

Príklady na cvičenia Aut. a gr.

Jan Hric, KTIML MFF UK

e-mail: Jan.Hric@mff.cuni.cz

<http://ktiml.ms.mff.cuni.cz/~hric/vyuka/.../cvaut.pdf>(, .ps)

28. května 2009

Motto: Všechny detaily jsou důležité, ale některé jsou důležitější.

Zapocet LS 08/2009: spravne vypracovat priklady: 9, 19, 21, 49 (2 podpriklady), 53, 60

Zapocet LS 04/2005: spravne vypracovat priklady:

- 1) moj 21 (z tejto zbierky); z prikladov <http://ktiml.ms.mff.cuni.cz/~bartak/prednasajuceho> - uvedene (cislo cvicenia).("cislo" prikladu).(podpriklad):
- 2) cv2."ekv".e),
- 3) cv3.1.i-"blizko mista",
- 4) cv3.f)."1 s kazdym stavom",
- 5) cv5.pr3.bod7,
- 6) cv10.2.{do Chomskeho NF},
- 7) cv6.2.analogicky:"posledna tretina slova z L" - sformulovat podmienku a skonstruovat 2KA,
- 9) cv10.2.{do Greibachovej NF}, rovnaka gram. s pr6),
- 11) cv10.3.bod2 .

Ak mate ciastocne spravnu pisomku (viz web) vypracujete priklady s neparnym (lichym) cislom.

1 Konštrukcia automatov

1 Úvodný tenisový automat Nad abecedou $\Sigma = (D, H)$, tj. $\{Domaci, Hostia\}$, vytvorte postupne automaty, ktoré rozoznávajú koniec prvej hry, výhru Domaceho hráča v prvej hre, koniec každej hry, koniec prvého setu, koniec zápasu.

Optimalizujte. Zmenšite počet stavov - zlučte stavy Zhoda, Výhoda, Nevýhoda s inými vhodnými.

(Ako môžu byť stavy označené? Musia byť očíslované alebo je prípustný reťazec?)

Z programátorského hľadiska: čo sú v KA deklarácie, čo je program. Uvedte príklad syntaktickej chyby. Aká je sémantika KA (význam), ako funkcia?

Programátorský idiom: mŕtvy stav.

2 Popis jazyka ad B.1.dole.d

Popisujú nasledujúce dve charakteristiky rovnaký jazyk: slová v L1 nekončia 00, slová v L2 končia 1 alebo 10. Zdovodnite.

3 Deliteľnosť a) Automat pre čísla deliteľné tromi v dvojkovej sústave.

a2) Varianty: Zvyšok po delení 5 je 1. a3) Deliteľnosť 7, 11, 13 v desiatkovej sústave.

b) Popíšte formálne automat A nad abecedou $\{a, b\}$ pre jazyk $L(A) = \{w; \exists k : |w| = 100k + 2\}$ (Pre formálny zápis sa hodí obecná formula. Výčet je nepraktický.) Pozn.: je to úloha na deliteľnosť v unárnej sústave.

c) Obecné, pre čísla v sústave so základom p určite zvyškovú triedu x po delení n. Prechodovú funkciu definujte vzorcom.

4 Navrh KA Navrhnite automat nad abecedou $\{a, b\}$, ktorý určuje, či slovo

a) obsahuje počet symbolov a deliteľný 3; a2) deliteľný 2 alebo 3; a3) deliteľný 2 a 3; a4) deliteľný 3 a nedeliteľný 2

b) končí aab; b2) začína aab; b3) obsahuje "aab"; b4) neobsahuje "aab"; b5) analogicky pre "baba"

c) končí aspoň dvomi b, pred ktorými je aspoň jedno a.

d) má dĺžku aspoň 2 a začína a končí tým istým symbolom; d2) má dĺžku aspoň 4 a prvé dva symboly sú zhodné ako posledné dva symboly; d3) prvé dva symboly sú zhodné ako posledné dva symboly;

e) slovo aaa končí na 5 pozícii od konca (- výhľad)

5 Navrhnite algoritmus, ktorý pre RJ zisťuje, či a) $L(A) = \emptyset$; b) $L(A) = L(B)$; c) $L(A) = X^*$; d) $L(A) \subset L(B)$; e) $L(A)$ je nekonečný

6 Príslušnosť do RJ Určite príslušnosť jazykov do triedy RJ. Jazyk slov nad $\{a, b\}$, $k, l, n \geq 0$ kde

a) $\#a = \#b$, tj. počet a sa rovná počtu b.

b) $|\#a - \#b| \leq n$ pre každý prefix slova

c) $a^k b^l$; $a^k b^l$, $k < l$; $a^k b^l$, $k > l$; $a^k b^l$, $k < l < 2k$; $a^k b^l a^k$; $a^k b a^k$ (b je oddelovač); $a^k b^k a^l$; $a^k a^k b^l$; ww , w je slovo; ww^R , w je slovo; ww^R , w je slovo z a^* ; a^{2n} ; a^{n^2} ; a^{2^n} ; a^p , p je prvocíslo;

7 Príslušnosť do RJ pomocou pravidiel. a) Tvrdenie: "Ak L1 je RJ a $L2 \subset L1$, potom L2 je RJ." neplatí. Zdovodnite.

b) Uvedte príklady jazykov, o ktorých môžete dokázať pomocou uzáverových vlastností, že sú, resp. nie sú RJ. (doplnok, prienik, zjednotenie, ... a kombinácie). $L1 \in RJ \wedge L2 \in RJ \rightarrow L1 \cap L2 \in RJ$ a $L1 \in RJ \wedge L1 \cap L2 \notin RJ \rightarrow L2 \notin RJ$.

8 Synchronizačná postupnosť Z daných stavov (resp. všetkých) sa dostať do nejakého spoločného.

a) Existuje pre dva ekvivalentné stavy nejaké synchronizačné slovo? A varianty:

b) (Existuje n, že) Sú všetky slová dlhšie než nejaké n synchronizačné?

c) Existuje pre každý prefix dlhšie synchronizačné slovo?

9 Iteračné lemma ZAPOCET 08/09 c1)

Sformulujte a dokážte variantu iteračnej lemy, ak opakované slovo v (z prednášky)

a) leží blízko konca slova

b) leží približne uprostred slova

c) leží za (c1), pred (c2) alebo okolo (c3) danej pozície p. Správne ošetríte podmienky na pozíciu p.

d) je pumpované v podčiarknutých písmenách. Sformulujte podmienky.

10 (Riadené) Zlyhanie dokazu Jazyk $0^k 1^l 1^l$ je regulárny. (Dokážte :-)

Pri dokazovaní sporom pomocou Nerodovej vety, že jazyk je neregulárny, musí niekde dokaz zlyhať. Ktorá časť zlyhá?

Tá istá otázka, pri dokazovaní pomocou Iteračnej lemy?

11 Ekvivalencia stavov Za akých podmienok môžeme ukončiť hľadanie ekvivalentných stavov. Sformulujte podmienky na

- stabilitu riešenia (tj. vzťah dvoch posledných iterácií)
- počet iterácií
- tvar (jednej) poslednej iterácie

Ak je stav uz jediný vo svojej triede, mozete ho opisovat "bez premyslania"?

12 Nedet. Aut. a) Vyjadrite zobecnenu prechodovu funkciu δ_N^* pre nedeterministicky automat. Aký je typ funkcie? (dve možnosti) Vyjadrite funkciu pomocou rekurzie podľa slova spredu, resp. zozadu, tj. slovo $u \in \Sigma^*$ rozoberáme na $av = u, a \in \Sigma, v \in \Sigma^*$, resp. $va = u$. b) Formálne definujte podmienku, kedy podmnožina stavov v podmnožinovej konštrukcii určuje koncový stav. c) Je možné pre nedeterministický KA jednoducho definovať automat, ktorý rozpoznáva doplnok jazyka? d) Ako je možné simulovať Nedet. KA primom, tj. bez prevodu na DKA? e) Pri definícii NKAutomatu obrázkom pre nejaký stav nemusí existovať pokračovanie podľa písmena. Ako je v tomto prípade e1) definovaná prechodová funkcia, e2) definovaný výpočet. e3) Čo môžete tvrdiť o prijatí slova týmto výpočtom.

13 Prevod zobecneneho nedeterministického automatu Aký má typ prechodová funkcia v zobecnenom nedeterministickom automate, tj. s λ -prechodmi. Može pri prevode na KA vzniknúť prázdna množina stavov? Prečo je výhodnejšie pri prevode na DKA λ -prechody aktivovať po vykonaní prechodu po písmene (tj. hladovo)? Zdovodnite z hľadiska vstupného stavu, výstupných stavov a veľkosti vytváranej množiny stavov. Ako sa zbavíte λ -prechodov, tj. ako prevediete ZNKA na NKA?

14 Odvodenie automatov Je daný ľubovoľný redukovaný KA A rozpoznávajúci jazyk L nad abecedou a, b . Navrhňte čo najmenší automat, ktorý rozpoznáva slová z L , ktoré a) začínajú na a ; b) končia na a ; c) obsahujú a ; d) neobsahujú a .

Ktoré skonštruované automaty sú (vždy) redukované? Ak nie sú, ktoré stavy môžu byť prípadne spojené? Može redukcia skonštruovaných automatov viesť k "dramatickému" poklesu počtu stavov?

15 Jazyky s chybami Navrhňte a formálne zapíšte konštrukciu, ktorá k ľub. DKA, resp. NKA, pre L vyrobí automat, ktorý prijíma slová z L

- s (práve) jedným vložením písmena "a": $L1 = \{uav | \exists u, v \in \Sigma^* : uv \in L\}$
- s najviac jedným vložením písmena "a": $L1 \cup L$
- s (práve) k vloženiami písmena "a": $L1 = \{u_0 a u_1 a \dots a u_k | \exists u_i \in \Sigma^* : u_0 u_1 \dots u_n \in L\}$
- s najviac k vloženiami písmena "a": $L1 = \{\dots\}$
- s jednou výmenou písmena "a" za "b": $L1 = \{uav | \exists u, v \in \Sigma^* : ubv \in L\}$
- s jednou výmenou písmena "a" za čokoľvek: $L1 = \{uav | \exists c \in \Sigma, u, v \in \Sigma^* : ucv \in L\}$
- s jednou výmenou ľubovoľného písmena za "b":

$L1 = \{ucv | \exists c \in \Sigma, u, v \in \Sigma^* : ubv \in L\}$

h) s jednou výmenou ľubovoľného písmena za ľubovoľné písmeno: $L1 = \{udv | \exists c, d \in \Sigma, u, v \in \Sigma^* : ucv \in L\}$

i) s jedným vypustením písmena "a": $L1 = \{uv | \exists u, v \in \Sigma^* : uav \in L\}$

j) s jedným vypustením ľub. písmena: $L1 = \{uv | \exists c \in \Sigma, u, v \in \Sigma^* : ucv \in L\}$

k) s jednou ľub. chybou vloženia, výmeny alebo vypustenia - snažte sa minimalizovať počet stavov pri konštrukcii.

l) s výmenou dvoch susedných písmen: $\{udcv | \exists u, v, c, d : ucdv \in L \wedge c, d \in \Sigma \wedge u, v \in \Sigma^*\}$

ad) Jazyky s chybami sú v cvičeniach: vypustenie písmena, vloženie ľub. písmena, vloženie a. Práve jedna chyba, najviac jedna chyba, najviac k chýb. Výmena dvoch písmen: $\{udcv | \exists u, v, c, d : ucdv \in L \wedge c, d \in \Sigma \wedge u, v \in \Sigma^*\}$

16 Časti slov K danému konečnému automatu A skonštruujte automat, ktorý prijíma jazyk

a) $\{u | \exists v, vu \in L\}$ - prípony slov z jazyka L , $\text{suff}(L)$

b) $\{u | \exists v, uv \in L\}$ - predpony, $\text{pref}(L)$

c) $\{u | \exists v, w, vuw \in L\}$ - spojitú časť slova, $\text{sub}(L)$

d) $\{u | \exists v, u \text{ je vybraná podpostupnosť z } v \& v \in L\}$ - vybrané p. (súvisí s paralelným spojením jazykov)

17 Cúvanie v automate Hm. "Lexikálna" analýza Fortranu.

b) rozpoznávanie L^R - jazyk obrátených slov (čítané odzadu dopredu)

18 Veľký nedet. aut. Dokážte, že pri zdeterminičtovaní pre $2n$ stavov NKA dostaneme 2^n stavov KA. (Alias $n \rightarrow 2^{n/2}$). Abeceda veľkosti n , automat uspeje, ak niektoré z písmen má párny počet výskytov. Navrhňte varianty na podobnom princípe.

Varianta, pre pevnú abecedu: niektoré (malé) prvočíslo delí dĺžku slova.

Z prednášok: slovo, ktoré má na pozícii n od konca písmeno 1, je prijaté.

19 nka2ka ZAPOCET 08/09 b)

Prevedte A na deterministický konečný automat a zredukujte. a) $A = (\{1, 2, 3, 4\}, \{a, b\}, \{1\}, D, \{3, 4\})$, (1 je počiatočný stav a 3, 4 su koncové)

$D = \{$
 $(1, a) \rightarrow \{2\}, (1, b) \rightarrow \emptyset,$
 $(2, a) \rightarrow \{3, 4\}, (2, b) \rightarrow \{1, 4\}$
 $(3, a) \rightarrow \emptyset, (3, b) \rightarrow \{1, 2\}$
 $(4, a) \rightarrow \emptyset, (4, b) \rightarrow \emptyset \}$

b) $A = (\{1, 2, 3, 4\}, \{a, b\}, \{1\}, D, \{3, 4\})$, (1 je počiatočný stav a 3, 4 su koncové)

$D = \{$
 $(1, a) \rightarrow \{2, 3\}, (1, b) \rightarrow \emptyset,$
 $(2, a) \rightarrow \{3, 4\}, (2, b) \rightarrow \{1, 4\}$
 $(3, a) \rightarrow \emptyset, (3, b) \rightarrow \{1, 2\}$
 $(4, a) \rightarrow \emptyset, (4, b) \rightarrow \emptyset \}$

2 Uzáverové vlastnosti RJ

20 Dokazovanie neregulárnosti jazykov

a) Dokážte: Ak $L1$ je RJ a $L1 \cap L2$ není RJ, potom $L2$ není RJ.

b) Pomocou a) dokážte, že jazyk $\{0^k 1^k 0^l 1^m 0^l; k, l, m \geq 0\}$ nie je regulárny.

c) Dokážte analogické tvrdenia k a) pre ostatné operácie na RJ.

21 Permutácia ZAPOCET 08/09: b)

Rozhodnite a dokážte, či RJ sú uzavreté na nasledujúce operácie: (skonštruujte automat alebo vyvráťte)

a) $\text{perm}(L) = \{u \mid \text{permutace}(u) \in L\}$ – permutácie

b) $L1 \parallel L2 = \{u \mid \exists v, w : u \in L1 \wedge v \in L2 \wedge w = u \mid v\}$, kde operácia \parallel odpovedá rezaniu dvoch balíčkov kariet (ľubovoľne) – (paralelné spojenie). Presnejšie: písmená slova s sa dajú ofarbiť dvomi farbami A a B tak, že podslovo s písmenami ofarbenými A , respektíve B , leží v $L1$, resp. $L2$.

c) $\text{cykl}(L) = \{u \mid \text{cyklicky_posun}(u) \in L\}$ – (všetky) slová vzniklé cyklickým posunom

$= \{u \mid \exists v, w : u = vw \wedge wv \in L\}$

c2) cyklický posun so zarážkou: $\text{cykl2}(L) = \{u \mid \exists v, w : u = v\#w \wedge wv \in L\}$

c3) $\text{cykl3}(L) = \{u \mid \exists v, w, x : u = vwx \wedge v.w^R.x \in L\}$ – slová vzniklé obrátením podsekvencie (z bioinformatiky)

varianty: reverzie možem požadovať len na začiatku alebo konci

d) $\text{jadro}(L1, L2) = \{u \mid \exists v, w : vuv \in L1 \wedge vw \in L2\}$ – zobecnenie $\text{sub}(L) = \text{jadro}(L, \Sigma^*)$

e) $\text{obal}(L1, L2) = \{u \mid \exists v, w, x : u = vw \wedge vxw \in L1 \wedge x \in L2\}$... a varianty

22 Analýza pomocou DP Popíšte, ako je možné metódou dynamického programovania rozpoznávať, či konkrétne slovo patrí do jazyka $L1 \parallel L2$ (paralelne) (bez explicitnej konštrukcie automatu), ak poznáme (nedeterministické) automaty pre jazyky $L1$ a $L2$.

Návod. Určite, v akých stavoch môžu byť automaty po prečítaní prvých i písmen slova.

Nech $s1$, resp. $s2$, je počet stavov $A(L1)$, resp. $A(L2)$, a k dĺžka slova. Aká je zložitosť ("simulácie") výpočtu pomocou dyn. programovania?

23 Kvocienty Hm. $*aabb*$ kvocientováno dvoupísmennými slovami, oboma smery. (viz úloha 3-r.2007)

24 Použitie derivácií a kvocientov

a1) Akú operáciu na jazyk použijete, ak v C v zápise hexadecimálnych čísel chcete odstrániť počiatočné $0x$.

b) Akú operáciu na jazyk použijete, ak v zápise reálnych čísel chcete odstrániť exponent.

c) Akú operáciu na jazyk použijete, ak v zápise celých čísel chcete odstrániť znamienko?

25 Použitie (derivácií a) kvocientov 2 Ako vyjadrite pomocou kvocientov k danému (regulárnemu) jazyku L :

a) jazyk všetkých prefixov $\text{pref}(L)$, a2) jazyk všetkých vlastných prefixov $\text{pref2}(L)$

b) jazyk všetkých suffixov $\text{suff}(L)$, b2) vlastné suffixy $\text{suff2}(L)$

c) jazyk všetkých spojitých podsekvencií $\text{subseq}(L)$, c2) vlastné podsekvencie $\text{subseq2}(L)$

26 Odvodené operácie

Aká je opačná (sprava inverzná) operácia k ľavej (resp. pravej) derivácii jazyka L podľa w ?

Aká je "približná" opačná operácia k ľavému (resp. pravému) kvocientu jazyka $L1$ podľa $L2$, pre dané $L2$?

3 Regulárne výrazy

27 Zápisy RV Otázka, či zápisy RV tvoria regulárny jazyk, nie je úplne správne formulovaná. Nájdite a vysvetlite problém a navrhnite spôsob jeho riešenia.

28 Prevod KA na RV ZAPOCET 07/08 c) Prevedte KA na RV: (viz úloha 2-r.2007)

a) $\text{Stavy}=\{1,2,3\}$, $\text{Abeceda}=\{a,b\}$, $\text{Init}=1$, $\text{Final}=\{2\}$, $\text{Delta}=\{1,a-1, 1,b-2, 2,a-1, 2,b-3, 3,a-3, 3,b-2\}$

b) $\text{Stavy}=\{1,2,3\}$, $\text{Abeceda}=\{a,b\}$, $\text{Init}=1$, $\text{Final}=\{2\}$, $\text{Delta}=\{1,a-2, 1,b-1, 2,a-3, 2,b-1, 3,a-3, 3,b-2\}$

c) (2008) $\text{Stavy}=\{1,2,3\}$, $\text{Abeceda}=\{a,b\}$, $\text{Init}=1$, $\text{Final}=\{3\}$, $\text{Delta}=\{1,a-3, 1,b-2, 2,a-3, 2,b-1, 3,a-2, 3,b-3\}$

z1) Vysvetlite, prečo 1) sa odstraňujú najprv slučky, 2) sa pridáva vstupná, resp. výstupná hrana, 3) sa spájajú multihraný, 4) sa odstraňuje vrchol naraz, a nie jednotlivé hrany 4a) je možné odstraňovať jednotlivé hrany? Prípadne za akých podmienok? Ak áno, popíšte a zdovodnite správnosť.

29 RV pre log. formule Utvorte RV popisujúci

a) log. formule v konjunktívnej (alebo disjunktívnej) normálnej forme, obsahujúce true, false a výrokové premenné. (Zápis zjednodušte :-)

b) ľahšie varianty: bez negácie, len s jednou binárnou spojkou

c) aké sú problémy so zápisom premenných a ako je možné ich vyriešiť?

d) vysvetlite neefektivitu zápisu z hľadiska jeho dĺžky, resp. zložitosti (porovnajete neskôr so zápisom gramatik)

e) je možné zapisovať obecné (výrokové) logické formule pomocou RV

30 Prevod RV na KA pomocou next

a) Definujte (matematicky, Prolog, Haskell) funkcie next, first, last, empty, isChar a prípadne ďalšie pomocné pomocou štrukturalnej indukcie (podľa tvaru RV) pre druhú metódu prevodu RV na KA (pomocou susednosti písmen). Funkcia first, resp. last, pre RV určuje výskyty písmen, ktoré môžu byť prvé, resp. posledné. Funkcia empty určuje, či RV prijíma prázdne slovo (a používa sa pri definícii ostatných funkcií). Funkcia next pre RV určuje, ktoré výskyty písmen smú byť za sebou, tj. výstup je množina (alebo zoznam) dvojíc.

b) Zavedte pojem výskytu RV a vysvetlite, prečo je potreba.

c) Aké máte programátorské možnosti realizovať RV – aspoň dve. Musíte ošetrovať zátvorky?

d) Pre regulárny výraz $a_1.b_2^*. (a_3 + b_4)^*$ s daným číslovaním výskytov písmen určite hodnoty funkcií next, first, last, empty pre všetky podvýrazy (pri zvolenom uzátvorkovaní).

e) Varianta: Očíslujeme nielen výskyty symbolov abecedy, ale aj pomocných symbolov vo výraze. Ako potom definujeme prechodové pravidlá pre pozície (včetně prechodov pomocou lambda)?

31 RV z PJ

Navrhnite RV pre

a1) reálne čísla, a2) celé čísla v C-notácii (aj hexa, octal)

b) reťazce v Pascale, C, ... (+sekv.stroj)

c) atomy (identifikátory) v Prologu

d) formátovacie reťazce v C (+sekv.stroj) - pre printf...

e) "kvalifikované" mená (ľavé strany priradenia) - ..., len "regulárnu časť"

f) komentáre

g) ... z XML (a HTML): tag, tag s atributmi, CDATA, komentare ...

h) rozne bežné entity: dátumy (v roznych formátoch), rozsahy

čísel, teplot (napr. §15.4-16.2)

i) súčasti mailov pre analýzu spamov: doménové mená, číselné IP adresy, http odkazy - na stránky, na obrázky, ...

32 Vlastnosti RV Rozhodnite, či platí pre každé (regulárne) jazyky L, L_1 a L_2 .

- $d_w(L) \subseteq L$
- $d_w(L) \subseteq w.L$
- $w.d_w(L) \subseteq L$
- $d_w(L) \supseteq L$
- $L1.(L1/L2) \supseteq L2$
- $L1.(L1/L2) \subseteq L2$
- $L1.(L1 \setminus L2) \supseteq L2$:-)
- $L1.(L1 \setminus L2) \subseteq L2$
- $L2.(L1/L2) \supseteq L1$
- $L2.(L1/L2) \subseteq L1$
- $L2.(L1 \setminus L2) \supseteq L1$
- $L2.(L1 \setminus L2) \subseteq L1$
- $L1^* + L2^* \supseteq (L1 + L2)^*$
- $L1^* + L2^* \subseteq (L1 + L2)^*$
- $(\lambda + L)^* \supseteq L^*$
- $(\lambda + L.L)^* \supseteq L^*$
- $(L + L.L)^* \supseteq L^+$
- $(\lambda + L)^* \supseteq L^*$
- $L1.L2 \subseteq L1^* + L2^*$
- $(L1.L2)^* \subseteq (L1 + L2)^*$
- $(L1.L2)^+ \subseteq (L1 + L2)^+$
- $L1^*.L2^* \subseteq (L1 + L2)^*$
- $L1^*.L2^* \supseteq (L1 + L2)^*$
- $\lambda \in L \Rightarrow L^+ = L^*$
- $L \setminus L^+ \supseteq L^+$
- aa) $L.L \supseteq \{w.w \mid w \in L\}$
- ab) $L.L \subseteq \{w.w \mid w \in L\}$

4 Dvojsmerné automaty

33 "Odmocnina" jazyka (6.2.) Nech L je jazyk prijímaný konečným automatom A . Zostrojte dvojsmerný (nedeterministický) konečný automat prijímajúci jazyk

- $\{\#u\# \mid uu \in L\}$ - druhá odmocnina
- a2) (poučné:) Popíšte, čo všetko si musí automat pamätať pri jednom prechode, aby rozpoznal jazyk L . Ako z daných informácií zistí, či prijíma prečítané slovo?
- b) $\{\#u\# \mid u^n \in L\}$ pre dane n
- c) $\{\#u\# \mid \exists n : u^n \in L\}$
- d) $\{\#v\# \mid uv \in L \wedge |u| = |v|\}$ - druhé polovicky slov, a varianty
- e) navrhnete jazyk(y), ktoré využívajú dvojsmernosť automатов "lokálne"
- f) $\{\#w\# \mid ww^R \in L\}$ - vhodný pri prevode do KA sa vhodné ilustrujú techniky (obojsstranne)
- g) DC: navrhnete ďalšie druhy podmienok (zobecnenia), ktoré je možné realizovať 2KA. Ktore z nich vyžadujú nedeterminizmus?

34 Male 2KA Skonstruujte 2KA automat, ktorý rozpoznáva $\#u\#$, ak L je jazyk obsahujúci slovo 1. "ab" (resp. 2. "aab", 3. "aaa") a a) $uu \in L$ b) $uu^R \in L$ c) $u^R u \in L$ d) $uu^R u \in L$

Pozn. Automaty musia kontrolovať vlastnosť súšmerne aj protišmerne (a aj rozdelené), preto typicky nebudú redukované.

35 Automaty s (1) kamienkom

- $\{\#w\# \mid \exists u, v : w = uv \wedge u^R uvv^R \in L\}$ - suvisla nejaka

polovička slova

- $\{\#w\# \mid \exists u, v : w = uv \wedge uu^R v \in L\}$ - povodny chybný príklad! :-)

c) $\{\#w\# \mid \exists u, v : w = uv \wedge vu \in L\}$ - cyklicky posun

Automaty s jedným kamienkom su (asi) silne ako RJ, s 2 kamienkami su silnejšie určite, rozpoznaju $(0^i 1^i)$.

$w = uv \wedge u^R v^R \in L$ - od koncov ku kamienku

$uu^R v^R v \in L, uvv \in L \dots$ a varianty

slovotvaru $(u|u^R|v|v^R)^* \in L$

gx) analogicky ku g) minuleho príkladu, navrhnete jazyky pre 2KA s 1 kamienkom, ktoré nie su priamociaro realizovateľne 2KA bez kamienku.

5 Sekvenčné stroje

36 Kontrola výstupu Navrhnete, ako skontrolovať výstup Mealyho a Mooreovho stroja konečným automatom. Vysvetlite použitie pre podporu nedeterminizmu a dokazovanie neexistencie sekvenčných strojov.

Vysvetlite vzťah ku generatívnym a analytickým gramatikám, resp. spôsobu ich použitia. (Pr 7.1)

Zobecnite pre prevody 1:n, m:1, m:n.

37 Prevody sústav Medzi bin, octal, hexa sústavou. Smer, okrajové podmienky. Vysvetlite vydávanie prázdného výstupu a skupiny znakov. B-O, B-H podobné, O-B sekvencie, O-H postupné. Stroje (po zliatí stavov) sa mierne líšia.

Je možný prevod medzi binárnou a ternárnou sústavou sekvenčným strojom? Návod: použite minulý príklad (a algebru). Je možný prevod medzi binárnou a unárnou sústavou sekvenčným strojom? Návod: dtto.

38 Oneskorenie výstupu, výhľad ZAPOCET 06/07: $(a+1$ alebo $a+2)$ and $a+3$

a) Navrhnete stroj, ktorý každú nulu na vstupe, pred ktorou su dve nuly, mení na 1. Zvyšne znaky nechava bez zmeny.

b) Navrhnete stroj, ktorý vydava diferencnu informáciu. Vstupna abeceda je 0, 1, vystupna je -1, 0, 1 a vystupna hodnota je rozdiel posledných dvoch znakov.

Navrhnete 1) Mealyho a 2) Mooreov stroj a nasledne prevedte na 3) ten druhý stroj.

39 Oneskorenie výstupu, výhľad Navrhnete, ako realizovať Mealyho a Mooreov stroj pre vydávanie identického výstupu oneskoreného o 3 taktov. Rozoberte okrajové podmienky, tj. počiatkové symboly a koncové symboly.

b) obecne: oneskorenie o n taktov.

40 Streamové operácie Sú tieto operácie realizovateľné na sekvenčnom stroji?

prevod C zápisu reťazca na postupnosť znakov

tokenizácia v kompilátore (identifikátory, čísla, operátory, reťazce, komentáre, ...)

CRC

blokové šifrovanie, lavínový efekt

prevody sústav (viz)

... hašovanie (pomocou xor a shift), ...

...

41 Stroj Aho-Corasicková Vysvetlite, v čom sa podobá a líši AC stroj od sekvenčných strojov. Rozoberte pamäťovú náročnosť.

6 Gramatiky

42 Návrh gramatik

slová začínajúce aba

pre jazyky $(i)^i$

jazyk obsahujúci rovnaký počet a a b

jazyk $\{0^i 1^j | i \leq j \leq 2i\}$, gramatiku zmeňte na jednoznačnú

jazyk $\{a^i b^i c^i\}$, (kontextová)

dobre uzátvorkované výrazy (len zátvorky), rozne druhy zátvoriek (to je jadro HTML)

fraktály: program pre čiarovú grafiku (včetně zmeny merítka) pre úsečku s "píkou" - súvislosť s master teoreom pre riešenie rekurzívnych rovníc.

zdvojenie slova: ak $S \rightarrow * \rightarrow \alpha$, potom $D \# S \# \rightarrow * \rightarrow \alpha \alpha$
nasobenie (DC): Navrhните system pravidiel, ktorý vetnu formu $\#M1^m 01^n 0\#$ prepíše na $\#1^{n \cdot m} 0\#$ (bez opustenia useku ohrančeného #).

jazyk a^{2^n} , (kontextová, prevedte na nevypustajúcu)

aritmetické (a ine) kombinatory: Navrhните system (meta)pravidiel, ktorý dovolí počítat hodnotu polynomu (a iných funkcií). Konkrétny polynom ako kombinátor.

výpočty funkcie f v (obecných) gramatikách: (čísla kódované obvykle unárne) 1) jazyk $L_1 = \{f(n) | n \in N\}$, 2) jazyk $L_2 = \{(n, f(n)) | n \in N\}$ s vhodným kódovaním dvojíc, 2b) ako vyberieme výslednú hodnotu f(n) pre dané n z L_2 ?

DC: CNF, porovnať s zápisom regulárnym výrazom (bez zátvoriek)

logické formule (s kvantifikátormi, zátvorkami), aritmetické výrazy, (priama rekurzia, tail-rekuzívny tvar)

pozn.: každý výskyt neterminálu je nezávislý (v bezkontextovej gramatike) a generuje čokoľvek "bez omedzení". Omedzenia (na použitie pravidla) musia byť popísané kontextom neterminálu (v rámci používania gramatik).

43 Gramatiky – príklady. Kde ste sa s gramatikami už stretli? Uvedte príklady "zo života" (aspon 3x).

44 Aritmetické výrazy Pr.7.2, varianty

a) Pridat unárne minus do aritmetických výrazov. Porovnať "pridanie" do lexikálnej a syntaktickej analýzy z hľadiska zmien v gramatike.

b) Overte, či je gramatika jednoznačná a prípadne odstráňte viacznačnosť. (Z hľadiska priorit a z hľadiska asociativity.)

c) Ako sa zmení gramatika, ak pridáte ďalšiu (binárnu) operáciu, napr. umocňovanie?

d) Ako by ste pristupovali k vytvoreniu gramatiky pre analýzu výrazov v Prologu (Haskell), tj. za situácie, že si môžete definovať nové operatory. (Ako sú možnosti implementácie? – najmä, tabulky)

e) Musíte takto obecné metódy použiť aj pri analýze výrazov v C++, keď môžete predefinovať použitie operatorov? rozšíriteľnosť gramatiky, prioritne rozlíšenie operatorov

45 Pregenerovanie a podgenerovanie "Overgeneration"

... k čomu to je dobré

overgeneration: ak nemáme presný popis jazyka, ale chceme určite zahrnúť dany podjazyk.

podgen.: namiesto celého jazyka budeme analyzovať len core-language; (do nej "zdrojaky" musíme previesť); analyticky a generatívne použitie ...

46 Core-language Jadro jazyka. ("Podgenerovanie")

Ako si pri písaní gramatiky (trochu) zľahčiť prácu

Porovnať: Pohľad tvorcu gramatiky a (nepriameho) užívateľa jazyka (generovaného gramatikou). Prirodzené a umelé (programovacie, vnorené) jazyky.

47 Nestandardne ulohy

Klasická uloha je tvorba gramatiky alebo automatu, ktorý má dané vlastnosti, popisane v nejakom inom formalizme.

Ina uloha je zistiť gramatiku z príkladov slov, pozitívnych (su v jazyku) aj negatívnych (nie su). Samozrejme, gramatika bude pregenerovávať.

48 retazcove operacie

tj. na zdrojakochoch. Hm.

7 Bezkontextové gramatiky

49 Prevod gramatiky ZAPOCET 08/09: 5+a), 5+b)

Danú gramatiku prevedte a) do Greibachovej normálnej formy, b) Chomského NF, c) do tvaru bez ľavej rekuzie.

1) Počiatočný symbol je E, pravidlá sú (gramatika pre binárne prefixné a postfixné operatory, so známou četnosťou)

$E \rightarrow oEE \mid EpE \mid EEq \mid T$

$T \rightarrow rF \mid Fs \mid (E, T)$

$F \rightarrow (E) \mid i$

2) Počiatočný symbol je E, pravidlá sú

$E \rightarrow EaE \mid T$

$T \rightarrow (E) \mid i \mid \lambda$

3) Počiatočný symbol je E, pravidlá sú

$E \rightarrow EaE \mid Tb \mid Fb$

$T \rightarrow (E) \mid i \mid \lambda$

4) Počiatočný symbol je S, pravidlá sú

$S \rightarrow SAa \mid a \mid Sb \mid SBb$

$A \rightarrow Ab \mid a$

5) Počiatočný symbol je E, pravidlá sú

$E \rightarrow EaE \mid T$

$T \rightarrow Tb \mid Fc$

$F \rightarrow (E) \mid i \mid \lambda$

50 Prevod gramatiky 3 Danú gramatiku prevedte a) do Greibachovej normálnej formy, b) Chomského NF, c) do tvaru bez ľavej rekuzie. Počiatočný symbol je S, pravidlá sú

$S \rightarrow ACE$

$A \rightarrow a \mid B \mid C$

$B \rightarrow b \mid A \mid \lambda$

$C \rightarrow c \mid D$

$D \rightarrow d \mid C \mid E$

$E \rightarrow e \mid FG \mid FH$

$F \rightarrow f \mid \lambda$

$G \rightarrow g$

b) Vysvetlite, prečo sa musí odstraňovať ľavá rekuzia pre jeden neterminál (alebo skupinu vzájomne rekurzívnych neterminálov) naraz.

$S \rightarrow Sa \mid Sb \mid c$

51 Zasobníková analýza Podľa gramatiky 2) vyššie analyzujte zasobníkovým automatom: a) $iaia$ b) $(iaia)(iaia)$ c) $a((i)a)$

52 Odstraňovanie nepriamej ľavej rekurzie pre BKJ. Sformulujte obecný postup pri odstraňovaní nepriamej ľavej rekurzie. Aplikujte na gramatiku:

S → ACE
 A → Bd | aA | C
 B → Ae | bB
 C → c

Gramatiky, kde všetky cykly v jednej komponente súvislosti (danej ľavým prepisom), sa dajú spracovať jednoduchšie (postupne).

b) V akom poradí je výhodné spracovávať silne súvislé komponenty?

53 BKJ jazyky ZAPOCET 08/09: c3, n2)

Rozhodnite (a dokážte), či jazyky patria do triedy BKJ, ak $i, j, k \geq 0$.

- a) $\{a^i b^j c^k | i \neq j \neq k\}$
- b) $\{a^i b^j c^k | i \neq k \vee j \neq k\}$
- c) $\{a^i b^j c^k | k = \max(i, j)\}$
- c1) $\{a^i b^j c^k | k = \min(i, j)\}$
- c2) $\{a^i b^k c^j | k = \min(i, j)\}$
- c3) $\{a^i b^k c^j | k = \max(i, j)\}$
- d) $\{a^i b^i c a^j b^j | \text{true}\}$
- e) $\{a^i b^i a^j b^j | \text{true}\}$
- e1) $\{L^+ | L = (a^i b^i)\}$
- f) $\{a^i b^j c^{i-j} | i > j\}$
- g) $\{a^i b^j c^k | k = \max(i - j, 0)\}$
- h) $\{a^i b^{i+j} a^j | \text{true}\}$
- i) $\{a^i b^j c^k | 1 \leq i \leq j \leq k\}$ na prednáške
- j) $\{a^{2^n} | n \in \mathbb{N}\}$
- k) $\{a^{2^n} | n \in \mathbb{N}\}$
- l) $\{ww | w \in \Sigma^*\}$
- m) $\{ww' | w, w' \in \Sigma^* \& |w| = |w'| \& w \neq w'\}$
- n) $\{w | w \in \{a, b, c, d\}^* \wedge \#a = \#b \wedge \#c = \#d\}$
- n2) $\{w | w \text{ je spravne uzatvorkovane slovo s dvomi druhmi zatvorek}\}$

54 Obecné pumping lemma pre BKJ. Sformulujte pumping lemma s podtrhnutými písmenami.

Dokážte, že jazyk $\{a^i b^j c^k | i \neq j \neq k \neq i\}$ nie je bezkontextový. Idea. Pumpujem a (podtrhnutím) a môžem si vybrať, či sa traťm do počtu b alebo počtu c. (Počet b,c v tvare $\#a + p!$, aby som sa traťm pri ľubovoľnom pumpovaní.)

55 BKJ - Vylúčenie veľkých dier Máte slovo s dĺžkou l. Dokážete pre vhodné (pevné) d zaručiť, že jazyk obsahuje slovo dĺžky najviac $l + d$?

56 Kontextové gramatiky a) Pre kontextové jazyky z príkladu 53 napíšte kontextové gramatiky. (Alebo aspoň ideu gr.) (poučné: b,c,k,l) b) Pre ostatné prípadne bezkontextové gramatiky.

57 Doplnok BKJ a) Dokážte, že doplnok jazyka $\{a^i b^i c^i\}$ je bezkontextový.
 b) Jazyk $\{a^i b^j c^k | i \neq k \vee j \neq k \vee i \neq j\}$ nie je doplnok jazyka sub a). Odstráňte chybu (správne).
 c) Dokážte, že doplnok jazyka $\{ww | w \in \Sigma^*\}$ je bezkontextový.
 d) Popíšte doplnok jazyka sub c) a vysvetlite, čím sa líši od $\{ww' | w, w' \in \Sigma^* \& |w| = |w'| \& w \neq w'\}$.
 e) Popíšte použitý trik pre vytváranie "dobrých" doplnkov obecné.

58 Opakovanie: BKJ

- a) Popíšte Dyckove jazyky.
- b) Charakterizujte BKJ pomocou Dyckových jazykov.
- c1) Je HTML Dyckov jazyk. c2) Je XML Dyckov jazyk.
- d) Popíšte unárnu reprezentáciu čísel.

59 Varianty Halting Problému Prevedte medzi základným H.P. a popísanými problémami. Ktorý smer prevodu je správny, ak chceme o novom probléme P' dokázať, že je nerozhodnuteľný?

- a) Popíšte, tj. zformulujte základný Halting Problém.
- b) Rozhodnúť, či sa TS zastaví na nejaké (tj. ľubovoľné) slovo.
- c) Rozhodnúť, či TS použije danú inštrukciu pri nejakom výpočte. (Eliminácia mŕtveho kódu)
- d) Formulujte tieto problémy pre klasické programovacie jazyky.

60 Zaradte do Chomskeho hierarchie ZAPOCET 07/08 a2), q), 08/09 q)

Rozhodnite (a dokážte), či jazyky patria do triedy BKJ, ak $i, j, k \geq 0$. viz aj 53

- a1) $\{a^i b^j | i = 2j\}$
- a2) $\{w | w \in \Sigma^* \& |w|_a = 2|w|_b\}$ Napiste aj BKG.
- a3) odvodte a2: $L \notin RJ$ pomocou a1)
- n) $\{wuw^R | w, u \in \Sigma^*\}$, $\Sigma = \{a\}$;-)
- o) $\{wuw^R | w, u \in \Sigma^*\}$, $\Sigma = \{a, b\}$
- p) $\{wuw | w, u \in \Sigma^*\}$, $\Sigma = \{a, b\}$
- q) $\{wuw^R | w \in \Sigma^+, u \in \Sigma^*\}$, $\Sigma = \{a, b\}$ Kde zlyha dokaz neprislusnosti pomocou pumpovania?
- r) $\{wuw^R | w \in \Sigma^*, u \in \Sigma^+\}$, $\Sigma = \{a, b\}$
- s) $\{ww | w \in \Sigma^*\}$, $\Sigma = \{a, b\}$
- s2) $\{ww | w \in \Sigma^*\}$, $\Sigma = \{a\}$
- t) $\{ww^R w | w \in \Sigma^*\}$, $\Sigma = \{a, b\}$

61 Schema gramatik V minulých príkladoch ste navrhli gramatiky napr. pre jazyky $a^{(2^n)}b$ a $a^{(n^2)}b$, čo su v podstate cisla v unarnej sustave. Navrhnite sposob (schema gramatik), ktorým umožni jednoducho generovať jazyky typu $a^{f(n)}$, kde $f(n)$ je nejaká dopredu dana aritmeticka funkcia, napr. polynom. (univerzálne operácie a "zdrojak" – reprezentáciu popisu, tj. kombinátory)

Idea: Skladanie operatorov a ich prevod do gramatik. Pozn: Obecná metóda prevodu jedného formalizmu na iný, pomocou elementarných operácií a "skladacích" operatorov. Tri úrovne analógie techník: 1. príklad (navrh) konkrétnej gramatiky, 2. príklad schémy gramatik, 3. príklad prevodu formalizmov. Táto myšlienka su vlastne kombinátory, zname z Haskellu a kombinatorickej logiky. Len obecné pracuju na stromoch a nie na retazcoch.

62 Eliminacia λ-pravidiel

Rozobrat jemnosti prevodu. Cyklus pri okamžitom odstránovaní lambda-pravidiel, a oprava (tabelaciou), iterácia do pevného bodu, davkove spracovanie ...

63 ad Prevod RG na KA, 7.x

Kolko druhov pravidiel rozlisujeme a rozne osetrujeme pri prevode?

Aky druh Konecneho Automatu prirodzene vzniká pri prevode?

Je mozne pri prevode zlcucovat stavy, ktore patria (pri standardnej konstrukcii) roznyh pravidlam, ale vedu z rovnakeho neterminalu cez rovnake pismeno abecedy. Ako vystizne (a jednoznacne) nazvete stavy, okrem anonymous1, anonymous2, ...

ad opacny prevod KA na RG, 7.x: Da sa pri prevode vyuzit informacia, ze prevadzany automat je deterministicky, na zjednodusenie algoritmu? (Tj. vyhnutie sa urcivym castiam algoritmu?)

64 Gramatika vs schema pravidiel

aky je rozdiel medzi gramatikou genereujucou $a^i b^j c^i$ a schemo pravidiel $S \rightarrow a^i b^j c^i$, ktora sa ma stat sucastou gramatiky. Pre ake hodnoty i mozete schemu pouzít?

65 Analyza BKJ Navrhnete algoritmus dynamickeho programovania, ktory bude efektívne zistovat, ci dane slovo patri do (nedeterministickeho) BKJ. Idea: analyzujte odspodu a vyuzite Chomskeho NF. Usek slova od-do je odvoditelny z neterminalu X . CYK - Už je v predn.

Ake zlepšenie prinesie lazy vyhodnocovanie zhora, s tabelaciou?

66 Analyza rekurzivnym zostupom

Pre BKG pravidla sa daju prepisat priamo do programu, napr. do Prologovskych klauzuli (vcetne nedeterminizmu), alebo do klasickych jazykov (obvykle pre deterministicke gramatiky).

Problemy s lavou rekurziou.

Priklad: pravidla $E \rightarrow T E1, E1 \rightarrow + T E1 - \lambda$ budu v Prolog-like jazyku

```
proc_E(...) :-
    proc_T(...),
    proc_E1(...).
proc_E1(CodeE1, ...) :-
    read_terminal('+'),
    proc_T(CodeT, ...),
    proc_E1(CodeE1a, ...),
    mkCodeE1(CodeT,CodeE1a, /* Code'+', */ CodeE1)
    % semantic action (+,+,-)
; read_lambda(...).
```

67 Oprava 2-NKA 6.2.d

Puvodni zadani druhého príkladu cviceni 6 obsahovalo príklad $\#u\#|u = wv \wedge ww^Rv \in L$, a po dohode jste jej nahradili aktualnim znenim $\#u\#|u = wv \wedge w^Rwv \in L$.

Mam pocit, ze lepsi bylo pozmenit slovní popis k jazykum: Necht L je jazyk prijimany konecnym automatem A . Sestrojte nedeterministicky automat prijimajici jazyk. Napoveda: Konstrukci zjednodusi, zacneteli nejprve konstrukci dvousmerneho automatu (je-li to mozne).

Puvodne uvedene zadani ukazuje, ze ne vzdy je mozno dvousmerny automat primocare pouzít, ze by se lepe hodil dvousmerny automat s kaminkem. Presto uvedena metoda vede k nalezeni nedeterministickeho automatu pro dany jazyk. (A otevira otazku sily dvousmerneho automatu s jednim kaminkem).

...
 $\dot{}$ Nejsm si uplne jisty, ze to vidim. Asi bych si nejdrive potreboval vyjasnit definici dvousmerneho automatu s kaminkem. Muzu tim kaminkem posouvat nebo ho mohu polozit jen jednou a potom uz jen pouzivat? Jak by to vypadalo formalne (myslim zapis instrukci)?

Samozrejme model, kde kaminek mohu polozit vicekrat je uzitecnejsi, ackoli jak tvrdim, stale se silou KA. Dva kaminky vedou k moznosti rozpoznavat napr $0^n 1^n$, takže jsou evidentne silnejsi.

Snad to zvladnu :) : $\delta : (Q \times \{0,1\}) \times (\Sigma \times \{0,1\}) \rightarrow (Q \times \{0,1\}) \times \{0,1\} \times \{L,R\}$ s tim, ze $(q, k_1)(a, k_2) \rightarrow$

$((q', k_3), k_4, smer)$ je v prechodove funkci jedine za podminky $k_1 + k_2 = k_3 + k_4$ (zakon zachovani kaminku). Interpretace pro simulaci je takova, ze pri navratu ve stavu q s k kaminky na policko s pismenem a , ktere jsem opustil instrukci $\rightarrow ((q_1, k_1), k_2, smer)$, lezi na policku k_2 kaminku, takže postupuji dle instrukce $(q, k), (a, k_2) \rightarrow$. Na policku kde jsem dosud nebyl nelezi kaminek nebo-li pouzivam $((q, k)(a, 0)) \rightarrow$. Automat s jednim kaminkem zacina ve "stavu" $(q_0, 1)$. (Pro dvoukaminkovy automat by tam bylo "vsude" $\times \{0, 1, 2\}$).

8 Metapriklady

68 Co sa (tiez) mozete/mate naucit na (tejto) skole a nie su na to specialne predmety?

Ako (ne)pisat nazvy prikladov :-)

Otazky: Klast spravne otazky a poskytovat spravne, strucne, ale presne odpovede. (zachytit a vyjadrit "jadro" problemu. Rozlisovat lahke (trivialne) a tazke otazky.

Komunikaciu, vysvetlovanie,

Prezentaciu: pripravu slajdov a reci, vlastnu prezentaciu, vyber vhodnych prikladov (a obrazkov), manazerske triky :-)

odhad narocnosti (dlhodobych) uloh: Na naucenie cinstiny nestaci noc pred skuskou (viz vtip).

naucit sa pouzivat naucene nastroje, rozlisovat vhodnost nastoov a vyberat vhodny nastroj

?? nachadzat analogie, zobecnovat, abstrahovat ?? ...

69 Vzťah nástrojov a problémov

Preco sa ucite viac technik (nastrojov) na riesenie jednoho problemu.

Cloveku, ktory ma len kladivo, pripada cely svet ako klinec.

70 Testy

Pr.: Pri odstranovani λ -pravidiel:

Q: Ktora mnozina neterminalov su prepisatelne na λ ? ... ani jedna z minulych - vypiste.

Q: Kolko pravidiel ma nova gramatika. ... iny pocet - uvedte.

====koniec automatov

9 Prolog/Hs

Hold ma priklady napadali nastriedacku.

71 Generovanie fraktálnych kriviek Generovanie a interpretacia popisu (do prednášok?)

72

73

a este Prolog/Hs dole ...