Vybrané partie z teórie jazykov

(poznámky z prednášok)

Michal Forišek
Jana Gajdošíková
Richard Kráľovič

Obsah

Disclaimer. Za chyby v texte zodpovedajú autori, ale za to, že im uveríte, si zodpovedáte sami.

Kapitola 1

Úvod

Pri formálnych jazykoch akýsi prvý stupeň abstrakcie predstavuje práca so slovami konkrétneho jazyka. Vyšším stupňom abstrakcie sú operácie so samotnými jazykmi. V ďalšom texte sa budeme zaoberať nasledujúcim stupňom abstrakcie – triedami jazykov.

Ukazuje sa totiž, že viaceré triedy jazykov vykazujú podobné vlastnosti. Ako nahliadneme v ďalšom texte, nie je to náhoda – ukážeme viacero výsledkov štýlu „ak trieda jazykov spĺňa určité predpoklady, potom o nej platia nasledovné tvrdenia“. Takýmto abstraktnejším prístupom si často môžeme ušetriť prácu pri skúmaní konkrétnych tried jazykov.

Asi najjednoduchším príkladom sú de Morganove zákony. Z nich o.i. vyplýva, že každá trieda jazykov uzavretá na zjednotenie a doplnok musí byť nutne uzavretá aj na prienik.

Teória jazykov nie je zďaleka jediným prípadom, kedy vyšší stupeň abstrakcie pomohol získať lepší nadhľad na problematiku. Iným príkladom môže byť algebra, kde postupné abstrakcie predstavujú čísla, algebraické štruktúry a teória kategórií.

1.1 Použité značenie

Pre najznámejšie triedy jazykov budeme používať nasledovné označenia:

$\mathscr{R}$	regulárne jazyky
$\mathscr{R}_0$	regulárne jazyky neobsahujúce ε
$\mathscr{L}_{lin}$	lineárne jazyky (generuje gramatika s pravidlami pod $N \times T^*N^{\leq 1}T^*$)
$\mathscr{L}_{dCF}$	deterministické bezkontextové jazyky (akceptuje DPDA stavom)
$\mathscr{L}_{CF}$	bezkontextové jazyky
$\mathscr{L}_{CS}$	kontextové jazyky (generuje CSG)
$\mathscr{L}_{ECS}$	rozšírené kontextové jazyky (akceptuje LBA)
$\mathscr{L}_{rec}$	rekurzívne jazyky
$\mathscr{L}_{RE}$	rekurzívne vyčísliteľné jazyky
$\mathscr{L}_{E0L}$	rozšírené Lindenmayerove jazyky s nulovou interakciou

Okrem štandardných množinových operácií s jazykmi ($\cup$, $\cap$, ^{C}, $\setminus$, atď.) budeme používať nasledovné operácie:

h	homomorfizmus
h_ε	nevymazávajúci homomorfizmus
h^{-1}	inverzný homomorfizmus
$\cap R$	prienik s regulárnym jazykom
$\cdot$	zreťazenie
$^+$	kladná iterácia
*	iterácia (Kleeneho *)

^{R} reverz (zrkadlový obraz)

$/_l$ ľavý kvocient: $L_1 /_l L_2 = \{w \mid \exists u \in L_2; uw \in L_1\}$ (neformálne: z ľavej strany slov z L_1 urezávame slová z L_2)

$/_r$ pravý kvocient: $L_1 /_r L_2 = \{w \mid \exists u \in L_2; wu \in L_1\}$ (neformálne: z pravej strany slov z L_1 urezávame slová z L_2)

1.2 Uzáverové vlastnosti

Začnime tým, že si zopakujeme uzáverové vlastnosti najznámejších tried jazykov pre najbežnejšie unárne a binárne operácie na jazykoch. Stručne ich zhrnieme v tabuľke 1.1. K netriviálnym uvedieme aspoň stručný komentár.

	$\cup$	$\cap$	$\cdot$	$+$	$*$	^{C}	$\cap R$	h	h_ε	h^{-1}
$\mathscr{R}$	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
$\mathscr{R}_0$	Y	Y	Y	Y	N	N	Y	N	Y	Y
$\mathscr{L}_{lin}$	Y	N	N	N	N	N	Y	Y	Y	Y
$\mathscr{L}_{dCF}$	N	N	N	N	N	Y	Y	N	N	Y
$\mathscr{L}_{CF}$	Y	N	Y	Y	Y	N	Y	Y	Y	Y
$\mathscr{L}_{CS}$	Y	Y	Y	Y	N	N	Y	N	Y	Y
$\mathscr{L}_{ECS}$	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
$\mathscr{L}_{rec}$	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
$\mathscr{L}_{RE}$	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
$\mathscr{L}_{E0L}$	Y	N?	Y	Y	Y	N?	Y	Y	Y	N

Tabuľka 1.1: Uzáverové vlastnosti známych tried jazykov

$\mathscr{L}_{dCF}, \cup\ \cap$ Jazyky $A = \{a^i b^c c^k \mid i = j\}$ a $B = \{a^i b^j c^k \mid j = k\}$ sú z $\mathscr{L}_{dCF}$, ale $A \cup B$ a $A \cap B$ nie.

$\mathscr{L}_{dCF}, \cdot$ Zoberme A, B ako vyššie, nech $0A = \{0w \mid w \in A\}$. Potom $0A \cup B$ je z $\mathscr{L}_{dCF}$, ale $\{0\}^*(0A \cup B)$ nie je, preto $\mathscr{L}_{dCF}$ nie je uzavretá ani len na zreťazenie s regulárnym jazykom. Podobná myšlienka (stratíme „tagy", ktoré nám zabezpečujú deterministickosť jazyka) sa dá použiť pri hľadaní protipríkladov na ďalšie uzáverové vlastnosti ($+$, $*$, ...), ukážeme si ju ešte na jednom príklade.

$\mathscr{L}_{dCF}, h_\varepsilon$ Jazyky $0A$, $1B$ (ako vyššie) a $0A \cup 1B$ sú z $\mathscr{L}_{dCF}$, ale keď zoberieme h taký, že $h(0) = h(1) = 0$ a $\forall x \in \{a, b, c\}; h(x) = x$, $h(0A \cup 1B) \notin \mathscr{L}_{dCF}$.

$\mathscr{L}_{dCF}, ^C$ Ku každému DPDA vieme zostrojiť ekvivalentný, ktorý vždy dočíta vstup. (Musíme mu zabrániť zaseknúť sa a cykliť sa na ε. Na ošetrenie prvej možnosti pridáme na spodok zásobníka nový symbol. Čo s cyklením? Buď sa nám niekedy zopakuje zásobník, alebo porastie do nekonečna. Platí, že $\exists$ konštanta K (závisiaca len od automatu) taká, že ak na ε-kroky narastie zásobník o viac ako K, tak už porastie do nekonečna. Na druhej strane vrchov zásobníkov do veľkosti K je len konečne veľa, keď spravíme viac ε-krokov ako je ich počet, zacyklili sme sa. Technické detaily vynechané.) Potom stačí zobrať nový, ktorého akceptačné stavy sú všetky okrem pôvodných akceptačných stavov (a doriešiť prechody na ε na konci slova).

$\mathscr{L}_{ECS}, ^C$ Innermannova-Szelepczényho veta. Použije sa technika „inductive counting", ktorá donúti LBA prezrieť všetky možné konfigurácie a akceptovať len vtedy, ak žiadna z nich nebola akceptujúca.

$\mathscr{L}_{ECS}, h$ Zoberme ľubovoľný jazyk $L \in \mathscr{L}_{RE}$, A nech je TS s jednou doprava nekonečnou páskou taký, že $L = L(A)$. Nech $B \notin \Sigma_L$. Zostrojíme LBA A' a homomorfizmus h také, že $h(L(A')) = L(A)$. Homomorfizmus h zvolíme tak, že $h(B) = \varepsilon$ a $\forall x \in \Sigma_L; h(x) = x$. A' bude pracovať nad abecedou $\Sigma_L \cup \{B\}$. Na začiatku skontroluje, či je vstup z $\Sigma_L^*\{B\}^*$, ak nie, zasekne sa. Ak áno, bude na ňom simulovať A (+zasekne sa, ak by mal opustiť pásku), B bude brať rovnako ako A berie blank. Nech $w \in L(A)$. Potom zjavne $\exists i; wB^i \in L(A')$. (Keď w doplníme sprava dostatočne veľa blankmi, tento priestor už bude A' stačiť.) Preto $L(A) \subseteq h(L(A'))$. Opačná inklúzia je podobne triviálna.

$\mathscr{L}_{E0L}, \cdot$ U väčšiny operácií, na ktoré je $\mathscr{L}_{E0L}$ uzavretá, vyzerá dôkaz podobne ako pre bezkontextové gramatiky. Ako príklad uvedieme zreťazenie. Nech E_1, E_2 sú E0L-systémy (v normálnom tvare, že terminály sa na nič nové neprepisujú a s disjunktnými množinami neterminálov) s počiatočnými neterminálmi σ_1, σ_2, nech σ, $\overline{\sigma}_1$, $\overline{\sigma}_2$ sú nové neterminály. Zostrojme E0L-systém E tak, že jeho počiatočný neterminál bude σ, bude mať všetky pravidlá z E_1, E_2 a navyše: $\sigma \to \overline{\sigma_1}\,\overline{\sigma_2}$, $\overline{\sigma_i} \to \overline{\sigma_i}$ (toto nám umožní počkať s odvodením jednej časti, ak by dĺžky odvodenia boli rôzne) a $\overline{\sigma_i} \to \sigma_i$. Zjavne $L(E) = L(E_1)L(E_2)$.

$\mathscr{L}_{E0L}, h^{-1}$ Jazyk $L = \{a^{2^n}\}$ je z $\mathscr{L}_{E0L}$, zoberme homomorfizmus taký, že $h(a) = a$, $h(b) = \varepsilon$. Potom $h^{-1}(L) \notin \mathscr{L}_{E0L}$. (Intuitívne: Aby počet a bol mocninou dvoch vieme zabezpečiť len tak, že niekoľkokrát sa počet vhodných symbolov zdvojnásobí – ak sa rozsynchronizujeme a prepíšeme len časť symbolov, už to nezachránime. Toto nám ale zhora obmedzí dĺžku odvodenia slova s $\#_a = 2^n$ na približne kn, kde k je počet neterminálov. A v takto dlhom odvodení nevieme vyrobiť ľubovoľne veľa b.)

$\mathscr{L}_{E0L}, ?$ Dôkaz by bol prinajmenšom nepríjemný, ale intuitívne je to takto.

Nás budú zaujímať hlavne tie triedy jazykov, ktoré majú rozumné uzáverové vlastnosti, t.j. sú uzavreté na väčšinu elementárnych operácií. Teda budú to napríklad triedy $\mathscr{R}$, $\mathscr{L}_{CF}$, $\mathscr{L}_{ECS}$, $\mathscr{L}_{rec}$ a $\mathscr{L}_{RE}$.

Kapitola 2

Charakterizácia bezkontextových jazykov

2.1 Chomského-Schützenbergerova veta

Už v počiatkoch rozvoja teórie jazykov bol jazyk všetkých dobre uzátvorkovaných výrazov jedným z najvýznamnejších príkladov bezkontextového jazyka, keďže štruktúra slov ľubovoľného bezkontextového jazyka v podstate zodpovedá uzátvorkovaným výrazom. (Každý neterminál bezkontextovej gramatiky akoby „od stredu“ generuje časť slova.) Tento neurčitý pocit o významnosti jazyka dobre uzátvorkovaných výrazov sa ukázal byť správny.

Veta 2.1.1 *(Chomsky, Schützenberger) Ku každému bezkontextovému jazyku L vieme nájsť prirodzené číslo n, regulárny jazyk R a homomorfizmus h také, že $L = h(D_n \cap R)$ (kde D_n je Dyckov jazyk – jazyk dobre uzátvorkovaných výrazov tvorených n typmi zátvoriek).*

Dôkaz. Každý úplný binárny strom vieme triviálne popísať dobre uzátvorkovaným výrazom. Myšlienka dôkazu bude spočívať v tom, že zoberieme pre L gramatiku G v prísnom Chomského normálnom tvare, potom vhodné dobre uzátvorkované výrazy budú zodpovedať stromom odvodenia v nej a zobrazíme ich na všetky slová, ktoré majú príslušný strom odvodenia.

Majme teda gramatiku G, ktorej každé pravidlo je tvaru $A \to BC$ alebo $A \to a$ ($A, B, C \in N$, $a \in T$). Zostrojme gramatiku G' nasledovne: neterminály necháme, terminály budú $[^i_\pi,]^i_\pi$ pre $\pi \in P$, $i \in \{1, 2\}$, pravidlá G zmeníme nasledovne: Namiesto $\pi = A \to BC$ bude v P' pravidlo $A \to [^1_\pi\ A\ ]^1_\pi\ [^2_\pi\ B\ ]^2_\pi$, namiesto $\pi = A \to a$ bude v P' pravidlo $A \to [^1_\pi\ ]^1_\pi\ [^2_\pi\ ]^2_\pi$.

Gramatika G' generuje dobre uzátvorkované výrazy, pričom každý ňou vygenerovaný výraz zodpovedá nejakému slovu z $L(G)$ spolu s jeho stromom odvodenia. Zjavne G' nevygeneruje všetky dobre uzátvorkované výrazy nad našou abecedou. Ktoré výrazy teda vygeneruje a ktoré nie?

- Na úvod, prvý znak musí byť $[^1_\pi$ pre nejaké π, ktoré má na ľavej strane počiatočný neterminál.
- Bezprostredne za každou $]^1_\pi$ musí nasledovať $[^2_\pi$.
- Za žiadnou $]^2_\pi$ nemôže bezprostredne nasledovať ľavá zátvorka.
- Pre každé $\pi = A \to BC$ ak nájdeme $[^1_\pi$, bezprostredne za ňou musí nasledovať $[^1_\varrho$ pre nejaké ϱ, ktoré má na ľavej strane B. Analogicky za $[^2_\pi$ musí ísť $[^1_\varrho$ pre nejaké ϱ, ktoré má na ľavej strane C.
- Pre $\pi = A \to a$ musia zátvorky s dolným indexom π nasledovať v správnom poradí bezprostredne za sebou.

On the other hand, ľubovoľný výraz spĺňajúci tieto podmienky vieme ľahko v G' vygenerovať. Nech teda $n = 2|P|$ a D_n je jazyk všetkých dobre uzátvorkovaných výrazov s našimi typmi zátvoriek. Prienikom s vhodným regulárnym jazykom R (ktorý kontroluje vyššie popísané podmienky) z neho vyrobíme $L(G')$.

Definujme teraz homomorfizmus h, ktorý pre $\pi = A \to a$ zobrazí $[^1_\pi$ na a a všetky ostatné zátvorky zobrazí na ε. Potom zjavne $h(L(G')) = L(G)$. Je teda $L = L(G) = h(D_n \cap R)$, q.e.d.

Technický detail. Pri úprave gramatiky na prísny Chomského normálny tvar sa nám z jazyka stratí ε, ak tam bolo. Ak potrebujeme mať vo výslednom jazyku ε, povolíme aj prázdny uzátvorkovaný výraz ako korektný (čiže bude $\varepsilon \in R$), inak nie.

Dôsledok 2.1.2 *Zjavne ku každému $n > 2$ vieme nájsť homomorfizmus g taký, že $D_n = g^{-1}(D_2)$ – g hovorí, ako zakódovať n typov zátvoriek pomocou dvoch typov. Môžeme teda vysloviť upravenú podobu vety 2.1.1: Ku každému bezkontextovému jazyku L vieme nájsť regulárny jazyk R a homomorfizmy g, h také, že $L = h(g^{-1}(D_2) \cap R)$.*

Poznámka. Tvrdenie v dôsledku 2.1.2 sa dá dokázať aj pomocou jemne pozmeneného zásobníkového automatu. Naše zásobníkové automaty mali doteraz akoby inštrukcie $push_x$ (pre $x \in \Gamma$) a *pop*. Nahraďme inštrukciu *pop* inštrukciami pop_x (pre $x \in \Gamma$). Toto nijako nezmení silu automatu, ten si jednoducho nedeterministicky vyberie správnu inštrukciu. Zoberme automat akceptujúci prázdnou pamäťou. Všimnime si teraz postupnosť inštrukcií *push* a *pop* počas výpočtu. Keď sa na inštrukcie $push_x$ a pop_x budeme dívať ako na ľavú a pravú zátvorku typu x, táto postupnosť bude nejaký dobre uzátvorkovaný výraz s $|\Gamma|$ typmi zátvoriek.

K danému jazyku L zoberme zásobníkový automat A, ktorý akceptuje L prázdnou pamäťou a používa len dva zásobníkové symboly x, y. Majme jazyk D_2, môžeme sa naň dívať ako na jazyk dobre uzátvorkovaných výrazov nad abecedou $\{push_x, pop_x, push_y, pop_y\}$. Každé slovo z tohto jazyka zodpovedá korektnej práci so zásobníkom. Zostrojme homomorfizmus g, ktorý zobrazí postupnosť krokov výpočtu na postupnosť inštrukcií na prácu so zásobníkom, ktoré používala.[1]

Potom $g^{-1}(D_2)$ sú všetky postupnosti krokov výpočtu, ktoré korektne pracujú so zásobníkom. Prienikom s vhodným regulárnym jazykom vyhodíme tie, kde na seba nenaväzujú stavy alebo kde stav na začiatku prvého kroku nie je začiatočný. Ostanú nám teda práve všetky akceptačné výpočty nášho automatu.

Homomorfizmus h bude zobrazovať postupnosť krokov výpočtu na postupnosť písmen, ktoré sa počas nej čítali zo vstupu. Zjavne keď zobrazíme týmto homomorfizmom jazyk všetkých akceptačných výpočtov automatu A, dostaneme práve $L(A)$.

Poznámka. Neskôr sa Greibachovej podaril ešte silnejší výsledok – nájsť bezkontextový jazyk taký, že každý bezkontextový jazyk je jeho vhodným inverzným homomorfizmom. Tento výsledok vlastne dáva tretí pohľad na bezkontextové jazyky (po bezkontexových gramatikách a zásobníkových automatoch) – keď sa nám to hodí, môžeme bezkontextový jazyk zadať nie gramatikou ani automatom, ale príslušným homomorfizmom.

2.2 Parikhova veta

Táto veta sa snaží presnejšie charakterizovať štruktúru bezkontextových jazykov. Zjednodušene povedané hovorí, že keď sa budeme pozerať iba na počty terminálov v slovách a zabudneme na ich poradie, výsledok bude nerozlíšiteľný od regulárneho jazyka.

Definícia 2.2.1 *Formálne definujme zobrazenie $\Psi : \Sigma^* \to \mathbb{N}^{|\Sigma|}$ nasledovne: Označme $|\Sigma| = k$. Nech $\Sigma = \{a_1, \dots, a_k\}$. Potom $\Psi(w) \stackrel{def}{=} (\#_{a_1}(w), \dots, \#_{a_k}(w))$. Teda Ψ priradí každému slovu vektor počtov jednotlivých písmen. Toto zobrazenie sa niekedy nazýva* Parikhovo *(Parikh map).*

Obrazom jazyka L v zobrazení Ψ bude množina $\Psi(L) \stackrel{def}{=} \{\Psi(w) \mid w \in L\}$.

Poznámka. Štruktúry $(\Sigma^*, \varepsilon, \cdot)$ a $(\mathbb{N}^k, \overline{0}, +)$ sú monoidy, prvý nie je komutatívny (ak $|\Sigma| > 1$), druhý áno. Nech $\equiv$ je najjemnejšia kongruencia na $(\Sigma^*, \varepsilon, \cdot)$ taká, že $\forall i, j; a_i a_j \equiv a_j a_i$, potom $(\mathbb{N}^k, \overline{0}, +)$ je izomorfný s $(\Sigma^*, \varepsilon, \cdot)/\equiv$.

[1] T.j. vstupná abeceda Σ_1 je množina možných krokov výpočtu, výstupná abeceda Σ_2 je množina inštrukcií.

Práve definované zobrazenie Ψ je monoidový homomorfizmus (spĺňa podmienky $\Psi(xy) = \Psi(x) + \Psi(y)$, $\Psi(\varepsilon) = \overline{0}$). Neskôr dokázané tvrdenie sa bude dať triviálne zovšeobecniť aj pre ľubovoľný iný homomorfizmus medzi týmito monoidmi.

Definícia 2.2.2 *Množinu* $A \subseteq \mathbb{N}^k$ *voláme* lineárnou*, ak existujú* $n \in \mathbb{N}$, $c, v_1, \ldots, v_n \in \mathbb{N}^k$ *také, že* $A = \{c + k_1v_1 + \ldots + k_nv_n \mid k_i \in \mathbb{N}\}$. *Množinu* A *voláme* semilineárnou*, ak je konečným zjednotením lineárnych množín.*

Poznámka. Iný pohľad na lineárne množiny: Nech $v_1, \ldots, v_n \in \mathbb{N}^k$, označme $\langle v_1, \ldots, v_n\rangle \stackrel{def}{=} \{k_1v_1 + \ldots + k_nv_n \mid k_i \in \mathbb{N}\}$ podmonoid generovaný $v_1, \ldots, v_n$. (Inými slovami $\langle v_1, \ldots, v_n\rangle$ je najmenší podmonoid, ktorý obsahuje $v_1, \ldots, v_n, \overline{0}$ a je uzavretý na $+$.) Tento podmonoid indukuje na $(N^k, \overline{0}, +)$ kongruenciu. Triedy každej takejto kongruencie nazveme lineárne množiny.

Príklad. Nech $L_1 = \{a^nb^{2n} \mid n \geq 0\}$, $L_2 = \{a^{2n+1}b^n \mid n \geq 0\}$. Potom $\Psi(L_1) = \{(n, 2n) \mid n \geq 0\}$ je lineárna množina, podobne $\Psi(L_2)$ je lineárna množina. $\Psi(L_1 \cup L_2)$ je semilineárna, ale nie lineárna množina. $\{(n, n^2) \mid n \geq 0\}$ nie je semilineárna množina.

Veta 2.2.1 *(Parikh) Pre každý bezkontextový jazyk* L *je množina* $\Psi(L)$ *semilineárna.*

Dôkaz. Pri dôkaze sa odvoláme na niekoľko liem, ktoré sú vyslovené a dokázané po jeho skončení.

Nech L je ľubovoľný bezkontextový jazyk, $G = (N, T, P, \sigma)$ gramatika preň v prísnom Chomského normálnom tvare.[2] Písmenami $\mathbf{S}, \mathbf{T}, \ldots$ budeme označovať stromy odvodenia, ktoré majú vo vnútorných vrcholoch neterminály, v listoch neterminály a terminály.

Definujme nasledujúce označenia:

$$\begin{aligned} \mathbf{kore\check{n}}(\mathbf{T}) &\stackrel{def}{=} \text{neterminál v koreni stromu } \mathbf{T} \\ \mathbf{spodok}(\mathbf{T}) &\stackrel{def}{=} \text{vetná forma tvorená terminálmi v listoch } \mathbf{T}\text{, čítané zľava doprava} \\ \mathbf{h\acute{l}bka}(\mathbf{T}) &\stackrel{def}{=} \text{dĺžka najdlhšej cesty z listu } \mathbf{T} \text{ do koreňa (počet hrán na nej)} \\ \mathbf{N}(\mathbf{T}) &\stackrel{def}{=} \text{množina neterminálov vyskytujúcich sa v } \mathbf{T} \end{aligned}$$

Strom odvodenia $\mathbf{T}$, v ktorom platí $\mathbf{spodok}(\mathbf{T}) = x.\mathbf{kore\check{n}}(\mathbf{T}).y$, kde $x, y \in T^*$ (a ktorý obsahuje aspoň dva vrcholy) budeme volať *pumpa*. Intuitívne, keď zoberieme ľubovoľný strom odvodenia slova z $L(G)$ a nájdeme v ňom neterminál A, môžeme v tomto mieste strom prerušiť a vložiť doň ľubovoľnú pumpu s koreňom A, takto dostaneme strom odvodenia dlhšieho slova. (Na tejto myšlienke je založená pumpovacia lema.) Tento proces budeme volať *napumpovanie*, resp. *vloženie pumpy*, opačný proces nazveme *vystrihnutie pumpy*.

Zjavne vystrihnutím pumpy z pumpy dostaneme opäť pumpu. Nazvime pumpu *základnou*, ak sa z nej nedá vystrihnúť žiadna menšia pumpa. Z lemy 2.2.2 (a toho, že G má len konečne veľa pravidiel) vyplýva, že základných púmp je pre konkrétnu gramatiku len konečne veľa.

Pre jednoduchší zápis dodefinujme Parikhovo zobrazenie nasledovne: Nech h je homomorfizmus taký, že $\forall \varphi \in N; h(\varphi) = \varepsilon$ a $\forall t \in T; h(t) = t$. Potom pre $w \in (N \cup T)^*$ definujeme $\Psi(w) = \Psi(h(w))$. Pre strom odvodenia $\mathbf{T}$ definujeme $\Psi(\mathbf{T}) = \Psi(\mathbf{spodok}(\mathbf{T}))$.

Definujme $\mathbf{S} \leq \mathbf{T}$, ak sa $\mathbf{T}$ dá vyrobiť z $\mathbf{S}$ konečným počtom vložení základnej pumpy (nie nutne rovnakej), pričom $\mathbf{N}(\mathbf{S}) = \mathbf{N}(\mathbf{T})$. Inými slovami vkladané pumpy nesmú do stromu odvodenia pridať nové neterminály. Takémuto postupu budeme hovoriť *konzervatívne napumpovanie*. Hovoríme, že $\mathbf{S}$ je $\leq$-minimálny, ak pre žiaden strom odvodenia $\mathbf{U} \neq \mathbf{S}$ neplatí $\mathbf{U} \leq \mathbf{S}$.

Myšlienka za tým, prečo definujeme konzervatívne napumpovanie, je nasledovná: akonáhle sa rozhodneme, že budeme pumpovať len konzervatívne, presne vieme, ktoré základné pumpy môžeme použiť, navyše každú z nich môžeme použiť ľubovoľne veľa krát. Parikhov obraz množiny stromov odvodenia, ktoré vzniknú z konkrétneho konzervatívnym napumpovaním bude lineárna množina. Toto tvrdenie formálne vyslovuje lema 2.2.4.

Nech teraz $\mathscr{B}$ je množina tých stromov odvodenia $\mathbf{T}$, ktoré sú $\leq$-minimálne, $\mathbf{kore\check{n}}(\mathbf{T}) = \sigma$ a $\mathbf{spodok}(\mathbf{T}) \in \Sigma^*$. Zjavne každý strom odvodenia, ktorý vznikne napumpovaním stromu z $\mathscr{B}$

[2] Zjavne nás nemusí trápiť, že prípadne stratíme ε z jazyka, do jazyka so semilineárnym obrazom vieme pridať ľubovoľný konečný počet slov bez porušenia semilineárnosti jeho obrazu.

je stromom odvodenia nejakého slova z $L(G)$. Ukážeme teraz, že každý strom odvodenia slova z $L(G)$ vznikne *konzervatívnym* napumpovaním nejakého stromu odvodenia z $\mathscr{B}$. Zoberme ľubovoľné $w \in L(G)$ a jeho ľubovoľný strom odvodenia $\mathbf{W}$. Platí $\mathbf{kore\check{n}}(\mathbf{W}) = \sigma$ a $\mathbf{spodok}(\mathbf{W}) \in \Sigma^*$. Nech $\mathbf{T}$ je (ľubovoľný) strom odvodenia s najmenším počtom vrcholov taký, že $\mathbf{T} \leq \mathbf{W}$. Potom $\mathbf{T}$ je $\leq$-minimálny a ľahko nahliadneme, že $\mathbf{kore\check{n}}(\mathbf{T}) = \sigma$ a $\mathbf{spodok}(\mathbf{T}) \in \Sigma^*$, teda $\mathbf{T} \in \mathscr{B}$.

Je teda

$$L(G) = \bigcup_{\mathbf{S} \in \mathscr{B}} \{\mathbf{spodok}(\mathbf{T}) \mid \mathbf{S} \leq \mathbf{T}\}$$

Z lemy 2.2.5 vyplýva, že množina $\mathscr{B}$ je konečná. Podľa lemy 2.2.4 je množina $\{\Psi(\mathbf{T}) \mid \mathbf{S} \leq \mathbf{T}\}$ lineárna. Preto množina

$$\Psi(L(G)) = \bigcup_{\mathbf{S} \in \mathscr{B}} \{\Psi(\mathbf{T}) \mid \mathbf{S} \leq \mathbf{T}\}$$

je semilineárna, q.e.d.

Lema 2.2.2 *Ak* $\mathbf{T}$ *je základná pumpa, tak* $\mathbf{h\acute{l}bka}(\mathbf{T}) \leq 2|N|$.

Dôkaz. Všimnime si cestu z jediného listu, ktorý obsahuje neterminál, do koreňa. Túto cestu nazveme *hlavná*. Zoberme vrcholy na nej okrem koreňa. Ak by niektoré dva obsahovali rovnaký neterminál, našli sme menšiu pumpu, ktorá sa dá vystrihnúť. Preto dĺžka tejto cesty je najviac $|N|$. Teraz vyjdime z ľubovoľného iného listu smerom do koreňa až kým neprídeme do vrcholu na hlavnej ceste. Opäť, keby boli medzi vrcholmi cez ktoré sme išli dva s rovnakým neterminálom, našli sme menšiu pumpu. Preto dĺžka cesty z listu s terminálom do najbližšieho vrcholu na hlavnej ceste je najviac $|N| + 1$, a teda dĺžka cesty z ľubovoľného listu do koreňa je najviac $2|N|$, q.e.d.

Lema 2.2.3 *Ak sa z* $\mathbf{T}$ *dá vystrihnúť pumpa, tak sa z* $\mathbf{T}$ *dá vystrihnúť základná pumpa.*

Dôkaz. Zoberme pumpu $\mathbf{S}$ s najmenším počtom vrcholov, ktorá sa dá vystrihnúť z $\mathbf{T}$. Ak by nebola základná, dala by sa z nej vystrihnúť menšia pumpa. Tá by sa ale dala vystrihnúť priamo z $\mathbf{T}$, čo je spor s voľbou S.

(Iný, intuitívnejší dôkaz: zoberme tú pumpu, ktorej koreň je v T najhlbšie, t.j. najbližšie pri konci odvodenia. Ak ich je viac, tak tú s najkratšou hlavnou cestou, ak ich je stále viac, tak ľubovoľnú z nich.)

Lema 2.2.4 *Množina* $M_{\mathbf{S}} = \{\Psi(\mathbf{T}) \mid \mathbf{S} \leq \mathbf{T}\}$ *je lineárna.*

Dôkaz. Nech $\mathbf{P_1}, \ldots, \mathbf{P_m}$ sú všetky základné pumpy také, že $\mathbf{N}(\mathbf{P_i}) \subseteq \mathbf{N}(\mathbf{S})$. Potom zjavne $M_{\mathbf{S}} = \{\Psi(\mathbf{S}) + k_1\Psi(\mathbf{P_1}) + \ldots + k_m\Psi(\mathbf{P_m}) \mid k_i \in \mathbb{N}\}$.

Lema 2.2.5 *Z lemy 2.2.2 vyplýva, že základných púmp je len konečne veľa, označme ich počet* p. *Nech* $\mathbf{T}$ *je* $\leq$*-minimálny, potom* $\mathbf{h\acute{l}bka}(\mathbf{T}) \leq (|N| + 1)(p + 1)$.

Dôkaz. Sporom. Majme v $\mathbf{T}$ dlhšiu cestu z koreňa do listu. Potom na nej vieme idúc od koreňa vyznačiť $p + 1$ úsekov, v každom bude $|N| + 1$ neterminálov. V každom úseku sa teda opakuje niektorý neterminál. Každý úsek teda obsahuje pumpu a tieto pumpy sú po dvoch disjunktné. Podľa lemy 2.2.3 každá z nich obsahuje niektorú základnú pumpu. Tých je ale len p, preto podľa Dirichletovho princípu existujú v $\mathbf{T}$ dva disjunktné výskyty niektorej základnej pumpy. Jeden z nich ale môžeme vystrihnúť bez toho, aby sme zmenili $\mathbf{N}(\mathbf{T})$ – to je spor s $\leq$-minimalitou $\mathbf{T}$.

Lema 2.2.6 *Ku každej semilineárnej množine* M *existuje regulárny jazyk* R *taký, že* $\Psi(R) = M$.

Dôkaz. K lineárnej množine triviálne zostrojíme regulárnu gramatiku, pre semilineárnu dostaneme výslednú gramatiku spojením gramatík pre lineárne množiny, ktorých je zjednotením.

Dôsledok 2.2.7 *Z vety 2.2.1 a lemy 2.2.6 vyplýva: Ku každému bezkontextovému jazyku* L *existuje regulárny jazyk* R *taký, že* $\Psi(L) = \Psi(R)$.

Kapitola 3

Abstraktné triedy jazykov

3.1 Definície

Definícia 3.1.1 *Nech Σ je potenciálne nekonečná množina symbolov, $\mathscr{L}$ nech je množina jazykov, ktorá obsahuje aspoň jeden neprázdny jazyk*[1] *a pre každý jazyk $L \in \mathscr{L}$ je $\Sigma_L \subseteq_{kon} \Sigma$. Potom dvojicu $(\mathscr{L}, \Sigma)$ voláme **trieda jazykov**. V ďalšom texte keď bude Σ implicitne zrejmá, prípadne nezaujímavá, budeme triedou jazykov niekedy volať množinu $\mathscr{L}$. (Uvedomte si, že podobné zjednodušenie už bežne používame u jazykov – formálne aj jazyk nie je len množina slov, ale aj abeceda.)*

Nasledujúce definície sú vždy vlastne dve definície v jednej – prvú dostaneme tak, že text v zátvorkách nečítame, druhú tak, že čítame všetok text. Podobný zápis budeme používať aj pri vetách o definovaných pojmoch. Skratka AFL znamená „abstract family of languages“ (abstraktná trieda jazykov).

Definícia 3.1.2 *Triedu jazykov, uzavretú na h_ε (h), h^{-1} a $\cap R$ voláme (úplné) **trio**.*

Definícia 3.1.3 *(Úplné) trio uzavreté na $\cup$ voláme (úplná) **semi-AFL**[2].*

Definícia 3.1.4 *(Úplnú) semi-AFL uzavretú na $\cdot$ a $+$ ($*$) voláme (úplná) **AFL**.*

3.2 Základné poznatky

Lema 3.2.1 *Každé trio je uzavreté na rozdiel s regulárnym jazykom.*

Dôkaz. Nech $\mathscr{L}$ je trio, nech $L \in \mathscr{L}$, $R \in \mathscr{R}$. Potom $R^C \in \mathscr{R}$ a $L \setminus R = L \cap R^C$, preto aj $L \setminus R \in \mathscr{L}$.

Lema 3.2.2 *Trio $\mathscr{L}$ obsahuje jazyk $\{\varepsilon\}$ iff $\mathscr{L}$ obsahuje L taký, že $\varepsilon \in L$.*

Dôkaz. Jeden smer je triviálny. Druhý vyplýva z uzavretosti $\mathscr{L}$ na prienik s jazykom $\{\varepsilon\} \in \mathscr{R}$.

Lema 3.2.3 *Nech $\mathscr{L}$ je trio obsahujúce jazyk L taký, že $\varepsilon \in L$. Potom $\mathscr{R} \subseteq \mathscr{L}$.*

Dôkaz. Majme ľubovoľný $R \in \mathscr{R}$. Nech $L \in \mathscr{L}$ je jazyk zo znenia lemy. Zoberme homomorfizmus $h : \Sigma_R^* \to \Sigma_L^*$ taký, že $\forall w; h(w) = \varepsilon$. Potom $h^{-1}(L) = \Sigma_R^*$ a $R = h^{-1}(L) \cap R$, preto $R \in \mathscr{L}$, q.e.d.

Veta 3.2.4 *Pre každé (úplné) trio $\mathscr{L}$ platí $\mathscr{R}_0 \subseteq \mathscr{L}$ ($\mathscr{R} \subseteq \mathscr{L}$).*

[1] Toto je technický detail – keby tento predpoklad nebol v definícií, musel by byť v znení väčšiny viet.
[2] Škoda že nie kvarteto :-)

Dôkaz. Ak $\mathscr{L}$ je úplné trio, zjavne obsahuje jazyk $\{\varepsilon\}$, tvrdenie potom vyplýva z lemy 3.2.3. Nech teda $\mathscr{L}$ je ľubovoľné trio. Ak nejaký $L \in \mathscr{L}$ obsahuje ε, opäť z lemy 3.2.3 $\mathscr{R} \subseteq \mathscr{L}$, a teda aj $\mathscr{R}_0 \subseteq \mathscr{L}$. V opačnom prípade zoberme ľubovoľný neprázdny $L \in \mathscr{L}$ (taký tam z definície aspoň jeden je), nech w je ľubovoľné jeho slovo. Zoberme ľubovoľný jazyk $R \in \mathscr{R}_0$, ukážeme, že $R \in \mathscr{L}$. Myšlienka: ukážeme, že $\Sigma_R^+ \in \mathscr{L}$, dokazované tvrdenie vyplynie z uzavretosti $\mathscr{L}$ na prienik s reg. jazykom.

Označme $\Sigma'_R = \{x' \mid x \in \Sigma_R\}$. (Toto značenie použijeme aj v niektorých ďalších dôkazoch.) Zvoľme homomorfizmus h nasledovne: $\forall x \in \Sigma'_R; h(x) = w$ a $\forall x \in \Sigma_R; h(x) = \varepsilon$. Zjavne $h^{-1}(L) \supseteq h^{-1}(w) \supseteq \{xu \mid x \in \Sigma'_R \land u \in \Sigma_R^*\}$ a $\varepsilon \notin h^{-1}(L)$.

Teraz zoberme nevymazávajúci homomorfizmus h_2 taký, že $\forall x \in \Sigma_R; h_2(x) = h_2(x') = x$. Zjavne $h_2(h^{-1}(L)) \subseteq \Sigma_R^+$ (lebo obrazy všetkých písmen sú zo Σ_R a $\varepsilon \notin h^{-1}(L)$). Ale $h_2(h^{-1}(L)) \supseteq h_2(\{xu \mid x \in \Sigma'_R \land u \in \Sigma_R^*\}) = \Sigma_R^+$. Preto $h_2(h^{-1}(L)) = \Sigma_R^+$.

Ale potom jazyk $R = h_2(h^{-1}(L)) \cap R$ patrí do tria $\mathscr{L}$, q.e.d.

Veta 3.2.5 *Každé (úplné) trio je uzavreté na zjednotenie s regulárnym bezepsilonovým (ľubovoľným regulárnym) jazykom.*

V tomto dôkaze prvýkrát použijeme predpoklad, že niekoľko jazykov má po dvoch disjunktné abecedy. Môžeme to predpokladať vďaka tomu, že máme k dispozícií (stačí nevymazávajúci) homomorfizmus – ak by abecedy neboli disjunkté, „preložíme“ homomorfizmami jazyky do nových, navzájom disjunktných abecied. Na konci konštrukcie potom „opačnými“ homomorfizmami preložíme písmená nových abecied na zodpovedajúce písmená pôvodných abecied. Tento postup budeme v budúcnosti vynechávať (na zjednodušenie dôkazu).

Dôkaz. Majme $L \in \mathscr{L}$, $R \in \mathscr{R}$, BUNV[3] nech Σ_L, Σ_R a Σ'_R sú po dvoch disjunktné. Ak $L = \emptyset$, $L \cap R = \emptyset \in \mathscr{L}$. Nech je teda L neprázdny, nech w je ľubovoľné slovo z L.

Zoberme homomorfizmus h taký, že $\forall x \in \Sigma_L; h(x) = x$, $\forall x \in \Sigma_R; h(x) = \varepsilon$ a $\forall x \in \Sigma'_R; h(x) = w$. V triu $\mathscr{L}$ máme jazyk $h^{-1}(L)$ aj jeho prienik s regulárnym jazykom $\Sigma_L^* \cup \Sigma'_R \Sigma_R^*$. Ľahko nahliadneme, že to je jazyk $L_1 = L \cup \Sigma'_R \Sigma_R^*$. (Zo Σ_L^* sú v ňom práve slová z L – zobrazia sa v h samé na seba. Všetky slová tvaru ux, kde $u \in \Sigma'_R$ a $x \in \Sigma_R^*$ sa v h zobrazia na w, preto v $h^{-1}(L)$ sú všetky slová z $\Sigma'_R \Sigma_R^*$.)

Teraz stačí zo slov jazyka L_1 vhodným homomorfizmom h' „odstrániť čiarku“. (Ak $\mathscr{L}$ je úplné trio, h' písmená s čiarkou zmaže, inak ich preloží na príslušné písmená bez čiarky.) Ale potom $h'(L_1) \cap (\Sigma_L^* \cup R) = L \cup R$ tiež patrí do $\mathscr{L}$, q.e.d.

Veta 3.2.6 *Každé trio je uzavreté na zreťazenie s regulárnym jazykom zľava aj sprava.*

Dôkaz. Majme $L \in \mathscr{L}$, $R \in \mathscr{R}$, BUNV nech Σ_L a Σ_R sú disjunktné. Zoberme homomorfizmus h taký, že $\forall x \in \Sigma_L; h(x) = x$ a $\forall x \in \Sigma_R; h(x) = \varepsilon$. Vieme, že v $\mathscr{L}$ máme jazyk $L_1 = h^{-1}(L) \cap (\Sigma_R^* \Sigma_L^* \Sigma_R^*) = \Sigma_R^* L \Sigma_R^*$. Ale potom aj $RL = L_1 \cap (R\Sigma_L^*)$ a $LR = L_1 \cap (\Sigma_L^* R)$ patria do $\mathscr{L}$, q.e.d.

Veta 3.2.7 *Nech I je potenciálne nekonečná množina, $\mathscr{L}_i$ (pre $i \in I$) nech sú (úplné) triá. Potom aj $\mathscr{L} = \bigcup_{i \in I} \mathscr{L}_i$ je (úplné) trio.*

Dôkaz. Potrebujeme ukázať, že $\mathscr{L}$ je uzavretá na h_ε (h), h^{-1} a $\cap R$. Zoberme ľubovoľný $L \in \mathscr{L}$. Potom $\exists i \in I; L \in \mathscr{L}_i$. Ale $\mathscr{L}_i$ je (úplné) trio, preto všetky jazyky, ktoré z L povolenými operáciami vyrobíme budú v $\mathscr{L}_i$, a teda aj v $\mathscr{L}$, q.e.d.

Veta 3.2.8 *Nech $\mathscr{L}_1$, $\mathscr{L}_2$ sú semi-AFL. Potom $\mathscr{L}_1 \cup \mathscr{L}_2$ je semi-AFL iff $\mathscr{L}_1 \subseteq \mathscr{L}_2$ alebo $\mathscr{L}_2 \subseteq \mathscr{L}_1$.*

Pred dôkazom si uvedomme, prečo neplatí podobný výsledok ako vo vete 3.2.7. Je to preto, že v semi-AFL už máme *binárnu* operáciu – zjednotenie, ktorá berie ako operandy dva jazyky. A tie vo všeobecnosti nemusia patriť do toho istého $\mathscr{L}_i$. A ako ukáže dôkaz, môžu z toho byť problémy.

[3] bez újmy na všeobecnosti

Dôkaz. Jeden smer je triviálny. BUNV nech $\mathscr{L}_1 \subseteq \mathscr{L}_2$. Potom $\mathscr{L}_1 \cup \mathscr{L}_2 = \mathscr{L}_2$ je semi-AFL. Opačný ukážeme sporom. Nech $\mathscr{L}_1 \cup \mathscr{L}_2$ je semi-AFL, nech $\mathscr{L}_1 \not\subseteq \mathscr{L}_2$ a $\mathscr{L}_2 \not\subseteq \mathscr{L}_1$. Potom existujú $L_1 \in (\mathscr{L}_1 \setminus \mathscr{L}_2)$ a $L_2 \in (\mathscr{L}_2 \setminus \mathscr{L}_1)$, BUNV nech majú disjunktné abecedy. (Prečo to môžeme predpokladať?) Keby $L = L_1 \cup L_2 \in \mathscr{L}_1$, tak by musel aj $L \cap \Sigma_{L_2}^* = L_2$ patriť do $\mathscr{L}_1$, čo ale neplatí. Preto $L_1 \cup L_2 \notin \mathscr{L}_1$ a analogicky $L_1 \cup L_2 \notin \mathscr{L}_2$. Potom ale $L_1 \cup L_2 \notin (\mathscr{L}_1 \cup \mathscr{L}_2)$, a teda $\mathscr{L}_1 \cup \mathscr{L}_2$ nie je uzavretá na $\cup$. Spor.

Príklad. Triedy $\mathscr{R}$ a $\mathscr{L}_{CS}$ sú semi-AFL, ale ich zjednotenie nie je. Najmenšia semi-AFL obsahujúca ich zjednotenie je $\mathscr{L}_{ECS}$.

Veta 3.2.9 *Každá semi-AFL uzavretá na + je AFL. (Teda $\cdot$ v definícií AFL je zbytočné.)*

Dôkaz. Nech $\mathscr{L}$ je semi-AFL uzavretá na $+$. Majme ľubovoľné $L_1, L_2 \in \mathscr{L}$, chceme ukázať, že aj $L_1L_2 \in \mathscr{L}$. BUNV nech L_1 a L_2 majú disjunktné abecedy. V $\mathscr{L}$ máme jazyk $A = (L_1 \cup L_2)^+$, a teda aj $A \cap (\Sigma_{L_1}^* \Sigma_{L_2}^*) = L_1L_2$, q.e.d.

Veta 3.2.10 *Nech $\mathscr{L}$ je trio (semi-AFL, AFL). Definujme triedy $\mathscr{L}_1$, $\mathscr{L}_2$ nasledovne:* $\mathscr{L}_1 = \{L \mid L \in \mathscr{L} \wedge \varepsilon \notin \mathscr{L}\}$, $\mathscr{L}_2 = L \cup \{L \cup \{\varepsilon\} \mid L \in \mathscr{L}\}$. *Potom:*

1. *$\mathscr{L}_1$, $\mathscr{L}_2$ sú triá (semi-AFL, AFL)*
2. *$\mathscr{L}_1 = \{L \setminus \{\varepsilon\} \mid L \in \mathscr{L}\}$*
3. *ak $\mathscr{L}$ je semi-AFL, tak $\mathscr{L} = \mathscr{L}_1$ alebo $\mathscr{L} = \mathscr{L}_2$*

Dôkaz. Prvé tvrdenie je triviálne, stačí overiť, že pomocou žiadnej z povolených operácií nevyrobíme v $\mathscr{L}_1$ ani $\mathscr{L}_2$ nič, čo tam nepatrí. (Vyrobíme zhruba to isté čo v $\mathscr{L}$, až na nejaký ε hore-dole.)

Druhé tvrdenie stačí dokázať pre triá. Nech $L \in \mathscr{L}_1$, potom $L \in \mathscr{L}$ a $L \setminus \{\varepsilon\} = L$, preto $L \in \{L \setminus \{\varepsilon\} \mid L \in \mathscr{L}\}$. Naopak nech $L \in \{L \setminus \{\varepsilon\} \mid L \in \mathscr{L}\}$. Potom buď $L \in \mathscr{L}$ alebo $(L \cup \{\varepsilon\}) \in \mathscr{L}$. Ale ak nastal druhý prípad, podľa lemy 3.2.1 aj $((L \cup \{\varepsilon\}) \setminus \{\varepsilon\}) = L \in \mathscr{L}$. Teda určite $L \in \mathscr{L}$ a keďže $\varepsilon \notin L$, $L \in \mathscr{L}_1$, q.e.d.

Nech $\mathscr{L}$ je semi-AFL a $\mathscr{L} \neq \mathscr{L}_1$, na dôkaz tretieho tvrdenia stačí ukázať, že potom $\mathscr{L} = \mathscr{L}_2$. Keďže $\mathscr{L} \neq \mathscr{L}_1$, $\exists L \in \mathscr{L}; \varepsilon \in \mathscr{L}$. Potom zjavne aj $L \cap \{\varepsilon\} = \{\varepsilon\} \in \mathscr{L}$ a keďže $\mathscr{L}$ je semi-AFL, $\forall L \in \mathscr{L}; (L \cup \{\varepsilon\}) \in \mathscr{L}$, čiže naozaj $\mathscr{L} = \mathscr{L}_2$, q.e.d.

Poznámka. Pre triá tretie tvrdenie vo všeobecnosti neplatí – napr. zoberme $\mathscr{L} = \mathscr{R} \cup \mathscr{L}_{CS}$, to je trio, ale $\mathscr{L}_1 = \mathscr{R}_0 \cup \mathscr{L}_{CS} = \mathscr{L}_{CS} \neq \mathscr{L}$ a $\mathscr{L}_2 = \mathscr{R} \cup \mathscr{L}_{ECS} = \mathscr{L}_{ECS} \neq \mathscr{R}$. (Resp. technickejšia konštrukcia: stačí vziať dve triá, ktorých (možno nekonečné) abecedy sú disjunktné, pričom jedno obsahuje a druhé neobsahuje jazyk obsahujúci ε.)

Cvičenie 3.2.1 *Ak $\mathscr{L}$ je (úplné) trio (semi-AFL, AFL), tak aj $\mathscr{L}^R$ je to isté.*

Cvičenie 3.2.2 *Každé (úplné) trio (semi-AFL, AFL) $\mathscr{L}$ obsahuje* najväčšie *(úplné) trio (semi-AFL, AFL) uzavreté na ^{R}.* (Najväčšie *znamená, že je nadmnožinou akéhokoľvek iného.)*

Cvičenie 3.2.3 *Nájdite príklad AFL $\mathscr{L}$, obsahujúcej jazyk $\{\varepsilon\}$, pre ktorú $\mathscr{L}' = \{\Sigma_1^* \setminus L \mid L \in \mathscr{L}, \Sigma_1 \subseteq_{kon} \Sigma_{\mathscr{L}}\}$ nie je AFL.*

Cvičenie 3.2.4 *Nech I je potenciálne nekonečná množina, $\mathscr{L}_i$ (pre $i \in I$) nech sú (úplné) triá (semi-AFL, AFL). Potom aj $\mathscr{L} = \bigcap_{i \in I} \mathscr{L}_i$ je (úplné) trio (semi-AFL, AFL).*

3.3 A-prekladače

V tejto časti sa budeme okrem iného zaoberať nasledujúcou otázkou: Nech $\mathscr{L}$ je ľubovoľná trieda jazykov. Ako nájsť najmenšie trio, ktoré túto triedu jazykov obsahuje? Ako uvidíme, pri odpovedi na túto otázku sa nám veľmi hodia a-prekladače, keďže sú schopné robiť všetky operácie, na ktoré má byť trio uzavreté.

Definícia 3.3.1 ***A-prekladač*** *je šestica $(K, \Sigma_1, \Sigma_2, H, q_0, F)$, kde K je konečná množina stavov, $F \subseteq K$ je množina akceptačných stavov, $q_0 \in K$ je počiatočný stav, Σ_1 je vstupná, Σ_2 je výstupná abeceda a $H \subseteq_{kon} K \times \Sigma_1^* \times \Sigma_2^* \times K$ je množina prechodových pravidiel.*

*A-prekladač, pre ktorý $H \subseteq_{kon} K \times \Sigma_1^* \times \Sigma_2^+ \times K$ (t.j. v každom kroku zapíše aspoň 1 písmeno) voláme **ε-free** (bez-ε; nevymazávajúci).*

Definícia 3.3.2 ***Konfigurácia*** *a-prekladača A je trojica (q, u, v), kde $q \in K_A$ je aktuálny stav, $u \in \Sigma_1^*$ je doteraz neprečítaná časť vstupu a $v \in \Sigma_2^*$ je doteraz zapísaná časť výstupu.*

Krok výpočtu *a-prekladača A je relácia $\vdash$ na jeho konfiguráciách definovaná nasledovne:*
$(q, u_1u_2, v_1) \vdash (p, u_2, v_1v_2) \iff (q, u_1, v_2, p) \in H_A$
Výpočet a-prekladača A budeme zapisovať ako postupnosť použitých štvoríc z H, každý výpočet teda bude slovo[4] z H^+.

Obrazom jazyka *L v zobrazení a-prekladačom A je jazyk:*
$A(L) = \{v \mid \exists u \in L, q_F \in F_A; (q_0, u, \varepsilon) \vdash^* (q_F, \varepsilon, v)\}$
Namiesto $A(\{w\})$ budeme niekedy písať $A(w)$.

Nám sa bude hodiť skutočnosť, že ľubovoľné zobrazenie a-prekladačom vieme zapísať pomocou vhodného homomorfizmu, inverzného homomorfizmu a prieniku s regulárnym jazykom. Toto v ďalšom ukážeme.

Definícia 3.3.3 *Pre a-prekladač A definujme homomorfizmy pr_i (pre $i \in \{1, 2, 3, 4\}$) z H^* do K^*, Σ_1^*, Σ_2^* a K^* nasledovne: $\mathrm{pr}_i((h_1, h_2, h_3, h_4)) = h_i$. Tieto homomorfizmy budeme volať **projekcie**. (Neformálne pr_i nám vráti i-tu zložku pravidel).*

Jazyk platných výpočtov *a-prekladača A je jazyk $V_A \subseteq H^+$, pričom $w = h_1h_2 \dots h_k$ patrí do V_A práve vtedy, keď: $\mathrm{pr}_1(h_1) = q_0$, $\mathrm{pr}_4(h_k) \in F_A$ a $\forall i$; $\mathrm{pr}_4(h_i) = \mathrm{pr}_1(h_{i+1})$. (Neformálne: Postupnosť pravidiel w predstavuje platný výpočet, ak začíname v počiatočnom stave, končíme v akceptačnom a počas výpočtu sa stav A mení len použitím pravidiel.)*

Lema 3.3.1 *Zjavne pre každý a-prekladač A je jazyk jeho platných výpočtov V_A regulárny.*

Lema 3.3.2 *Pre každý a-prekladač A a jazyk L je $A(L) = \mathrm{pr}_3(\mathrm{pr}_2^{-1}(L) \cap V_A)$.*

Dôkaz. $\mathrm{pr}_2(L)$ sú tie postupnosti štvoríc z H_A, ktorých druhé projekcie dokopy tvoria slovo z L. Teda $\mathrm{pr}_2^{-1}(L) \cap V_A$ sú práva všetky platné výpočty A na slovách z L. Z každého z nich zoberieme tretiu projekciu, t.j. slovo, ktoré pri ňom zapísal na výstup.

Uvedomte si, že pr_2, pr_3 a V_A nezávisia od L, iba od A. Takisto si všimnite, že ak A je ε-free, pr_3 je nevymazávajúci homomorfizmus.

Veta 3.3.3 *K ľubovoľným (ε-free) a-prekladačom A_1, A_2 existuje (ε-free) a-prekladač A_3 taký, že $\forall L; A_3(L) = A_2(A_1(L))$.*

Myšlienka dôkazu. Pre jednoduchosť môžeme upraviť a-prekladače A_1, A_2 do tzv. 1-bounded normálneho tvaru – v každom kroku číta aj píše nanajvýš 1 písmeno. A_3 bude simulovať oba, pričom výstup z A_1 si bude pamätať v stave a následne ho spracúvať A_2.

Definícia 3.3.4 *Najmenšie trio jazykov obsahujúce triedu jazykov $\mathscr{L}$ budeme značiť $\mathfrak{M}(\mathscr{L})$. Najmenšie úplné trio jazykov obsahujúce triedu jazykov $\mathscr{L}$ budeme značiť $\widehat{\mathfrak{M}}(\mathscr{L})$. Nech L je jazyk, potom namiesto $\mathfrak{M}(\{L\})$ budeme písať $\mathfrak{M}(L)$.*

Veta 3.3.4 *Nech $\mathscr{L}$ je trieda jazykov, potom platí: $\mathfrak{M}(\mathscr{L}) = \{M(L) \mid M$ je ε-free a-prekladač$\}$, $\widehat{\mathfrak{M}}(\mathscr{L}) = \{M(L) \mid M$ je a-prekladač$\}$.*

[4] Z technických príčin zakážeme prázdny výpočet, silu a-prekladača to neovplyvní.

Dôkaz. Zobrazenie (ε-free) a-prekladačom vieme podľa lemy 3.3.2 zapísať pomocou h (h_ε), h^{-1} a $\cap R$. Preto uvedená trieda jazykov je určite podmnožinou $\widehat{\mathfrak{M}}(\mathscr{L})$ ($\mathfrak{M}(\mathscr{L})$).

Na druhej strane každý jazyk v $\widehat{\mathfrak{M}}(\mathscr{L})$ ($\mathfrak{M}(\mathscr{L})$) vieme vyrobiť z nejakého jazyku z $\mathscr{L}$ konečným počtom uvedených operácií. Každú z nich vieme realizovať nejakým a-prekladačom. Podľa vety 3.3.3 však a-prekladače vieme „skladať“, preto pre každý jazyk J z $\widehat{\mathfrak{M}}(\mathscr{L})$ ($\mathfrak{M}(\mathscr{L})$) existuje ($\varepsilon$-free) a-prekladač A a jazyk $L \in \mathscr{L}$ také, že $J = A(L)$, čím sme dokázali aj opačnú inklúziu.

Niekedy sa nám môže zdať požiadavka, aby a-prekladač bol ε-free, prisilná. Napríklad u kontextových gramatík sme vedeli „umazať“ niekoľko symbolov z vetnej formy tak, že sme prefíkane použili poschodové symboly. Ukáže sa, že za rozumných obmedzení môžeme vymazávať aj pri použití a-prekladačov.

Definícia 3.3.5 *Nech M je a-prekladač, L jazyk. Potom M je* ***ohraničene vymazávajúci*** *na L (ε-output bounded under non-ε input in L) ak existuje také k, že platí:*

Pre všetky $n \in \mathbb{N}$, $x_1, \dots, x_n \in \Sigma_L^*$ platí: Ak existujú p_i také, že $\forall i; (p_i, x_i, \varepsilon, p_{i+1}) \in H_M$ a existujú z_1, z_2, y také, že $z_1 x_1 x_2 \dots x_n z_2 \in L$ a $(q_0, z_1 x_1 x_2 \dots x_n z_2, \varepsilon) \vdash^* (p_1, x_1 x_2 \dots x_n z_2, y)$, tak $|x_1 x_2 \dots x_n| \leq k$.

Teda ak sa a-prekladač dostane do situácie, v ktorej môže niekoľko krokov po sebe písať na výstup ε, nesmie počas týchto krokov zo vstupu prečítať (a umazať) viac ako k symbolov.

Veta 3.3.5 *Nech $\mathscr{L}$ je trio, $L \in \mathscr{L}$ je jazyk a A je a-prekladač ohraničene vymazávajúci vzhľadom na L. Potom $A(L) \setminus \{\varepsilon\} \in \mathscr{L}$. Ak navyše $\mathscr{L}$ je uzavretá na $\cup\{\varepsilon\}$ alebo ak z $\varepsilon \in A(x)$ vyplýva $x = \varepsilon$, aj $A(L) \in \mathscr{L}$.*

Myšlienka dôkazu. Aby sme ukázali, že $A(L) \setminus \{\varepsilon\} \in \mathscr{L}$, potrebujeme zostrojiť ε-free a-prekladač A', ktorý robí v podstate to isté ako A. Druhá časť vety potom vyplynie z podmienok v zadaní.

Podobnú definíciu ako pre a-prekladače môžeme spraviť aj pre homomorfizmy.

Definícia 3.3.6 *Nech L je jazyk, h je homomorfizmus zo Σ_L^*. Hovoríme, že h je ohraničene vymazávajúci na L ak existuje také k, že platí: Nech $xyz \in L$ a $h(y) = \varepsilon$. Potom $|y| \leq k$.*

Veta 3.3.6 *Každé trio obsahujúce $\{\varepsilon\}$ je uzavreté na ohraničene vymazávajúce homomorfizmy.*

Jedna z výhod a-prekladačov spočíva v tom, že sa s nimi pracuje omnoho pohodlnejšie ako s pôvodnými operáciami. Ukážeme si teraz niekoľko príkladov.

Veta 3.3.7 *Nech $\mathscr{L}$ je úplné trio, nech $L \in \mathscr{L}$, $R \in \mathscr{R}$. Potom $(L /_l R)$ a $(L /_r R)$ (ľavý a pravý kvocient L podľa R) patria do $\mathscr{L}$.*

Dôkaz. Triviálne zostrojíme a-prekladač, ktorý L preloží na $(L /_r R)$ – kopíruje vstup na výstup, v nejakom okamihu sa rozhodne, že ide kontrolovať, zvyšok vstupu už len číta a simuluje konečný automat pre R. Ak si dobre tipol, akceptuje. Pre ľavý kvocient analogicky.

Rozmyslite si, ako by vyzeral dôkaz tohto tvrdenia pomocou pôvodných operácií (h, h^{-1}, $\cap R$).

Veta 3.3.8 *Nech $\mathscr{L}$ je trio, nech $L \in \mathscr{L}$, $w \in \Sigma_L^+$. Potom aj $(L /_l \{w\}) \setminus \{\varepsilon\}$ a $(L /_r \{w\}) \setminus \{\varepsilon\}$ patria do $\mathscr{L}$. (T.j. môžeme zo začiatku aj konca slov vyškrtnúť ľubovoľné neprázdne slovo.)*

Dôkaz. Analogicky ako v predchádzajúcej vete, použijeme ohraničene vymazávajúci a-prekladač.

Veta 3.3.9 *Nie každé úplné trio je uzavreté na kvocient.*

Dôkaz. Uvažujme napr. triedu bezkontextových jazykov $\mathscr{L}_{CF}$. Nájdeme jazyky L_1, L_2 také, že $L_1 /_l L_2 \notin \mathscr{L}_{CF}$. Nech $L_1 = \{a^i b a^{2i} b | i \geq 1\}^*$. Nech $L_2 = abL_1$. Zjavne L_1 aj L_2 bezkontextové sú. Je ľahko nahliadnuť, že $(L_1 /_l L_2) \cap \{a^i b | i \geq 0\} = \{a^{2^n} b | n \geq 1\}$. Tento jazyk bezkontextový nie je. Kedže jazyk $\{a^i b | i \geq 0\}$ je regulárny, jazyk $L_1 /_l L_2$ nemôže byť bezkontextový. Dôkaz pre pravý kvocient je analogický.

Cvičenie 3.3.1 *Ukážte, že k ľubovoľným dvom homomorfizmom h_1, h_2 existuje homomorfizmus h taký, že $\forall L; h(L) = h_2(h_1(L))$. Rozhodnite, či platia podobné tvrdenia o inverzných homomorfizmoch a prienikoch s regulárnym jazykom.*

Cvičenie 3.3.2 *Ukážte, že k homomorfizmu h_1 a regulárnemu jazyku R_1 existujú homomorfizmus h_2 a regulárny jazyk R_2 také, že $\forall L; h_1(L \cap R_1) = h_2(L) \cap R_2$.*

Cvičenie 3.3.3 *Ukážte, že k inverznému homomorfizmu h_1^{-1}, homomorfizmu h_2 a regulárnemu jazyku R existuje a-prekladač A taký, že $\forall L; h_1^{-1}(h_2(L) \cap R) = A(L)$.*

3.4 Substitúcie

Homomorfizmus je zobrazenie h zo Σ_1^* do Σ_2^* také, že $\forall u, v; h(uv) = h(u)h(v)$. Táto vlastnosť hovorí, že homomorfizmus je jednoznačne určený obrazmi písmen zo Σ_1. Prirodzeným zovšeobecnením homomorfizmu bude substitúcia.

Definícia 3.4.1 ***Substitúcia*** *je zobrazenie τ zo Σ_1^* do $2^{\Sigma_2^*}$ také, že $\forall u, v; \tau(uv) = \tau(u)\tau(v)$.*
Ak sú všetky jazyky $\tau(a)$ ($a \in \Sigma_1$) regulárne (bezkontextové, atď.), hovoríme, že substitúcia je regulárna (bezkontextová, atď.).

Rozdiel oproti homomorfizmu je ten, že obrazom slova nie je slovo, ale jazyk. Zjavne aj substitúcia je jednoznačne určená obrazmi písmen zo Σ_1.

Homomorfizmus sme vedeli rozšíriť aj na jazyky: $h(L) = \{h(w) \mid w \in L\}$. Podobne vieme na jazyky rozšíriť aj substitúciu: $\tau(L) = \bigcup_{w \in L} \tau(w)$.

Substitúciu vieme rozšíriť aj na triedy jazykov, a to nasledovne:
$\mathrm{SUB}(\mathscr{L}_1, \mathscr{L}_2) = \{\tau(L) \mid L \in \mathscr{L}_1 \ \wedge \ \forall a \in \Sigma_L; (\tau(a) \in \mathscr{L}_2 \ \wedge \ \varepsilon \notin \tau(a))\}$
$\widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_2) = \{\tau(L) \mid L \in \mathscr{L}_1 \ \wedge \ \forall a \in \Sigma_L; \tau(a) \in \mathscr{L}_2\}$

Teda pri našej definícií SUB predstavuje „bez-ε“ a $\widehat{\mathrm{SUB}}$ ľubovoľnú substitúciu, pričom výslednú triedu jazykov dostaneme tak, že do jazykov z $\mathscr{L}_1$ substitujeme jazyky z $\mathscr{L}_2$.

Príklad. $\widehat{\mathrm{SUB}}(\mathscr{L}_{CF}, \mathscr{L}_{CF}) = \mathscr{L}_{CF}$, $\widehat{\mathrm{SUB}}(\mathscr{L}_{CS}, \mathscr{R}) = \mathscr{L}_{RE}$, $\mathrm{SUB}(\mathscr{L}_{CS}, \mathscr{R}_0) = \mathscr{L}_{CS}$.

Substitúcia je zovšeobecnením viacerých operácií, s ktorými sme sa už stretli.

Príklad. Nech $\tau(a) = L_1$, $\tau(b) = L_2$. Potom $\tau(\{a, b\}) = L_1 \cup L_2$, $\tau(\{ab\}) = L_1 L_2$, $\tau(\{a\}^*) = L_1^*$.

Veta 3.4.1 *Nech $\mathscr{L}$ je trieda jazykov, potom $\widehat{\mathrm{SUB}}(\mathscr{R}_0, \mathscr{L})$, resp. $\widehat{\mathrm{SUB}}(\mathscr{R}, \mathscr{L})$ je najmenšia (v zmysle „je podmnožinou akejkoľvek inej“) trieda jazykov, obsahujúca $\mathscr{L}$ a uzavretá na $\cup$, $\cdot$ a $+$ (resp. $*$).*

Dôsledok 3.4.2 *Nech $\mathscr{L}$ je AFL. Potom $\widehat{\mathrm{SUB}}(\mathscr{R}_0, \mathscr{L}) = \mathscr{L}$.*

Veta 3.4.3 *Pre každé trio $\mathscr{L}$ platí $\widehat{\mathrm{SUB}}(\mathscr{L}, \mathscr{R}_0) = \mathscr{L}$. Pre každé úplné trio $\mathscr{L}$ platí $\widehat{\mathrm{SUB}}(\mathscr{L}, \mathscr{R}) = \mathscr{L}$.*

Lema 3.4.4 *Pre každé úplné trio $\mathscr{L}_1$ a triedu jazykov $\mathscr{L}_2$ platí $\widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_2) \subseteq \widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_3)$, pričom $\mathscr{L}_3 = \{L - \{\varepsilon\} | L \in \mathscr{L}_2\}$.*

Veta 3.4.5 *Nech $\mathscr{L}_1$ je úplné trio, $\mathscr{L}_2$ je trieda jazykov uzavretá na unárnu operáciu $\setminus\{\varepsilon\}$. Potom $\widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_2) = \mathrm{SUB}(\mathscr{L}_1, \mathscr{L}_2)$.*

Veta 3.4.6 *Každé trio uzavreté na bezepsilonovú substitúciu je AFL.*

Cvičenie 3.4.1 *Nájdite AFL $\mathscr{L}_1$, $\mathscr{L}_2$ také, že:*

1. $\widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_2) \neq \widehat{\mathrm{SUB}}(\mathscr{L}_2, \mathscr{L}_1)$

2. $\mathrm{SUB}(\mathscr{L}_1, \mathscr{L}_2) \neq \mathrm{SUB}(\mathscr{L}_2, \mathscr{L}_1)$

(Hint: *skúste triedy* $\mathscr{R}$ *a* $\mathscr{L}_{CS}$*).*

Definícia 3.4.2 ***Inverzná substitúcia*** *τ^{-1} k substitúcií τ je zobrazenie zo $2^{\Sigma_2^*}$ do $2^{\Sigma_1^*}$ také, že $\tau^{-1}(L) = \{w \mid w \in \Sigma_1^* \wedge \tau(w) = L\}$. Analogicky ako substitúciu vieme aj inverznú substitúciu rozšíriť na triedy jazykov.*

Cvičenie 3.4.2 *Ukážte, že každé trio je uzavreté na τ^{-1} pre ľubovoľnú substitúciu τ konečnými množinami.* (Hint: *použite A-prekladač)*

Definícia 3.4.3 *Trio $\mathscr{L}$ je* ***kvázi-úplné****, ak trio $\{L, L \cup \{\varepsilon\} \mid L \in \mathscr{L}\}$ je úplné.*

Príkladom tria, ktoré nie je ani kvázi-úplné, sú kontextové jazyky.

Cvičenie 3.4.3 *Ukážte, že každá kvázi-úplná AFL $\mathscr{L}$ uzavretá na operáciu $\cap$ je uzavretá aj na operáciu τ^{-1}, kde τ je substitúcia jazykmi z $\mathscr{L}$.*

Označenie $\mathbf{S}(\mathscr{L})$ resp. $\widehat{\mathbf{S}}(\mathscr{L})$ budeme označovať najmenšiu (úplnú) semi-AFL obsahujúcu $\mathscr{L}$.
$\mathbf{F}(\mathscr{L})$ resp. $\widehat{\mathbf{F}}(\mathscr{L})$ budeme označovať najmenšiu (úplnú) AFL obsahujúcu $\mathscr{L}$.
$\mathbf{H}(\mathscr{L})$ resp. $\widehat{\mathbf{H}}(\mathscr{L})$ je množina $\{h(L) \mid L \in \mathscr{L},\ h$ je (bezepsilonový) homomorfizmus $\}$ (teda je to najmenšia množina, obsahujúca $\mathscr{L}$ a uzavretá naň).

Lema 3.4.7 *Nech $\mathscr{L}$ je trieda jazykov, ktoré všetky obsahujú ε, potom $\mathbf{F}(\mathscr{L}) = \widehat{\mathrm{SUB}}(\mathscr{R}, \mathfrak{M}(\mathscr{L}))$ a $\widehat{\mathbf{F}}(\mathscr{L}) = \widehat{\mathrm{SUB}}(\mathscr{R}, \widehat{\mathfrak{M}}(\mathscr{L}))$.*

Veta 3.4.8 *Pre každú triedu jazykov $\mathscr{L}$ platí $\mathbf{F}(\mathscr{L}) = \widehat{\mathrm{SUB}}(\mathscr{R}_0, \mathfrak{M}(\mathscr{L}))$ a $\widehat{\mathbf{F}}(\mathscr{L}) = \widehat{\mathrm{SUB}}(\mathscr{R}_0, \widehat{\mathfrak{M}}(\mathscr{L})) = \widehat{\mathrm{SUB}}(\mathscr{R}, \widehat{\mathfrak{M}}(\mathscr{L}))$.*

Veta 3.4.9 *Pre každú triedu jazykov $\mathscr{L}$ platí:*
$\mathbf{S}(\mathscr{L}) = \{\bigcup_{i=1}^{n} M_i(L_i) \mid n \geq 1,\ L_i \in \mathscr{L},\ M_i$ *je bezepsilonový A-prekladač* $\}$
$\widehat{\mathbf{S}}(\mathscr{L}) = \{\bigcup_{i=1}^{n} M_i(L_i) \mid n \geq 1,\ L_i \in \mathscr{L},\ M_i$ *je A-prekladač* $\}$.

Lema 3.4.10 *Ak $\mathscr{L}$ je trieda jazykov uzavretých na ^{R}, potom aj $\mathbf{H}(\mathscr{L})$, $\widehat{\mathbf{H}}(\mathscr{L})$ sú uzavreté na ^{R}.*

Veta 3.4.11 *Pre každú triedu jazykov $\mathscr{L}$ platí*

- *Ak $\mathscr{L}$ je trio, potom $\widehat{\mathbf{H}}(\mathscr{L}) = \widehat{\mathrm{SUB}}(\mathscr{L}, \mathscr{R})$*
- *Ak $\mathscr{L}$ je trio, semi-AFL resp. AFL potom $\widehat{\mathbf{H}}(\mathscr{L}) = \widehat{\mathbf{O}}(\mathscr{L})$, kde $\mathbf{O}$ je $\mathfrak{M}, \mathbf{S}$ resp. $\mathbf{F}$.*

Cvičenie 3.4.4 *Ak trieda $\mathscr{L}$ je uzavretá na $\cap\mathscr{L}_{lin}$ je potom aj $\widehat{\mathbf{F}}(\mathscr{L})$ uzavretá na $\cap\mathscr{L}_{lin}$?*

3.5 Vytváranie základných operácií

Definícia 3.5.1 *Na triedach jazykov definujeme operáciu $\wedge$* ***(wedge)*** *nasledovne:*
$\mathscr{L}_1 \wedge \mathscr{L}_2 = \{L_1 \cap L_2 \mid L_i \in \mathscr{L}_i\}$

Pozorovanie. Skutočnosť, že triá sú uzavreté na prienik s regulárnym jazykom vieme teraz zapísať: pre všetky triá $\mathscr{L}$ platí $\mathscr{L} \wedge \mathscr{R} = \mathscr{L}$. Trieda $\mathscr{L}$ je uzavretá na prienik iff $\mathscr{L} \wedge \mathscr{L} \subseteq \mathscr{L}$.

Lema 3.5.1 *Bezkontextové jazyky nad jednopísmenovou abecedou sú regulárne.*

Dôkaz. Nad jednoprvkovou abecedou je podľa Parikhovej vety (2.2.1) množina dĺžok slov zjednotením konečného počtu aritmetických postupností. Ľahko zostrojíme konečný automat, ktorý bude takýto jazyk akceptovať.

Veta 3.5.2 *Nech $\mathscr{L}_1$, $\mathscr{L}_2$ sú triá. Potom $\mathscr{L}_1 \wedge \mathscr{L}_2$* **nemusí** *byť trio.*

Dôkaz. $\mathscr{L}_1 \wedge \mathscr{L}_2$ nemusí byť uzavretá na h_ε. Položme $\mathscr{L}_1 = \mathscr{L}_2 = \mathscr{L}_{CF}$. Zoberme teraz dva bezkontextové jazyky, ktorých prienik je dosť zložitý. Keď tento prienik zobrazíme homomorfizmom, ktorý všetko prepisuje na a, dostaneme jazyk, ktorý je nad abecedou $\{a\}$ a nie je bezkontextový.

Tento jazyk ale nepatrí do $\mathscr{L}_1 \wedge \mathscr{L}_2$. Musel by totiž byť prienikom nejakých dvoch bezkontextových jazykov nad abecedou $\{a\}$. Bezkontextové jazyky nad jednopísmenkovou abecedou sú však podľa lemy 3.5.1 regulárne, a teda uzavreté na prienik. Preto náš nebezkontextový jazyk takto nevyrobíme.

Konkrétny príklad: Nech $L_p = \{a^iba^{i+2}b \mid i \geq 0\}$, $L_1 = bL_p^*$ $L_2 = L_p^*a^*b$. Zjavne $L_1, L_2 \in \mathscr{L}_{CF}$, ale $L_1 \cap L_2 = \{a^0ba^2ba^4b \dots a^{4k-2}ba^{4k}b \mid k > 0\}$ a keď zoberieme vyššie spomínaný homomorfizmus h, dostávame $h(L) = \{a^{(2k+1)^2}\}$. Tento jazyk zjavne nie je bezkontextový, a teda podľa vyššie uvedenej úvahy neleží v $\mathscr{L}_1 \wedge \mathscr{L}_2$

Lema 3.5.3 *Ak $\mathscr{L}_1$, $\mathscr{L}_2$ sú trio, semi-AFL, AFL, potom $\mathbf{H}(\mathscr{L}_1 \wedge \mathscr{L}_2)$ je trio, semi-AFL, AFL.*

Dôkaz. Zrejme $\mathbf{H}(\mathscr{L}_1 \wedge \mathscr{L}_2)$ je uzavretá na h_ε. Nie je ťažké overiť, že $h(L_1 \cap L_2) \cap R = h(L_1 \cap h^{-1}(R) \cap L_2)$. Aby sme tvrdenie dokázali pre triá, zostáva nám dokázať, že pre ľubovoľné homomorfizmy h a h_1 a ľubovoľné jazyky L_1 a L_2 existujú jazyky L_3 a L_4 a homomorfizmus h_2 taký, že $h_1^{-1}(h(L_1 \cap L_2)) = h_2(L_3 \cap L_4)$.

Nech Σ_u, Σ_v sú také, že $h : \Sigma_{L_1,L_2}^* \to \Sigma_u^*$, $h_1 : \Sigma_v^* \to \Sigma_u^*$.

Definujme $\Sigma_w = \Sigma_v \times \Sigma_{L_1,L_2}^{\leq k}$ pre nejaké vhodné k. Definujme:

$$R = \Big\{[b_1, x_1] \dots [b_n, x_n] \in \Sigma_w^* \;\Big|\; h_1(b_1 \dots b_n) = h(x_1 \dots x_n),$$
$$\forall i \exists y_i;\ |y_i| \leq k \ \wedge\ h_1(b_1 \dots b_i) = h(x_1 \dots x_i)y_i \Big\}$$

Inými slovami, R je jazyk nad Σ_w taký, že prvé zložky homomorfované h_1 sa rovnajú druhým zložkám homomorfovaným h a navyše rozdiel homomorfovaných dĺžok nikdy nie je príliš veľký. Jazyk R je regulárny.

Nie je ťažké ukázať, že $h_1^{-1}(h(L_1 \cap L_2)) = pr_1(pr_2^{-1}(L_1 \cap L_2) \cap R) = pr_1(pr_2^{-1}(L_1) \cap (pr_2^{-1}(L_2) \cap R))$. Podmienka na „malé pretŕčanie“ v jazyku R nám platnosť tvrdenia nepokazí, pretože v R vždy určite existuje vhodné slovo[5].

Dokázali sme teda platnosť tvrdenia pre triá. Ak by sme používali $\widehat{\mathbf{H}}$ namiesto $\mathbf{H}$, získali by sme analogické tvrdenie pre úplne triá. Rozšírenie pre semi-AFL je jednoduché. Pre homomorfizmy h_1 a h_2 a jazyky L_1, L_2, L_3 a L_4 potrebujeme nájsť homomorfizmus h_3 a jazyky L_5 a L_6 také, že $h_1(L_1 \cap L_2) \cup h_2(L_3 \cap L_4) = h_3(L_5 \cap L_6)$. Použitím čiarkovacieho homomorfizmu h' môžme definovať $L_5 = L_1 \cup h'(L_3)$ a $L_6 = L_2 \cup h'(L_4)$. Homomorfizmus h_3 bude odčiarkovávací homomorfizmus.

Pre AFL operácie sa tvrdenie dokáže analogicky.

Veta 3.5.4 *Pre každé trio $\mathscr{L}_1$ a každú AFL $\mathscr{L}_2$ platí: $\widehat{\mathrm{SUB}}(\mathscr{L}_1, \mathscr{L}_2) \subseteq \widehat{\mathbf{H}}(\mathscr{L}_1 \wedge \mathscr{L}_2)$*

Dôkaz. Uvažujme homomorfizmus h_1 taký, že $h_1(a') = a$, $h_1(a) = \varepsilon$. Nech $L \in \mathscr{L}_1$, potom $h_1^{-1}(L) \in \mathscr{L}_1$. Nech $\tau(a)$ je substitúcia jazykmi z $\mathscr{L}_2$, potom $(\bigcup_a a'\tau(a))^+ \in \mathscr{L}_2$.

Nech h_2 je homomorfizmus taký, že $h_2(a') = \varepsilon$, $h_2(a) = a$). Je vidno, že $\tau(L) = h_2(h_1^{-1}(L) \cap (\bigcup_a a'\tau(a))^+) \in \widehat{\mathbf{H}}(\mathscr{L}_1 \wedge \mathscr{L}_2)$.

Dôsledok 3.5.5 *Úplná AFL uzavretá na prienik je uzavretá na substitúciu.*

Cvičenie 3.5.1 *Nech $\mathscr{L}$ je úplná AFL. Dokážte, že ak $\mathscr{L} \wedge \mathscr{L}_{lin} \subseteq \mathscr{L}$, potom $\mathscr{L} \wedge \mathscr{L} \subseteq \mathscr{L}$ (teda $\mathscr{L}$ je uzavretá na operáciu wedge).*

Veta 3.5.6 *Každá trieda jazykov $\mathscr{L}$ uzavretá na inverzný homomorfizmus, prienik s regulárnym jazykom, a dĺžku zachovávajúci homomorfizmus je trio.*

[5]Druhá zložka písmen zo Σ_w môže byť aj nulová.

Dôkaz. Treba dokázať, že trieda jazykov $\mathscr{L}$ je uzavretá na nevymazávajúci homomorfizmus. Nech h je nevymazávajúci homomorfizmus, teda nech $h(a) = b_a w_a$. Definujme $h_1(a) = b'_a$, $h_2(b') = b'$, $h_2(c) = \varepsilon$, $h_3(b') = b$, $h_3(b) = b$. Jazyk $R = (\bigcup_a b'_a w_a)^+$ je zjavne regulárny. Pre každý jazyk $L \in \mathscr{L}$ platí $h(L) = h_3(h_2^{-1}(h_1(L)) \cap R) \in \mathscr{L}$. Hotovo.

Lema 3.5.7 *Ak $\mathscr{L}$ je trieda bezepsilonových jazykov uzavretá na $\cdot, h_\varepsilon, h^{-1}$ potom $\{c\} \in \mathscr{L}$ pre každé $c \in \Sigma$.*

Dôkaz. Keďže $\mathscr{L}$ je trieda jazykov, existuje $L \neq \emptyset \in \mathscr{L}$ teda existuje $w \in L,\ w \neq \varepsilon$. Označme si písmená abecedy a takto označkovaný jazyk[6] zreťazme s L, nech $L_1 = L \cdot \overline{L} \in \mathscr{L}$. Platí $w\overline{w} \in L_1$ a žiadna mocnina slova $w\overline{w}$ nepatrí do L_1. Na tento jazyk aplikujeme h_2^{-1} taký, že $h_2(c) = w\overline{w}$. Potom $h_2^{-1}(L \cdot \overline{L}) = \{c\}$.

Lema 3.5.8 *Nech $\mathscr{L}$ je trieda jazykov uzavretá na h_ε, h^{-1}. Potom ak $cLd \in \mathscr{L}$, kde $c, d \notin \Sigma_L$ a L je bezepsilonový, potom $L \in \mathscr{L}$.*

Dôkaz. Ošetríme najprv špeciálny prípad $L = \emptyset$. Potom $cLd = \emptyset$ čo patrí do $\mathscr{L}$.
Zadefinujme $\Sigma_1 = \{[c,a] \mid a \in \Sigma_L\} \cup \{[a,d] \mid a \in \Sigma_L\} \cup \{[c,a,d] \mid a \in \Sigma_L\} \cup \Sigma_L$ a homomorfizmus $h_1 : \Sigma_1^* \to (\Sigma_L \cup \{c,d\})^*$ takto $h_1(a) = a,\ h_1([c,a]) = ca,\ h_1([a,d]) = ad,\ h_1([c,a,d]) = cad$. Zrejme $h_1^{-1}(cLd)$ spojí krajné písmená so znakmi c, d. Teraz použijeme druhý homomorfizmus h_2 definovaný $h_2(a) = h_2([c,a]) = h_2([a,d]) = h_2([c,a,d]) = a$ na odmazanie znakov.
Platí $h_2(h_1^{-1}(cLd)) = L$.

Lema 3.5.9 *Nech $\mathscr{L}$ je trieda bezepsilonových jazykov uzavretá na $\cdot, h_\varepsilon, h^{-1}$. Potom $\mathscr{L}$ je uzavretá na $\cap R$ (teda $\mathscr{L}$ je trio).*

Dôkaz. Potrebujeme ukázať, že ak $L \in \mathscr{L}$ a $R \in \mathscr{R}$ (nech A je taký DFA, že $L(A) = R$), potom $L \cap R \in \mathscr{L}$.
Zoberme si jazyk cRd. Ten je akceptovaný DFA A', ktorý vznikol z A pridaním dvoch stavov q'_0, q'_F a niekoľkých vhodných hrán. Nech $h_1 : \Sigma_1 \to \Sigma_L \cup \{c,d\}$, kde $\Sigma_1 = \{[q,a,p] \mid p,q \in K,\ a \in \Sigma_L \cup \{c,d\}\}$ je homomorfizmus definovaný takto: $h_1([q'_0, c, q_0]) = c$, $h_1([q,d,q'_F]) = d$ pre všetky $q \in F$ a $h_1([q,a,p]) = a$ pre všetky $a \in \Sigma_L$ a $p, q \in K$.
Potom jazyk $L_1 = h_1^{-1}(cLd)$ je jazyk, ktorý obsahuje slová tvaru $[q'_0, c, q_0][q_{x_1}, a_{y_1}, q_{z_1}] \dots [q, d, q'_F]$, v ktorých zreťazenie a_{y_i} tvorí slovo z L. My chceme vybrať práve tie z nich, ktoré predstavujú akceptačné výpočty.

Pre jednoduchší zápis použijeme v ďalšom texte štandardné značenie $\widehat{\delta}$, ktoré predstavuje funkciu definovanú nasledovne: $\widehat{\delta}(q,a) = \delta(q,a)$, pre $n > 1$ je $\widehat{\delta}(q, w_1 \dots w_n) = \widehat{\delta}(\delta(q, w_1), w_2 \dots w_n)$

Dvomi homomorfizmami budeme chcieť kontrolovať následnosť stavov. Definujme abecedu
$$\Sigma_2 = \{[q, a, \delta(q,a), b, \widehat{\delta}(q,ab)] \mid a, b \in \Sigma_L \cup \{c,d\}\}$$
$$\cup \{[q, a, \delta(q,a), b, \widehat{\delta}(q,ab), d, \widehat{\delta}(q,abd)] \mid a, b \in \Sigma_L \cup \{c\}\}.$$
Symboly tejto abecedy teda zodpovedajú ľubovoľným dvom, resp. posledným trom krokom výpočtu. Homomorfizmus $h_2 : \Sigma_2^* \to \Sigma_1^*$ je homomorfizmus, ktorý prirodzeným spôsobom rozbije slová nad Σ_2 na zodpovedajúce slová nad Σ_1.

Podobne definujme abecedu $\Sigma_3 = \{[q, a, \delta(q,a), b, \widehat{\delta}(q,ab)] \mid a, b \in \Sigma_L \cup \{d\}\} \cup \{[q'_0, c, q_0]\} \cup \{[q, d, q'_F] \mid q \in F\}$. Symboly tejto abecedy zodpovedajú prvému kroku výpočtu, poslednému kroku výpočtu a ľubovoľným dvom krokom inde ako na začiatku výpočtu. Tiež definujeme homomorfizmus $h_3 : \Sigma_3^* \to \Sigma_1^*$ podobne ako h_2.

Potom jazyk $L_2 = h_3 h_3^{-1} h_2 h_2^{-1}(L_1)$ je jazyk, ktorý popisuje platný výpočet v automate A' na slove z cLd. To znamená, že toto slovo je aj z cRd. Teda $L_3 = h_1(L_2) = c(L \cap R)d$. V $\mathscr{L}$ teda máme jazyk $c(L \cap R)d$, potom ale podľa lemy 3.5.8 aj $L \cap R \in \mathscr{L}$ (keďže L a teda aj $L \cap R$ je bezepsilonový).

Veta 3.5.10 *Nech $\mathscr{L}$ je trieda bezepsilonových jazykov uzavretá na $\cdot, h_\varepsilon, h^{-1}$. Potom $\mathscr{L}$ je uzavretá na $\cap R, \cup$ (teda $\mathscr{L}$ je semi-AFL).*

[6] $\overline{L} = h_1(L)$, kde $h_1(a) = \overline{a}\ \forall a \in \Sigma$

Dôkaz. Uzavretosť $\mathscr{L}$ na $\cap R$ vieme z lemy 3.5.9. Potrebujeme ešte ukázať, že pre $L_1, L_2 \in \mathscr{L}$ platí, že $L_1 \cup L_2 \in \mathscr{L}$. Ak je niektorý z nich prázdny, tvrdenie triviálne platí. V opačnom prípade nech $w_i \in L_i$. Štandardým spôsobom zostrojíme k jazykom L_1, L_2 ich očiarkované verzie L_1', L_2''. Potom $L_3 = (L_1' \cdot L_2'') \cap (w_1' \cdot \Sigma^*_{L_2''} \cup \Sigma^*_{L_1'} \cdot w_2'') = L_1' \cdot w_2'' \cup w_1' \cdot L_2''$. Nech h_1 je homomorfizmus definovaný nasledovne $h_1(\overline{a}') = a'w_2''$, $h_1(a') = a'$, $h_1(\overline{a}'') = w_1'a''$ a $h_1(a'') = a''$. Ďalej nech h_2 je homomorfizmus definovaný nasledovne $h_2(\overline{a}') = h_2(a') = h_2(\overline{a}'') = h_2(a'') = a$. Potom $h_2(h_1^{-1}(L_3)) = L_1 \cup L_2$.

Dôsledok 3.5.11 *Nech $\mathscr{L}$ je trieda jazykov uzavretá na $\cdot, h_\varepsilon, h^{-1}, +, \setminus\{\varepsilon\}, \cup\{\varepsilon\}$. Potom $\mathscr{L}$ je AFL.*

Dôkaz. Podobne ako vo vete 3.5.10 potrebujeme dokázať uzavretosť $\mathscr{L}$ na $\cap R$ a $\cup$. Už vieme, že pre bezepsilonové jazyky uzavretosť platí. Pre jazyky obsahujúce ε využijeme uzavretosť $\mathscr{L}$ na $\setminus\{\varepsilon\}, \cup\{\varepsilon\}$ – vieme z nich ε odstrániť, spraviť príslušnú operáciu (už s bezepsilonovými jazykmi, takže výsledok bude z $\mathscr{L}$) a v prípade potreby do výsledku ε pridať.

Definícia 3.5.2 ***Hviezdna substitúcia*** *τ na L je substitúcia taká, že existuje $c \in \Sigma_L$ také, že $\tau(a) = c^*ac^*$ pre všetky $a \in \Sigma_L$.*

Definícia 3.5.3 *Homomorfizmus h je* ***ohraničené vymazávanie symbolu*** *na L, ak pre nejaké $k \geq 0$ platí $L \subseteq (\Sigma_1\{c^i \mid 0 \leq i \leq k\})^*$, kde $c \notin \Sigma_1$ a $h(c) = \varepsilon$, $h(a) = a$ pre všetky $a \in \Sigma_1$.*

Veta 3.5.12 *Nech $\mathscr{L}$ je trieda jazykov uzavretá na hviezdnu substitúciu, ohraničené vymazávanie symbolu, dĺžku zachovávajúci h a $\cap R$. Ak $\mathscr{L}$ je uzavretá na $\cup R$, alebo $\mathscr{L}$ je bezepsilonová, potom $\mathscr{L}$ je uzavretá aj na h^{-1}.*

Dôsledok 3.5.13 *Ak je trieda jazykov $\mathscr{L}$ uzavretá na $\cup R, \cap R$ a substitúciu regulárnymi množinami, potom $\mathscr{L}$ je uzavretá aj na h^{-1}.*

Kapitola 4

Abstraktné triedy automatov

V tejto kapitole sa budeme zaoberať automatmi, konkrétne nedeterministickými a jednosmernými. Ako fungujú, v čom sa líšia a čo je vlastne podstatné. Vieme, že každý typ automatu, s ktorým sme sa stretli mal vstup, konečnostavovú riadiacu hlavu a nejakú pamäť. Zistíme, že trieda automatov je daná práve spôsobom práce s touto pamäťou.

4.1 Definície

Definícia 4.1.1 ***AFA-schéma*** *je štvorica* (Γ, I, f, g), *kde* Γ *je abeceda pracovných symbolov (môže byť aj nekonečná),* I *je množina symbolov – inštrukcií (môže byť aj nekonečná) a* f, g *sú funkcie pracujúce s pamäťou, pričom* $f : \Gamma^* \times I \to \Gamma^* \cup \{\emptyset\}$ *mení pamäť (zapisuje do nej) a* $g : \Gamma^* \to 2^{\Gamma^*}$ *je čítacia funkcia,*[1] *ktorá na základe obsahu pamäte vráti nejaké informácie o nej. Musia byť tiež splnené podmienky:*

1. $g(\varepsilon) = \{\varepsilon\}$ *a* $\varepsilon \in g(\gamma) \iff \gamma = \varepsilon$
2. *pre každé* $\gamma \in \Gamma^*$ *existuje prvok* $1_\gamma \in I$ *taký, že* $f(\gamma', 1_\gamma) = \gamma'$ *pre všetky* $\gamma' : \gamma \in g(\gamma')$, *t.j. musí existovať pre každý stav pamäte inštrukcia, ktorá tento stav nezmení.*
3. *pre každé* $u \in I$ *existuje konečná množina* $\Gamma_u \subseteq \Gamma$ *taká, že ak* $\gamma \in \Gamma_1^*$ *a* $f(\gamma, u) \neq \emptyset$ *potom* $f(\gamma, u) \in (\Gamma_1 \cup \Gamma_u)^*$, *teda každá inštrukcia môže pridávať iba konečne veľa rôznych písmen.*

Definícia 4.1.2 ***Abstraktná trieda*** *nedeterm. jednosmerných automatov je dvojica* $(\Omega, \mathscr{D})$, *kde*

1. $\Omega = (K, \Sigma, \Gamma, I, f, g)$, *kde* K *je nekonečná množina stavov,* Σ *je nekonečná abeceda vstupných symbolov*[2] *a* (Γ, I, f, g) *je AFA-schéma*
2. $\mathscr{D}$ *je trieda všetkých automatov* $D = (K_D, \Sigma_D, \delta, q_0, F)$ *takých, že*
   - (a) K_D, Σ_D *sú konečné podmnožiny* K, Σ, q_0 *je počiatočný stav,* $F \subseteq K_D$ *je množina akceptačných stavov*
   - (b) $\delta : K_D \times (\Sigma_D \cup \{\varepsilon\}) \times g(\Gamma^*) \to 2^{K_D \times I}_{kon}$ *je prechodová funkcia automatu taká, že množina* $G_D = \{\gamma \in g(\Gamma^*) \mid \delta(q, a, \gamma) \neq \emptyset,\ q \in K_D,\ a \in \Sigma_D \cup \{\varepsilon\}\}$ *je konečná (alebo ekvivalentne neprázdna časť* δ*-fcie je konečná)*

Definícia 4.1.3 ***Konfigurácia automatu*** D *je prvok z* $(K_D \times \Sigma_D^* \times \Gamma^*)$.

Definícia 4.1.4 ***Krok výpočtu*** *je relácia* $\vdash_D$ *definovaná na konfiguráciách takto:* $(q, aw, \gamma) \vdash_D (p, w, \gamma')$ *práve vtedy, ak existuje* $\gamma'' \in g(\gamma)$ *a* $u \in I$ *také, že* $(p, u) \in \delta(q, a, \gamma'')$, *kde* $\gamma' = f(\gamma, u)$.

Definícia 4.1.5 ***Jazyk akceptovaný automatom*** D *je*
$L(D) = \{w \in \Sigma_D^* \mid (q_0, w, \varepsilon) \vdash_D^* (q, \varepsilon, \varepsilon),\ q \in F\}$.

---

[1] Vie sa o aspoň jednom type automatov (Listautomata), kde je pri čítaní z pamäte nedeterminizmus potrebný. Ide o automaty, ktoré sa nedeterministicky rozhodnú, aký veľký úsek pamäti vrátia

[2] kvôli možnosti pridávania nových symbolov a stavov

Definícia 4.1.6 ***Jazyk akceptovaný akceptačným stavom automatom*** *D je* $L_f(D) = \{w \in \Sigma_D^* \mid (q_0, w, \varepsilon) \vdash_D^* (q, \varepsilon, \gamma),\ q \in F\}$.

Definícia 4.1.7 *Definujme reláciu $\vdash_D^-$ na konfiguráciách takto: $(q, w, \gamma) \vdash_D^- (p, w', \gamma')$ práve vtedy, ak $(q, w, \gamma) = (p, w', \gamma')$ alebo $(q, w, \gamma) \vdash_D (p, w', \gamma')$ alebo existujú $n \geq 1$ a (p_i, w_i, γ_i), $\gamma_i \neq \varepsilon$ (pre $i = 1 \dots n$) také, že $(q, w, \gamma) \vdash_D (p_1, w_1, \gamma_1) \vdash_D \dots \vdash_D (p_n, w_n, \gamma_n) \vdash_D (p, w', \gamma')$.*

Neformálne táto relácia predstavuje výpočet, počas ktorého nie je nikdy prázdna pamäť.

Definícia 4.1.8 ***Jazyk akceptovaný prázdnou pamäťou automatom*** *D je* $\widetilde{L}(D) = \{w \in \Sigma_D^* \mid (q_0, w, \varepsilon) \vdash_D^- (q, \varepsilon, \varepsilon),\ q \in K_D\}$.

Označenie. $\mathscr{L}(\mathscr{D}) = \{L(D) \mid D \in \mathscr{D}\}$, $\mathscr{L}_f(\mathscr{D}) = \{L_f(D) \mid D \in \mathscr{D}\}$ a $\widetilde{\mathscr{L}}(\mathscr{D}) = \{\widetilde{L}(D) \mid D \in \mathscr{D}\}$. (Teda napr. $\mathscr{L}(\mathscr{D})$ je trieda jazykov, akceptovaných automatmi z $\mathscr{D}$.)

Príklady Skúsime si zadefinovať niektoré známe triedy automatov.

1. $\Omega_{PDA} = (K, \Sigma, \Gamma, I, f, g)$, kde Γ obsahuje aspoň dva symboly, $I = \Gamma^*$ (prípadne $I = \{i_w \mid w \in \Gamma^*\}$), $g(\varepsilon) = \varepsilon$, $g(yZ) = \{Z\}$, $f(\varepsilon, u) = u$ a $f(yZ, u) = yu$.
   Sú splnené podmienky z definície: $1_\gamma = \gamma$ a inštrukcie nemôžu pridávať nekonečne veľa znakov $\Gamma_u = u$.
   Táto trieda sa dá upraviť na konečnú triedu. Na ekvivalentnú silu stačí $|\Gamma| = 2$, lebo znaky môžeme zakódovať a $I = \{i_\varepsilon, i_Z, i_ZX \mid Z, X \in \Gamma\}$ čo je tiež zrejmé.
   Tiež vieme, že $\mathscr{L}(\mathscr{D}) = \mathscr{L}_f(\mathscr{D}) = \widetilde{\mathscr{L}}(\mathscr{D}) = \mathscr{L}_{CF}$.
2. Nech $\Gamma = \{Z\}$, $I = \{0\}$, $f(\gamma, 0) = \gamma$ pre všetky $\gamma \in \Gamma^*$, $g(\varepsilon) = \varepsilon$ a $g(Z^n) = \emptyset$.
   Zrejme sú splnené podmienky definície $1_\gamma = 0$ pre všetky $\gamma \in \Gamma^*$ a $\Gamma_u = \emptyset$.
   Takto vytvorená trieda zodpovedá triede konečných automatov Ω_{FSA}, lebo na začiatku máme prázdnu pamäť a už to nemám čím zmeniť.
3. Nech $\Gamma = \{Z\}$, $I = Z^*$, $f(\varepsilon, Z^k) = Z^k$, $f(Z^l, Z^k) = Z^{l+k-1}$, $g(\varepsilon) = \varepsilon$ a $g(Z^n) = Z$.
   Zrejme sú splnené podmienky definície $1_{Z^n} = Z^n$, $1_\varepsilon = \varepsilon$ a $\Gamma_u = Z$.
   Dostávame triedu počítadlových automatov Ω_C.

Cvičenie 4.1.1 *Skúste si podobne zadefinovať aj Turingove stroje. Nezabudnite pridať špeciálny symbol $\mathcal{B}$.*

4.2 Vzťah AFL a AFA

Lema 4.2.1 *Pre každú AFA $\mathscr{D}$ platí $\widehat{\mathrm{SUB}}(\mathscr{R}_0, \mathscr{L}(\mathscr{D})) \subseteq \mathscr{L}(\mathscr{D})$, tj pre každé $R \in \mathscr{R}_0$ a τ také, že $\tau(a) \in \mathscr{L}\mathscr{D}$ pre každé a, platí $\tau(R) \in \mathscr{L}(\mathscr{D})$.*

Dôsledok 4.2.2 *Trieda $\mathscr{L}(\mathscr{D})$ je uzavretá na operácie $\cup, \cdot, +$.*

Lema 4.2.3 *Trieda $\mathscr{L}(\mathscr{D})$ je uzavretá na $\cap R$.*

Lema 4.2.4 *Trieda $\mathscr{L}(\mathscr{D})$ je uzavretá na h^{-1}.*

Lema 4.2.5 *Trieda $\mathscr{L}(\mathscr{D})$ je uzavretá na h.*

Dôsledok 4.2.6 *Trieda $\mathscr{L}(\mathscr{D})$ je úplná AFL pre každú AFA $\mathscr{D}$.*

Dôkaz. Vyplýva bezprostredne z dôsledku 4.2.2 a z liem 4.2.3, 4.2.4 a 4.2.5.

4.3 Vzťah prieniku jazykov a viacpáskových AFA

Definícia 4.3.1 *Nech $(\mathscr{B}, \prec_{\mathscr{B}})$ je usporiadaná (konečná?) množina. Nech $\{\Omega_\beta, \mathscr{D}_\beta \mid \beta \in \mathscr{B}\}$ je množina viacpáskových AFA, kde $\Omega_\beta = (K, \Sigma, \mathcal{A}_\beta, \prec_\beta, \mu_\beta)$, pričom $\mathcal{A}_\beta$ je množina indexov, $\prec_\beta$ usporiadanie na nej a μ_β indexovaná množina AFA-schém.*

Potom $\bigwedge_{\beta\in\mathscr{B}} \mathscr{D}_\beta$ bude označovať viacpáskovú AFA $(\Omega, \bigwedge_{\beta\in\mathscr{B}} \mathscr{D}_\beta)$, kde $\Omega = (K, \Sigma, \mathcal{A}, \prec, \mu)$, pričom $\mathcal{A} = \bigcup_{\beta\in\mathscr{B}} \mathcal{A}_\beta \times \{\beta\}$, $\mu(\alpha, \beta) = \mu_\beta(\alpha)$ a $(\alpha, \beta) \prec (\alpha', \beta')$ iff ($\beta = \beta' \wedge \alpha \prec_\beta \alpha'$ alebo $\beta \prec_{\mathscr{B}} \beta'$).

Neformálne: automat z výslednej triedy dostaneme tak, že zoberieme vhodné automaty z pôvodných tried a dáme mu všetky ich pamäte (pásky). Tie budú fungovať presne rovnako ako fungovali v pôvodných automatoch. Zvyšok konštrukcie je len o tom, aby boli aj pásky výsledného automatu nejako zoradené a pod.

Lema 4.3.1
$\mathscr{L}(\mathscr{D}_1) \wedge \mathscr{L}(\mathscr{D}_2) \subseteq \mathscr{L}(\mathscr{D}_1 \wedge \mathscr{D}_2)$
$\mathscr{L}^t(\mathscr{D}_1) \wedge \mathscr{L}^t(\mathscr{D}_2) \subseteq \mathscr{L}^t(\mathscr{D}_1 \wedge \mathscr{D}_2)$

Dôkaz. Nech $L_1 = L(D_1)$, $L_2 = L(D_2)$, kde $D_i \in \mathscr{D}_i$ sú nejaké automaty z príslušných tried. Triviálne (kartézsky súčin stavov, zjednotenie pások) k nim zostrojíme automat $D_3 \in \mathscr{D}_3$ taký, že $L(D_3) = L(D_1) \cap L(D_2)$.

Ak sú navyše D_1, D_2 quasi-realtime s konštantami k, l, takto zostrojený D_3 je tiež quasi-realtime s konštantou $k + l$.

Lema 4.3.2
$\mathbf{H}(\mathscr{L}^t(\mathscr{D}_1) \wedge \mathscr{L}^t(\mathscr{D}_2)) \ = \ \mathscr{L}^t(\mathscr{D}_1 \wedge \mathscr{D}_2)$
$\widehat{\mathbf{H}}(\mathscr{L}(\mathscr{D}_1) \wedge \mathscr{L}(\mathscr{D}_2)) \ = \ \widehat{\mathbf{H}}(\mathscr{L}^t(\mathscr{D}_1) \wedge \mathscr{L}^t(\mathscr{D}_2)) \ = \ \mathscr{L}(\mathscr{D}_1 \wedge \mathscr{D}_2)$

Dôkaz. V prvom tvrdení inklúzia $\subseteq$ vyplýva z lemy 4.3.1.

V druhom tvrdení pri dôkaze prvej rovnosti môžeme použiť rovnaký trik ako pri dôkaze $\widehat{\mathbf{H}}(NTIME(n)) = \mathscr{L}_{RE}$ – pridáme kopu symbolov navyše, tie následne homomorfizmom umažeme.

Pozrime sa na dôkaz poslednej rovnosti. Problém je s inklúziou $\supseteq$, opačná opäť vyplýva z lemy 4.3.1. Majme automat $D_3 \in \mathscr{D}_1 \wedge \mathscr{D}_2$. Spomínaný problém je v tom, že tento automat nemôžeme jednoducho rozbiť na tie dva, z ktorých vznikol. Totiž mohli by existovať slová, ktoré každý z nich (nedeterministicky) vie akceptovať, ale ktoré D_3, ktorý overuje viac, akceptovať nevie.

Nech $D_3 = (K, \Sigma, \delta, q_0, F, \nu)$, kde $\nu = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_l)$. Pre všetky $q, a, \gamma_1, \ldots, \gamma_{k+l}$ také, že $|\delta(q, a, \gamma_1, \ldots, \gamma_{k+l})| \geq 1$ si jednotlivé možnosti očíslujeme.

Nech teraz $[q, a, \gamma_1, \ldots, \gamma_{k+l}, m]$ (kde m je od 1 do počtu príslušných možností) sú nové symboly v Σ. Naše automaty D_1, D_2 budú spracúvať slová nad touto zložitou abecedou. Budeme chcieť overiť, či dané slovo nad touto abecedou zodpovedá akceptačnému výpočtu automatu D_3. Pritom automat D_1 bude pomocou svojich k pások overovať, či je tento výpočet konzistentný na prvých k páskach, D_2 bude podobne overovať zvyšné pásky.

Každé slovo akceptované D_1 aj D_2 teda zodpovedá nejakému akceptačnému výpočtu D_3. Teraz už len potrebujeme vhodným homomorfizmom prepísať naše komplikované symboly na príslušné písmená čítané zo vstupu a sme hotoví.

K opačnej inklúzií v prvom tvrdení: pri dôkaze použijeme presne rovnakú úvahu, iba na konci musíme použiť vhodne (ako?) obmedzene vymazávajúci homomorfizmus.

Veta 4.3.3 *Nech $\mathscr{D}_1, \ldots, \mathscr{D}_n$ sú viacpáskové AFA, potom platia analogické tvrdenia ako v leme 4.3.1 aj pre wedge n tried automatov.*

Triedu automatov s k počítadlami budeme označovať $\mathscr{D}_{kCFA}$, triedu zásobníkových automatov $\mathscr{D}_{PDA}$. Zjavne keď $\mathscr{D}_1 = \ldots = \mathscr{D}_n = \mathscr{D}_{1CFA}$, tak $\bigwedge \mathscr{D}_i = \mathscr{D}_{nCFA}$.

Veta 4.3.4 $\mathscr{L}(\mathscr{D}_{2CFA}) = \mathscr{L}_{RE}$

Dôkaz. Táto veta síce príliš nesúvisí s aktuálnou témou, ale ide o jeden z klasických výsledkov teórie formálnych jazykov. Triviálne vieme simulovať nekonečnú pásku Turingovho stroja pomocou dvoch zásobníkov. Podobne vieme simulovať zásobník pomocou dvoch countrov – jeho obsah zakódujeme do čísla v $|\Gamma|$-árnej sústave (kde Γ je zásobníková abeceda), pritom vrch zásobníka je najmenej významná cifra. Vkladanie a výber zo zásobníka zodpovedajú násobeniu, sčítaniu a deleniu so zvyškom, ktoré vieme za pomoci druhého countra triviálne robiť.

Tým sme dokázali $\mathscr{L}(\mathscr{D}_{4CFA}) = \mathscr{L}_{RE}$. Teraz si už stačí uvedomiť, že obsah konečného počtu countrov vieme zakódovať do jedného. V našom prípade budeme napr. namiesto 4 countrov s hodnotami c_i mať v jednom countri číslo $2^{c_1}3^{c_2}5^{c_3}7^{c_4}$ a druhý counter bude slúžiť na to, aby sme s prvým vedeli robiť potrebné aritmetické operácie.

Veta 4.3.5 *Nech $\mathscr{D}_1 = \ldots = \mathscr{D}_n = \mathscr{D}_{PDA}$. Potom $\mathscr{L}^t(\bigwedge \mathscr{D}_i) = \mathscr{L}^t(\mathscr{D}_{PDA} \wedge \mathscr{D}_{PDA} \wedge \mathscr{D}_{PDA})$.*

Poznámka. Otvorený problém je či v predchádzajúcej vete stačia dva zásobníky.

Definícia 4.3.2 *Hovoríme, že AFA $(\Omega, \mathscr{D})$, kde $\Omega = (K, \Sigma, \Gamma, I, f, g)$, je **reverzibilná**, ak $\forall u \in I, \xi \in g(\Gamma^*)\ \exists v_{u,\xi} \in I$ taká, že $\forall \gamma, \gamma'$ ak $\xi \in g(\gamma)$ a $f(\gamma, u) = \gamma'$, tak $f(\gamma', v_{u,\xi}) = \gamma$.*

Neformálne povedané požadujeme, aby ku každej inštrukcii existovala „inverzná“, ktorá bude závisieť len od aktuálnej hodnoty funkcie g a bude vracať pamäť späť do súčasného stavu.

Veta 4.3.6 *Pre všetky reverzibilné AFA $\mathscr{D}$ sú triedy $\mathscr{L}(\mathscr{D})$ a $\mathscr{L}^t(\mathscr{D})$ uzavreté na reverz.*

4.4 Hlavné triedy jazykov

Definícia 4.4.1 *Hovoríme, že trieda jazykov $\mathscr{L}$ je **hlavná** AFL, ak existuje jazyk L taký, že $\mathscr{L} = \mathbf{F}(L)$. Analogicky ak $\mathscr{L} = \widehat{\mathbf{F}}(L)$, hovoríme, že $\mathscr{L}$ je úplná hlavná AFL. Podobne definujeme hlavnú semi-AFL a hlavné trio.*

Pozorovanie. Z vety 2.1.2 vyplýva, že $\mathscr{L}_{CF}$ je úplná hlavná AFL.

Pozorovanie. $\mathscr{R}$ je úplná hlavná AFL generovaná ľubovoľným neprázdnym regulárnym jazykom. $\mathscr{R}_0$ je hlavná AFL generovaná ľubovoľným neprázdnym bezepsilonovým regulárnym jazykom.

Lema 4.4.1 *Nech L_1, L_2 sú jazyky, nech $c \notin \Sigma_{L_1} \cup \Sigma_{L_2}$, nech $\mathbf{O} \in \{\mathbf{S}, \mathbf{F}\}$. Potom:*
$\widehat{\mathbf{O}}(\{L_1, L_2\}) = \widehat{\mathbf{O}}(L_1 \cup cL_2)$
$\mathbf{O}(\{L_1, L_2\}) = \mathbf{O}(L_1 \cup cL_2 \cup (L_2 \cap \{\varepsilon\}))$

Veta 4.4.2 *Každé (úplné) hlavné trio je (úplná) hlavná semi-AFL.*

Dôkaz. Nech $\mathscr{L}$ je (úplné) hlavné trio, $L \in \mathscr{L}$ nech je jeho ľub. generátor. Nech $\mathscr{L}_1, \mathscr{L}_2 \in \mathscr{L}$. Keďže L je generátor, existujú a-prekladače M_1, M_2 také, že $L_i = M_i(L)$. Zostrojme a-prekladač M, ktorý sa po spustení nedeterministicky rozhodne, či bude fungovať ako M_1 alebo ako M_2. Zjavne $M(L) = L_1 \cup L_2$.

Cvičenie 4.4.1 *Nech L_1, L_2 sú jazyky. Existuje vždy jazyk L taký, že $\mathfrak{M}(\{L_1, L_2\}) = \mathfrak{M}(L)$?*

Veta 4.4.3 *$\mathscr{L}$ je hlavná AFL iff existuje konečná množina jazykov $\mathscr{U}$ taká, že $\mathscr{L} = \mathbf{F}(\mathscr{U})$. (Resp. $\mathscr{L} = \widehat{\mathbf{F}}(\mathscr{U})$ pre úplnú AFL.)*

Pozorovanie. Identické tvrdenie ako veta 4.4.3 platí aj pre (úplné) semi-AFL, dôkaz je taktiež identický.

Lema 4.4.4 *Existuje AFL, ktorá nie je hlavná.*

Dôkaz. Každý jazyk hlavnej AFL vznikne konečným počtom operácií. Možných operácií je spočítateľne veľa. Potom ale aj jazykov v hlavnej AFL musí byť spočítateľne veľa. (Prečo?) Preto žiadna AFL, ktorá obsahuje nespočítateľne veľa jazykov, nemôže byť hlavná. Napr. teda nie je hlavná AFL obsahujúca všetky jazyky (nad nejakou abecedou).

Veta 4.4.5 *$\mathscr{L}$ je AFL, ktorá nie je hlavná iff existuje nekonečná postupnosť AFL $\mathscr{L}_1 \subsetneq L_2 \subsetneq \dots$ taká, že $\mathscr{L} = \bigcup \mathscr{L}_i$.*

Dôkaz. Nech $\mathscr{L}$ je AFL, ktorá nie je hlavná. Nech $L_1 \in \mathscr{L}$ je ľubovoľný, položme $\mathscr{L}_1 = \mathbf{F}(L)$. Keďže $\mathscr{L}$ nie je hlavná ale $\mathscr{L}_1$ je, existuje $L_2 \in (\mathscr{L} \setminus \mathscr{L}_1)$. Položme $\mathscr{L}_2 = \mathbf{F}(\{L_1, L_2\})$. Podľa vety 4.4.3 je aj $\mathscr{L}_2$ hlavná AFL. Analogicky postupujeme ďalej.

Opačným smerom nech $\mathscr{L} = \bigcup \mathscr{L}_i$. Keby $\mathscr{L}$ bola hlavná, má generátor L. Ten ale patrí do niektorého $\mathscr{L}_i$. Potom ale $\mathscr{L} = \mathscr{L}_i \subsetneq L_{i+1} \subseteq \mathscr{L}$, čo je spor.

Dôsledok 4.4.6 *$\mathscr{L}_{rec}$ nie je hlavná AFL.*