Formální jazyky I

Chomského hierarchie formálních jazyků. Regulární jazyky, jejich reprezentace a převody mezi nimi. Varianty konečných automatů. Nedeterminismus a determinizace automatů. Uzávěrové vlastnosti regulárních jazyků.

IB102/IB005

Uvod

Abeceda a jazyk

Abecedou rozumieme lubovolnu konecnu mnozinu Σ. Jej prvky nazyvame znaky (pismena alebo symboly).
Priklad: {a,b}, {0,1,…,9}, $\varnothing$

Slovo (retazec) nad abecedou Σ je lubovolna konecna postupnost znakov z Σ. Dlzka slova v je pocet znakov v slove (znacime #(v)), pocet vyskytu znaku a v slove znacime $\#_a(v)$. Specialny pripad je prazdne slovo s nulovou delkou, znacime ho ε.
Priklad: pre Σ = {a,b} je slovo aabb, aa, b. Pre v = aabb, #(v) = 4 a $\#_a(v)$ = 2

Mnozina vsetkych slov nad Σ znacime Σ^*, mnozinu vsetkych neprazdnych slov znacime Σ^+.
Priklad: $\{a\}^* = \{\varepsilon,a,aa,aaa,aaaa,...\}$, $\{a\}^+ = \{a,aa,aaa,aaaa,...\}$
Specialne: $\varnothing^* = \{\varepsilon\}$ a $\varnothing^+ = \varnothing$.

Jazyk nad abecedou Σ je libovolna mnozina slov nad Σ (i.e. podmnoziny Σ^*). Mozu byt konecne aj nekonecne.
Priklad: $\Sigma = \{0, 1\}$, potom L nad Σ je {10, 1, 011101}. L nad {a,b} definovany ako $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ je zas nekonecny.
Specialne: $\varnothing$ je L nad libovolnou Σ.

Operace nad jazykmi

Kedze jazyku su len mnoziny, mozeme aplikovat mnozinove operacie: zjednotenie, prienik a rozdiel.
Priklad: Ak L je nad Σ, K je nad Δ , tak $L \cup K$ je nad $\Sigma \cup \Delta$.

Dalej definujeme (pre jazyky L je nad Σ, K je nad Δ)
- Zretazenie: $K.L = \{uv \mid u \in K, v \in L\}$ nad $\Sigma \cup \Delta$
- i-ta mocnina jazyka L: $L^0 = \{\varepsilon\}$ a $L^{i+1} = L.L^i$
- Iterace jazyka L: $L^* = \bigcup_{i=0}^{\infty} L^i$.
- Doplnok: co–L = $\Sigma^* \setminus L$.
- Substituce, homomorfizmus (prip. inverzni), zrkadlovy obraz…

Specialne: $\varnothing.L = L.\varnothing = \varnothing$ a $\{\varepsilon\}.L = L.\{\varepsilon\} = L$, $\varnothing^0 = \{\varepsilon\}$, $\varnothing^i = \varnothing$ $(i \in N)$, $\{\varepsilon\}^j = \{\varepsilon\}$ $(j \in N_0)$.

Definice 1.2. *Gramatika* $\mathcal{G}$ je čtveřice (N, Σ, P, S), kde

- N je neprázdná konečná množina *neterminálních symbolů* (stručněji: *neterminálů*).
- Σ je konečná množina *terminálních symbolů (terminálů)* taková, že $N \cap \Sigma = \emptyset$. Sjednocením N a Σ obdržíme množinu *všech symbolů* gramatiky, kterou obvykle označujeme symbolem V.
- $P \subseteq V^*NV^* \times V^*$ je konečná množina *pravidel*. Pravidlo (α, β) obvykle zapisujeme ve tvaru $\alpha \to \beta$ (a čteme jako "α přepiš na β").
- $S \in N$ je speciální *počáteční* neterminál (nazývaný také *kořen gramatiky*).

Poziadavky na pravidla (α,β) : α musi obsahovat aspon jeden neterminal.
β moze byt aj prazna (ε).
Vetna forma gramatiky G: prvky mnoziny (N a Σ) * , ktore ide odvodit z pociatocneho neterminalu za pomoci pravidiel gramatiky.
Formalne: α ∈ (N a Σ) * je vetna forma $\Longleftrightarrow$ S $\Rightarrow^*$ α.
Vetna forma bez neterminalu je veta. Mnozina vsetkcyh viet gramatiky je jazyk generovany gramatikou: L(G) = {w ∈ Σ^* | S $\Rightarrow^*$ w}

Chomského hierarchie formálních jazyků.

- 4 skupiny na zaklade omedzenia na tvar pravidiel pre gramatiku

typ 0 Libovolná gramatika je gramatikou typu 0; na tvar pravidel se nekladou žádné omezující požadavky. Někdy též se takové gramatiky označují jako gramatiky bez omezení či frázové gramatiky (phrase grammars).

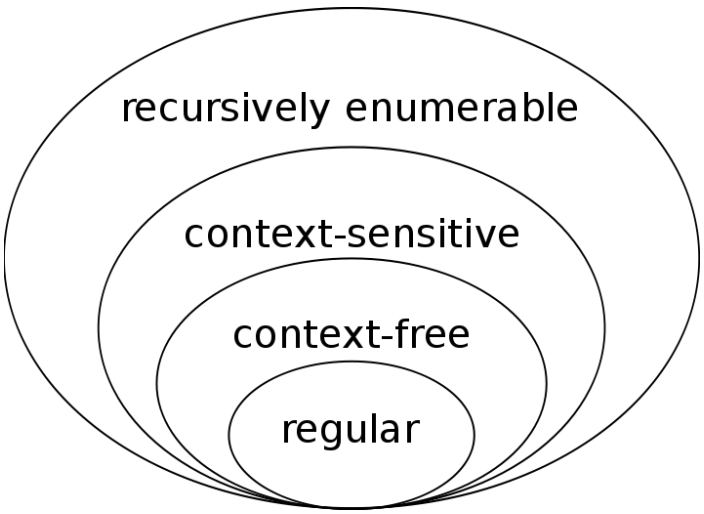
typ 1 Gramatika je typu 1 (nebo též *kontextová*[1], Context-Sensitive, CSG, méně často též *monotónní*), jestliže pro každé její pravidlo $\alpha \to \beta$ platí $|\alpha| \leq |\beta|$ s eventuelní výjimkou pravidla $S \to \varepsilon$, pokud se S nevyskytuje na pravé straně žádného pravidla.

typ 2 Gramatika je typu 2 (též *bezkontextová*, Context-Free, CFG), jestliže každé její pravidlo je tvaru $A \to \alpha$, $|\alpha| \geq 0$.

typ 3 Gramatika je typu 3 (též *regulární* či *pravolineární*[2]), jestliže každé její pravidlo je tvaru $A \to aB$ nebo $A \to a$ s eventuelní výjimkou pravidla $S \to \varepsilon$, pokud se S nevyskytuje na pravé straně žádného pravidla. [3]

Mapovanie gramatika : jazyk:
Jazyk je:
- regularny, ak ho generuje gramatika typu 3
- bezkontextovy, ak ho generuje gramatika typu 2
- kontextovy, ak ho generuje gramatika typu 1
- typu 0, ak ho generuje gramatika typu 0

To znamena: Hierarchie gramatik urcuje prislusnu hierarchiu jazykov.



Pozor: obrazok ukazuje vlastnost: ak je jazyk regularny, potom je aj context-free. Nehovori nic o velkosti jazyka (napr Σ^* je regularny, $a^n b^n$ je “mensi”, ale je context-free)

Prehlad vsetkych rozhodnutelnych a nerozhodnutelnych problemov tykajucich sa tried jazykov Chomskeho hierarchie:

	R	DCF	CF	CS	Rec	RE
Je $L(\mathcal{G})$ prázdný? konečný?	R	R	R	N	N	N
Je $L(\mathcal{G}) = \Sigma^*$?	R	R	N	N	N	N
Je $L(\mathcal{G}) = R$? (R je regulární množina)	R	R	N	N	N	N
Je $L(\mathcal{G}_1) = L(\mathcal{G}_2)$?	R	R	N	N	N	N
Je $L(\mathcal{G}_1) \supseteq L(\mathcal{G}_2)$?	R	N	N	N	N	N
Je $L(\mathcal{G})$ regulární jazyk?	ano	R	N	N	N	N
Je průnik dvou jazyků jazyk téhož typu?	ano	N	N	ano	ano	ano
Je sjednocení dvou jazyků jazyk téhož typu?	ano	N	ano	ano	ano	ano
Je komplement jazyka jazyk téhož typu?	ano	ano	N	ano	ano	N
Je zřetězení dvou jazyků jazyk téhož typu?	ano	N	ano	ano	ano	ano
Je gramatika $\mathcal{G}$ víceznačná?	R	N	N	N	N	N

Regulární jazyky, jejich reprezentace a převody mezi nimi.

Jazyk L nazveme regularnim ak je rozpoznatelny nejakym konecnym automatom (vid. konecne automaty nizsie). Pripadne, ak je generovany gramatikou typu 3 (vid. definicia Chomskeho hierarchie).

Pumping lemma:

Lemma 2.13 (o vkládání). *Nechť L je regulární jazyk. Pak existuje $n \in \mathbb{N}$ takové, že libovolné slovo $w \in L$, jehož délka je alespoň n, lze psát ve tvaru $w = xyz$, kde $|xy| \leq n$, $y \neq \varepsilon$ a $xy^i z \in L$ pro každé $i \in \mathbb{N}_0$. (Číslo n se neformálně nazývá pumpovací konstanta.)*

Pozor: Pumping lemma je v tvare implikacie v tvare L jeregularni⇒ Q (zvysne tvrenie), a teda plati, ze existuju neregularne jazyky, ktore splnaju pumping lemmu.
Pri dokazovani, ze L nie je regularni pouzivame kontrapositivni formu: ¬Q ⇒ L neni regularni. Kucharka: dokaz sporom: Predpokladame ze L je regularni, ukazeme ze:

- pro lib. n ∈ N (pumpovacia konstanta)
- existuje slovo w ∈ L s dlzkou aspon n
- pri lub. rozlozeni slova w na 3 casti: x,y,z , take ze |xy| ≤ n a y nie je ε,
- existuje i ∈ N_0: xy^iz nepatri do L.

Priklad:

Příklad 2.14. *Ukážeme, že $L = \{a^p \mid p$ je prvočíslo$\}$ nad abecedou $\{a\}$ není regulární.*

Důkaz. Pro dosažení sporu předpokládejme, že L je regulární. Buď $n \in N$ libovolné (pumpovací konstanta z PL). Jelikož prvočísel je nekonečně mnoho, existuje prvočíslo p, které je větší nebo rovno n; zvolme $w = a^p$ patřící do L. Při jakémkoli rozdělení w na podslova x, y, z musí být $y = a^k, k \geq 1$. Napumpujeme-li y $p+1$-krát, dostaneme: $xy^{p+1}z = xyy^pz = xyzy^p = a^p a^{kp} = a^{p(k+1)}$, což je jistě slovo, které nepatří do jazyka L, protože $p(k+1)$ není prvočíslo – dostáváme tedy spor s naším předpokladem, že L je regulární. Podle PL tedy L regulární není. □

Myhillova-Nerodova veta (nebralo sa na lahsich automatoch):

Věta 2.28 (Myhillova-Nerodova). *Nechť L je jazyk nad Σ, pak tato tvrzení jsou ekvivalentní:*

1. *L je rozpoznatelný konečným automatem.*
2. *L je sjednocením některých tříd rozkladu určeného pravou kongruencí na Σ^* s konečným indexem.*
3. *Relace $\sim_L$ má konečný index.*

Varianty konečných automatů (minimalni, nedeterministicky, ε-kroky).

Definice 2.1. *Konečný automat* (Finite Automaton, FA) $\mathcal{M}$ je pětice $(Q, \Sigma, \delta, q_0, F)$, kde

- Q je neprázdná konečná množina *stavů*.
- Σ je konečná množina *vstupních symbolů*, nazývaná také *vstupní abeceda*.
- $\delta : Q \times \Sigma \to Q$ je parciální *přechodová funkce*.
- $q_0 \in Q$ je *počáteční stav*.
- $F \subseteq Q$ je množina *koncových stavů*.

Abychom mohli definovat jazyk přijímaný daným FA $\mathcal{M}$, zavedeme *rozšířenou* přechodovou funkci $\hat{\delta} : Q \times \Sigma^* \to Q$, definovanou induktivně vzhledem k délce slova ze Σ^*:

- $\hat{\delta}(q, \varepsilon) = q$ pro každý stav $q \in Q$.
- $\hat{\delta}(q, wa) = \begin{cases} \delta(\hat{\delta}(q, w), a) & \text{je-li } \hat{\delta}(q, w) \text{ i } \delta(\hat{\delta}(q, w), a) \text{ definováno,} \\ \bot & \text{jinak.} \end{cases}$

Rozsirena prechodova funckia induktivne definuje prechod nad celym slovom za pomoci prechodu nad jednym znakom. Potom (M je konecny automat):

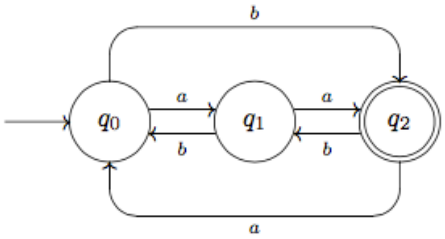
$$L(M) = \{ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F \}$$

Priklad:

Příklad 2.3. *Nechť* $\mathcal{M} = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$ *je FA, kde*

$$\delta(q_0, a) = q_1 \qquad \delta(q_0, b) = q_2$$
$$\delta(q_1, a) = q_2 \qquad \delta(q_1, b) = q_0$$
$$\delta(q_2, a) = q_0 \qquad \delta(q_2, b) = q_1$$

Pak $L(\mathcal{M}) = \{w \in \{a, b\}^* \mid (\#_a(w) - \#_b(w)) \bmod 3 = 2\}$.



Minimalni konecny automat
(k lubovolenmu konecnemu automatu A) je automat s najmensim poctom stavov a totalnou prechodovou funckiou, ktory rozpoznava regularny jazyk L(A).

Tvrdenie o existencii mininalneho konecneho automatu plynie z Myhillovy-Nerodovy vety. Prevod ide uskutocnit *algoritmicky*. Dokaz korektnosti tohto postupu je pomocou kongruencii nad stavmi. Hladame jazykovo eqivalentne stavy: Stavy p, q nazveme jazykovo ekvivalentne ($p \equiv q$), ak $L(p) = L(q)$.

Postup: odstranime nedosazitelne stavy. Potom "spajame" stavy do mnozin podla relace ≡. Na zaciatku je kazdy stav sam. Stavy su v relaci $\equiv_0$ ak su oba koncove (akceptuju teda rovnake slova = ε). Postupne hladame $\equiv_i$ az kym nenajdeme fixpoint.

Priklad:

	$\mathcal{M}$	a	b
$\rightarrow$	1	2	–
	2	3	4
$\leftarrow$	3	6	5
	4	3	2
$\leftarrow$	5	6	3
$\leftarrow$	6	2	–
	7	6	1

	$\mathcal{M}'$	a	b
$\rightarrow$	1	2	N
	2	3	4
$\leftarrow$	3	6	5
	4	3	2
$\leftarrow$	5	6	3
$\leftarrow$	6	2	N
	N	N	N

(odstranenie nedosiahnutelnych)

	$\equiv_0$	a	b
I	1	I	I
	2	II	I
	4	II	I
	N	I	I
II	3	II	II
	5	II	II
	6	I	I

	$\equiv_1$	a	b
I	1	II	I
	N	I	I
II	2	III	II
	4	III	II
III	3	IV	III
	5	IV	III
IV	6	II	I

	$\equiv_2$	a	b
I	1	III	II
II	N	II	II
III	2	IV	III
	4	IV	III
IV	3	V	IV
	5	V	IV
V	6	III	II

(rozklad podla ≡i)

Relace $\equiv$ je tedy v tomto případě rovna relaci $\equiv_2$. Minimální automat pro jazyk $L(\mathcal{M})$ vypadá takto:

	$\mathcal{M}/\equiv$	a	b
$\rightarrow$	I	III	II
	II	II	II
	III	IV	III
$\leftarrow$	IV	V	IV
$\leftarrow$	V	III	II

Nedeterministicke konecne automaty

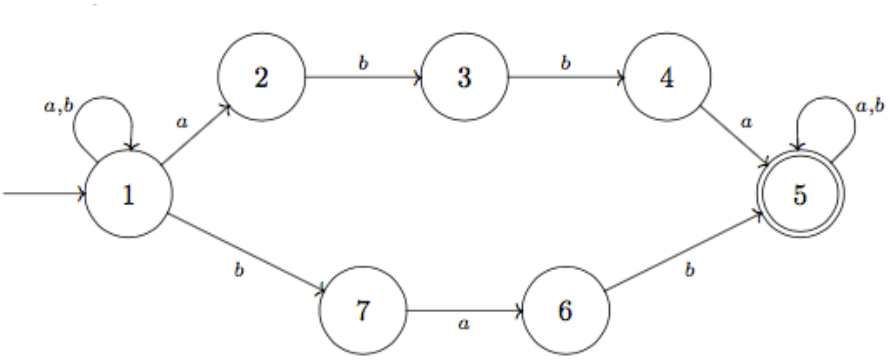
$\delta : Q \times \Sigma \rightarrow Q$ zmenime na $\delta : Q \times \Sigma \rightarrow 2^Q$. A teda z jedneho stavu mozme pod jednym symbolom prejst do viacerych stavov, a teda do urcitej podmnoziny Q.

Podobně jako v případě deterministických automatů zavedeme *rozšířenou* přechodovou funkci $\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q$:

- $\hat{\delta}(q, \varepsilon) = \{q\}$
- $\hat{\delta}(q, wa) = \bigcup_{p \in \hat{\delta}(q,w)} \delta(p, a)$

A potom plati (M je nedeterministiky automat): $L(M)=\{ w\in\Sigma^* \mid (\hat{\delta}(q_0,w) \cap F) /=\emptyset \}$
Nedeterminizmus nam pomaha konstruovat “krajsie” automaty:

$L = \{w \in \{a, b\}^* \mid w$ obsahuje podslovo abba nebo bab$\}$

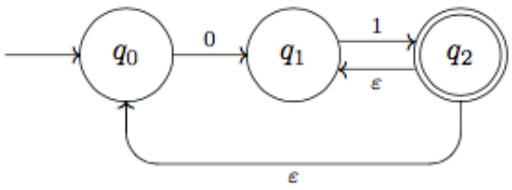


Plati ale: Pro kazdy NFA $M = (Q, \Sigma, \delta, q_0, F)$ existuje ekvivalentny DFA. (sekcia Nedeterminizmus a determinizace automatu).

Automaty s ε-kroky

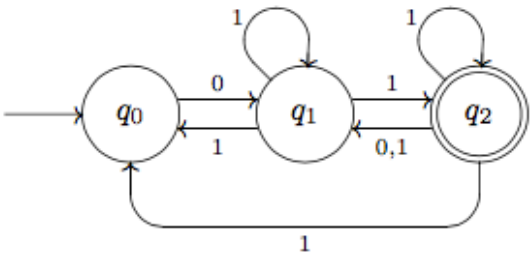
Automat takto moze svoj stav zmenit aj bez precitania vstupenho symbolu (a teda uplne samovolne). Ako znacenie pouzivame prechod pod ε.

Prechodovu funckiu teda redefinujeme:
$\delta : Q \times(\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$.



Pozor: rozsirena prechodova funckia uz potom nie je tak intuitivna. V kazdom stave musime zahrnut este aj stavy, do ktorych sa dostaneme pomocou ε prechodu (predtym nez nacitame dalsi symbol).

Plati: ku kazdemu automatu M s ε-kroky existuje ekvivalentny nedeterministicky automat M.
Hladame cesty tvaru:
$p_1 \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} p_m \xrightarrow{a} q_1 \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} q_n$
, pre ktore pridame pravidlo $p_1 \xrightarrow{a} q_n$.



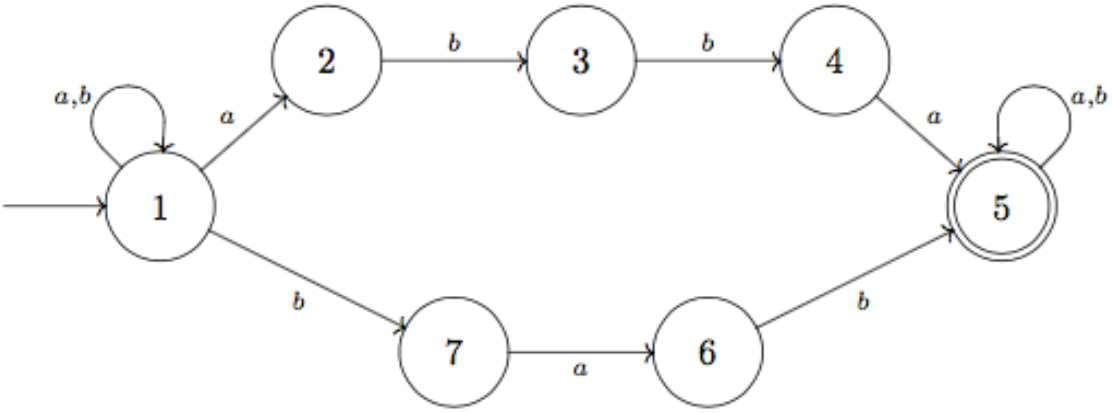
Nedeterminismus a determinizace automatů.

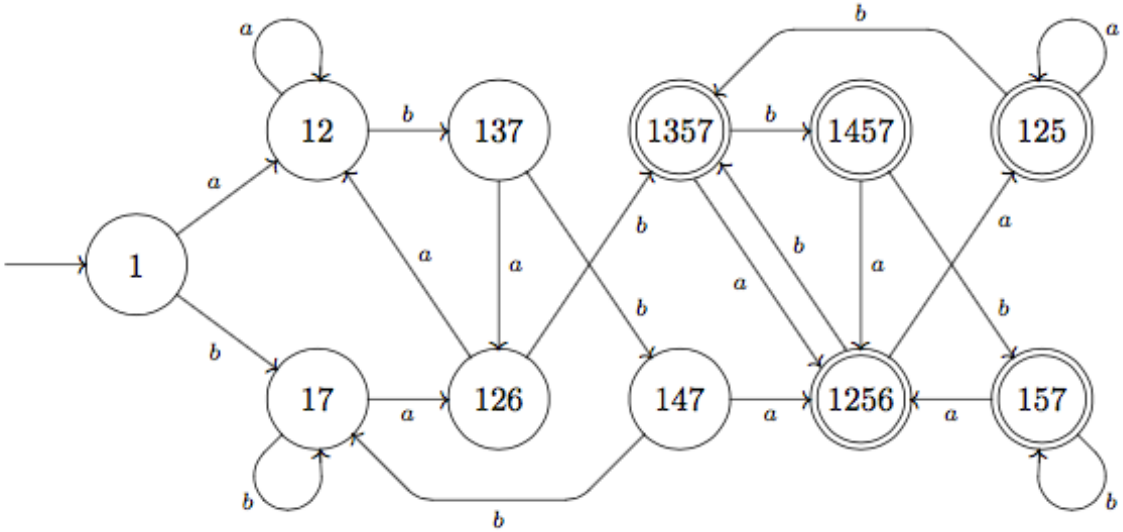
Pro kazdy NFA M = (Q, Σ, δ, q_0, F) existuje ekvivalentny DFA.

Důkaz. Nechť $\mathcal{M}' = (Q', \Sigma, \delta', \{q_0\}, F')$ je deterministický konečný automat, kde:

- $Q' = 2^Q$, tj. stavy automatu $\mathcal{M}'$ jsou všechny podmnožiny Q.
- Přechodová funkce δ' je definována předpisem $\delta'(P, a) = \bigcup_{q \in P} \delta(q, a)$
- Množina koncových stavů F' je tvořena právě těmi podmnožinami Q, které obsahují alespoň jeden prvek množiny F.

Zřejmě $\mathcal{M}'$ je deterministický konečný automat (dokonce s totální přechodovou funkcí).





To, ze akceptuju ten isty jazyk sa da ukazat vzhladom k dlzke odvodeneho slova w (indukciou), ale je asi nad ramec statnic (aj casovo). Pozor na narast stavov (exponencialny). Automat sa da nasledne este minimalizovat.

Uzávěrové vlastnosti regulárních jazyků.

1. Trieda regularnych jazykov je uzavrena na zjednotenie, prienik a rozdiel.

$L_1 \cup L_2$: Majme A_1, A_2, plati $L_1 \cup L_2 = A_1 \cup A_2$,ktory je definovany ako bud parelelna kompozicia, kde akcaptujeme prave vtedy ked aspon jeden automat dosiahne akceptujuci stav, alebo urobime kartezsky sucin stavov, prechodova relacia je dvojica, a akceptujeme stavmi, kde aspon jeden z povodnych bol akceptujuci.

$L_1 \cap L_2$: Majme A_1, A_2, plati $L_1 \cup L_2 = A_1 \cup A_2$,podobne ako zjednotenie, ale musia akceptovat oba.

$L_1 - L_2$: Majme A_1, A_2, plati $L_1 \cup L_2 = A_1 \cup A_2$,podobne, ale akceptujeme len ak prvy akceptoval a druhy zamietol, pri sucine su akceptujuce stavy dvojica (p,q), p patri A_1 a je akceptujuci (patri do F_1) a q patri A_2 a je neakceptujuci (nepatri do F_2).

2. Trieda regularnych jazykov je uzavrena na komplement.

Plati: co–$L = \Sigma^* - L$
Postup: vymenime koncove a nekoncove stavy: co-$A = (Q,\Sigma,\delta,q_0,Q - F)$ k povodnemu $A = (Q,\Sigma,\delta,q_0,F)$.

3. Trieda regularnych jazykov je uzavrena na zretazenie.

Pridame ε-krok pre koncove stavy z A_1 do pociatocnych z A_2.

4. Trieda regularnych jazykov je uzavrena na iteraciu.

Pridame novy stav q, ktory bude po novom jedyni koncovy. Zavedieme ε-krok pre koncove stavy z A_1 do q, a taktiez ε-krok z q do pociatocneho stavu.

5. Trieda regularnych jazykov je uzavrena na substituciu, homomorfizmus… (uz prilis specificke, no time for that)

Priklady:

5.1 Rozhodněte, zda platí: jsou-li jazyky $L_1, L_2, L_3, \ldots$ regulární, pak i jazyk

$$\bigcup_{i=1}^{\infty} L_i$$

je regulární jazyk.

Neplatí. Jazyky $L_i = \{a^i b^i\}$ pro každé $i > 0$ jsou konečné a tudíž regulární, ale $\bigcup_{i=1}^{\infty} L_i = \{a^n b^n \mid n > 0\}$ není regulární.

a) L_1 je regulární, L_2 je neregulární $\Rightarrow L_1 \cap L_2$ je neregulární
b) L_1 je regulární, L_2 je neregulární $\Rightarrow L_1 \cap L_2$ je regulární
c) L_1 je regulární, L_2 je neregulární $\Rightarrow L_1 \setminus L_2$ je neregulární
d) L_1 je regulární, L_2 je neregulární $\Rightarrow L_1 \setminus L_2$ je regulární
e) L_1 je regulární, L_2 je neregulární $\Rightarrow L_2 \setminus L_1$ je neregulární
f) L_1 je regulární, L_2 je neregulární $\Rightarrow L_2 \setminus L_1$ je regulární

a) {ab} je regularni jazyk, $a^n b^n$ nie je. Ich prienik je {a,b}
b) Σ^* je regularni jazyk, $a^n b^n$ nie je. Ich prienik je $a^n b^n$
c) {ab} je regularni jazyk, $a^n b^n$ nie je. vysledok je $\varnothing$
d) Σ^* je regularni jazyk, $a^n b^n$ nie je. vysledok je co-$a^n b^n$
e) ….