obe ulohy mozu obsahovat len informacie o tvaroch slov alebo aj morfologicke informacie (rod, cislo, pad, osobu...), ktore sa hodia pri generovani.

Cielom je navrh obecnych datovych struktur a interpretu, nie popis celeho jazyka. Navrhnuté dat. struktury by mali byt schopne popisat podobne ulohy: chemicke nazvoslovie (si'rovny' - siric"ity', uhelnaty' - uhlic"ity'), zdrobneliny (socha - soska), anglicke vzory: zdvojovanie spoluhlasok, vypustane e.

Idea popisu: pattern matching - porovnavanie retazcov (slov) so vzorkami

651. Eliza, cesky (2002)

Program cita vety a podla najdenych klucovych slov reaguje frazami zo slovnika. Frazy pripadne opakuju cast vstupnej vety.

Hm. a) Cesky

priklad slovnika, pre vetu daneho tvaru.

```
[mam, rad:0-1, slova(X)] ->
  [naozaj, mas, rad, slova(X)]
; [prekvapujes, ma, ',', ze, mas, rad, slova(X)]
```

Mozna reprezentacia cestiny pre analyzu je koren slova a mozny pocet pismen koncovky, dany intervalom. V priklade vyssie zahrnie aj vetu "Mam rada ...".

b) Anglicky. Navrhnut rozsirenia.

652. Konverzacka hra. (2003)

a) Na principe "adventury", analogicky Elize. Navrhnut netrivialne rozsirenie, napr. rozne modely chovania pre rozne "miestnosti", pre roznych ludi.

b) Spolupraca skupiny.

653. Generator pohadek (2004, Bojar)

Program bude z vestaveneho slovnika generovat gramaticky spravne ceske vety. Musi byt schopen generovat vety obsahujici:

- shodu podmětu s přísudkem - předložka plus shodný přívlastek k podstatnému jménu (podle toho velkého červeného medvěda) - vedlejší větu vztahovou (medvěd, který usnul) - řečnickou otázku a odpověď (Byl ten medvěd zlý? Nebyl.) - zájmeno, pokud už v předchozím textu vygeneroval něco, k čemu se to zájmeno může vztahovat

Program se zavolá jednoduše: chci pohádku o délce deseti vět. Výstupem je seznam slov a interpunkce.

Doporučené vestavěné limity: (Ač lze snadno nastavit jinak!) - maximální počet přívlastků: 3 - maximální počet vnořených vedlejších vět: 3 - maximální vzdálenost mezi podstatným jménem a zájmenem: přes 1 větu

a) česky b) anglicky, tj. bez morfológie.

653.2 Generator popisov (2004)

(Povodne: generator dokumentacie) Z danej semantickej siete vygenerovat ceske (anglicke) vety podla danyh omedzeni. Sem. siet je (multi)graf s ohodnotenymi (pomenovanymi) hranami a vrcholmi. Hrany a vrcholy su niekoľkych *druhov* a vrcholy obsahuju *atributy*.

Omedzenia pre generaciu urcuju:

- tvar vydavanych viet (pocet privlastkov, pocet vno-renych viet, celkovu velkost vety a/alebo pocet popisanych vrcholov, pouzitie zamen (viz 653))

- strategiu prechodu grafom (do hlbky, do sirky, tematicky ...)

- druhy vydavanych informaci: filter - projekcia, co nas zaujima a ma byt na vystupe

- pripadne zakazanu, pozadovanu a preferovanu po-vrchovu syntax

Hrany obsahuju informaciu o vystupnom syntaktickom tvare (povrchovej syntaxi); tvarov moze byt niekoľko. Atributy mozu byt multihodnoty. Jazyk nemusí byt dokonaly. Je mozne / vhodne vyuzit "modifikatory", ktore vylepsia vystup: v(miestnost) - i [v,miestnosti], od(cas) -i [od,casu], plural(kruh)-i [kruhy]

Priklad, popis grafu v tvare RDF: jvrchol_i hrana jvrchol_j

```
<PRG005> druh <prednaska>
<PRG005> je_v <S3>, syntax [vr(1),je, v, vr(2)],
  [vr(1),kona, sa, v, vr(2)]
<S3> druh <miestnost>
<PRG005> nazov <Neproceduralne programovanie>,
  syntax ...
<PRG005> od <9:00>
<9:00> druh <cas>
<PRG005> do <10:30>
<10:30> druh <cas>
<PRG005> pre multi([I/2/31,I/2/32,...])
<I/2/31> druh <kruh> ...
```

Trivialny vystup je "Prednaska PRG005 je v !miestnost! S3. Prednaska PRG005 ma nazov Neproceduralne programovanie. Prednaska je od !cas! 9:00 do !cas! 10:30. ...". Vykricnikmi su oznacene miesta, kde by sa hodili modifikatory.

Lepsi vystup (ale nie dokonaly) pre prilis nazorny filter [vypisujDruhy([prednaska]), pouziZamena([prednaska]), vynechajHrany([do]), syntax(nazovvedlajsouVetou)] bude: "Prednaska PRG005, ktora sa nazyva Neproceduralne programovanie, kona sa v S3. Tato prednaska je od 9:00 pre I/2/31, ... I/2/39 a I/2/40".

a) Cestina vs. b) anglictina.

654. Prevod semantickej siete na RDF (2004)
- a) Tam a naspat. Semanticcka siet viz 653.2. RDF (viz <http://w3c.org>) su trojice (objekt, atribut, hodnota), hodnoty su atomicke alebo ine objekty.
 - b) Vcetne jednoduchych operacii na na RDF (hladanie najkratsich ciest daneho tvaru (reg. vyrazom))
 - c) Prevod medzi XML (viz 655) a sem. sietou
655. Rozbor publikacnich udaju (2004).
viz. Vypis v XML. napr.

```
<InProceedings name="enti_mis:2003">
  <author no='1'>Bojar, Ondřej</author>
  <author no='2'>Brom, Cyril</author>
  ...
  <title>ENTI -- Simulátor přirozeného
    prostředí lidského světa</title>
  <BookTitle>MIS 2003</BookTitle>
  <Pages from='3' to='14' />
  <Publisher>MATFYZPRESS</Publisher>
  <PubAddress>Praha</PubAddress>
  <Year>2003</Year>
  <Month>January 18--25</Month>
  <Note>MSM113200006, LN00A063</Note>
}
```

vice o XML: <http://www.w3c.org/>

(orig. Bojar): Program dostane na vstup řetězec obsahující publikační údaje (např. "Ondřej Bojar, Cyril Brom, Milan Hladík, Mikuláš Vejlupek, Vojtěch Toman, and David Voňka. 2003. ENTI - Simulátor přirozeného prostředí lidského světa. In MIS 2003, pages 3-14. MATFYZPRESS, January 18-25, 2003. MSM113200006, LN00A063." ap.). Program bude postupně navrhovat *smysluplná* čtení těch údajů a vypisovat je ve formátu BibTeXu, např.:

```
@InProceedings{enti_mis:2003,
  author = {Bojar, Ondřej and Brom, Cyril
    and Hladík, Milan and Vejlupek, Mikuláš
    and Toman, Vojtěch and Voňka, David},
  title = "{ENTI -- Simulátor přirozeného
    prostředí lidského světa}",
  BookTitle = {MIS 2003},
  Pages = {3--14},
  Publisher = {MATFYZPRESS},
  PubAddress = {Praha},
  Year = {2003},
  Month = {January 18--25},
  Note = {MSM113200006, LN00A063}
}
```

Vce o BibTeXu na webu. Např. <http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/btxdoc.html>

656. Rozbor inzeratov. (2004)
viz 655. Rozbor inzerátov z novin. Výstup v strukturovanom tvare (XML).

Domeny: a) byty; b) auta; c) práca d) a ďalšie

657. Eliza matfyzak (2006)
Program s princípom Eliza (příklad 651), určený pre Deň otvorených dverí, ktorý má záujemcom o štúdium na MFF poskytnúť relevantné informácie k ich otázkam.

658. Interpret vyslovnosti (2006)
a) Program na základe písanej formy slova určí jeho vyslovovanú podobu. Pravidlá pre zmeny výslovnosti sú reprezentované v dátovej štruktúre (1) alebo programom (2).

Interpret je obecný (motivovaný češtinou), ale podľa vhodne zadaných pravidiel dokáže vyslovovať napríklad (správne) nemecky. Jadro programu je navrhnuť vhodnú dostatočne obecnú podobu pravidiel a využívať ju pri transformácii slov.

Pravidlá typu (1) zahrňujú zmeny, ktoré je možné zadať výčtom, tj. pravidla 1:1, N:1, 1:N a M:N písmen. Pravidlá dovoľia ošetriť koniec (a začiatok) slova.

Pravidlá typu (2) realizujú napr. spodobňovanie (dlhých) skupín písmen. (Nie je vhodné ich zadávať výčtom)

b) Výslovnosť celej vety.

660. Transformácie XML (2004)

a) Pre dokument v XML reprezentovaný v prologovskej štruktúre (!) navrhnete popis obecných transformácií dokumentu do XML. XML-strom reprezentujte pomocou n -árneho stromu. SWI-Prolog obsahuje podpory pre prácu s XML formátom, načítanie XML je pomocou `load_xml_file/2`.

a1) jednoduché transformácie zahrňujú pridanie atributu, elementu, update hodnoty atributu alebo elementu, vypustenie, presun nahor, nadol. Atribut čítač. Zmena atributu na element a naopak.

a2) podmienené operácie: pridanie len do určitých elementov splňujúcich podmienku.

a3) Zložitejšie operácie. ...

b) Realizujte aj výstup XML do obvyklej formy.

661. Notacia N3

Notacia N3 je iný spôsob zapisu informácii, ktoré sa zapisuju v XML.

a) Vypis v Standardnom tvare. (asi jednoduche)

663. Integrovanie (2009)

a) Skúšanie základných postupov a integrácia per partes pomocou prehľadávania možností. Zvláda polynomy, goniometrické fcie, log a exp, operátory +, *, -, (*c). V bez zložitých úprav.

b) zložené funkcie, c) substitúcie

A DALSIE:

-xxx Backpropagation Hm.

- (28.11.2002, RK) Eliza; zjednodušovanie výrazov, formálne derivovanie; práca s polynomami viac premenných (aj spoločný deliteľ); zebra; fraktály - generovanie výrazov pre plotter; prevod algebraických notácií, prevod formule do prenexného tvaru; kombinatorické algoritmy; generovanie všetkých struktur nad konečnou množinou (jaká jsou neisomorfní částečná uspořádání nad množinou o N prvcích) - viz 260; Gaussova eliminace; práce s permutacemi; "expertní systém"; manipulace s konečnými automaty; ...

[1] L. Kučera: Kombinatorické algoritmy, SNTL, Praha [2] M Chytil: Automaty a gramatiky, SNTL, Praha, 1984

PISOMKOVE príklady 15. a 19.11.1999:

Za neúspešne napísanie písomky máte odovzdať niekoľko príkladov. Vyberte si z ľahkých započtových (napr. 201,203,205,210,320,321...). Odovzdajte na papieri, za dva týždne (6.12. resp. 10.12.).

Nevhodne ako započtove príklady, typicky príliš ľahké.

Polynómy - úvod: (Kapitola z počítačovej algebry)

Základné spôsoby reprezentácie polynómov (vektorov, matic, ...) viacerých premenných môžeme rozdeliť

A) podľa toho, či sú uvedené všetky koeficienty na:

A.1) riedké - sú uvedené len nenulové koeficienty s príslušnou mocninou u premennej/premenných

A.2) husté - sú uvedené všetky koeficienty v rastúcom alebo klasajúcom poradí. V prípade rastúceho poradia nemusíme ani uvádzať stupeň polynomu.

B) podľa spôsobu zhlukovania premenných:

B.1) rekurzívne - polynóm viacerých premenných je zapísaný ako polynóm jednej premennej, ktorý má ako koeficienty polynómy v ostatných premenných

B.2) distributívne - polynóm je zapísaný ako súčet termov, z ktorých je každý súčin premenných a koeficientu - obvykle čísla.

B.3) zmiešané

C) Hornerovo schéma

101. (1) Prevod polynomu do žiadanej formy.

Zadanie: Vstup je ľubovoľne uzatvorený výraz s číslami, premennými a operáciami '+', '-', '*' a '^' (tj. polynóm), a zoznam premenných, ktorý udáva dôležitosť premenných. V príklade nižšie bude hlavný polynóm v premennej 'x', jeho koeficienty v 'y'. Pr. volania: ?-uprav((x-2*y)^2*(3*z+x), [x,y], X).

Úloha: Preveďte vstup na: a) riedky rekurzívny zápis (1) b) riedky distributívny zápis (1) c) hustý rekurzívny zápis (1) d) hustý distributívny zápis (1)

e) riedke Hornerovo schéma (2)

f) husté Hornerovo schéma (2)

g1) (2) Premenné sú zadané v zozname zoznamov. Každý prvok, tj. zoznam premenných, reprezentuje jednu úroveň rekurzívneho vnorenia polynómov, premenné v zozname budú roznásobené. Pr. Zoznam [[x,y],[u,v]] znamená, že výsledok bude roznásobený polynóm v x a y a jeho koeficienty budú roznásobené polynómy v u a v. Výsledok je hustý zápis.

g2) dtto. Výsledok je riedky zápis.

102. (1)-ľahké. Prevod polynomu do riedkeho distributívneho tvaru

Riedky distributívny tvar polynomu viacerých premenných je zadaný zoznamom premenných, ktoré sa v polynóme vyskytujú, a zoznamom sčítancov - faktorov. Každý faktor je v tvare zoznamu, jeho hlava je absolútny člen a ostatné prvky sú exponenty jednotlivých premenných podľa poradia daného zoznamom. Zhodné mocniny u všetkých premenných sa nevyskytujú dvakrát (v takom prípade môžeme sčítať absolútne členy).

Pr. polynóm $(2x^3+4)y$ je pri danom poradí premenných [y,x] zapísaný ako [[4,1,0],[2,1,3]], tj. $4y^1x^0+2y^1x^3$

Úloha: Vstup programu je zoznam premenných a polynóm v týchto premenných, ktorý obsahuje operácie sčítanie, odčítanie, násobenie a umocňovanie na prirodzené číslo. Preveďte polynóm do riedkeho distributívneho tvaru. Uvedomte si, že umocňovať a násobiť môžete aj celé polynómy, napr. $((x+2)*(y*z))^5$.

110. (1) Datová štruktúra trie.

Slúži na reprezentáciu postupností nad úplne usporiadanou množinou (napr. číslami, znakmi ...). Každý vrchol reprezentuje prefix nejakej postupnosti. Trie je (binárny) vyhľadávací strom podľa jednej zložky postupnosti, ktorý obsahuje v každom vrchole „trie“ pre postupnosti s dosiaľ nájdeným prefixom. (pomôcka: neprázdny trie má 4 zložky: ľavý a pravý pod-trie pre vyhľadávanie podľa prvej zložky, informáciu pre tu končiaci prefix a pod-trie pre dlhšie postupnosti s týmto prefixom)

a) Implementovať vyhľadávanie, pridávanie, (uberaenie) b) (a) + Výroba vyváženého trie z daných postupností

111. (1) Implementácia algoritmu Aho-Corasicková. dopredná funkcia, spätná funkcia, interpret zvolenej d.š.

120. Hľadanie podgrúp.

Úvod. Grupa je daná, podobne ako grafy, nejakým abstraktným rozhraním, napr. pomocou predikátov $\text{elem}(G,X)$, $\text{elems}(G,Xs)$, $\text{unit}(G,X)$, $\text{inv}(G,X,IX)$ a $\text{oper}(G,X,Y,XY)$, ktoré určujú pre danú grupu G postupne (backtrackingom) prvok, všetky prvky, jednotku, inverziu a binárnu operáciu. Podstatné na rozhraní je, že grupová operácia (a ostatné súčasti) môžu

byť dané tabuľkou alebo predpisom výpočtu (napr. grupa násobenia modulo 1001).

a) Je daná grupa a množina generátorov. Úlohou je nájsť najmenšiu generovanú podgrupu a vyjadriť každý jej prvok pomocou generátorov.

b) Pre iné algebraické štruktúry ...

b1) napr. teleso

201. Izomorfizmus grafov.

a) (1) slepým prehľadávaním

b) algoritmus rozkladu na triedy so zhodnými stupňami vrcholov [1], viz 240.

ba) orientované

bb) neorientované

202. (1) Rozklad orientovaného grafu na silne súvislé komponenty. - efektívne

203. (0) Redukcia orientovaného grafu podľa daných tried vrcholov.

204. (1) Testovanie k-súvislosti grafu. a) hranovej: odobratie k-1 hrán nezvačší počet komponent b) vrcholovej: "- k-1 vrcholov "- c) rozklad na 2-súvislé komponenty

206. Maximálne párovanie.

a) (0) prehľadávaním

b) (2) lepší algoritmus [1]

c) (1) párovanie v bipartitnom grafe

210. (1) Rozklad na Eulerovské grafy.

Pre daný graf vydať minimálny počet sledov (čiार jedným ťahom), ktoré graf nakreslia.

211. (1) Maximálny sled.

Pre daný hranovo ohodnotený graf zistiť sled s maximálnym ohodnotením.

220. Minimálne cesty.

a) Pre hranovo ohodnotený (orientovaný) graf s n vrcholmi sa spočítajú minimálne cesty medzi všetkými vrcholmi takto: Pole i, j v matici $n \times n$ obsahuje v k-tom kroku dĺžku minimálnej cesty z i do j cez medzilahlé vrcholy 1 až k. Na začiatku $k=0$ a matica obsahuje len priame cesty - hrany. Na konci $k=n$ a cesta môže byť ľubovoľná. Matica v $k+1$. kroku sa spočíta z matice v k. kroku. (pozn.: zložitosť $O(n^3)$)

Naviac rekonštrukcia minimálnej cesty z matice.

221. (1) Minimálna kostra

a) Pre daný hranovo ohodnotený graf naprogramujte hľadanie minimálnej kostry b) dtto, ale maximálnej

301. (1) Šifrovanie zrkadlovou mriežkou. Mriežka sa namiesto otáčania prevracia, najprv podľa zvislej osy, potom vodorovnej, a nakoniec zvislej. Ďalšie prevrátenie podľa vodorovnej osy by ju dostalo do pôvodnej polohy. Šifrovaná postupnosť je dlhšia ako počet polí v mriežke. Je daná (správna) mriežka a vstup, výstup je postupnosť matíc písmen, prípadne linearizovaná.

302. (1) Výroba zrkadlovej mriežky. Postupne pri backtrackingu vydávať jednotlivé mriežky, ktoré: a) (1) nemajú políčka spojené hranou alebo rohom. b) majú najviac daný počet takých spojení

310. (1) Prelievanie vody. (Aplikácia prehľadávania grafu)

a) Je daných n nádob (n je vstupný parameter!), ich objemy, počiatkové naplnenie a požadované koncové naplnenie. Úlohou je nájsť nejakú postupnosť preliatí, ktoré vedú z počiatkového do koncového stavu.

b) dtto, ale najkratšiu postupnosť

c) Okrem nádob je ešte nevyčerpatelný zdroj a nepreplnitelný kanál. (Nádoby sú na začiatku typicky prázdne)

d) dtto (c), ale najkratšie riešenie

311. (1) Lloydova 15.

V štvorci 4×4 je 15 kostičiek s číslami a jedno voľné pole.

a) poskladať

b) (a) nadaný počet ťahov

c) dosažitelnosť dvoch pozícií

d) (c) na daný počet ťahov

312. (1) Zmenšená Lloydova 9.

Ako úloha 311, ale na 3×3 .

313. (1) Rubikova kocka.

a) poskladať

b) (a) na daný počet ťahov

c) dosažitelnosť dvoch pozícií

d) (c) na daný počet ťahov

320. (1) Výroba magických štvorcov

a) (0) nediagonálnych

b) (1) diagonálnych

c) dokonalých - aj prerušené uhlopriečky majú zhodný súčet Úlohu neriešate backtrackingom. Chcete vydať všetky štvorce, resp. určený počet, v datovej štruktúre.

321.(1) Generovanie latinských štvorcov

Latinský štvorec rádu n obsahuje čísla 1 až n, každé n krát. Sú usporiadané tak, že v žiadnom stĺpci ani riadku sa žiadne číslo nevyskytuje dvakrát.

Naprogramujte generovanie latinského štvorca daného rádu. Snažte sa o rýchle vylúčenie nevyhovujúcich možností, tj. každé číslo, ktoré pridávate do štruktúry štvorca, skontrolujte na čo najväčší počet podmienok voči dosiaľ vygenerovanej časti štvorca.

a) (1) riešenia vydávajte backtrackingom

b) Riešenia vydať v datovej štruktúre, všetky alebo určený počet.

321. (1) Generovanie ortogonálneho latinského štvorca

Latinský štvorec rádu n obsahuje čísla 1 až n , každé n krát. Sú usporiadané tak, že v žiadnom stĺpci ani riadku sa žiadne číslo nevyskytuje dvakrát.

Ortogonalný latinský štvorec obsahuje dvojice čísel (i,j) , $1 \leq i,j \leq n$, každú jedenkrát. V prvej aj druhej zložke je to latinský štvorec.

Naprogramujte generovanie ortogonálneho latinského štvorca daného rádu. Snažte sa o rýchle vylúčenie nevhodujúcich možností, tj. každé číslo alebo dvojicu, ktorú pridávate do štruktúry štvorca, skontrolujte na čo najväčší počet podmienok voči dosiaľ vygenerovanej časti štvorca.

a) (1) Riešenia vydávajte backtrackingom
b) Riešenia vydajte v datovej štruktúre, všetky alebo určený počet.

343. (1) Riešenie sústav algebrogramov.

a) S operáciami $+$, $-$, $*$

b) dtto, $+$ presné celočíselné delenie, (div, mod ...)

350. (1) Refazce domina.

Sú dané dominové kostičky a prirodzené číslo - počet refazcov. Úlohou je zistiť, maximálne koľko kostičiek sa dá poskladať do daného počtu refazcov.

510. (1) Výpočet hradlovej siete.

Acyklickej. a) (1) logickej b) (1) aritmetickej c) s externe, genericky zadanými operáciami Menám operácií odpovedajú prologovské procedúry a pri výpočte sa predávajú ako parametre.

511. (1) Usporiadanie hradlovej siete. (alias topologické triedenie)

Hradlová sieť je množina hradiel, kde každé hradlo má jediný výstup, svoje jedinečné meno, operáciu a zoznam vstupov (s daným poradím). Vstupy sú buď mená iných hradiel alebo mená vstupných bodov siete. Sieť je acyklická, tzn. žiadne hradlo nezávisí na svojom výstupe.

Vstup programu je hradlová sieť. Úlohou je usporiadať hradlá tak, aby každé hradlo bolo až za hradlami, ktoré počítajú jeho vstupy.