Teoretická informatika

TIN 2017/2018

prof. RNDr. Milan Češka, CSc.

ceska@fit.vutbr.cz

prof. Ing. Tomáš Vojnar, Ph.D.

vojnar@fit.vutbr.cz

sazba dr. A. Smrčka, Ing. P. Erlebach, Ing. P. Novosad

Vysoké učení technické v Brně
Fakulta informačních technologií
Božetěchova 2, 612 66 Brno

Referenční literatura

❖ Předmět vychází zejména z následujících zdrojů:

- Češka, M.: Teoretická informatika, učební text FIT VUT v Brně, 2002. `http://www.fit.vutbr.cz/study/courses/TIN/public/Texty/ti.pdf`
- Kozen, D.C.: Automata and Computability, Springer-Verlag, New York, Inc, 1997. ISBN 0-387-94907-0
- Černá, I., Křetínský, M., Kučera, A.: Automaty a formální jazyky I, učební text FI MU, Brno, 1999.
- Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, Addison Wesley, 2. vydání, 2000. ISBN 0-201-44124-1
- Gruska, J.: Foundations of Computing, International Thomson Computer Press, 1997. ISBN 1-85032-243-0
- Bovet, D.P., Crescenzi, P.: Introduction to the Theory of Complexity, Prentice Hall Europe, Pearson Education Limited, 1994. ISBN 0-13-915380-2
- Reisig, W.: Petri Nets, An Introduction, Springer Verlag, 1985. ISBN: 0-387-13723-8

Jazyky a jejich reprezentace, algebra formálních jazyků

Formální jazyky

- Prvotní pojmy: symbol, abeceda.

Definice 1.1 Nechť Σ je abeceda. Označme Σ^* množinu všech konečných posloupností w tvaru:

$$w = a_1 a_2 \ldots a_n, a_i \in \Sigma \text{ pro } i = 1, \ldots, n.$$

Posloupnosti w nazýváme řetězce nad abecedou Σ. Dále definujeme délku $|w|$ řetězce $w : |w| = n$. Množina Σ^* obsahuje také speciální řetězec ε, pro který platí $|\varepsilon| = 0$. ε se nazývá prázdný řetězec.

❖ Na množině Σ^* zavedeme operaci $\cdot$ takto:
Jsou-li dány dva řetězce w, w' z Σ^* (nad abecedou Σ):

$$w = a_1 a_2 \ldots a_n,$$
$$w' = a'_1 a'_2 \ldots a'_m \; n, m \geq 0,$$

pak $w \cdot w' = a_1 a_2 \ldots a_n a'_1 a'_2 \ldots a'_m \; (= ww')$.

Operace $\cdot$ se nazývá zřetězení nebo konkatenace.
Pro w, w', w'' platí:

1. $|ww'| = |w| + |w'|$,
2. $w(w'w'') = (ww')w''$ tj. asociativnost konkatenace,
3. $w\varepsilon = \varepsilon w = w$ tj. ε je jednotkový prvek vzhledem k operaci $\cdot$.

❖ Terminologie:

- Σ^* se nazývá iterace abecedy Σ.
- $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ se nazývá pozitivní iterace abecedy Σ.
- Dále zavádíme pojmy: prefix, sufix, podřetězec, a^i, w^R.

Věta 1.1 Algebraická struktura $< \Sigma^+, \cdot >$, resp. $< \Sigma^*, \cdot, \varepsilon >$ je pologrupa, resp. monoid.

❖ Alternativní způsob definice množiny Σ^*:

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots \cup \Sigma^n \cup \dots$$
$$= \{\varepsilon\} \cup \Sigma \cup \Sigma \times \Sigma \cup \dots \underbrace{(\Sigma \times \Sigma \times \dots \times \Sigma)}_{n-krát} \cup \dots$$

Definice 1.2 Množinu L, pro kterou platí $L \subseteq \Sigma^*$ (resp. $L \subseteq \Sigma^+$) nazýváme formálním jazykem nad abecedou Σ.

❖ Příklady jazyků:

- $L_1 = \{0^n 1^n \mid n \geq 0\}$
- $L_2 = \{ww^R \mid w \in \{0,1\}^+\}$
- $L_3 \equiv$ progr. jazyk Pascal

Operace nad jazyky

Jazyk je množina $\rightarrow$ jsou definovány všechny množinové operace nad jazyky např.:
$L' = \Sigma_L^* \setminus L$ — komplement jazyka L

Definice 1.3 Nechť L_1 je jazyk nad abecedou Σ_1, L_2 je jazyk nad abecedou Σ_2. Součinem (konkatenací) jazyků L_1 a L_2 nad abecedou $\Sigma_1 \cup \Sigma_2$ rozumíme jazyk

$$L_1 \cdot L_2 = \{xy \mid x \in L_1 \wedge y \in L_2\}$$

Příklad 1.1 Nechť $P = \{A, B, \ldots, Z, a, b, \ldots, z\}$, $c = \{0, 1, \ldots, 9\}$ jsou abecedy, $L_1 = P$ a $L_2 = (P \cup C)^*$ jazyky nad P resp. $P \cup C$.
Jaký jazyk určuje součin $L_1 L_2$?

Iterace a pozitivní iterace

Definice 1.4 Nechť L je jazyk. Iterací L^* jazyka L a pozitivní iterací L^+ jazyka L definujeme takto:

1. $L^0 = \{\varepsilon\}$
2. $L^n = L \cdot L^{n-1}$ pro $n \geq 1$
3. $L^* = \bigcup_{n \geq 0} L^n$
4. $L^+ = \bigcup_{n \geq 1} L^n$

Příklad 1.2 $L_1 = \{(p\}, L_2 = \{,p\}, L_3 = \{)\}$

$$L_1 L_2^* L_3 = \{(p),\ (p,p),\ (p,p,p),\ \ldots\}$$

❖ *Poznámka 1:* Operátor $*$ se nazývá také Kleene star.

❖ *Poznámka 2:* Všimněte si, že (pozitivní) iterace abecedy odpovídá (pozitivní) iteraci jazyka tvořeného větami délky jedna.

Definice 1.5 Algebraická struktura $< A,\ +,\ \cdot,\ 0,\ 1 >$ se nazývá polookruh, jestliže:

1. $< A,\ +,\ 0 >$ je komutativní monoid,
2. $< A,\ \cdot,\ 1 >$ je monoid,
3. pro operaci $\cdot$ platí distributivní zákon vzhledem k $+$:
   $\forall a, b, c \in A : a(b + c) = ab + ac, (a + b)c = ac + bc$.

Věta 1.2 Algebra jazyků $< 2^{\Sigma^*},\ \cup,\ \cdot,\ \emptyset,\ \{\varepsilon\} >$, kde $\cup$ je sjednocení a $\cdot$ konkatenace jazyků tvoří polookruh.

Důkaz.

1. $< 2^{\Sigma^*},\ \cup,\ \emptyset >$ je komutativní monoid ($\cup$ je komutativní a asociativní operace a $L \cup \emptyset = \emptyset \cup L = L$ pro všechna $L \in 2^{\Sigma^*}$).
2. $< 2^{\Sigma^*},\ \cdot,\ \{\varepsilon\} >$ je monoid:
   $L \cdot \{\varepsilon\} = \{\varepsilon\} \cdot L = L$ pro všechna $L \in 2^{\Sigma^*}$.

Důkaz pokračuje dále.

Pokračování důkazu.

3. Pro všechny $L_1, L_2, L_3 \in 2^{\Sigma^*}$:
$L_1(L_2 \cup L_3) = \{xy \mid (x \in L_1) \wedge (y \in L_2 \vee y \in L_3)\} =$
$= \{xy \mid (x \in L_1 \wedge y \in L_2) \vee (x \in L_1 \wedge y \in L_3)\} =$
$= \{xy \mid x \in L_1 \wedge y \in L_2\} \cup \{xy \mid x \in L_1 \wedge y \in L_3\} =$
$= L_1L_2 \cup L_1L_3$.

□

Věta 1.3 Je-li L jazyk, pak platí:

1. $L^* = L^+ \cup \{\varepsilon\}$
2. $L^+ = L \cdot L^* = L^* \cdot L$

Důkaz.

1. Zřejmé z definice L^* a L^+.
2. Důsledek Věty 1.2 (platnosti distributivního zákona).

□

Gramatiky

❖ Pozn. Reprezentace jazyků – problém reprezentace, způsoby reprezentace.

Definice 1.6 Gramatika G je čtveřice $G = (N,\ \Sigma,\ P,\ S)$, kde

1. N je konečná množina nonterminálních symbolů.
2. Σ je konečná množina terminálních symbolů, kde $N \cap \Sigma = \emptyset$.
3. P je konečná podmnožina kartézského součinu

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$$

   nazývaná množina přepisovacích pravidel
4. $S \in N$ je výchozí (startovací) symbol gramatiky G.

❖ Prvek $(\alpha, \beta) \in P$ je přepisovací pravidlo a zapisuje se ve tvaru

$$\alpha \rightarrow \beta,$$

kde α je levá strana, β je pravá strana pravidla $\alpha \rightarrow \beta$.

Příklady:

- $G_1 = (\{A, S\}, \{0, 1\}, P_1, S)$

$$
\begin{array}{llll}
P_1: & S & \rightarrow & 0A1 \\
 & 0A & \rightarrow & 00A1 \\
 & A & \rightarrow & \varepsilon
\end{array}
$$

- $G_2 = (N_2, \Sigma_2, P_2, I)$
  $N_2 = \{I, P, C\}$
  $\Sigma_2 = \{a, b, \ldots, z, 0, 1, \ldots, 0\}$

$$
\begin{array}{llllllll}
P_2: & I & \rightarrow & P & & & & \\
 & I & \rightarrow & IP & & & & \\
 & I & \rightarrow & IC & & & & \\
 & P & \rightarrow & a & & C & \rightarrow & 0 \\
 & P & \rightarrow & b & & C & \rightarrow & 1 \\
 & & \vdots & & & & \vdots & \\
 & P & \rightarrow & z & & C & \rightarrow & 9
\end{array}
$$

❖ Konvence 1: Obsahuje-li množina P přepisovací pravidla tvaru

$$\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \ldots, \alpha \rightarrow \beta_n$$

pak pro zkrácení budeme používat zápisu

$$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$$

❖ Konvence 2: Pro zápis symbolů a řetězců budeme užívat této úmluvy:

1. a, b, c, d reprezentují terminální symboly.
2. A, B, C, D, S reprezentují nonterminální symboly, S výchozí symbol.
3. $U, V, \ldots, Z$ reprezentují terminální nebo nonterminální symboly.
4. $\alpha, \beta, \ldots, \omega$ reprezentují řetězce z množiny $(N \cup \Sigma)^*$.
5. $u, v, w, \ldots, z$ reprezentují řetězce z Σ^*.

Definice 1.7 Nechť $G = (N, \Sigma, P, S)$ je gramatika a nechť λ, μ jsou řetězce z $(N \cup \Sigma)^*$. Mezi λ a μ platí binární relace $\underset{G}{\Rightarrow}$, zvaná přímá derivace, můžeme-li řetězce λ a μ vyjádřit ve tvaru

$$\lambda = \gamma\alpha\delta$$
$$\mu = \gamma\beta\delta$$

$\gamma, \delta \in (N \cup \Sigma)^*$ a $\alpha \rightarrow \beta$ je nějaké přepisovací pravidlo z P. Pak píšeme

$$\lambda \underset{G}{\Rightarrow} \mu \text{ nebo}$$
$$\lambda \Rightarrow \mu.$$

Poznámka.

1. Je-li $\alpha \rightarrow \beta$ pravidlo z P, pak $\alpha \Rightarrow \beta$.
2. relace $\Rightarrow^{-1}$ se nazývá (přímá) redukce.

Definice 1.8 Nechť $G = (N, \Sigma, P, S)$ je gramatika a $\Rightarrow$ relace přímé derivace na $(N \cup \Sigma)^*$. Relace $\overset{+}{\Rightarrow}$ označuje tranzitivní uzávěr relace $\Rightarrow$ a nazývá se relací derivace. Platí-li $\lambda \overset{+}{\Rightarrow} \mu$, pak existuje posloupnost

$$\lambda = \nu_0 \Rightarrow \nu_1 \Rightarrow \ldots \Rightarrow \nu_n = \mu,\ n \geq 1,$$

která se nazývá derivací délky n.

- Relace $\overset{*}{\Rightarrow}$ označuje reflexivní a tranzitivní uzávěr relace $\Rightarrow$:

$$\lambda \overset{*}{\Rightarrow} \mu \qquad \Rightarrow \qquad \lambda \overset{+}{\Rightarrow} \mu \text{ nebo } \lambda = \mu$$

Příklad 1.3 Derivace v gramatice G_1, resp. G_2, ze strany 11:

- V gramatice G_1:
  - Pravidlo $0A \rightarrow 00A1$ implikuje $0^n A1^n \Rightarrow 0^{n+1} A1^{n+1}$,
  - tedy $0A1 \overset{*}{\Rightarrow} 0^n A1^n$ pro libovolné $n > 0$.

- V gramatice G_2:
  - $I \Rightarrow IP \Rightarrow IPP \Rightarrow ICPP \Rightarrow PCPP \Rightarrow aCPP \Rightarrow a1PP \Rightarrow a1xP \Rightarrow a1xy$,
  - tj. $I \overset{+}{\Rightarrow} a1xy$.

Definice 1.9 Nechť $G = (N, \Sigma, P, S)$ je gramatika. Řetězec $\alpha \in (N \cup \Sigma)^*$ nazýváme větnou formou, platí-li $S \stackrel{*}{\Rightarrow} \alpha$. Větná forma, která obsahuje pouze terminální symboly se nazývá věta.

Jazyk $L(G)$ generovaný gramatikou G je množina:

$$L(G) = \{w \mid S \stackrel{*}{\Rightarrow} w \wedge w \in \Sigma^*\}$$

Příklad 1.4

$$L(G_1) = \{0^n 1^n \mid n > 0\}$$

protože

$$\begin{aligned} &S \Rightarrow 0A1 \\ &S \stackrel{*}{\Rightarrow} 0^n A 1^n && \text{(viz předchozí příklad)} \\ &S \stackrel{*}{\Rightarrow} 0^n 1^n && \text{(pravidlo } A \rightarrow \varepsilon\text{)} \end{aligned}$$

Chomského hierarchie

Je definována na základě tvaru přepisovacích pravidel:

- Typ 0 – obecné (neomezené) gramatiky:

$$\alpha \to \beta \qquad \alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*, \beta \in (N \cup \Sigma)^*$$

- Typ 1 – kontextové gramatiky:

$$\alpha A \beta \to \alpha \gamma \beta \qquad A \in N,\ \alpha, \beta \in (N \cup \Sigma)^*, \gamma \in (N \cup \Sigma)^+$$

nebo $S \to \varepsilon$, pakliže se S neobjevuje na pravé straně žádného pravidla

(Alternativní definice definující stejnou třídu jazyků: $\alpha \to \beta,\ |\alpha| \leq |\beta|$ nebo $S \to \varepsilon$ omezené jako výše.)

- Typ 2 – bezkontextové gramatiky:

$$A \to \alpha \qquad A \in N,\ \alpha \in (N \cup \Sigma)^*$$

- Typ 3 – pravé lineární gramatiky:

$$A \to xB \qquad \text{nebo}$$
$$A \to x \qquad A, B \in N,\ x \in \Sigma^*$$

Definice 1.10 Jazyk generovaný gram. typu i, $i = 0, 1, 2, 3$, se nazývá jazykem typu i.

Existuje synonymní označení jazyků:

- Jazyk typu 0 — rekurzivně vyčíslitelný jazyk.
- Jazyk typu 1 — kontextový jazyk.
- Jazyk typu 2 — bezkontextový jazyk.
- Jazyk typu 3 — regulární jazyk.

Věta 1.4 Nechť $\mathcal{L}_i$ značí třídu všech jazyků typu i.
Pak platí:

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$$

Důkaz.
Důkaz plyne z definice tříd Chomského hierarchie jazyků.

□

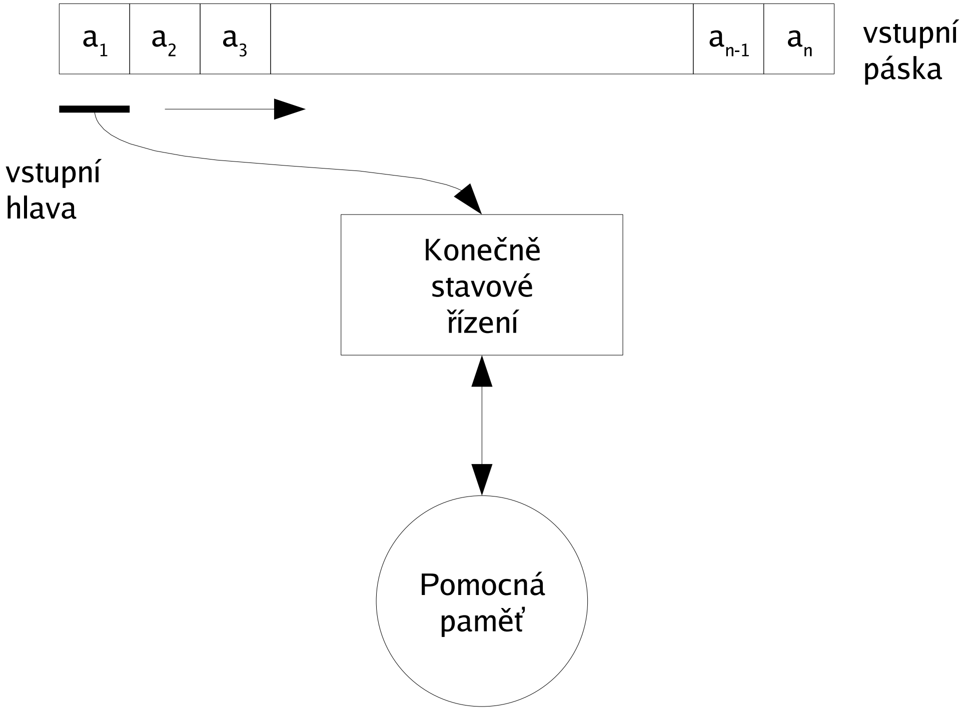
Věta 1.5 Platí:

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$$

Důkaz později.

Automaty

Anglický termín — recognizer.



❖ Klasifikace:

- podle mechanismu a funkce čtecí hlavy,
- pomocné paměti,
- určenosti přechodů.

Jazyky typu 3 — regulární jazyky

- Význam regulárních jazyků.
- Prostředky specifikace regulárních jazyků:
  - gramatikou typu 3 (a jejími modifikacemi)

    např. $G = (\{A, S\},\ \{0, 1\},\ \{S \to 0S, S \to 1A, A \to 0\},\ S)$

  - konečným automatem

    např. $M = (\{q_0, q_1, q_F\},\ \{0, 1\},\ \delta,\ q_0,\ \{q_F\}),$ $\quad \delta :$

    $$\begin{aligned} \delta(q_0, 0) &= \{q_0\} \\ \delta(q_0, 1) &= \{q_1\} \\ \delta(q_1, 0) &= \{q_F\} \\ \ldots \end{aligned}$$

  - regulárním výrazem

    např. $(1 + 0^+1)0$

Konečný automat

Definice 1.11 Konečným automatem (KA) rozumíme jednocestný iniciální automat M specifikovaný 5-ticí

$$M = (Q, \Sigma, \delta, q_0, F), \qquad \text{kde:}$$

1. Q je konečná množina stavů,
2. Σ je konečná vstupní abeceda,
3. δ je funkce přechodů (přechodová funkce) tvaru $\delta : Q \times \Sigma \to 2^Q$,
4. $q_0 \in Q$ je počáteční stav,
5. $F \subseteq Q$ je množina koncových stavů.

Je-li $\forall q \in Q\ \forall a \in \Sigma : |\delta(q, a)| \leq 1$, pak M nazýváme deterministickým konečným automatem (zkráceně DKA), v případě, že $\exists q \in Q\ \exists a \in \Sigma : |\delta(q, a)| > 1$ pak nedeterministickým konečným automatem (zkráceně NKA).

Deterministický konečným automat je také často specifikován 5-ticí

$$M = (Q, \Sigma, \delta, q_0, F), \qquad \text{kde:}$$

- δ je parciální funkce tvaru $\delta : Q \times \Sigma \to Q$,
- a význam ostatních složek zůstává zachován.

Je-li přechodová funkce δ totální, pak M nazýváme úplně definovaným deterministickým konečným automatem.

Dále budeme obvykle pracovat s touto specifikací DKA.

Lemma 1.1 Ke každému DKA M existuje „ekvivalentní“ úplně definovaný DKA M'.

Důkaz. (idea) Množinu stavů automatu M' rozšíříme o nový, nekoncový stav (anglicky označovaný jako *SINK* stav) a s využitím tohoto stavu doplníme prvky přechodové funkce δ' automatu M' tak, aby byla totální. $\square$

Příklad 1.5

$$M_1 = (\{q_0, q_1, q_2, q_F\},\ \{0,1\},\ \delta,\ q_0,\ \{q_F\})$$

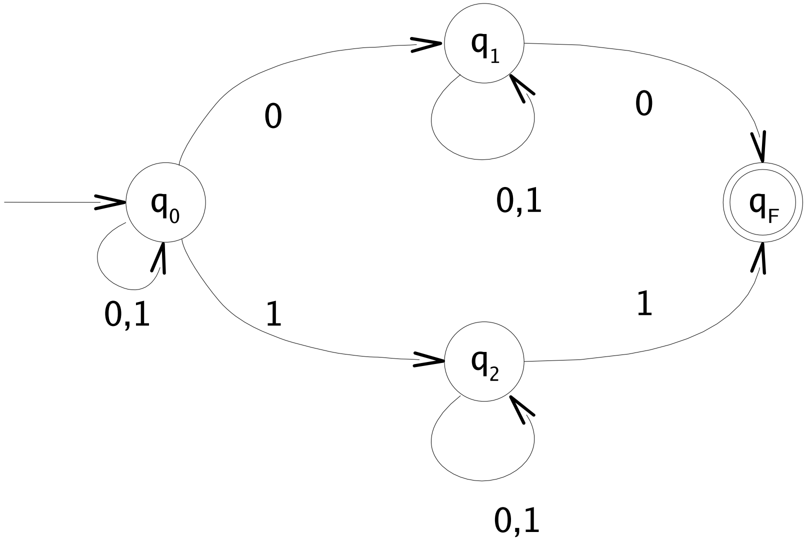
$$\begin{array}{lll} \delta: & \delta(q_0,0) = \{q_0,q_1\} & \delta(q_0,1) = \{q_0,q_2\} \\ & \delta(q_1,0) = \{q_1,q_F\} & \delta(q_1,1) = \{q_1\} \\ & \delta(q_2,0) = \{q_2\} & \delta(q_2,1) = \{q_2,q_F\} \\ & \delta(q_F,0) = \emptyset & \delta(q_F,1) = \emptyset \end{array}$$

- Alternativní způsoby reprezentace funkce δ:

1. maticí (přechodů)

	0	1
q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$
q_1	$\{q_1, q_F\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2, q_F\}$
q_F	$\emptyset$	$\emptyset$

2. diagramem přechodů



Definice 1.12 Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat (tj. NKA).

- Konfigurace C konečného automatu M je uspořádaná dvojice

$$C = (q, w), \quad (q, w) \in Q \times \Sigma^*$$

kde q je aktuální stav, w je dosud nezpracovaná část vstupního řetězce.

- Počáteční konfigurace je konfigurace $(q_0, a_1 a_2 \dots a_n)$.

- Koncová konfigurace je konfigurace $(q_F, \varepsilon), \quad q_F \in F$.

- Přechodem automatu M rozumíme binární relaci $\underset{M}{\vdash}$ v množině konfigurací C

$$\underset{M}{\vdash} \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*)$$

která je definována takto:

$$(q, w) \underset{M}{\vdash} (q', w') \overset{def.}{\iff} w = aw' \ \wedge \ q' \in \delta(q, a) \ \ pro \ q, q' \in Q, a \in \Sigma, w, w' \in \Sigma^*$$

Relace $\overset{k}{\underset{M}{\vdash}}$, $\overset{+}{\underset{M}{\vdash}}$, $\overset{*}{\underset{M}{\vdash}}$ mají obvyklý význam, tj. $k-$tá mocnina relace $\vdash$, tranzitivní a tranzitivní a reflexivní uzávěr relace $\vdash$.

❖ Řetězec w přijímaný NKA M je definován takto: $(q_0, w) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon), \;\; q \in F$.

❖ Jazyk $L(M)$ přijímaný NKA M je definován takto:

$$L(M) = \{w \mid (q_0, w) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon) \wedge q \in F\}.$$

Příklad 1.6 Uvažujme NKA M_1 z příkladu 1.5. Platí:

$$(q_0, 1010) \vdash (q_0, 010) \vdash (q_1, 10) \vdash (q_1, 0) \vdash (q_f, \varepsilon)$$

a tedy: $(q_0, 1010) \overset{*}{\vdash} (q_f, \varepsilon)$

Neplatí například $(q_0, \varepsilon) \overset{*}{\vdash} (q_f, \varepsilon)$

Vyjádření jazyka $L(M_1)$:

$$L(M_1) = \{w \mid w \in \{0,1\}^* \wedge w \text{ končí symbolem, který je již v řetězci } w \text{ obsažen}\}$$

$\square$

Ekvivalence NKA a DKA

Věta 1.6 Každý NKA M lze převést na DKA M' tak, že

$$L(M) = L(M').$$

Důkaz.

1. Nalezneme algoritmus převodu $M \to M'$ (níže).
2. Ukážeme, že $L(M) = L(M')$ tj. ukážeme, že platí:
   (a) $L(M) \subseteq L(M')$ a současně,
   (b) $L(M') \subseteq L(M)$.

$\square$

Převod NKA na ekvivalentní DKA

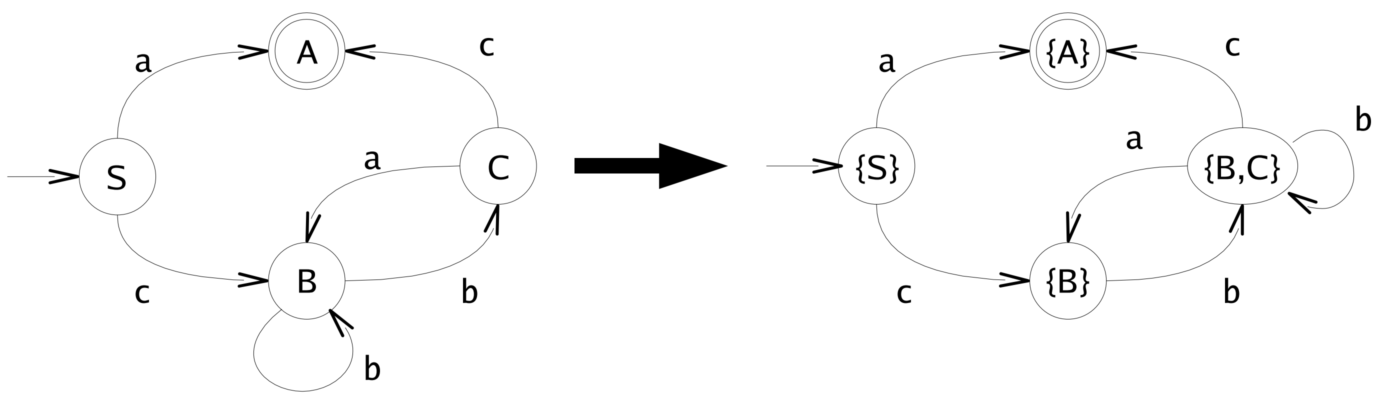
Algoritmus 1.1

- Vstup: NKA $M = (Q, \Sigma, \delta, q_0, F)$
- Výstup: DKA $M' = (Q', \Sigma, \delta', q'_0, F')$
- Metoda:
  1. Polož $Q' = 2^Q \setminus \{\emptyset\}$.
  2. Polož $q'_0 = \{q_0\}$.
  3. Polož $F' = \{S \mid S \in 2^Q \wedge S \cap F \neq \emptyset\}$.
  4. Pro všechna $S \in 2^Q \setminus \{\emptyset\}$ a pro všechna $a \in \Sigma$ polož:
     - $\delta'(S, a) = \bigcup\limits_{q \in S} \delta(q, a)$, je-li $\bigcup\limits_{q \in S} \delta(q, a) \neq \emptyset$.
     - Jinak $\delta'(S, a)$ není definována.

Příklad 1.7 Uvažujme NKA $M_2 = (\{S, A, B, C\},\ \{a, b, c\},\ \delta,\ S,\ \{A\})$
$\delta : \quad \delta(S, a) = \{A\} \quad \delta(S, c) = \{B\} \quad \delta(B, b) = \{B, C\} \quad \delta(C, a) = \{B\} \quad \delta(C, c) = \{A\}$

K nalezení funkce δ' příslušného DKA aplikujeme zkrácený postup, využívající skutečnosti, že řada stavů z 2^Q může být nedostupných:

1. Počáteční stav: $\{S\}$
2. $\delta'(\{S\}, a) = \{A\}$ — koncový stav
   $\delta'(\{S\}, c) = \{B\}$
3. $\delta'(\{B\}, b) = \{B, C\}$
4. $\delta'(\{B, C\}, a) = \delta(B, a) \cup \delta(C, a) = \{B\}$
   $\delta'(\{B, C\}, b) = \{B, C\}$ $\delta'(\{B, C\}, c) = \{A\}$



Konečné automaty a jazyky typu 3

Definice 1.13

❖ Gramatika $G = (N, \Sigma, P, S)$ s pravidly tvaru:

$$\begin{array}{ll} A \rightarrow xB & A, B \in N, x \in \Sigma^* \text{ nebo} \\ A \rightarrow x & x \in \Sigma^*, \end{array}$$

resp. tvaru:

$$\begin{array}{ll} A \rightarrow Bx & A, B \in N \\ A \rightarrow x & x \in \Sigma^* \end{array}$$

se nazývá pravá lineární, resp. levá lineární, gramatika.

❖ Gramatika $G = (N, \Sigma, P, S)$ s pravidly tvaru:

$$\begin{array}{lll} & A \rightarrow aB & A, B \in N, a \in \Sigma \\ & A \rightarrow a & a \in \Sigma \\ \text{případně} & S \rightarrow \varepsilon & \text{pokud se } S \text{ neobjevuje na pravé straně žádného pravidla} \end{array}$$

resp. s pravidly tvaru:

$$\begin{array}{lll} & A \rightarrow Ba & A, B \in N, a \in \Sigma \\ & A \rightarrow a & a \in \Sigma \\ \text{případně} & S \rightarrow \varepsilon & \text{pokud se } S \text{ neobjevuje na pravé straně žádného pravidla} \end{array}$$

se nazývá pravá regulární, resp. levá regulární, gramatika.

Poznámka. Gramatika $G = (N, \Sigma, P, S)$ s pravidly tvaru $A \to xBy$ nebo $A \to x$, kde $A, B \in N$ a $x, y \in \Sigma^*$ se nazývá lineární gramatika.
Označme:

- $\mathcal{L}_{PL}$ — všechny jazyky generované pravými lineárními gramatikami,
- $\mathcal{L}_{LL}$ — všechny jazyky generované levými lineárními gramatikami,
- $\mathcal{L}_{L}$ — všechny jazyky generované lineárními gramatikami.

Platí:

$$\mathcal{L}_{PL} = \mathcal{L}_{LL} \text{ a } \mathcal{L}_{PL} \subset \mathcal{L}_{L}$$

$\square$

Lemma 1.2 Každá pravá lineární gramatika $G = (N, \Sigma, P, S)$ může být převedena na gramatiku $G' = (N', \Sigma', P', S')$, kde P' obsahuje pouze pravidla tvaru

$$A \to aB \qquad A, B \in N', a \in \Sigma \text{ nebo}$$
$$A \to \varepsilon$$

tak, že $L(G) = L(G')$.

Důkaz. Množinu P' vytvoříme takto:

1. Pravidla z P tvaru

$$A \to aB \qquad A, B \in N', a \in \Sigma \text{ nebo}$$
$$A \to \varepsilon$$

   zařadíme do P'.

2. Každé pravidlo tvaru

$$A \to a_1 a_2 \dots a_n B, \quad n \geq 2$$

   z P nahradíme v P' soustavou pravidel:

$$\begin{array}{l} A \to a_1 A_1 \\ A_1 \to a_2 A_2 \\ \vdots \\ A_{n-1} \to a_n B \end{array}$$

Důkaz pokračuje dále.

Pokračování důkazu.

3. Každé pravidlo tvaru

$$A \to a_1 a_2 \dots a_n, \quad n \geq 1$$

   z P nahradíme v P' soustavou pravidel:

$$\begin{array}{l} A \to a_1 A'_1 \\ A'_1 \to a_2 A'_2 \\ \vdots \\ A'_{n-1} \to a_n A'_n \\ A'_n \to \varepsilon \end{array}$$

4. Odstraníme (zbývající) tzv. jednoduchá pravidla tvaru $A \to B$. Nyní se již snadno dokáže ekvivalence G a G' tj. $L(G) = L(G')$

$\square$

Příklad 1.8 Uvažujme gramatiku s pravidly

$$\underline{S \to abcA} \mid \underline{aB}$$
$$A \to B \mid \underline{\varepsilon}$$
$$B \to \underline{cA} \mid \underline{bbc}$$

Aplikací Lemmy 1.1 obdržíme pravidla ekvivalentní gramatiky. Nové nonterminály budeme označovat $X, Y, \ldots$:

$$\begin{array}{ll} \underline{S \to aX} \mid \underline{aB} & \underline{A \to B} \mid \underline{\varepsilon} \\ \underline{X \to bY} & B \to \underline{cA} \mid \underline{bZ} \\ \underline{Y \to cA} & \underline{U \to cV} \\ \underline{Z \to bU} & \underline{V \to \varepsilon} \end{array}$$

Po odstranění jednoduchého pravidla $A \to B$ dostaneme výslednou gramatiku:

$$\begin{array}{ll} S \to aX \mid aB & \underline{A \to cA \mid bZ} \mid \varepsilon \\ X \to bY & B \to cA \mid bZ \\ Y \to cA & U \to cV \\ Z \to bU & V \to \varepsilon \end{array}$$

Převod gramatiky typu 3 na NKA

Věta 1.7 Nechť $\mathcal{L}_M$ je množina (třída) všech jazyků přijímaných konečnými automaty a nechť L je libovolný jazyk typu 3 ($L \in \mathcal{L}_3$). Pak existuje konečný automat M takový, že:

$$L = L(M), \text{ tj. } \mathcal{L}_3 \subseteq \mathcal{L}_M.$$

Důkaz.

1. Podle věty 1.7 můžeme předpokládat, že $L = L(G)$, kde gramatika G obsahuje pouze pravidla tvaru:

   $$A \rightarrow aB \text{ nebo } A \rightarrow \varepsilon$$

2. Ke gramatice $G = (N, \Sigma, P, S)$ sestrojíme NKA $M = (Q, \Sigma, \delta, q_0, F)$ takto:
   (a) $Q = N$
   (b) $\Sigma = \Sigma$
   (c) $\delta : \delta(A, a)$ obsahuje B, jestliže $A \rightarrow aB$ je v P
   (d) $q_0 = S$
   (e) $F = \{A \mid A \rightarrow \varepsilon \text{ je v } P\}$

Důkaz pokračuje dále.

Pokračování důkazu.

3. Matematickou indukcí ukážeme, že $L(G) = L(M)$. Indukční hypotézu formulujeme obecněji ve tvaru:

$$\forall A \in N : A \overset{i+1}{\underset{G}{\Rightarrow}} w \quad \Longleftrightarrow \quad (A, w) \overset{i}{\underset{M}{\vdash}} (C, \varepsilon) \text{ pro } C \in F, w \in \Sigma^*$$

Pro $i = 0$ dostáváme

$$A \Rightarrow \varepsilon \quad \Longleftrightarrow \quad (A, \varepsilon) \underset{0}{\vdash} (A, \varepsilon) \text{ pro } A \in F$$

a tvrzení tedy platí.

Nyní předpokládejme, že dokazovaná hypotéza platí pro $i > 0$ a položme $w = ax$, kde $a \in \Sigma$ a $|x| = i - 1$.

Důkaz pokračuje dále.

Pokračování důkazu.

3. pokračování.

   Dále předpokládejme $A \Rightarrow aB \overset{i}{\Rightarrow} ax$,

   z indukční hypotézy plyne $B \overset{i}{\Rightarrow} x \iff (B, x) \overset{i-1}{\vdash} (C, \varepsilon), C \in F$

   a z definice funkce δ: $A \Rightarrow aB \iff B \in \delta(A, a)$

   Dohromady tedy

$$A \Rightarrow aB \overset{i}{\Rightarrow} ax = w' \iff (A, ax) \vdash (B, x) \overset{i-1}{\vdash} (C, \varepsilon), C \in F$$

   tedy $A \overset{i+1}{\Rightarrow} w' \iff (A, w') \overset{i}{\vdash} (C, \varepsilon), C \in F$

   tj. tvrzení platí i pro $i + 1$.

   Pro případ $A = S$ je dokázaná hypotéza tvrzením věty, tj.

$$\forall w' \in \Sigma^* : S \overset{*}{\Rightarrow} w' \iff (S, w') \overset{*}{\vdash} (C, \varepsilon), C \in F, \text{ tj. } L(G) = L(M)$$

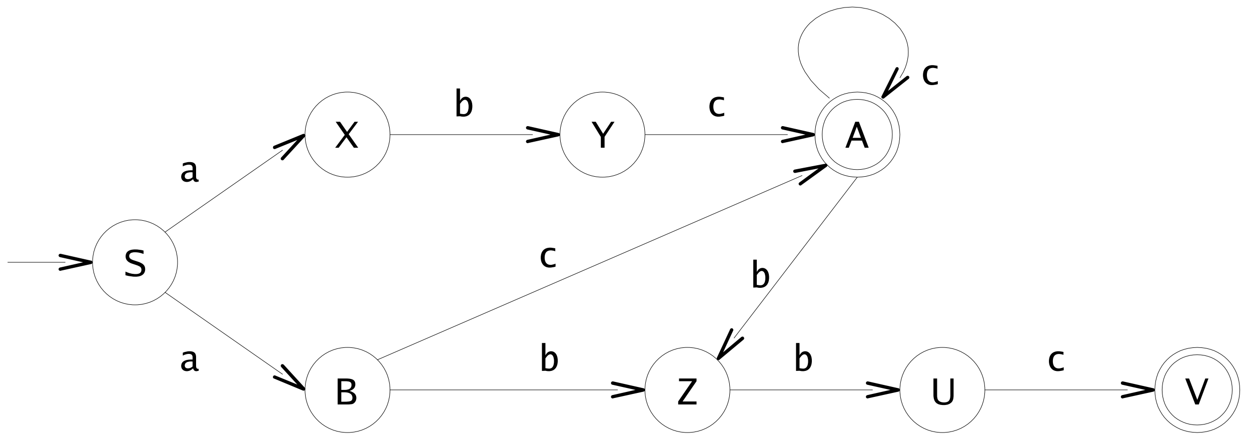
$\square$

Příklad 1.9

Gramatika z příkladu 1.8 $G = (\{S, A, B, U, V, X, Y, Z\},\ \{a, b, c\},\ P,\ S)$, má pravidla P:

$$\begin{array}{ll} S \to aX \mid aB & \qquad A \to cA \mid bZ \mid \varepsilon \\ X \to bY & \qquad B \to cA \mid bZ \\ Y \to cA & \qquad U \to cV \\ Z \to bU & \qquad V \to \varepsilon \end{array}$$

Takové gramatice odpovídá konečný automat:



Převod NKA na gramatiku typu 3

Věta 1.8 Nechť M je NKA. Pak existuje gramatika G typu 3 taková, že:

$L(M) = L(G)$, tj. $\mathcal{L}_M \subseteq \mathcal{L}_3$.

Důkaz. Nechť $M = (Q, \Sigma, \delta, q_0, F)$. Předpokládejme, že M je NKA. Nechť $G = (Q, \Sigma, P, q_0)$ je gramatika, jejíž pravidla jsou definována takto:

1. je-li $\delta(q, a) = r$, pak P obsahuje pravidlo $q \rightarrow ar$
2. je-li $p \in F$, pak P obsahuje pravidlo $p \rightarrow \varepsilon$
3. jiná pravidla množina P neobsahuje.

G je zřejmě typu 3 a indukcí lze dokázat, že platí $L(G) = L(M)$.

$\square$

Příklad 1.10 Uvažujme KA $M_3 = (\{A, B, C, D\},\ \{a, b, c\},\ \delta,\ A,\ \{C, D\})$, kde

$$\begin{array}{lll}\delta: & \delta(A, a) = B & \delta(C, c) = D \\ & \delta(B, b) = A & \delta(D, a) = A \\ & \delta(B, c) = B & \delta(D, b) = D \\ & \delta(B, a) = C & \end{array}$$

Gramatika G typu 3, která generuje jazyk $L(M_3)$, má tvar:

$$G = (\{A, B, C, D\},\ \{a, b, c\},\ P,\ A)$$

$$\begin{array}{lll}P: & A \to aB & C \to cD \mid \varepsilon \\ & B \to bA \mid cB \mid aC & D \to aA \mid bD \mid \varepsilon\end{array}$$

Po odstranění ε-pravidel (algoritmus viz přednáška 4), získáme ekvivalentní pravou regulární gramatiku G' s pravidly:

$$\begin{array}{ll}A \to aB & C \to cD \mid c \\ B \to bA \mid cB \mid aC \mid a & D \to aA \mid bD \mid b\end{array}$$

Regulární množiny a výrazy

Regulární množiny

Definice 1.14 Nechť Σ je konečná abeceda. Regulární množinu nad Σ definujeme rekurzívně takto:

1. $\emptyset$ (tj. prázdná množina) je regulární množina nad Σ,
2. $\{\varepsilon\}$ je regulární množina nad Σ,
3. $\{a\}$ je regulární množina nad Σ pro všechny $a \in \Sigma$,
4. jsou-li P a Q regulární množiny nad Σ, pak také
   (a) $P \cup Q$,
   (b) $P.Q$,
   (c) P^*

   jsou regulární množiny nad Σ.
5. Žádné jiné množiny, než ty, které lze získat pomocí výše uvedených pravidel, nejsou regulárními množinami.

Příklad 1.11 $L = (\{a\} \cup \{d\}).(\{b\}^*).\{c\}$ je regulární množina nad $\Sigma = \{a, b, c, d\}$.

Regulární výrazy

Definice 1.15 Regulární výrazy nad Σ a regulární množiny, které označují, jsou rekurzívně definovány takto:

1. $\emptyset$ je regulární výraz označující regulární množinu $\emptyset$,
2. ε je regulární výraz označující regulární množinu $\{\varepsilon\}$,
3. a je regulární výraz označující regulární množinu $\{a\}$ pro všechny $a \in \Sigma$,
4. jsou-li p, q regulární výrazy označující regulární množiny P a Q, pak
   (a) $(p + q)$ je regulární výraz označující regulární množinu $P \cup Q$,
   (b) (pq) je regulární výraz označující regulární množinu $P.Q$,
   (c) (p^*) je regulární výraz označující regulární množinu P^*.
5. Žádné jiné regulární výrazy nad Σ neexistují.

❖ Konvence:

1. Regulární výraz p^+ značí regulární výraz pp^*.
2. Abychom minimalizovali počet používaných závorek, stanovujeme priority operátorů:
   1. $*$, $+$ (iterace – nejvyšší priorita),
   2. $.$ (konkatenace),
   3. $+$ (alternativa).

Příklad 1.12

1. 01 odpovídá $\{01\}$.
2. 0^* odpovídá $\{0\}^*$.
3. $(0+1)^*$ odpovídá $\{0,1\}^*$.
4. $(0+1)^*011$ značí množinu řetězců nad $\{0,1\}$ končících 011.
5. $(a+b)(a+b+0+1)^*(0+1)$ značí množinu řetězců nad $\{a,b,0,1\}$, které začínají symbolem a nebo b a končí symbolem 0 nebo 1.