VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

KLASIFIKÁCIA GRAMATÍK, JAZYKOV A AUTOMATOV (ZÁKLADNÉ TYPY A VYUŽITIE)

CLASSIFICATION OF GRAMMARS, LANGUAGES AND MACHINES (BASIC TYPES, USE)

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE MICHAELA KLČOVÁ

AUTHOR

VEDOUCÍ PRÁCE Mgr. MARTINA BOBALOVÁ, Ph.D.

SUPERVISOR

BRNO 2009

Vysoké učení technické v Brně
Fakulta podnikatelská

Akademický rok: 2008/2009
Ústav informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Klčová Michaela

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Klasifikácia gramatík, jazykov a automatov (základné typy a využitie)

v anglickém jazyce:

Classification of Grammars, Languages and Machines (Basic Types, Use)

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Seznam odborné literatury:

ČERNÁ,I., KŘETINSKÝ,M. a KUČERA,A. Automaty a formální jazyky I. Brno: Masarykova univerzita, Fakulta informatiky, 2002. 159 s.

GÉSCEG,F. a PEÁK,I. Algebraic theory of automata. 1. vyd. Budapest: Akadémiai Kiadó, 1972. 326 s. ISBN 2-6198-3497-1.

GRUSKA,J. Foundations of computing. 1. vyd. International Thompson Computer Press. 716 s. ISBN 1-85032-243-0

HOPCROFT,J.E. a ULLMAN,J.D. Formálne jazyky a automaty. 1. vyd. Bratislava : Alfa, 1978. 343 s. MDT 519.682.1.

LEPISTO,T. a SALOMAA,A. Automata, languages and programming. 1. vyd. Berlin: Springer-Verlag, 1988. 742 s. ISBN 3-540-19488-6.

ROSEN,K.H. Discrete mathematics and its applications. 4. vydanie. Singapore: MCGraw- Hill Book Co, 1999. ISBN 0-07-289905-0

Vedoucí bakalářské práce: Mgr. Martina Bobalová, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2008/2009.

L.S.

Ing. Jiří Kříž, Ph.D.	doc. RNDr. Anna Putnová, Ph.D., MBA
Ředitel ústavu	Děkan fakulty

V Brně, dne 24.05.2009

Abstrakt

Práca je napísaná ako stručný prehľad základnej teórie k danej téme s rozšírením na spracovanie postupov a riešenie príkladov.

Práca je rozdelená na dve základne časti. Prvá časť je zameraná na základné teoretické poznatky o jazykoch, gramatikách a automatoch. Podrobne je v nej vysvetlená teória vzťahujúcu sa k formálnym jazykom a operáciám nad nimi, ku gramatikám a ich rozdeleniu a ku konečným automatom, spolu s rozdelením na deterministické a nedeterministické.

Druhá časť je venovaná spracovaniu, vysvetleniu a spočítaniu príkladov týkajúcich sa danej témy od jednoduchých až po zložité.

Abstract

This bachelor's thesis is written as a brief list of fundamental theory on the given topic with extensions of working out methods and solving examples.

The thesis is divided into two basic parts. The first part is focused on the essential theoretical pieces of knowledge about languages, grammars and machines. There is a particular explanation of the theory related to formal languages and operations on languages, grammars and their separation, and to finite state machines, together with dividing them into deterministic and nondeterministic ones.

The second part is devoted to working out, explanation and calculation of the examples applied to the given topic - from the simple ones to the difficult ones.

Kľúčové slová

Abeceda, deterministický konečný automat, formálne jazyky, gramatika, Chomského hierarchia, konečný automat, nedeterministický konečný automat, operácie nad abecedou, operácie nad jazykmi, pumping lemma.

Key words

Alphabet, deterministic finite state machine, formal languages, Grammar, Chomsky's Hierarchy, finite state machine, nondeterministic finite state machine, operations on Alphabet, operations on languages, pumping lemma.

Bibliografická citácia

KLČOVÁ, M. *Klasifikácia gramatík, jazykov a automatov (základné typy a využitie)*. Brno: Vysoké učení technické v Brne, Fakulta podnikateľská, 2009, 55 s. Vedúci bakalárskej práce Mgr. Martina Bobalová, Ph.D.

Čestné prehlásenie

Prehlasujem, že táto práca je mojim pôvodným autorským dielom, ktoré som vypracovala samostatne. Všetky zdroje, ktoré som použila alebo z nich čerpala, v práci riadne citujem s uvedením úplného odkazu na príslušný vzor.

V Brne dňa ....

podpis

Poďakovanie

Chcela by som poďakovať vedúcej práce Mgr. Martine Bobalovej, Ph.D, za vedenie a pomoc pri písaní bakalárskej práce.

OBSAH

Úvod

Bakalárska práca sa zaoberá kapitolou teoretickej informatiky týkajúcou sa jazykov, automatov a gramatík. Táto problematika je súčasť predmetu diskrétna matematika vyučovanom na Fakulte podnikateľskej Vysokého učenia technického v Brne. Je jej však venovaný len malý priestor v rámci tohto predmetu. Jazyky, automaty a gramatiky sú základom informatiky a preto je potrebné kvalitné a presné pochopenie ich princípov.

Práca je rozdelená na dve základné časti, prehľad teórie a spracovaná zbierka príkladov. Snahou prvej časti práce je zovšeobecnenie a zostručnenie poznatkov zo základných literatúr tak, aby podávala základný, stručný a ľahko pochopiteľný prehľad teórie. Má slúžiť študentom Fakulty podnikateľskej ako podklad pre pochopenie učiva a vytvoriť základ pre následnú schopnosť spracovania príkladov.

Druhá časť je spracovaná ako zbierka príkladov. Je v nej výber príkladov od jednoduchých, pre ľahké pochopenie základnej problematiky, až po zložité, pre zvládnutie skúšky z tohto predmetu. Ku každému príkladu je vysvetlenie postupu riešenia spolu so samotným riešením.

1 Vymedzenie problému a cieľ práce

1.1 Vymedzenie problému a návrh realizácie

V každej spoločnosti vrátane univerzít sa objavujú určité prednosti a nedostatky. Nedostatkom, ktorý je riešený v tejto bakalárskej práci je malé množstvo času a priestoru venované výučbe v oblasti jazykov, automatov a gramatík v predmete diskrétna matematika vyučovanom na Fakulte podnikateľskej Vysokého učenia technického v Brne. Práca sa zaoberá klasifikáciou jazykov, gramatík a automatov. Je v nej spracovaný zovšeobecnený teoretický prehľad základných poznatkov, ktoré sú dôležité pre následné spracovanie príkladov potrebných pre dôkladné pochopenie témy a zvládnutie učiva. Spracovávaním príkladov je vytvorený ucelený súbor, ktorý kompletne zahrňuje vyššie uvedené témy. K tomuto súboru príkladov sú vytvorené postupy ich riešení a samotné riešenia.

V práci sú využité hlavne tieto 2 základné metódy:

Systematický popis

V práci je pomocou systematického popisu realizovaný pokus priblížiť študentom teóriu vzťahujúcu sa k formálnym jazykom a operáciám nad nimi, teóriu gramatík a ich rozdelenia a teóriu konečných automatov spolu s ich rozdelením na deterministické a nedeterministické. Ďalej je v práci veľmi prehľadne popísaná dôkazová metóda regularity jazyka, pummping lemma.

Zovšeobecňovanie

Práca je napísaná v čo najjednoduchšej forme, aby dotyčný čitateľ (študent) získal všeobecný prehľad o teórii automatov, gramatík a jazykov. Je v nej zovšeobecnená a zhrnutá teória k danej téme tak, aby dávala zmysel a bola v zjednodušenej forme dobre pochopiteľná.

1.2 Cieľ práce

Cieľom bakalárskej práce je priblíženie základných typov gramatík, jazykov a automatov, ich základné rozdelenie a využitie v ekonomickej a informatickej praxi. Rozšírenie výučby predmetu diskrétna matematika na Fakulte podnikateľskej v Brne.

Bližšie vysvetlenie danej tematiky a spracovanie postupov a riešenia príkladov od jednoduchých až po zložitejšie.

2 Stručný úvod do teórie jazykov, automatov a gramatík

Prvé základy formálnych jazykov vznikli vďaka americkému matematikovi Noamovi Chomskému, ktorý pri štúdiu prirodzených jazykov vytvoril matematický model gramatiky a jazyka. Po Chomského definícii využili základné objekty gramatiky Backus a Nauer a vytvorili Backus- Nauerovu formu. Táto forma je definíciou syntaxe pre programovací jazyk Algol 60. S ďalším vývojom sa aplikovala teória jazykov v oblasti prekladačov programovacích jazykov.

V dnešnej informaticky zameranej dobe predstavujú formálne jazyky veľmi dôležitú oblasť. Pod formálnymi jazykmi chápeme matematické štruktúry vznikajúce formalizáciou a zovšeobecňovaním prirodzených a umelých jazykov, hlavne jazykov programovacích.

2.1 Abeceda

V bežnom živote slovo abeceda znamená určitú sadu znakov (písmen), ktorými sa graficky vyjadrujú hlásky. Z hlások sa skladajú slová, slúžiace ako dorozumievací prostriedok ľudí.

Táto práca sa venuje pojmu abeceda a ďalším spolu súvisiacim pojmom z matematického hľadiska. A práve z tohto hľadiska sa abecedou rozumie ľubovoľná neprázdna konečná množina, ktorej prvky sa nazývajú znaky, písmená alebo symboly abecedy. V literatúre sa abeceda označuje gréckym písmenom Σ. Príkladom abecedy môže byť binárna abeceda {0,1}, množina číslic {0,1,…,9}, hexadecimálna abeceda {0,1,...,9,A,...,F}, atď. [2]

2.2 Slovo

Slovo (taktiež reťazec) v nad abecedou Σ je možné získať vytvorením konečnej dĺžky symbolov, ktoré túto abecedu tvoria (napr. aaba je slovo nad abecedou {a,b}).

Počet znakov, ktoré slovo tvoria označujeme | v | a nazývame ho dĺžkou slova. Počet výskytov symbolu a v slove v sa označuje $| v |_a$ ($\#_a$ (v)). Ak teda v = aabbaa, potom $| v |_a = 4$.

Prázdnou postupnosťou znakov (čiže prázdnym slovom) rozumieme slovo ktorého dĺžka | v | = 0. Takéto slovo sa označuje λ.

Množinu všetkých slov nad abecedou Σ označujeme Σ^* a množinu všetkých neprázdnych slov Σ^+.

Slovo u je podslovom slova v ak existujú slová x,y také, že v = xuy. Pokiaľ x = λ, tak potom slovo u je prefixom, čiže predponou slova v. Ak y = λ, vravíme, že slovo u je sufixom, čiže príponou slova v. [2]

2.2.1 Operácie nad slovami

Častou operáciou so slovami je ich zreťazenie, tzv. spojenie do jedného slova. Zreťazenie slova $u = a_1a_2...a_n$ so slovom $v = b_1b_2...b_n$ sa označuje u.v alebo častejšie taktiež uv a $uv = a_1a_2...a_n\ b_1b_2...b_n$. Operácia zreťazenie je asociatívna, čiže u.(v.w) = (u.v).w pre ľubovoľné slová u,v,w. Pri zreťazení sa prázdne slovo λ chová ako jednotkový prvok, čiže u. λ = λ.u = u.

Ďalším spôsobom zreťazenia je takzvané i- násobné zreťazenie slova v, označované ako v^i. Pre i = 0 sa $v^0 = \lambda$, pre i = 1 sa $v^1 = v$, pre i = 2 sa $v^2 = vv$ atď. Všeobecne teda platí, že $v^i = vv^{i-1}$.

Pri zreťazení ma exponent vždy väčšiu prioritu než samotne zreťazenie.

2.3 Jazyk

Formálny jazyk nad abecedou Σ je ľubovoľná množina slov nad touto abecedou, čiže podmnožina Σ^*. Napríklad {a, ab, aab, b, bbb} je jazyk nad abecedou {a, b}, prázdna množina λ je jazykom nad ľubovoľnou abecedou Σ. Jazyk je obvykle označovaný písmenom L.

Na rozdiel od abecedy môže byť jazyk ako konečný, tak aj nekonečný. Konečným jazykom rozumieme jazyk, ktorý obsahuje konečný počet slov. Naproti tomu nekonečný jazyk, je jazyk, ktorý obsahuje nekonečné množstvo slov. Z toho plynie, že konečné jazyky je možné zadať výpisom prvkov, ale nekonečné iba s pomocou nejakej obmedzujúcej podmienky, ktorá charakterizuje slová obsiahnuté v takomto jazyku.

2.3.1 Operácie nad jazykmi

Každý formálny jazyk je množina slov nad určitou abecedou a preto je možné s jazykmi prevádzať štandardné množinové operácie ako prienik, zjednotenie, rozdiel alebo doplnok. Nad formálnymi jazykmi sa ale zavádzajú aj ďalšie špeciálne operácie ako napríklad zreťazenie, iterácia alebo zrkadlový obraz jazyka.

- *Štandardné množinové operácie*

  Ak L_1 je ľubovoľný jazyk nad abecedou Σ_1 a L_2 je ľubovoľný jazyk nad abecedou Σ_2, môžeme na ne aplikovať štandardné množinové operácie zjednotenia, prieniku a rozdielu. Výsledkom je vždy jazyk nad abecedou $\Sigma_1 \cup \Sigma_2$. Doplnkom jazyka L_1 je jazyk co-$L_1 = \Sigma_1^* - L_1$.
- *Zreťazenie jazykov*

  Zreťazením jazykov L_1 a , L_2 kde L_1 je jazyk nad abecedou Σ_1 a L_2 jazyk nad abecedou Σ_2, je jazyk $L_1 . L_2 = \{uv \mid u \in L_1, v \in L_2\}$ nad abecedou $\Sigma_1 \cup \Sigma_2$. Operácia zreťazenie je asociatívna.
- *I- tá mocnina jazyka*

  I- tá mocnina je definovaná pre každé $i \in N_0$. L^i označuje i- tu mocninu jazyka L a platí : $L^0 = \{\lambda\}$, $L^{i+1} = L.L^i$.
- *Iterácia jazyka*

  Iteráciou jazyka L je jazyk $L^* = \bigcup_{i=0}^{\infty} L^i$. Jedná sa o zjednotenie všetkých mocnín daného jazyka.
- *Pozitívna iterácia jazyka*

  Pozitívna iterácia jazyka L je jazyk $L^+ = \bigcup_{i=1}^{\infty} L^i$. V tomto prípade sa jedná o zjednotenie všetkých mocnín daného jazyka okrem prázdneho slova λ, čiže 0- tej mocniny.
- *Zrkadlový obraz jazyka*

Zrkadlovým obrazom jazyka L je jazyk $L^R = \{u^R \mid u \in L\}$, kde u^R je zrkadlový obraz slova u. Napríklad u = aaba, potom zrkadlovým obrazom slova u je u^R = abaa.

2.4 Gramatika

Formálna gramatika umožňuje vygenerovanie všetkých slov patriacich do daného jazyka. Preto sú gramatiky považované za generatívne systémy. Gramatika je zložená z množiny prepisovacích pravidiel P, pomocou ktorých môže byť každé slovo vygenerované z dopredu určeného počiatočného symbolu S. Generácia požadovaného slova prebieha tak, že sa začne s počiatočným symbolom S, na ktorý sa aplikuje príslušne prepisovacie pravidlo P. Na takto získaný reťazec sa opäť aplikuje vhodné prepisovacie pravidlo P a to až do doby, kým sa nevygeneruje požadované slovo.

Formálna gramatika je štvorica G = {Σ , T, S, P}, kde

- Σ – je abeceda
- T- je konečná množina terminálnych symbolov (terminálov) taká, že $N \cap \Sigma = \emptyset$. Zjednotením N a Σ dostaneme množinu všetkých symbolov gramatiky, tzv. celková abeceda, ktorú obvykle označujeme V.
- S- je počiatočný symbol (neterminál). $S \in N$. Nazýva sa taktiež koreňom gramatiky.
- P- je konečná množina prepisovacích pravidiel. $P \subseteq V^*NV^* \times V^*$. Pravidlo (α, β) sa obvykle zapisuje v tvare $\alpha \rightarrow \beta$. Pre každé pravidlo $\alpha \rightarrow \beta$ z P platí, že α obsahuje aspoň jeden neterminál.

Jazyk $L(G) = \{w \in T^*; S \Rightarrow^* w\}$ je jazyk gramatiky G, čiže jazyk vygenerovaný gramatikou G. [7]

2.4.1 Chomského hierarchia gramatík a jazykov

Na základe rôznych obmedzení na tvar pravidiel rozdelil Noam Chomsky gramatiky do štyroch skupín (typov).

Chomského hierarchia teda rozlišuje tieto štyri základné typy gramatík $G = \{\Sigma, T, S, P\}$:

- *Typ 0*

  Ide o ľubovoľnú gramatiku. Na tvar pravidiel sa nekladú žiadne obmedzujúce požiadavky. Tieto gramatiky sú niekedy označované ako frázové gramatiky.

- *Typ 1*

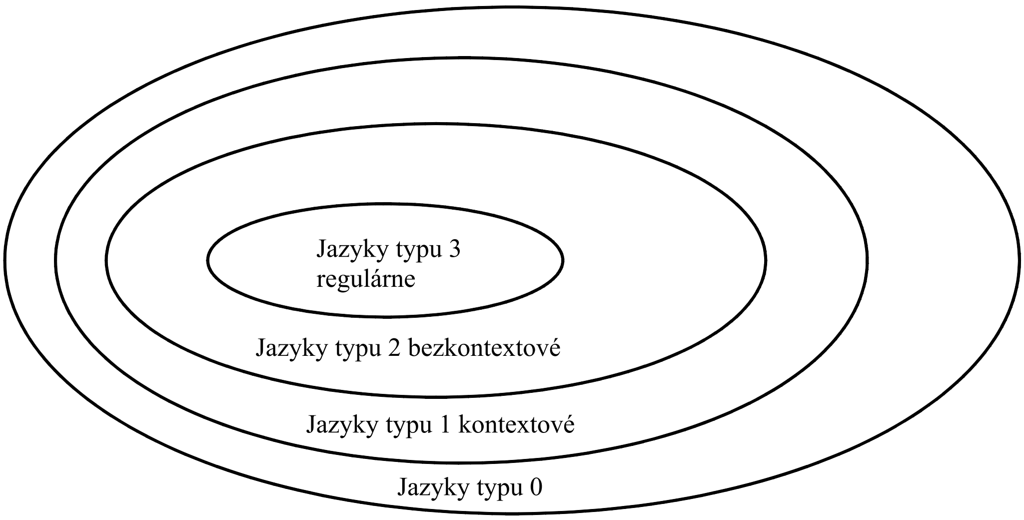
  Kontextová gramatika - pre každé prepisovacie pravidlo z P: $\alpha \rightarrow \beta$ platí $|\alpha| \leq |\beta|$, s výnimkou tvaru pravidla $S \rightarrow \lambda$, pokiaľ sa S nevyskytuje na pravej strane žiadneho pravidla. Gramatiku typu jedna nazývame taktiež neskracujúcou, pretože ňou vygenerované slová sa buď predlžujú alebo zostávajú rovnako dlhé, nikdy sa však neskracujú.

- *Typ 2*

  Bezkontextová gramatika - každé jej pravidlo je tvaru $A \rightarrow \alpha$, kde $|\alpha| \geq 1$ s výnimkou pravidla $S \rightarrow \lambda$, pokiaľ sa S nevyskytuje na pravej strane žiadneho pravidla.

- *Typ 3*

  Regulárna gramatika - každé jej pravidlo je tvaru $A \rightarrow aB$ alebo $A \rightarrow a$ s výnimkou $S \rightarrow \lambda$, pokiaľ sa S nevyskytuje na pravej strane žiadneho pravidla.



Obrázok 1: Chomského hierarchia [7]

Hierarchičnosť gramatík spočíva v tom, že každá regulárna gramatika je bezkontextová, každá bezkontextová gramatika je kontextová a každá kontextová gramatika je typu 0. Opačne to samozrejme neplatí.

Hierarchia gramatík určuje aj príslušnú hierarchiu jazykov. Jazyk L je regulárny (prípadne bezkontextový, kontextový, typu 0), pokiaľ existuje regulárna (prípadne bezkontextová, kontextová, typu 0) gramatika G taká, že L(G) = L. [7]

2.5 Automat

Konečný automat je teoretický výpočtový model, s ktorým sa už každý z nás určite stretol, akurát si to neuvedomil. V skutočnosti je to model používaný pri riadení činností napríklad automatu na kávu, bankomatu, automatických otváraní dverí alebo svetelnej križovatky.

2.6 Konečný automat

Pod pojmom konečný automat (KA) chápeme model, ktorý je možné využiť pre modelovanie systémov s konečným počtom stavov a vstupných podnetov. Je možné ho rozdeliť na deterministický a nedeterministický.

Konečný automat je pätica A = (S, Σ, f, s_0, F), kde

- S- je konečná neprázdna množina stavov,
- Σ- je abeceda vstupných symbolov (vstupov),
- f- je parciálna prechodová funkcia, f: $S \times \Sigma \rightarrow S$,
- s_0- je počiatočný stav , $s_0 \in S$,
- F- je množina koncových stavov, $F \subseteq S$. [7]

2.7 Rozšírená prechodová funkcia

Majme zápis prechodovej funkcie $f(s_i, a) = s_j$. Tento zápis vyjadruje, že automat zo stavu s_i po prečítaní symbolu a prejde do stavu s_j. Vystihuje ho zobrazenie prechodovej funkcie zápisom $f: S \times \Sigma \rightarrow S$.

Pre umožnenie definovania jazyka prijatého konečným automatom, je potrebné zaviesť rozšírenú prechodovú funkciu $f^*: S \times \Sigma^* \rightarrow S$. Táto funkcia je definovaná vzhľadom k dĺžke slova z Σ^* takto:

- $f^*(s, \lambda) = s$ pre všetky slová $s \in S$,
- $f^*(s, va) = f(f^*(s, v),a)$, ak je $f^*(s, v)$ a $f(f^*(s, v),a)$ definované,

  $\perp$ inak.

Symbol $\perp$ značí, že funkcia nie je definovaná. [2]

2.8 Jazyk prijatý automatom

Aby bol jazyk prijatý (akceptovaný) konečným automatom A, musí byť tvorený takými slovami, pod ktorými automat prejde z počiatočného stavu s_0 do niektorého z koncových stavov z F.

$L(A) = \{v \in \Sigma^* \mid f^*(s_0, v) \in F\}$, jazyk ktorý je takto akceptovaný nejakým konečným automatom sa nazýva regulárny. [2]

2.9 Stavy dosiahnuteľné automatom

V konečnom automate sa nachádzajú stavy, ktoré sú dosiahnuteľné alebo nedosiahnuteľné. Nedosiahnuteľné stavy sú pre automat zbytočné a je vhodné ich eliminovať.

Stavy dosiahnuteľné automatom sú také, ktoré spĺňajú: Nech $A = (S, \Sigma, f, s_0, F)$ je konečný automat. Stav $s \in S$ je dosiahnuteľný, ak existuje slovo $v \in \Sigma^*$ také, že $f^*(s_0, v) = s$.

Stav automatu je nedosiahnuteľný, pokiaľ nie je dosiahnuteľný. [2]

2.10 Nadbytočné stavy

Popri stavoch nedosiahnuteľných konečným automatom existujú aj stavy nadbytočné. Sú to stavy, z ktorých sa žiadnym spôsobom nedá dostať do žiadneho z koncových stavov automatu.

Nech A = (S, Σ, f, s_0, F) je konečný automat. Stav s ∈ S je nadbytočný, ak z tohto stavu neexistuje žiadna prechodová funkcia $f^*(s_0, v')$ = s_f, kde v′ je doteraz neprečítaná časť vstupného slova a s_f ≡ F.

2.11 Reprezentácia konečných automatov

Konečný automat A = (S, Σ, f, s_0, F) je možné reprezentovať viacerými spôsobmi. Najpoužívanejšie sú vymenovanie prvkov a pravidiel prechodovej funkcie, tabuľka prechodov a stavový diagram.

- *Reprezentácia vymenovaním prvkov a pravidiel prechodovej funkcie*

Je daný konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, {s_2, s_3 }), kde pre f platí:

$f(s_0, a) = s_2$, $f(s_2, a) = s_1$,
$f(s_0, b) = s_1$, $f(s_2, b) = s_3$,
$f(s_1, a) = s_0$, $f(s_3, a) = s_3$,
$f(s_1, b) = s_1$, $f(s_3, b) = s_1$.

- *Reprezentácia tabuľkou prechodov*

Prechodovú funkciu konečného automatu je možné jednoduchým spôsobom prezentovať tabuľkou prechodov. Záhlavie stĺpcov tvoria symboly vstupnej abecedy, záhlavie riadkov jednotlivé stavy automatu. Počiatočné stavy sú označované →, koncové stavy ← a stavy, ktoré sú súčasne počiatočné aj koncové symbolom ↔.

Vnútorné pole tabuľky na riadku s_i a stĺpci a vyjadruje výsledok prechodovej funkcie $f(s_i, a)$. Pokiaľ pre danú dvojicu nie je definovaná, uvádza sa znak - .

Je daný konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, {s_2, s_3 }), kde pre f platí:

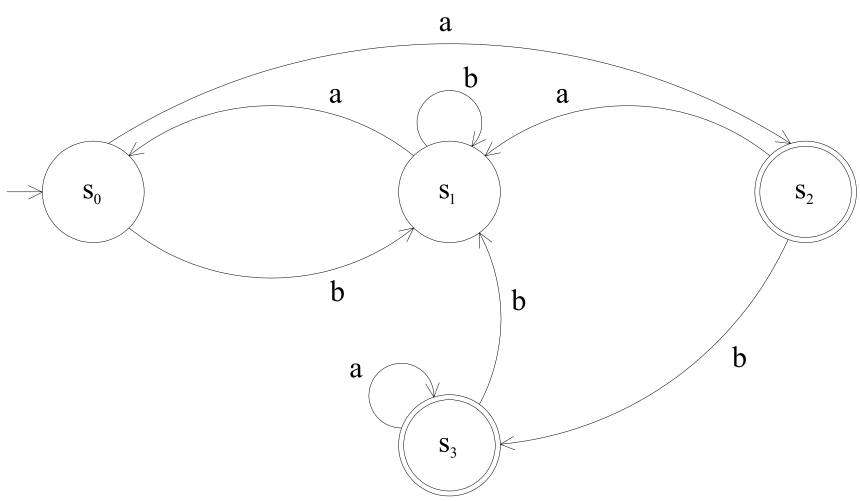
Tabuľka 1: Reprezentácia konečného automatu tabuľkou prechodov

	a	**b**
→ s_0	s_2	s_1
s_1	s_0	s_1
← s_2	s_1	s_3
← s_3	s_3	s_1

- *Reprezentácia stavovým diagramom*

Táto grafická reprezentácia pomocou stavového diagramu je najpoužívanejšia. Diagram je tvorený množinou uzlov, z ktorej každý predstavuje určitý stav automatu a orientovanými hranami medzi týmito uzlami. Hrany sú ohodnotené symbolmi vstupnej abecedy. Počiatočný stav je označený →, koncový stav dvojitým kruhom.

Je daný konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, {s_2, s_3 }), kde pre f platí:



Obrázok 2: Reprezentácia konečného automatu stavovým diagramom

2.12 Totálna prechodová funkcia

Ku každému automatu A existuje ekvivalentný automat A´ s totálnou prechodovou funkciou. Znamená to, že automat z každého stavu prejde do iného stavu pod všetkými vstupnými symbolmi.

Totálna prechodová funkcia je definovaná: Máme automat $A = (S, \Sigma, f, s_0, F)$. Automat A´ zostrojíme tak, že k automatu A pridáme jeden nový nekoncový stav q, do ktorého nasmerujeme všetky chýbajúce prechody. K stavu q taktiež pridáme prechod, ktorý v ňom začína aj končí, takzvanú slučku, pre všetky vstupné symboly.

Automat $A' = (S \cup \{q\}, \Sigma, f', s_0, F)$, kde $q \notin S$ a f' je definované:

$$f'(s, a) = \begin{cases} f(s, a), \text{ ak je } f(s, a) \text{ definované} \\ q, \text{ inak.} \end{cases}$$ [2]

2.13 Deterministický konečný automat

Deterministickým automatom sa chápe taký konečný automat, u ktorého je prechod zo súčasného stavu do nového stavu po príchode vstupného symbolu určený jednoznačne. Čo znamená, že automat pod jedným vstupným symbolom prejde zo súčasného stavu iba do jedného nového stavu.

2.14 Nedeterministický konečný automat

Nedeterministický konečný automat sa od deterministického konečného automatu líši tým, že nemusí mať jednoznačne daný prechod z aktuálneho stavu do nového stavu po príchode vstupného symbolu. V tomto prípade je vymedzené množina stavov, do ktorých môže automat prejsť. Automat si teda v priebehu svojho výpočtu môže vyberať, do ktorého stavu z množiny stavov po príchode vstupného symbolu prejde. Ďalšia odlišnosť je v tom, že nedeterministický konečný automat má vymedzenú aj množinu počiatočných stavov, na rozdiel od deterministického konečného automatu, ktorý má počiatočný stav iba jeden.

2.15 Pumping lemma

Ak chceme zistiť či je daný jazyk regulárny, môžeme sa k nemu pokúsiť skonštruovať príslušný automat. Ak sa to podarí, znamená to, že daný jazyk skutočne regulárny je. Ak sa to však nepodarí, môže to znamenať že daný jazyk regulárny nie je, že automat neexistuje. Dokázať to môžeme pomocou vety nazývanej pumping lemma, ktorá je nutnou podmienkou pre regularitu jazyka.

Pumping lemma však nie je dostačujúcou podmienkou pre dôkaz regularity jazyka. Pomocou nej môžeme dokázať, že jazyk regulárny nie je, nie však, že regulárny je. Aby sme skutočne dokázali, že daný jazyk je regulárny, musíme použiť Myhill- Nerodovu vetu.

Dôkaz pomocou pumping lemmy: Pokiaľ je jazyk regulárny, tak preň platí:

- $\exists \ n \in N$
- $\forall \ v \in L, |v| \geq n$
- $\exists \ xyz$, kde $|xy| \leq n, |y| \geq 1$
- $\forall \ i \in N_0 : v' = xy^i z \in L$

Číslo n sa neformálne nazýva pumpovacia konštanta.

Pre dôkaz pomocou pumping lemmy sa využíva negácia regularity jazyka:

- $\forall \ n \in N$
- $\exists \ v \in L, |v| \geq n$
- $\forall \ xyz$, kde $|xy| \leq n, |y| \geq 1$
- $\exists \ i \in N_0 : v' = xy^i z \notin L$

Po dokázaní vyššie uvedených podmienok je možné s istotou tvrdiť, že jazyk nie je regulárny. Nie je však možné dokázať, že jazyk regulárny je. [2]

2.16 Využitie v ekonomickej a informatickej praxi

Jazyky, automaty a gramatiky sú využívané i v ekonomickej praxi a to prakticky i v každom systéme z dôvodu samotnej logičnosti návrhu. Nielen ako vlastná podstata funkčnosti použitého jazyka, ale aj ako nutný teoretický podklad samotného fungovania vzťahov a súvislostí jednotlivých ekonomických vlastností systému. Väzby ekonomických pravidiel musia byť jednoznačne dané v priamych súvislostiach, a musí byť možné formálne ich zapísať. Iba takto definovaný systém je možné vytvoriť a aplikovať. Je teda nutné už pri návrhu ekonomického či iného systému definovať väzby jednotlivých sekcií a položiek v matematicky definovateľných súvislostiach. V konečnom dôsledku sa tak buduje potencionálne ľahko rozšíriteľný matematický základ pre budovanie systémových nadstavieb a vytvorenie všeobecných šablón pre ekonomické účely.

V informatickej praxi je bežné používať a formalizovať návrhy pomocou formálnych jazykov. Jazyky, automaty a gramatiky sú samotnou podstatou zdrojového kódu. Nielen programovacie jazyky a z nich vychádzajúce informačné systémy sú postavene na tejto teórii, ale aj samotné prekladače jazykov. Využívajú sa na spracovanie prirodzeného jazyka a návrh a popis hardvéru napríklad v strojoch a automatoch.

3 Zbierka príkladov s vlastným návrhom riešenia

3.1 Slovo

Príklad číslo 1

Je daná abeceda $\Sigma = \{0, 1\}$. Vypíšte všetky slová, ktoré majú dĺžku 3.

Riešenie:

Počet týchto slov je $2^3 = 8$ a sú to : {000, 001, 010, 100, 111, 110, 101, 011}.

Príklad číslo 2

Je daná abeceda $\Sigma = \{x, y\}$. Nad touto abecedou určite dvojice slov, ktoré spĺňajú nasledujúce podmienky:

- $|u| = |v| = 3$
- $|u|_x = |v|_y$

Riešenie:

Ide o príklad v ktorom je potrebné vypísať dvojice slov nad abecedou $\Sigma = \{x, y\}$, ktoré spĺňajú podmienky, že slová u a v majú dĺžku 3 a počet výskytov symbolu x v slove u sa musí rovnať počtu výskytov symbolu y v slove v.

Medzi dvojice, ktoré spĺňajú zadané podmienky patria:

u_1 = xxx	v_1 = yyy	u_{11} = xyy	v_{11} = yxx
u_2 = xxy	v_2 = yyx	u_{12} = xyy	v_{12} = xyx
u_3 = xxy	v_3 = yxy	u_{13} = xyy	v_{13} = xxy
u_4 = xxy	v_4 = xyy	u_{14} = yxy	v_{14} = yxx
u_5 = xyx	v_5 = yyx	u_{15} = yxy	v_{15} = xyx
u_6 = xyx	v_6 = yxy	u_{16} = yxy	v_{16} = xxy
u_7 = xyx	v_7 = xyy	u_{17} = yyx	v_{17} = yxx
u_8 = yxx	v_8 = yyx	u_{18} = yyx	v_{18} = xyx
u_9 = yxx	v_9 = yxy	u_{19} = yyx	v_{19} = xxy
u_{10} = yxx	v_{10} = xyy		

Príklad číslo 3

Slovo u = abab. Určite :

a) všetky prefixy slova u

b) všetky sufixy slova u

c) všetky podslová slova u, ktoré obsahujú rovnaký počet znakov a, b.

Riešenie:

a) Prefixov slova u je dohromady 5 a sú to: λ, a, ab, aba, abab

b) Sufixov slova u je dohromady taktiež 5 a sú to: λ, b, ab, aba, abab

c) Podslová slova u, ktoré obsahujú rovnaký počet symbolov a,b: λ, ab, ba, abab

3.2 Operácie nad slovami

Príklad číslo 4

Nad abecedou Σ = {0, 1} určite všetky slová u, ktoré spĺňajú nasledujúce podmienky:

- $| u | = 4$
- $u^R = u$

Riešenie:

Podmienky pre dĺžku slova u = 4 a zrkadlový obraz slova u spĺňajú tieto 4 slová:

$u_1 = 0000$ $u_3 = 0110$

$u_2 = 1111$ $u_4 = 1001$

Príklad číslo 5

Sú zadané slová u = aab, v = abba, w = abab, x = 0100, y = 0011, z = 111. Určite:

a) u.v.z

b) z.v.x

c) w.z.u.x

d) v.y.w

e) z.y.x.w.v.u

f) λ.u.v.w

Riešenie:

Vo všetkých prípadoch a – f ide o zreťazenie slov. Riešením teda je:

a) u.v.z = aababba111

b) z.v.x = 111abba0100

c) w.z.u.x = abab111aab0100

d) v.y.w = abba0011abab

e) z.y.x.w.v.u = 11100110100abababbaaab

g) Prázdne slovo sa pri zreťazení chová ako jednotkový prvok, takže:

λ.u.v.w = aababbaabab

Príklad číslo 6

Explicitne rozpíšte postupnosť znakov ktorá vznikne zreťazením:

a) slova $u = 0^2(10)^2$ so slovom $v = 1^4$

b) slova $u = aba^3$ so slovom $v = c(ab)^2$

c) slova $u = ab(aba)^3$ so slovom $v = aab^2$

d) slova $u = 11(01)^2$ so slovom $v = \lambda$

Riešenie:

Pri zreťazení má exponent vždy väčšiu prioritu ako samotné zreťazenie. Riešením je:

a) u.v = 0010101111

b) u.v = abaaacabab

c) u.v = ababaabaabaaabb

d) u.v = 110101

3.3 Jazyk

Príklad číslo 7

Slovne popíšte jazyky :

a) $L_1 = \{a, b, c\}^*$

b) $L_2 = \{abc\}^*$

c) $L_3 = \{a, b, c\}^+$

d) $L_4 = \{a\}^*.\{b\}^3.\{c\}^*$

e) $L_5 = \{a\}^*.\{b\}^*.\{c\}^*$

f) $L_6 = \{a, b, c\}^*.\{a\}.\{a, b, c\}^*$

g) Rozhodnite či $L_1 = L_5$

Riešenie:

a) Jazyk L_1 obsahuje všetky slová nad abecedou $\Sigma = \{a, b, c\}$. Je to jazyk obsahujúci všetky možné slova, ktoré vzniknú kombináciou symbolov a, b, c. Tento jazyk obsahuje aj prázdne slovo λ, pretože množina symbolov $\{a, b, c\}$ je na *.

b) Jazyk L_2 obsahuje všetky slová, ktoré sú tvorene mocninami slova abc. Čiže $abc^0 = \lambda$, $abc^1 = abc$, $abc^2 = abcabc$ atď.

c) Jazyk L_3 obsahuje všetky slová vznikajúce kombináciou symbolov a, b, c. Neobsahuje však prázdne slovo λ, pretože množina symbolov $\{a, b, c\}$ je na +.

d) Jazyk L_4 obsahuje všetky slová tvoriace kombináciu slov $\{a\}^*.bbb.\{c\}^*$. Najkratšie slovo tohto jazyka je slovo bbb, ktoré vznikne v prípade jeho zreťazenia s $a^0bbbc^0 = \lambda bbb\lambda = bbb$. Ďalšie slová sú tvorené kombináciou rôzneho počtu symbolov a zreťazených so slovom bbb zreťazených s rôznym počtom symbolov b. Formálne sa tento jazyk dá zapísať ako $\{xyz \mid x = a^i, y = b^3, z = c^j; i, j \in N_0\}$

e) Jazyk L_5 obsahuje všetky slová tvorené rôznym počtom symbolov a reťazených s rôznym počtom symbolov b zreťazených s rôznym počtom symbolov c. Do tohto jazyka patrí aj prázdne slovo λ. Formálne sa tento jazyk dá zapísať ako $\{xyz \mid x = a^i, y = b^j, z = c^k; i, j, k \in N_0\}$

f) Jazyk L_6 obsahuje všetky slová tvorené rôznou kombináciou symbolov z $\Sigma = \{a, b, c\}$ zreťazených so symbolom a zreťazených s rôznou kombináciou symbolov z $\Sigma = \{a, b, c\}$. Najkratším slovom tohto jazyka je slovo a.

g) Jazyky L_1 a L_5 sa nerovnajú. Jazyk L_1 obsahuje všetky slová nad abecedou $\Sigma = \{a, b, c\}$ a jazyk L_5 obsahuje všetky slová tvorené rôznym počtom symbolov a reťazených s rôznym počtom symbolov b zreťazených s rôznym počtom symbolov c. Jazyk L_5 je obsiahnutý v jazyku L_1.

Príklad číslo 8

Určite jazyk L nad abecedou $\Sigma = \{0, 1\}$ taký, že slová tohto jazyka obsahujú rovnaký počet symbolov 0 i 1.

Riešenie:

Riešením je formálny zápis jazyka $L = \{u \in \{0, 1\}^* \mid |u|_0 = |u|_1\}$.

3.4 Operácie nad jazykmi

Príklad číslo 9

Sú zadané jazyky $L_1 = \{01, 1, 010\}$, $L_2 = \{1, 100, 010\}$, $L_3 = \{yx, xx, x\}$, $L_4 = \{y, yx, yy\}$, $L_5 = \{yy, yxy, y\}$. Určite:

a) $L_1 . L_2$

b) $L_2 . L_1$

c) $L_3 . L_1$

d) $L_4 . L_3 . L_5$

e) $\lambda . L_1$

Riešenie:

a) Zreťazenie jazyka L_1 s jazykom L_2 sa rieši nasledujúcim spôsobom. Vezmeme prvé slovo z jazyka L_1 a zreťazíme ho s každým slovom z jazyka L_2. Týmto spôsobom vzniknú slová 011, 01100, 01010. Ďalej vezmeme druhé slovo z jazyka L_1 a zreťazíme ho s každým slovom z jazyka L_2 a týmto spôsobom pokračujeme až kým nevyčerpáme všetky slová z jazyka L_1. $L_1 . L_2 = \{011, 01100, 01010, 11, 1100, 1010, 0101, 010100, 010010\}$

b) $L_2 . L_1 = \{101, 11, 1010, 10001, 1001, 100010, 01001, 0101, 010010\}$

c) $L_3 . L_1 = \{yx01, yx1, yx010, xx01, xx1, xx010, x01, x1, x010\}$

d) $L_4 . L_3 . L_5 = \{yyxyy, yyxyxy, yyxy, yxxyy, yxxyxy, yxxy, yxyy, yxyxy, yxy, yxyxyy, yxyxyxy, yxyxy, yxxxyy, yxxxyxy, yxxxy, yxxyy, yxxyxy, yxxy, yyyxyy, yyyxyxy, yyyxy, yyxxyy, yyxxyxy, yyxxy, yyxyy, yyxyxy, yyxy\}$

e) Zreťazením jazyka L_1 s prázdnym slovom λ dostaneme opäť jazyk L_1.

$\lambda \, . \, L_1 = \{01, 1, 010\}$

Príklad číslo 10

Je daný jazyk $L_1 = \{11, 10, 01, 00\}$ a jazyk $L_2 = \{11, 111, 010, 01\}$ nad abecedou $\Sigma = \{0, 1\}$. Určite:

a) $L_1 \cup L_2$

b) $L_1 \cap L_2$

c) co-L_1

Riešenie:

a) Pri zjednotení jazykov sa postupuje rovnakým spôsobom ako pri zjednotení množín. Preto riešením zjednotenia jazyku L_1 s jazykom L_2 bude:

$L_1 \cup L_2 = \{11, 10, 01, 00, 111, 010\}$

b) Pri prieniku jazykov sa taktiež postupuje rovnakým spôsobom ako pri prieniku množín. Preto riešením bude:

$L_1 \cap L_2 = \{11, 01\}$

c) Doplnkom jazyka L_1 bude jazyk, ktorý obsahuje všetky slová nad abecedou $\Sigma = \{0, 1\}$, okrem slov, ktoré obsahuje samotný jazyk L_1.

co-$L_1 = \{1, 0, 111, 110, 101, 011, 100, 010, 001, ...\}$

Príklad číslo 11

Je daný jazyk $L = \{1, 0, 11, 00\}$ nad abecedou $\Sigma = \{0, 1\}$. Určite:

a) L^*

b) L^+

Riešenie:

a) Jazyk L^* znamená, že ide o iteráciu jazyka L. Pod pojmom iterácia jazyka L rozumieme zjednotenie všetkých mocnín jazyka L.

$L^* = \{\lambda , 1, 0, 11, 00, 1111, 0000, 111, 000, 111111, 000000, \ldots\}$

b) Jazyk L^+ znamená, že ide o pozitívnu iteráciu jazyka L. Pozitívnou iteráciou jazyka L chápeme zjednotenie všetkých mocnín jazyka L okrem nultej mocniny, čiže prázdneho slova λ.

Príklad číslo 12

Určite zrkadlový obraz daných jazykov:

a) $L_1 = \{\lambda , 1, 01, 001, 0100\}$

b) $L_2 = \{10, 011, 1101, 1110\}$

c) $L_3 = \{001, 0011, 01001\}$

Riešenie:

a) Zrkadlový obraz jazyka L_1 je jazyk $L_1^R = \{\lambda , 1, 10, 100, 0010\}$

b) Zrkadlový obraz jazyka L_2 je jazyk $L_2^R = \{01, 110, 1011, 0111\}$

c) Zrkadlový obraz jazyka L_3 je jazyk $L_3^R = \{100, 1100, 10010\}$

3.5 Gramatika

Príklad číslo 13

Je zadaná gramatika $G = (\{S, A, B\}, \{a, b\}, S, P)$, kde

$P = \{S \rightarrow Aa \mid B,$

$A \rightarrow Aa \mid a,$

$B \rightarrow Bb \mid b\}$

Určite či sa zo slova AaBb dá odvodiť slovo:

a) aaabbb

b) aabb

c) aaabb

Riešenie:

a) Slovo aaabbb je možné odvodiť nasledujúcim spôsobom pomocou relácií priameho odvodenia:

- na začiatku využijeme pravidlo $A \rightarrow Aa$ a slovo upravíme na: $AaBb \Rightarrow_G AaaBb$.
- pomocou toho istého pravidla dostaneme slovo $AaBb \Rightarrow_G AaaBb \Rightarrow_G aaaBb$.
- potom pomocou pravidla $B \rightarrow Bb$ dostaneme slovo $AaBb \Rightarrow_G AaaBb \Rightarrow_G aaaBb \Rightarrow_G aaaBbb$

- využitím pravidla $B \rightarrow b$ odvodíme $AaBb \Rightarrow_G AaaBb \Rightarrow_G aaaBb \Rightarrow_G aaaBbb \Rightarrow_G aaabbb$

b) Slovo aabb je možné odvodiť nasledujúcim spôsobom pomocou relácií priameho odvodenia:

- ako prvé využijeme pravidlo $A \rightarrow a$ a slovo upravíme na: $AaBb \Rightarrow_G aaBb$.
- pomocou pravidla $B \rightarrow b$ dostaneme $AaBb \Rightarrow_G aaBb \Rightarrow_G aabb$

c) Slovo aaabb je možné odvodiť nasledujúcim spôsobom pomocou relácií priameho odvodenia:

- na začiatku využijeme pravidlo $A \rightarrow Aa$ a slovo upravíme na $AaBb \Rightarrow_G AaaBb$
- pomocou toho istého pravidla upravíme slovo do podoby $AaBb \Rightarrow_G AaaBb \Rightarrow_G aaaBb$
- ako posledné využijeme pravidlo $B \rightarrow b$ a dostaneme $AaBb \Rightarrow_G AaaBb \Rightarrow_G aaaBb \Rightarrow_G aaabb$

Príklad číslo 14

Je zadaná gramatika $G = (\{S, A, B\}, \{0, 1\}, S, P)$, kde
$P = \{S \rightarrow A0 \mid B1 \mid 0 \mid 1,$

$A \rightarrow A0 \mid 1,$

$B \rightarrow B1 \mid 0\}$

Určite, čí sú nasledujúce slová akceptované gramatikou G:

a) 0

b) 1000

c) 01111

d) 101

Riešenie:

a) Pomocou relácií priameho odvodenia zistíme, či je slovo 0 generované gramatikou G:

- využitím pravidla $S \rightarrow 0$ získame slovo $S \Rightarrow_G 0$ a zistili sme, že slovo 0 je generované gramatikou G.

b) Pomocou relácií priameho odvodenia zistíme, či je slovo 1000 generované gramatikou G:

- využitím pravidla $S \rightarrow A0$ sme dostali slovo $S \Rightarrow_G A0$
- ďalej sme aplikovali pravidlo $A \rightarrow A0$ a získali sme $S \Rightarrow_G A0 \Rightarrow_G A00$
- pomocou toho istého pravidla dostaneme $S \Rightarrow_G A0 \Rightarrow_G A00 \Rightarrow_G A000$
- nakoniec využitím pravidla $A \rightarrow 1$ dostaneme hľadané slovo $S \Rightarrow_G A0 \Rightarrow_G A00 \Rightarrow_G A000 \Rightarrow_G 1000$. Slovo 1000 je generované gramatikou G.

c) Pomocou relácií priameho odvodenia zistíme, či je slovo 01111 generované gramatikou G:

- ako prvé aplikujeme pravidlo $S \rightarrow B1$ a dostaneme $S \Rightarrow_G B1$
- potom s trojnásobným využitím pravidla $B \rightarrow B1$ získame $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G B1111$
- nakoniec s využitím pravidla $B \rightarrow 0$ dostaneme slovo $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G B1111 \Rightarrow_G 01111$. Slovo 01111 je teda generované gramatikou G.

d) Pomocou relácií priameho odvodenia zistíme, či je slovo 101 generované gramatikou G:

1)

- využitím pravidla $S \rightarrow B1$ dostaneme $S \Rightarrow_G B1$
- ako ďalšie využijeme pravidlo $B \rightarrow 0$ a dostaneme $S \Rightarrow_G B1 \Rightarrow_G 01$
- výsledné slovo je 01, čo nie je to, ktoré hľadáme – nesprávny spôsob

2)

- využitím pravidla $S \rightarrow B1$ dostaneme $S \Rightarrow_G B1$
- opätovným využitím pravidla $S \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11$
- je evidentné že zo slova B11 už nedostaneme slovo 101 – nesprávny spôsob

3)

- využitím pravidla $S \rightarrow A0$ dostaneme $S \Rightarrow_G A0$
- ako ďalšie využijeme pravidlo $A \rightarrow 1$ a dostaneme $S \Rightarrow_G A0 \Rightarrow_G 10$
- výsledné slovo je 10, čo nie je to, ktoré hľadáme – nesprávny spôsob

4)

- využitím pravidla $S \rightarrow A0$ dostaneme $S \Rightarrow_G A0$
- opätovným využitím pravidla $S \rightarrow A0$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A00$

- je evidentné, že zo slova A00 už žiadnym spôsobom neostaneme slovo 101 – nesprávny spôsob
- Slovo 101 nie je generované gramatikou G.

Príklad číslo 15

Určite jazyk generovaný gramatikou G = ({S}, {0, 1}, S, P), kde

P = {S → 0S1 | 01}

Riešenie:

- ako prvý krok je možné využiť dve prepisovacie pravidlá S → 01 a S → 0S1 a dostaneme dve varianty slov:

  S $\Rightarrow_G$ 01 - dostaneme slovo 01

  $\Rightarrow_G$ 0S1
- na slovo 0S1 je opäť možné aplikovať tie isté prepisovacie pravidlá:

  0S1 $\Rightarrow_G$ 0011 - dostaneme slovo 0011

  $\Rightarrow_G$ 00S11
- na slovo 00S11 je analogicky možné opäť aplikovať prepisovacie pravidlá a dostaneme dve varianty slova:

  00S11 $\Rightarrow_G$ 000111 – dostaneme slovo 000111

  $\Rightarrow_G$ 000S111
- z výpočtov je jasne vidieť, že gramatika G generuje jazyk L(G) = $\{0^n1^n \mid n \geq 1\}$, pretože vygenerované slová sú 01, 0011, 000111 ...

Príklad číslo 16

Určite jazyk generovaný gramatikou G = ({S, A, B}, {0, 1}, S, P), kde:

a) P = {S → A0 | B1,

A → A1 | 0,

B → B1 | 0}

b) P = {S → AB | AA,

A → B0 | 01,

B → 1}

c) P = {S → A | B,

$A \rightarrow A0 \mid 1,$

$B \rightarrow B1 \mid 0\}$

d) $P = \{S \rightarrow A1 \mid B1,$

$A \rightarrow A1 \mid 0,$

$B \rightarrow B1 \mid 1\}$

Riešenie:

a) Pomocou relácií priameho odvodenia určíme jazyk L(G). Odvodzovanie jazyka generovaného gramatikou G má na začiatku dve vetvy, po ktorých sa môžeme vydať. Jedna začína pravidlom $S \rightarrow A0$ a druhá pravidlom $S \rightarrow B1$.

1) Prvá vetva:

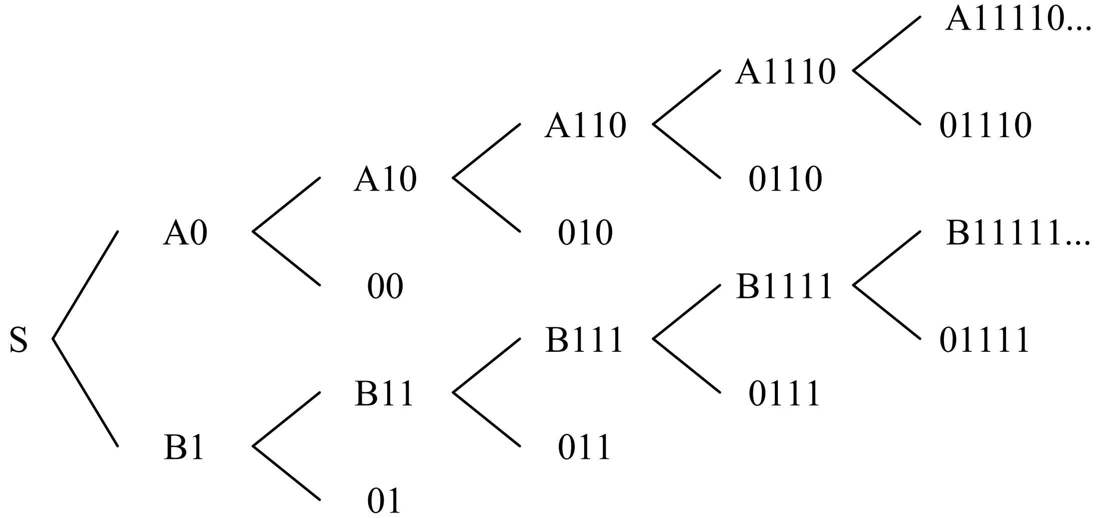
- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow A0$.
- Dostaneme slovo $S \Rightarrow_G A0$.
- Na slovo A0 je možné aplikovať dve prepisovacie pravidlá $A \rightarrow A1$ a $A \rightarrow 0$. Využitím pravidla $A \rightarrow 0$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G 00$ – odvodili sme prvé slovo 00. Využitím pravidla $A \rightarrow A1$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10$.
- Na slovo A10 je možné aplikovať tie isté prepisovacie pravidlá $A \rightarrow A1$ a $A \rightarrow 0$. Využitím pravidla $A \rightarrow 0$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10 \Rightarrow_G 010$ – odvodili sme druhé slovo 010. Využitím pravidla $A \rightarrow A1$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10 \Rightarrow_G A110$.
- Na slovo A110 je možné opäť aplikovať tie isté prepisovacie pravidlá. Využitím pravidla $A \rightarrow 0$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10 \Rightarrow_G A110 \Rightarrow_G 0110$ – odvodili sme tretie slovo 0110. Využitím pravidla $A \rightarrow A1$ dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10 \Rightarrow_G A110 \Rightarrow_G A1110$.
- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie prvej časti jazyka nám stačí využitie pravidla $A \rightarrow 0$ a dostaneme $S \Rightarrow_G A0 \Rightarrow_G A10 \Rightarrow_G A110 \Rightarrow_G A1110 \Rightarrow_G 01110$ – odvodili sme štvrté slovo 01110.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 01^n0, kde $n \geq 0$.

2) Druhá vetva:

- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow B1$.
- Dostaneme slovo $S \Rightarrow_G B1$.

- Na slovo B1 je možné aplikovať dve prepisovacie pravidlá $B \rightarrow B1$ a $B \rightarrow 0$. Využitím pravidla $B \rightarrow 0$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G 01$ – odvodili sme prvé slovo 01. Využitím pravidla $B \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11$.
- Na slovo B11 je možné aplikovať tie isté prepisovacie pravidlá $B \rightarrow B1$ a $B \rightarrow 0$. Využitím pravidla $B \rightarrow 0$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G 011$ – odvodili sme druhé slovo 011. Využitím pravidla $B \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111$.
- Na slovo B111 je možné opäť aplikovať tie isté prepisovacie pravidlá. Využitím pravidla $B \rightarrow 0$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G 0111$ – odvodili sme tretie slovo 0111. Využitím pravidla $B \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G B1111$.
- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie druhej časti jazyka nám stačí využitie pravidla $B \rightarrow 0$ a dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G B1111 \Rightarrow_G 01111$ – odvodili sme štvrté slovo 01111.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 01^m, kde $m \geq 1$.

3) Jazyk generovaný gramatikou G je $L(G) = \{01^n0, 01^m \mid n \geq 0, m \geq 1\}$.



Obrázok 3: Názorná schéma odvodenia jazyka, príklad č. 16, a)

b) Pomocou relácií priameho odvodenia určíme jazyk L(G). Odvodzovanie jazyka generovaného gramatikou G má na začiatku dve vetvy, po ktorých sa môžeme vydať. Jedna začína pravidlom $S \rightarrow AB$ a druhá pravidlom $S \rightarrow AA$.

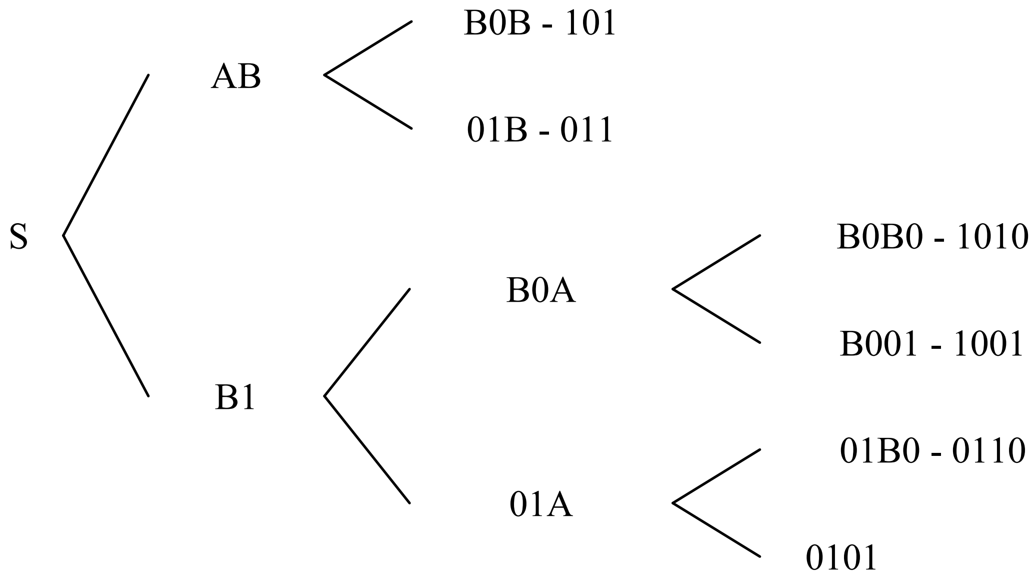
1) Prvá vetva:

- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow AA$.
- Dostaneme slovo $S \Rightarrow_G AA$.
- Na slovo AA je možné aplikovať dve prepisovacie pravidlá $A \rightarrow B0$ a $A \rightarrow 01$. Využitím pravidla $A \rightarrow B0$ dostaneme $S \Rightarrow_G AA \Rightarrow_G B0A$. Využitím pravidla $A \rightarrow 01$ dostaneme $S \Rightarrow_G AA \Rightarrow_G 01A$.
- Na slovo B0A je možné aplikovať prepisovacie pravidlá $A \rightarrow B0$ a $A \rightarrow 01$. Využitím pravidla $A \rightarrow B0$ dostaneme $S \Rightarrow_G AA \Rightarrow_G B0A \Rightarrow_G B0B0$. Využitím pravidla $A \rightarrow 01$ dostaneme $S \Rightarrow_G AA \Rightarrow_G B0A \Rightarrow_G B001$.
- Na slovo B0B0 je možná aplikovať pravidlo $B \rightarrow 1$. Po jeho aplikácii dostaneme $S \Rightarrow_G AA \Rightarrow_G B0A \Rightarrow_G B0B0 \Rightarrow_G 1010$ – odvodili sme prvé slovo jazyka 1010.
- Na slovo B001 je taktiež možné aplikovať pravidlo $B \rightarrow 1$. Dostaneme tak $S \Rightarrow_G AA \Rightarrow_G B0A \Rightarrow_G B001 \Rightarrow_G 1001$ – odvodili sme druhé slovo 1001.
- Na slovo 01A je možné aplikovať dve prepisovacie pravidlá $A \rightarrow B0$ a $A \rightarrow 01$. Využitím pravidla $A \rightarrow B0$ dostaneme $S \Rightarrow_G AA \Rightarrow_G 01A \Rightarrow_G 01B0$. Využitím pravidla $A \rightarrow 01$ dostaneme $S \Rightarrow_G AA \Rightarrow_G 01A \Rightarrow_G 0101$ – odvodili sme tretie slovo 0101.
- Aplikáciou pravidla $B \rightarrow 1$ na slovo 01B0 dostaneme $S \Rightarrow_G AA \Rightarrow_G 01A \Rightarrow_G 01B0 \Rightarrow_G 0110$ – odvodili sme štvrté slovo 0110.

2) Druhá vetva:

- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow AB$.
- Dostaneme slovo $S \Rightarrow_G AB$.
- Na toto slovo je možné aplikovať dve prepisovacie pravidlá $A \rightarrow B0$ a $A \rightarrow 01$. Aplikáciou $A \rightarrow B0$ dostaneme $S \Rightarrow_G AB \Rightarrow_G B0B$. Aplikáciou $A \rightarrow 01$ dostaneme $S \Rightarrow_G AB \Rightarrow_G 01B$.
- Aplikáciou pravidla $B \rightarrow 1$ na slovo B0B dostaneme $S \Rightarrow_G AB \Rightarrow_G B0B \Rightarrow_G 101$ – odvodili sme piate slovo 101.
- Aplikáciou pravidla $B \rightarrow 1$ na slovo 01B dostaneme $S \Rightarrow_G AB \Rightarrow_G 01B \Rightarrow_G 011$ – odvodili sme šieste slovo 011.

3) Týmto sme vyčerpali všetky možné varianty odvodenia slov v jazyku L(G). Jazyk L(G) je teda tvorený slovami L(G) ={101, 011, 1010, 1001, 0110, 0101}. Tento jazyk je konečný, obsahuje konečný počet prvkov, ktoré sme vymenovali.



Obrázok 4: Názorná schéma odvodenia jazyka, príklad č. 16, b)

c) Pomocou relácií priameho odvodenia určíme jazyk L(G). Odvodzovanie jazyka generovaného gramatikou G má na začiatku dve vetvy, po ktorých sa môžeme vydať. Jedna začína pravidlom $S \rightarrow A$ a druhá pravidlom $S \rightarrow B$.

1) Prvá vetva:

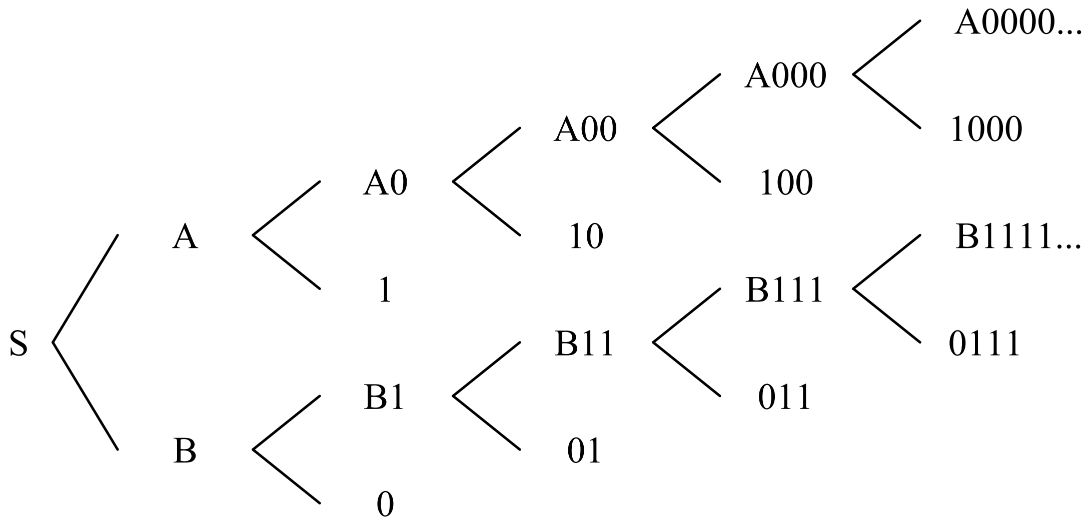
- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow A$.
- Dostaneme slovo $S \Rightarrow_G A$.
- Na slovo A je možné aplikovať dve prepisovacie pravidlá $A \rightarrow A0$ a $A \rightarrow 1$. Využitím pravidla $A \rightarrow 1$ dostaneme $S \Rightarrow_G A \Rightarrow_G 1$ – odvodili sme prvé slovo 1. Využitím pravidla $A \rightarrow A0$ dostaneme $S \Rightarrow_G A \Rightarrow_G A0$.
- Na slovo A0 je možné aplikovať tie isté prepisovacie pravidlá $A \rightarrow A0$ a $A \rightarrow 1$. Využitím pravidla $A \rightarrow 1$ dostaneme $S \Rightarrow_G A \Rightarrow_G A0 \Rightarrow_G 10$ – odvodili sme druhé slovo 10. Využitím pravidla $A \rightarrow A0$ dostaneme $S \Rightarrow_G A \Rightarrow_G A0 \Rightarrow_G A00$.
- Na slovo A00 je možné opäť aplikovať tie isté prepisovacie pravidlá. Využitím pravidla $A \rightarrow 1$ dostaneme $S \Rightarrow_G A \Rightarrow_G A0 \Rightarrow_G A00 \Rightarrow_G 100$ – odvodili sme tretie slovo 100. Využitím pravidla $A \rightarrow A0$ dostaneme $S \Rightarrow_G A \Rightarrow_G A0 \Rightarrow_G A00 \Rightarrow_G A000$.

- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie prvej časti jazyka nám stačí využitie pravidla A → 1 a dostaneme $S \Rightarrow_G A \Rightarrow_G A0 \Rightarrow_G A00 \Rightarrow_G A000 \Rightarrow_G 1000$ – odvodili sme štvrté slovo 1000.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 10^n, kde $n \geq 0$.

2) Druhá vetva:

- Na začiatku využijeme prepisovacie pravidlo S → B.
- Dostaneme slovo $S \Rightarrow_G B$.
- Na slovo B je možné aplikovať dve prepisovacie pravidlá B → B1 a B → 0. Využitím pravidla B → 0 dostaneme $S \Rightarrow_G B \Rightarrow_G 0$ – odvodili sme prvé slovo 0. Využitím pravidla B → B1 dostaneme $S \Rightarrow_G B \Rightarrow_G B1$.
- Na slovo B1 je možné aplikovať tie isté prepisovacie pravidlá B → B1 a B → 0. Využitím pravidla B → 0 dostaneme $S \Rightarrow_G B \Rightarrow_G B1 \Rightarrow_G 01$ – odvodili sme druhé slovo 01. Využitím pravidla B → B1 dostaneme $S \Rightarrow_G B \Rightarrow_G B1 \Rightarrow_G B11$.
- Na slovo B11 je možné opäť aplikovať tie isté prepisovacie pravidlá. Využitím pravidla B → 0 dostaneme $S \Rightarrow_G B \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G 011$ – odvodili sme tretie slovo 011. Využitím pravidla B → B1 dostaneme $S \Rightarrow_G B \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111$.
- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie druhej časti jazyka nám stačí využitie pravidla B → 0 a dostaneme $S \Rightarrow_G B \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111 \Rightarrow_G 0111$ – odvodili sme štvrté slovo 0111.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 01^n, kde $n \geq 0$.

3) Jazyk generovaný gramatikou G je $L(G) = \{10^m, 01^n \mid m, n \geq 0\}$.



Obrázok 5: Názorná schéma odvodenia jazyka, príklad č. 16, c)

d) Pomocou relácií priameho odvodenia určíme jazyk L(G). Odvodzovanie jazyka generovaného gramatikou G má na začiatku dve vetvy, po ktorých sa môžeme vydať. Jedna začína pravidlom $S \rightarrow A1$ a druhá pravidlom $S \rightarrow B1$.

1) Prvá vetva:

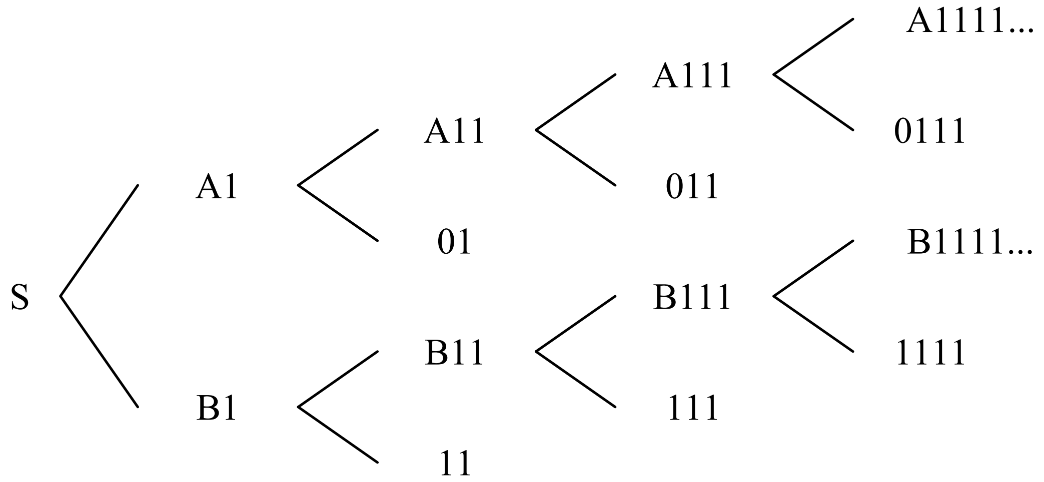
- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow A1$.
- Dostaneme slovo $S \Rightarrow_G A1$.
- Na slovo A1 je možné aplikovať dve prepisovacie pravidlá $A \rightarrow A1$ a $A \rightarrow 0$. Využitím pravidla $A \rightarrow 0$ dostaneme $S \Rightarrow_G A1 \Rightarrow_G 01$ – odvodili sme prvé slovo 01. Využitím pravidla $A \rightarrow A1$ dostaneme $S \Rightarrow_G A1 \Rightarrow_G A11$.
- Na slovo A11 je možné aplikovať tie isté prepisovacie pravidlá $A \rightarrow A1$ a $A \rightarrow 0$. Využitím pravidla $A \rightarrow 0$ dostaneme $S \Rightarrow_G A1 \Rightarrow_G A11 \Rightarrow_G 011$ – odvodili sme druhé slovo 011. Využitím pravidla $A \rightarrow A1$ dostaneme $S \Rightarrow_G A1 \Rightarrow_G A11 \Rightarrow_G A111$.
- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie prvej časti jazyka nám stačí využitie pravidla $A \rightarrow 0$ a dostaneme $S \Rightarrow_G A1 \Rightarrow_G A11 \Rightarrow_G A111 \Rightarrow_G 0111$ – odvodili sme štvrté slovo 0111.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 01^n, kde $n \geq 1$.

2) Druhá vetva:

- Na začiatku využijeme prepisovacie pravidlo $S \rightarrow B1$.
- Dostaneme slovo $S \Rightarrow_G B1$.

- Na slovo B1 je možné aplikovať dve prepisovacie pravidlá $B \rightarrow B1$ a $B \rightarrow 1$. Využitím pravidla $B \rightarrow 1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G 11$ – odvodili sme prvé slovo 11. Využitím pravidla $B \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11$.
- Na slovo B11 je možné aplikovať tie isté prepisovacie pravidlá $B \rightarrow B1$ a $B \rightarrow 1$. Využitím pravidla $B \rightarrow 1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G 111$ – odvodili sme druhé slovo 111. Využitím pravidla $B \rightarrow B1$ dostaneme $S \Rightarrow_G B1 \Rightarrow_G B11 \Rightarrow_G B111$.
- Opäť môžeme aplikovať obe prepisovacie pravidlá, ale pre odvodenie druhej časti jazyka nám stačí využitie pravidla $B \rightarrow 1$ a dostaneme $S \Rightarrow_G B1 \Rightarrow_G B1\,1 \Rightarrow_G B111 \Rightarrow_G 1111$ – odvodili sme štvrté slovo 1111.
- Je evidentné, že táto vetva generovania slov vytvára slová typu 1^m, kde $m \geq 2$.

3) Jazyk generovaný gramatikou G je $L(G) = \{01^n, 1^m \mid n \geq 1, m \geq 2\}$.



Obrázok 6: Názorná schéma odvodenia jazyka, príklad č. 16, d)

Príklad číslo 17

Navrhnite gramatiku G, ktorá generuje zadaný jazyk:

a) Jazyk, ktorý obsahuje rovnaký počet a zreťazený s rovnakým počtom b. $L(G) = \{a^n b^n \mid n \geq 0\}$

b) Jazyk, ktorý obsahuje slová začínajúce symbolom b zreťazené s ľubovoľným počtom symbolov a. $L(G) = \{ba^n \mid n \geq 0\}$

c) Jazyk, ktorý obsahuje slová s párnym počtom a zreťazených s nepárnym počtom b + prázdne slovo. $L(G) = \{a^{2m} b^{2n-1} \mid m, n \geq 0\}$

d) Jazyk, ktorý obsahuje slová zložené ľubovoľne zo symbolov a a b dĺžky najmenej 3. $L(G) = \{w \mid w \in \{a, b\}^*, \mid w \mid \geq 3\}$

Riešenie:

a) Gramatika, ktorá generuje jazyk $L(G) = \{a^n b^n \mid n \geq 0\}$ obsahuje:

$G = (S, \{a, b\}, S, P)$, kde :

$P = \{S \rightarrow aSb \mid \lambda\}$

b) Gramatika generujúca jazyk $L(G) = \{ba^n \mid n \geq 0\}$ bude určite obsahovať aspoň dva terminály, aby sme oddelili začiatok slova so symbolom b a zvyšok slova obsahujúci ľubovoľný počet symbolov a.

$G = (\{S, A\}, \{a, b\}, S, P)$, kde :

$P = \{S \rightarrow b \mid bA,$

$A \rightarrow Aa \mid a\}$

c) Gramatika generujúca jazyk $L(G) = \{a^{2m} b^{2n+1} \mid m, n \geq 0\}$ obsahuje:

$G = (\{S, A, B\}, \{a, b\}, S, P)$, kde :

$P = \{S \rightarrow AB\ ,$

$A \rightarrow aa \mid aaA \mid \lambda,$

$B \rightarrow b \mid bbB \mid \lambda\}$

d) Gramatika generujúca jazyk $L(G) = \{w \mid w \in \{a, b\}^*, \mid w \mid \geq 3\}$ obsahuje:

$G = (\{ S, A, B \},\ a, b\}, S, P)$, kde :

$P = \{S \rightarrow aA \mid bA,$

$A \rightarrow aB \mid bB,$

$B \rightarrow aB \mid bB \mid a \mid b\}$

3.6 Konečné automaty

Príklad číslo 18

Je daný konečný automat, ktorý pracuje nad abecedou $\Sigma = \{a, b\}$ a je tvorený stavmi s_0, s_1, s_2, s_3. Automat zo stavu s_0 prejde po príchode symbolu a do stavu s_1 a po príchode symbolu b do stavu s_2, zo stavu s_1 prejde po príchode symbolu a do stavu s_0 a po príchode symbolu b do stavu s_2, zo stavu s_2 prejde po príchode symbolu a do stavu s_3 a po príchode symbolu b do stavu s_1 a zo stavu s_3 prejde po príchode symbolu a do stavu

s_2 a po príchode symbolu b do stavu s_0. Stav s_0 je počiatočný stav automatu a stav s_3 je konečný stav automatu. Znázornite automat A :

a) Vymenovaním prvkov a pravidiel prechodovej funkcie

b) Tabuľkou prechodov

c) Stavovým diagramom

Riešenie:

a) Konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, s_3), kde pre f platí:

$f(s_0, a) = s_1$, $f(s_2, a) = s_3$,

$f(s_0, b) = s_2$, $f(s_2, b) = s_1$,

$f(s_1, a) = s_0$, $f(s_3, a) = s_2$,

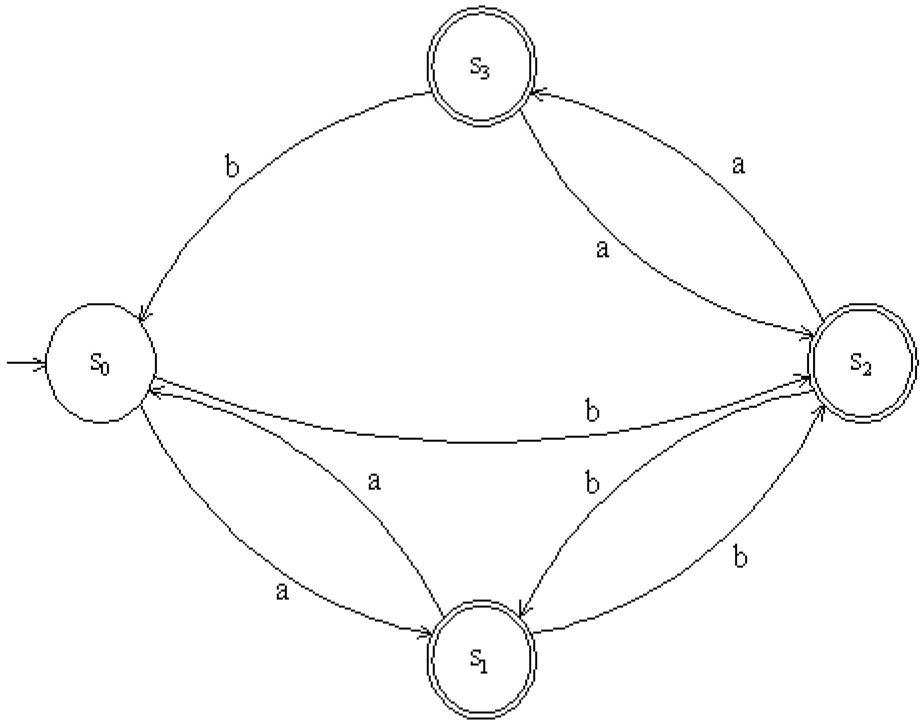
$f(s_1, b) = s_2$, $f(s_3, b) = s_0$.

b) Konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, s_3), kde pre f platí:

Tabuľka 2: Tabuľka prechodov, príklad č. 18

	a	**b**
$\rightarrow s_0$	s_1	s_2
s_1	s_0	s_2
s_2	s_3	s_1
$\leftarrow s_3$	s_2	s_0

c) Konečný automat A = ({s_0, s_1, s_2, s_3}, {a, b}, f, s_0, s_3), kde pre f platí:



Obrázok 7: Automat reprezentovaný stavovým diagramom, príklad č. 18

Príklad číslo 19

Je zadaný jazyk L(A) = {w ∈ {a, b}* | | w $|_a$ mod 2 ≠ 0}. Je to jazyk, ktorého automat prijíma všetky slová nad abecedou Σ = {a, b}, ktoré obsahujú nepárny počet výskytov symbolu a. Určite:

a) Konečný automat A vymenovaním prvkov a pravidiel prechodovej funkcie

b) Konečný automat A tabuľkou prechodov

c) Konečný automat A stavovým diagramom

Riešenie:

a) Konečný automat akceptujúci slová z jazyka L(A) zadaný vymenovaním prvkov je: A = ({s_0, s_1}, {a, b}, f, s_0, s_1), kde pre f platí:

$f(s_0, a) = s_1$, $f(s_1, a) = s_0$,

$f(s_0, b) = s_0$, $f(s_1, b) = s_1$.

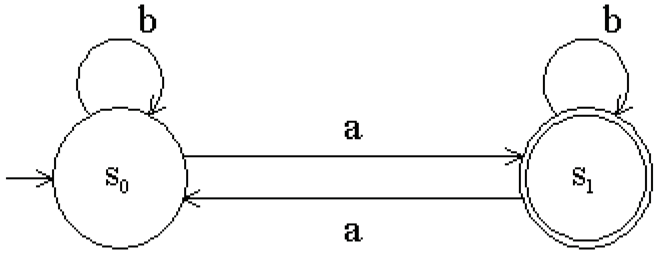
b) Konečný automat akceptujúci slová z jazyka L(A) zadaný tabuľkou prechodov je: A = ({s_0, s_1}, {a, b}, f, s_0, s_1), kde pre f platí:

c)

Tabuľka 3: Tabuľka prechodov, príklad č. 19

	a	**b**
→ s_0	s_1	s_0
← s_1	s_0	s_1

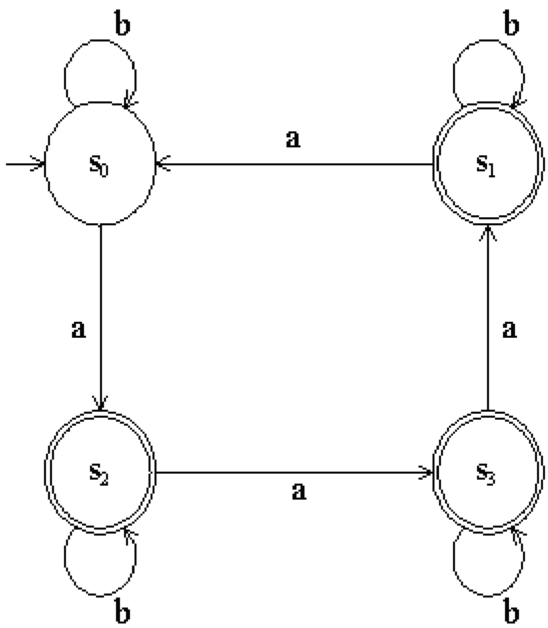
d) Konečný automat akceptujúci slová z jazyka L(A) zadaný stavovým diagramom je: A = ({s_0, s_1}, {a, b}, *f*, s_0, s_1), kde pre *f* platí:



Obrázok 8: Automat reprezentovaný stavovým diagramom, príklad č. 19

Príklad číslo 20

Určite jazyk prijímaný konečným automatom A, ktorého stavový diagram vyzerá nasledovne:



Obrázok 9: Automat reprezentovaný stavovým diagramom, príklad č. 20

Riešenie:

- Z obrázku je vidieť, že prvé slovo, ktoré automat prijíma je slovo $b^m a b^n$, kde m, n ≥ 0.
- Druhým akceptovaným slovom je slovo $b^m a b^n a b^k$, kde kde m, n, k ≥ 0.
- Tretím akceptovaným slovom je slovo $b^m a b^n a b^k a b^l$, kde m, n, k, l ≥ 0.
- Štvrtým akceptovaným slovom je slovo $b^m a b^n a b^k a b^l a b^o a b^p$, kde m, n, k, l, o, p ≥ 0.
- Týmto spôsobom, by sme mohli pokračovať ďalej, ale pre určenie jazyka prijímaného konečným automatom A nám stačia vyššie uvedené štyri slova. Je vidieť, že automat akceptuje slová, ktoré majú počet výskytov symbolov a nedeliteľných 4. Ide teda o jazyk $L(A) = \{w \in \{ a, b \}^* \mid |w|_a \bmod 4 \neq 0\}$.

Príklad číslo 21

Pre konečný automat A, ktorý je zadaný vymenovaním prvkov a pravidlami prechodovej funkcie, vytvorte jeho reprezentáciu pomocou tabuľky prechodov a stavového diagramu. Automat A je zadaný nasledovne: $A = (\{s_0, s_1, s_2, s_3\}, \{a, b, c, d\}, f, s_0, \{s_1, s_3\})$, kde pre f platí:

$f(s_0, a) = s_1$,	$f(s_2, a) = s_3$,
$f(s_0, b) = s_1$,	$f(s_2, b) = s_2$,
$f(s_0, c) = s_2$,	$f(s_2, c) = s_0$,
$f(s_0, d) = s_2$,	$f(s_2, d) = s_2$,
$f(s_1, a) = s_1$,	$f(s_3, a) = s_3$,
$f(s_1, b) = s_1$,	$f(s_3, b) = s_1$,
$f(s_1, c) = s_1$,	$f(s_3, c) = s_1$,
$f(s_1, d) = s_1$,	$f(s_3, d) = s_1$.

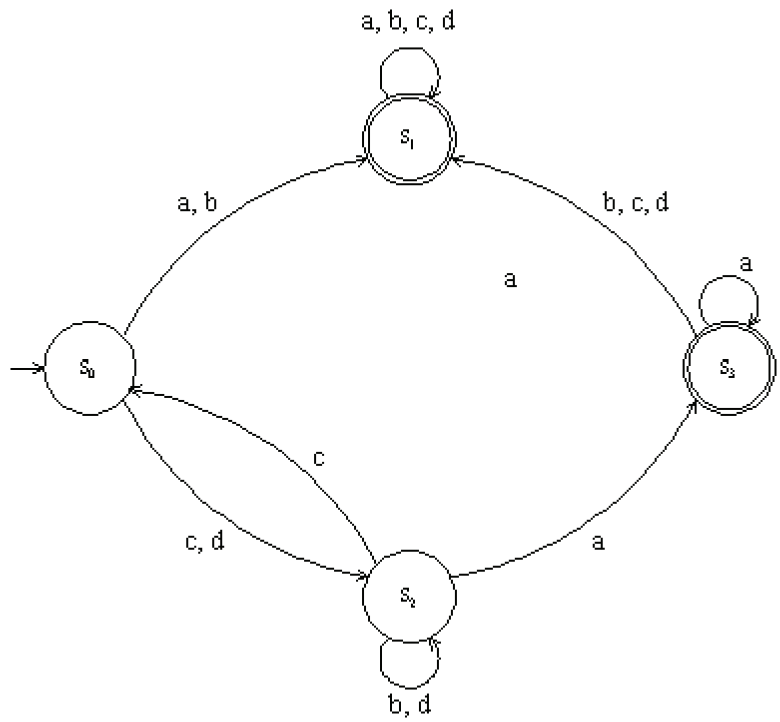
Riešenie:

- Reprezentácia konečného automatu tabuľkou prechodov:

Tabuľka 4: Tabuľka prechodov, príklad č. 21

	a	**b**	**c**	**d**
s_0	s_1	s_1	s_2	s_2
s_1	s_1	s_1	s_1	s_1
s_2	s_3	s_2	s_0	s_2
s_3	s_3	s_1	s_1	s_1

- Reprezentácia konečného automatu stavovým diagramom:

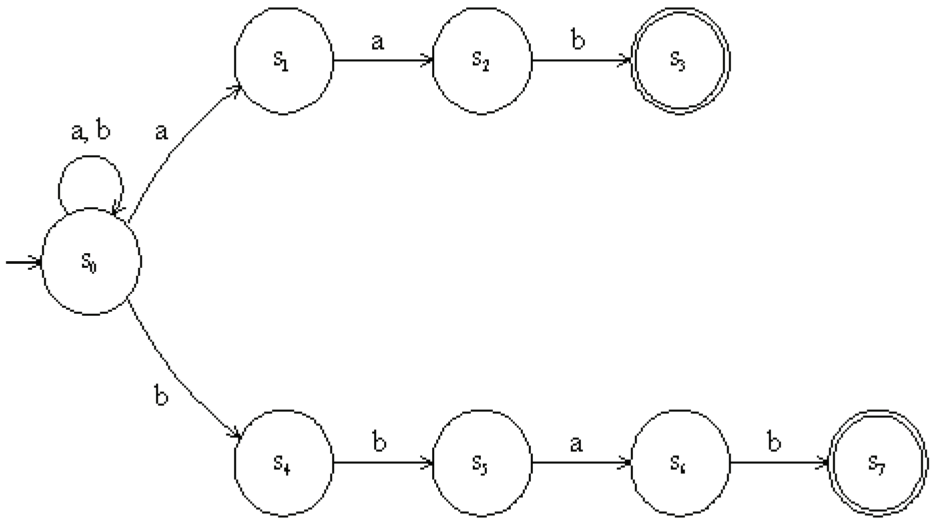


Obrázok 10: Automat reprezentovaný stavovým diagramom, príklad č. 21

Príklad číslo 22

Navrhnite nedeterministický konečný automat A, ktorý prijíma jazyk všetkých slov nad abecedou Σ = {a, b}, ktoré končia sufixom aab alebo bbab.

Riešenie:

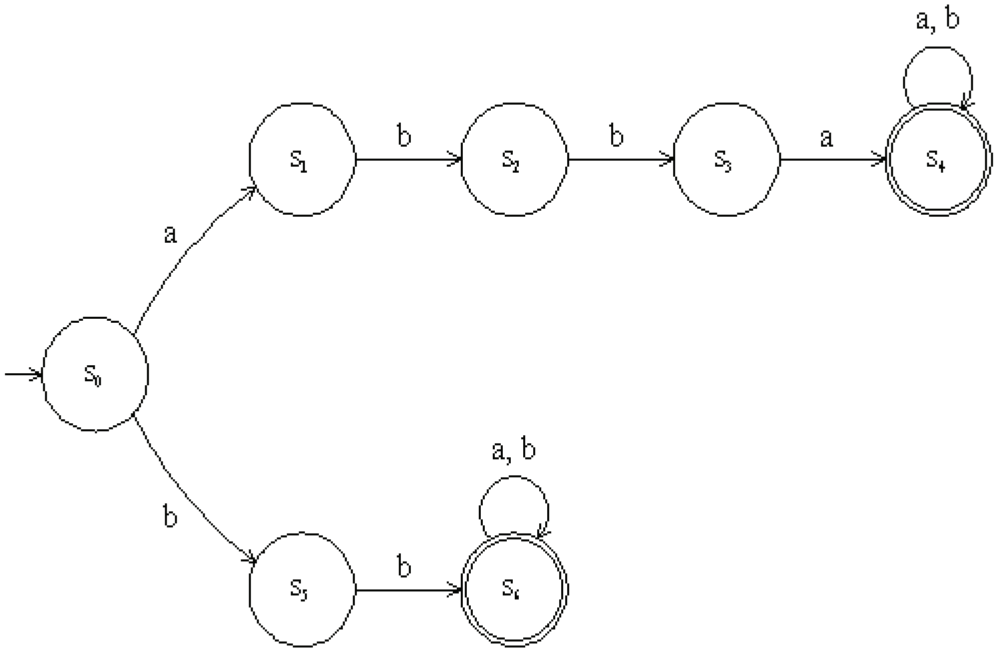


Obrázok 11: Automat reprezentovaný stavovým diagramom, príklad č. 22

Príklad číslo 23

Navrhnite nedeterministický konečný automat A, ktorý prijíma jazyk všetkých slov nad abecedou Σ = {a, b}, ktoré začínajú prefixom abba alebo bb.

Riešenie:



Obrázok 12: Automat reprezentovaný stavovým diagramom, príklad č. 23

Príklad číslo 24

K zadanému deterministickému konečnému automatu A zostrojte totálny konečný automat A´.

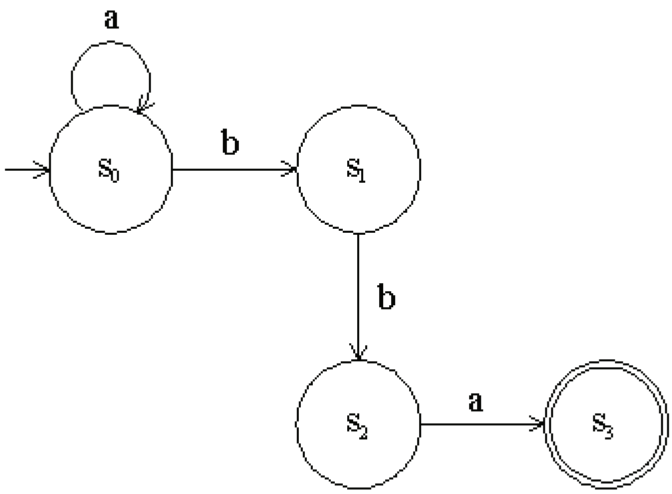
Tabuľka prechodov A:

Tabuľka 5: Tabuľka prechodov, príklad č. 24

	a	**b**
s_0	s_0	s_1
s_1	-	s_2
s_2	s_3	-
s_3	-	-

Riešenie:

- Ako prvé k automatu A zadanému tabuľkou prechodov zostrojíme stavový diagram.



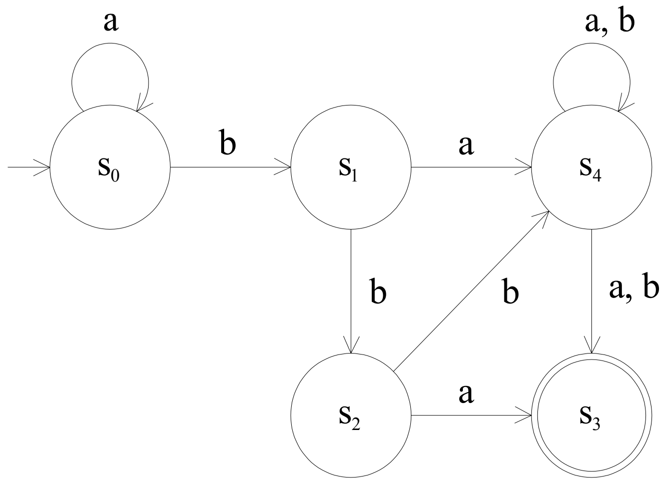
Obrázok 13: Automat reprezentovaný stavovým diagramom, príklad č. 24

- Z tabuľky prechodov automatu A je možné jednoducho zistiť, pre ktoré stavy nie je definovaná prechodová funkcia k jednotlivým vstupom. Vytvoríme teda nový stav s_4 , do ktorého nasmerujeme prechody pre tieto stavy.
- Tabuľka prechodov A´:

Tabuľka 6: Tabuľka prechodov, príklad č. 24

	a	**b**
s_0	s_0	s_1
s_1	s_4	s_2
s_2	s_3	s_4
s_3	s_4	s_4
s_4	s_4	s_4

- Podľa tabuľky prechodov automatu A´ zostrojíme jeho stavový diagram.



Obrázok 14: Automat reprezentovaný stavovým diagramom, príklad č. 24

3.7 Pumping lemma

Príklad číslo 25

Pomocou vety o vkladaní (pumping lemma) dokážte, že jazyk $L = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$ nie je regulárny.

Riešenie:

Pre dôkaz pomocou pumping lemmy sa využíva negácia regularity jazyka:

- $\forall\ n \in N$
- $\exists\ v \in L, |v| \geq n$
- $\forall\ xyz$, kde $|xy| \leq n, |y| \geq 1$
- $\exists\ i \in N_0 : v' = xy^iz \notin L$

Dôkaz že jazyk L nie je regulárny:

- nech n je ľubovoľné číslo z N
- zvolíme slovo $w = a^nb^n$, $|w| \geq n$
- nech rozdelenie je ľubovoľné podľa podmienok $|xy| \leq n$, $|y| \geq n$
- $x = a^k, k \geq 0$
- $y = a^l, l \geq 1, k+l \leq n$
- $z = a^{n-k-l}b^n$, - ide o zvyšok symbolov a zreťazených so symbolmi b
- zvolíme $i = 2$

- $w' = xy^2z = a^k . a^{2l} . a^{n-k-l}b^n = a^{k+2l+n-k-l} . b^n = a^{n+l} . b^n$
- $n+l \neq n$
- $w' \notin L$

Týmto sme dokázali, že jazyk L nie je regulárny.

4 Záver

Táto práca mala dosiahnuť vytýčený cieľ, priblížiť základné typy gramatík, automatov a jazykov a ich základné rozdelenie. Bližšie vysvetliť danú tematiku a spracovať postupy a riešenia príkladov od jednoduchých až po zložité. Práca bola pre prehľadnosť a použiteľnosť rozdelená na dve základné časti a to teoretický prehľad a riešenú zbierku príkladov.

Teoretický prehľad danej tematiky bol spracovaný stručne a prehľadne. Bude slúžiť študentom ako predloha pre správne pochopenie základnej problematiky. Je v ňom vysvetlené čo je to abeceda, slovo a jazyk, čo slúži k následnému pochopeniu, čo je to gramatika a automat generujúci danú gramatiku. Ďalej je teória rozšírená o vysvetlenie totálnej a rozšírenej prechodovej funkcie, deterministického a nedeterministického konečného automatu, stavov dosiahnuteľných automatom a nadbytočných stavov automatu. V kapitole o pumping lemme je vysvetlená dôkazová metóda regularity, respektíve neregularity jazyka.

Zbierka bola napísaná ako súbor príkladov spolu s podrobnými popismi postupov a riešení. Nachádzajú sa v nej príklady týkajúce sa všetkých základných kapitol potrebné pre zvládnutie učiva a absolvovanie skúšky. K príkladom je podrobný popis postupu riešenia, aby bolo jasné ako sa príklad rieši a ako sa k správnemu riešeniu krok po kroku dopracovať.

Snahou bolo vytvoriť stručnú, prehľadnú a pre študentov použiteľnú zbierku príkladov slúžiacu pre podporu výučby predmetu Diskrétna matematika.

5 Zoznam použitej literatúry

[1] ADÁMEK, J. a TRNKOVÁ, V. *Automata and Algebras in Categories*. 1. vyd. Praha: Kluwer Academic Publisher, 1990. 474 s. ISBN 0-7923-0010-6.

[2] ČERNÁ, I., KŘETINSKÝ, M. a KUČERA, A. *Automaty a formální jazyky I*. Brno: Masarykova univerzita, Fakulta informatiky, 2002. 159 s.

[3] GÉSCEG, F. a PEÁK, I. *Algebraic theory of automata.* 1. vyd. Budapest: Akadémiai Kiadó, 1972. 326 s. ISBN 2-6198-3497-1.

[4] GRUSKA, J. *Foundations of computing.* 1. vyd. International Thompson Computer Press. 716 s. ISBN 1-85032-243-0

[5] HOPCROFT, J. E. a ULLMAN, J. D. *Formálne jazyky a automaty*. 1. vyd. Bratislava : Alfa, 1978. 343 s. MDT 519.682.1.

[6] LEPISTO, T. a SALOMAA, A. *Automata, languages and programming*. 1. vyd. Berlin: Springer- Verlag, 1988. 742 s. ISBN 3-540-19488-6.

[7] MEZNÍK, I. *Diskrétní matematika*. 1. vyd. Brno: Akedemické nakladatelství CERM, 2004. 74 s. ISBN 80-214-2754-X.

[8] BIGGS, N.L. *Discrete Mathematics.* 2. vyd. Oxford University Press, 2002. 440 s. ISBN: 978-0-19-850717-8.

[9] ROSEN, K. H. *Discrete mathematics and its applications.* 4. vydanie. Singapore: MCGraw- Hill Book Co, 1999. ISBN 0-07-289905-0

6 Zoznam obrázkov a tabuliek

Zoznam obrázkov

Zoznam tabuliek