Teorie jazyků a automatů

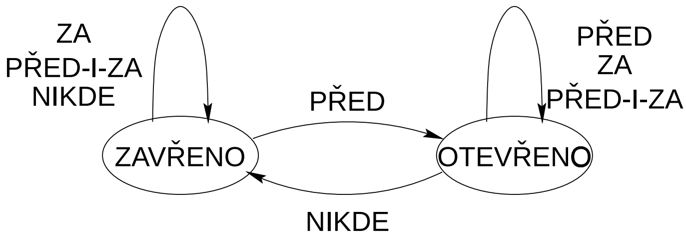
- konečné automaty, regulární jazyky
- bezkontextové gramatiky a jazyky, zásobníkové automaty
- Turingovy stroje, Chomského hierarchie

Vyčíslitelnost a složitost

- algoritmicky rozhodnutelné problémy
- časová (a paměťová) složitost algoritmů a problémů

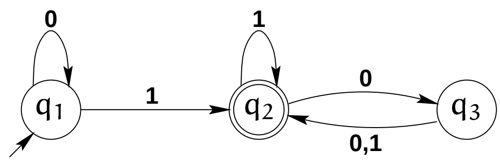
---

http://www.cs.vsb.cz/jancar
(http://www.cs.vsb.cz/jancar/TJAA/tjaa.htm)
Petr.Jancar@vsb.cz



	PŘED	ZA	PŘ-I-ZA	NIKDE
ZAV	OTEV	ZAV	ZAV	ZAV
OTEV	OTEV	OTEV	OTEV	ZAV

ements



přijímá:
1101, 010101, ...

nepřijímá:
0110, 0010, ...

Konečný automat (KA)

$A = (Q, \Sigma, \delta, q_0, F)$

Q ... konečná množina *stavů*

Σ ... konečná množina (vstupních) *symbolů* (*abeceda*)

$\delta : Q \times \Sigma \to Q$ ... *přechodová funkce*

$q_0 \in Q$ ... *počáteční stav*

$F \subseteq Q$ ... množina *přijímajících (koncových) stavů*

abeceda .... označujeme Σ, Γ, Δ, ...

prvky abecedy (*písmena*) ... označujeme $a, b, c, \ldots$

slovo .... $u, v, w, \ldots$, $u = a_1 a_2 \ldots a_n$

délka slova ... $|u|$

prázdné slovo ... ε ($|\varepsilon| = 0$)

zřetězení slov ... $u \cdot v$, stručněji uv

značení $u^0 = \varepsilon$, $u^1 = u$, $u^2 = uu$, ... $u^n = uu..u$ (n-krát)

u je *podslovem* (*prefixem*, *sufixem*) slova v $\Leftrightarrow_{df}$ ...

Σ^* ... množina všech slov v abecedě Σ

Jazyk nad abecedou Σ ... $L \subseteq \Sigma^*$

$A = (Q, \Sigma, \delta, q_0, F)$

$\delta^* : Q \times \Sigma^* \to Q$

1. $\delta^*(q, \varepsilon) = q$,
2. $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

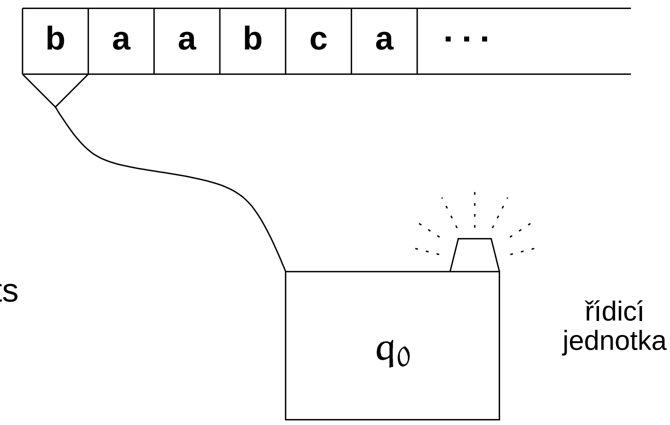
Slovo $w \in \Sigma^*$ je *přijímáno* automatem A $\Leftrightarrow_{df}$ $\delta(q_0, w) \in F$

Jazykem rozpoznávaným (přijímaným) automatem A rozumíme jazyk

$L(A) = \{w \mid$ slovo w je přijímáno $A\}$

Jazyk L je *regulární* (tj. rozpoznatelný konečným automatem) $\Leftrightarrow_{df}$ existuje KA A tž. $L(A) = L$

ements



Konfigurace konečného automatu $A = (Q, \Sigma, \delta, q_0, F)$:

$K = (q, w)$ $(q \in Q, w \in \Sigma^*)$

Na množině všech konfigurací automatu A definujeme relaci $\vdash_A$ takto:

$(q, aw) \vdash_A (q', w) \Leftrightarrow_{df} \delta(q, a) = q'$

$K_1 \vdash_A K_2$ čteme ...

Výpočet automatu A:

$K_0, K_1, K_2, \ldots, K_n$, kde $K_i \vdash_A K_{i+1}$ pro $i = 0, 1, \ldots, n{-}1$

výpočet $K_0, K_1, K_2, \ldots, K_n$ je *přijímajícím výpočtem* pro slovo w $\Leftrightarrow_{df}$
$K_0 = (q_0, w)$,
$K_n = (q, \varepsilon)$ pro nějaký $q \in F$

relace ρ *na množině* M ... $\rho \subseteq M \times M$

$x\rho y$ místo $(x, y) \in \rho$

ρ je *reflexivní* $\Leftrightarrow_{df}$ $\forall x \in M : x\rho x$

ρ je *tranzitivní* $\Leftrightarrow_{df}$
$\forall x, y, z \in M : (x\rho y \wedge y\rho z) \Longrightarrow x\rho z$

Reflexivním a tranzitivním uzávěrem relace ρ ... nejmenší relace ρ', která obsahuje ρ a je reflexivní i tranzitivní

$\vdash_A^*$ je reflexivním a tranzitivním uzávěrem relace $\vdash_A$

$K_1 \vdash_A^* K_2$ čteme ...

Slovo $w \in \Sigma^*$ *je přijímáno* KA A $\Leftrightarrow_{df}$ $(q_0, w) \vdash^* (q, \varepsilon)$ pro nějaký $q \in F$.

$\delta(q, w) = q'$ ... $(q, w) \vdash^* (q', \varepsilon)$

$q \xrightarrow{w} q'$

Stav q automatu $(Q, \Sigma, \delta, q_0, F)$ je *dosažitelný* $\Leftrightarrow_{df}$ existuje $w \in \Sigma^*$ tž. $\delta(q_0, w) = q$.

Množina *dosažitelných stavů* automatu $(Q, \Sigma, \delta, q_0, F)$ je *nejmenší* množina $K \subseteq Q$ splňující tyto dvě podmínky:

1/ $q_0 \in K$,

2/ jestliže $q \in K$ a $q' = \delta(q, a)$ pro nějaké $a \in \Sigma$, potom $q' \in K$.

Tvrz. Ke každému KA A lze zkonstruvat KA A', v němž každý stav je dosažitelný a $L(A') = L(A)$.

Tvrz. Existuje algoritmus, který pro zadaný KA A rozhodne, zda $L(A)$ je neprázdný.

Tvrz. Pro konečný automat A s n stavy je $L(A)$ neprázdný právě tehdy, když existuje $w \in L(A)$ délky menší než n ($|w| < n$).

Tvrz. Pro konečný automat A s n stavy je $L(A)$ nekonečný právě tehdy, když existuje $w \in L(A)$ splňující
$n \leq |w| < 2n$.

Relace ρ na množině A je *(částečným) uspořádáním* $\Leftrightarrow_{df}$ ρ je

- reflexivní ($\forall x \in A : x\rho x$)
- tranzitivní ($\forall x, y, z \in A :$ $(x\rho y \wedge y\rho z) \Longrightarrow x\rho z)$)
- antisymetrická ($\forall x, y \in A :$ $(x\rho y \wedge y\rho x) \Longrightarrow x = y)$)

Uspořádáním obvykle myslíme úplné (neboli lineární) uspořádání, což znamená navíc

- $\forall x, y \in A : (x\rho y \vee y\rho x)$

Obvykle: pro uspořádání znaky jako $\leq$
příslušné 'ostré' $<$
$x < y \;\; \Leftrightarrow_{df} \;\; x \leq y \wedge x \neq y$

Pro $\Sigma = \{a_1, a_2, \ldots, a_k\}$ s (abecedním) uspořádáním prvků (např. $a_1 < a_2 < \ldots < a_k$) lze přirozeně definovat uspořádání na Σ^* (podle délky a abecedně); označme $\leq_L$.

Např. pro $\Sigma = \{a, b\}$, kde $a < b$, máme

$\varepsilon <_L a <_L b <_L aa <_L ab <_L ba <_L bb <_L aaa <_L aab <_L \ldots$

Tvrz. Pro abecedu Σ je Σ^* nekonečná spočetná množina.

Pozn.: Proto i konečných automatů (a regulárních jazyků) je spočetně mnoho !

Z Cantorovy věty – mohutnost M je ostře menší než mohutnost $\mathcal{P}(M)$:

Tvrz. Jazyků nad (neprázdnou) abecedou Σ je nespočetně mnoho.

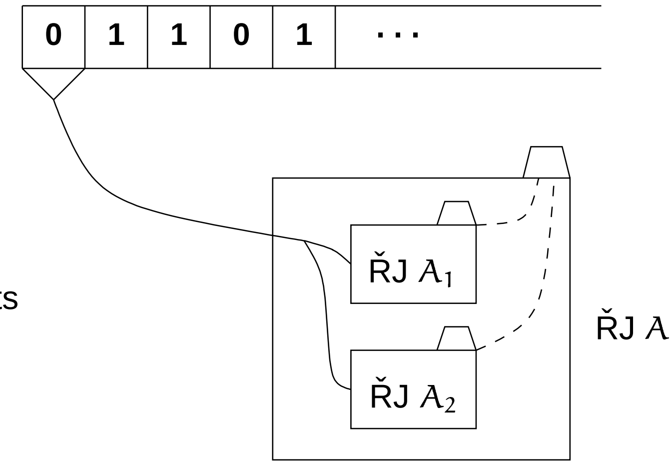
Máme-li dán konečný automat $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ bez nedosažitelných stavů a uspořádání (prvků) abecedy Σ (které indukuje uspořádání $<_L$ na Σ^*), pak řekneme, že $\mathcal{A}$ je v *normovaném tvaru*, jestliže

- $Q = \{1, 2, \ldots, n\}$ (pro nějaké $n \geq 1$),
- 1 je počáteční stav,
- označíme-li pro každý stav $i \in \{1, 2, \ldots, n\}$ symbolem u_i nejmenší slovo v uspořádání $<_L$, pro něž $\delta(1, u_i) = i$, pak pro $i < j$ platí $u_i <_L u_j$.

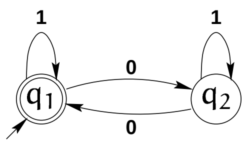
KA $\mathcal{A}$ pro rozpoznávání jazyka

$L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$

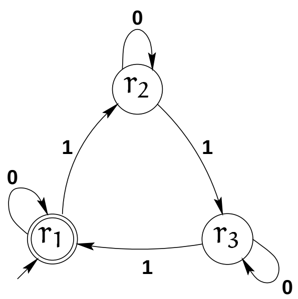
ements



ements

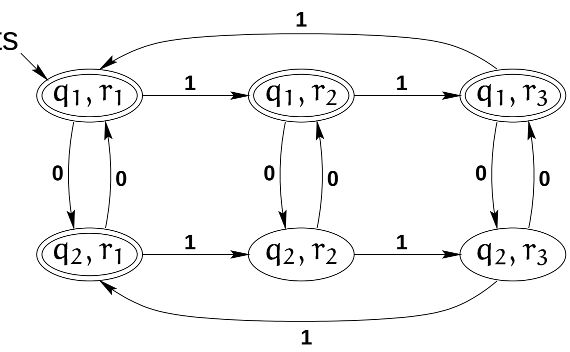


$\mathcal{A}_1$



$\mathcal{A}_2$

ements



Věta. Jestliže $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak $(L_1 \cup L_2)$ i $(L_1 \cap L_2)$ jsou regulární.

Důkaz. Nechť $L_1 = L(\mathcal{A}_1)$, $L_2 = L(\mathcal{A}_2)$ pro konečné automaty $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$,
$\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.

Definujme $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ takto:

- $Q = Q_1 \times Q_2$
- pro každé $q_1 \in Q_1$, $q_2 \in Q_2$, $a \in \Sigma$:
  $\delta(\,(q_1, q_2), a\,) = (\,\delta_1(q_1, a), \delta_2(q_2, a)\,)$
- $q_0 = (q_{01}, q_{02})$

Platí: pro vš. $q_1 \in Q_1$, $q_2 \in Q_2$, $w \in \Sigma^*$:
$\delta(\,(q_1, q_2), w\,) = (\,\delta_1(q_1, w), \delta_2(q_2, w)\,)$.

Pro docílení $L(\mathcal{A}) = L_1 \cup L_2$ položíme
- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

pro docílení $L(\mathcal{A}) = L_1 \cap L_2$ položíme
- $F = F_1 \times F_2$.

Regulárními operacemi s jazyky nazýváme operaci

- *sjednocení*
  $L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$
- *zřetězení*
  $L_1 \cdot L_2 = \{uv \mid u \in L_1, v \in L_2\}$
- *iterace*
  $L^* = \bigcup_{n=0}^{\infty} L^n$,
  kde L^n je definováno induktivně:
  $L^0 = \{\varepsilon\}$, $L^{n+1} = L \cdot L^n$.

Pozn. L^* lze definovat také např. takto: $L^* = \{w \mid$ ex. $n \geq 0$ a slova $u_1, u_2, \ldots, u_n \in L$ tak, že $w = u_1 u_2 \ldots u_n\}$.

Nedeterm. konečný automat (NKA)

$A = (Q, \Sigma, \delta, I, F)$

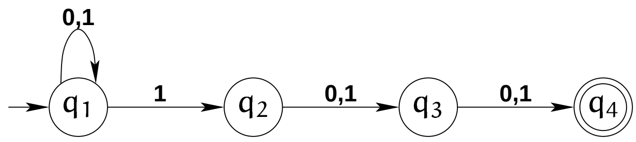
Q ... konečná množina *stavů*

Σ ... konečná *abeceda*

$\delta : Q \times \Sigma \to \mathcal{P}(Q)$ ... *přechodová funkce*

$I \subseteq Q$ ... množina *počátečních stavů*

$F \subseteq Q$ ... množina *přijímajících stavů*



přijímá např.: **01101**

nepřijímá např.: **011010**

Konfigurace NKA $A = (Q, \Sigma, \delta, I, F)$:

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

Na množině všech konfigurací automatu A definujeme relaci $\vdash_A$ takto:

$(q, aw) \vdash_A (q', w) \;\Leftrightarrow_{df}\; \delta(q, a) \ni q'$

$K_1 \vdash_A K_2$ čteme ...

Výpočet automatu A:

$K_0, K_1, K_2, \ldots, K_n$, kde $K_i \vdash_A K_{i+1}$ pro $i = 0, 1, \ldots, n-1$

výpočet $K_0, K_1, K_2, \ldots, K_n$ je *přijímajícím výpočtem* pro slovo $w \quad \Leftrightarrow_{df}$
$K_0 = (r, w)$ pro nějaký $r \in I$,
$K_n = (q, \varepsilon)$ pro nějaký $q \in F$

Slovo $w \in \Sigma^*$, tvaru $w = a_1 a_2 \ldots a_n$, je *přijímáno* NKA $A = (Q, \Sigma, \delta, I, F)$ právě když ex. stavy $q_0, q_1, \ldots, q_n$ takové, že

- $q_0 \in I$
- $q_n \in F$
- $\delta(q_{i-1}, a_i) \ni q_i$ pro $i = 1, 2, \ldots, n$

Přechodovou funkci můžeme přirozeně zobecnit na

$$\delta : Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

resp. ještě obecněji na

$$\delta : \mathcal{P}(Q) \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

A sice induktivně:

1/ $\delta(K, \varepsilon) = K$
2/ $\delta(K, wa) = \bigcup_{q \in \delta(K, w)} \delta(q, a)$

Tedy $L(A) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}$

Věta. NKA rozpoznávají právě regulární jazyky (tedy právě ty jako DKA).

Důkaz. a/ DKA $A = (Q, \Sigma, \delta, q_0, F)$ je de facto spec. NKA

(formálně parametr q_0 nahradíme $\{q_0\}$ a funkci δ nahradíme δ', kde $\delta'(q, a) = \{\, \delta(q, a)\,\}$)

b/ Pro NKA $A = (Q, \Sigma, \delta, I, F)$ lze definovat

DKA $A_1 = (\mathcal{P}(Q), \Sigma, \delta_1, I, F_1)$, kde

$\delta_1(K, a) = \delta(K, a)$ ($K \in \mathcal{P}(Q)$)

$F_1 = \{\, K \subseteq Q \mid K \cap F \neq \emptyset \}$

$L(A) = L(A_1)$ je zřejmé:

$w \in L(A) \iff \delta(I, w) \cap F \neq \emptyset \iff \delta_1(I, w) \in F_1 \iff w \in L(A_1)$

Zobecněný NKA (ZNKA)
(NKA s ε-přechody)

$A = (Q, \Sigma, \delta, I, F)$

Q ... konečná množina *stavů*

Σ ... konečná *abeceda*

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ ... *přechodová funkce*

$I \subseteq Q$ ... množina *počátečních stavů*

$F \subseteq Q$ ... množina *přijímajících stavů*

Slovo $w = a_1 a_2 \dots a_n$ je *přijímáno* ZNKA A $\Leftrightarrow_{df}$ existují stavy
$q_1^0, q_2^0, \dots, q_{m_0}^0$,

$q_1^1, q_2^1, \dots, q_{m_1}^1$,

$\dots$,

$q_1^n, q_2^n, \dots, q_{m_n}^n$

($m_i \geq 1$ pro $i = 0, 1, 2, \dots, n$)

tak, že

1/ $q_1^0 \in I$

2/ $q_{m_n}^n \in F$

3/ $\delta(q_{m_{i-1}}^{i-1}, a_i) \ni q_1^i$ pro vš. $i = 1, 2, \dots, n$

4/ $\delta(q_{j-1}^i, \varepsilon) \ni q_j^i$ pro vš. $i = 0, 1, 2, \dots, n$, $j = 2, 3, \dots, m_i$

Konfigurace ZNKA $A = (Q, \Sigma, \delta, I, F)$:

$K = (q, w)$ ($q \in Q$, $w \in \Sigma^*$)

Na množině všech konfigurací automatu A definujeme relaci $\vdash_A$ takto:

$(q, w) \vdash_A (q', w) \Leftrightarrow_{df} \delta(q, \varepsilon) \ni q'$

a pro $a \in \Sigma$

$(q, aw) \vdash_A (q', w) \Leftrightarrow_{df} \delta(q, a) \ni q'$

Výpočet automatu A:

$K_0, K_1, K_2, \dots, K_n$, kde $K_i \vdash_A K_{i+1}$ pro $i = 0, 1, \dots, n-1$

výpočet $K_0, K_1, K_2, \dots, K_n$ je *přijímajícím výpočtem* pro slovo w $\Leftrightarrow_{df}$
$K_0 = (r, w)$ pro nějaký $r \in I$,
$K_n = (q, \varepsilon)$ pro nějaký $q \in F$

ZNKA $A = (Q, \Sigma, \delta, I, F)$

$E(K)$ ... množina stavů dosažitelných z K jen pomocí ε-šipek:

$E(K)$ je nejmenší množina splňující

1/ $K \subseteq E(K)$,

2/ jestliže $q \in E(K)$ a $q' \in \delta(q, \varepsilon)$, potom $q' \in E(K)$.

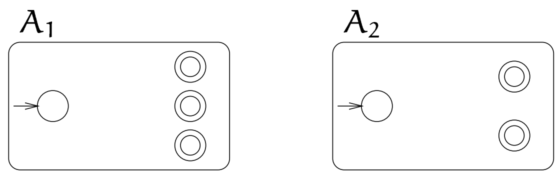
Rozšiřme $\delta : \mathcal{P}(Q) \times \Sigma^* \longrightarrow \mathcal{P}(Q)$ takto:

1/ $\delta(K, \varepsilon) = E(K)$,

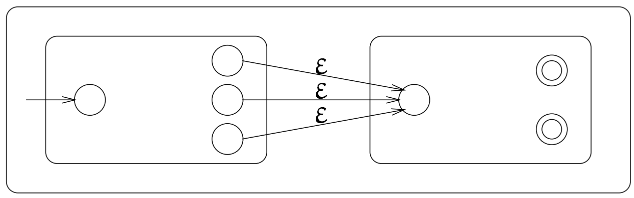
2/ $\delta(K, wa) = \bigcup_{q \in \delta(K,w)} E(\delta(q, a))$.

Tedy $L(A) = \{ w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset \}$

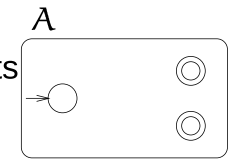
Věta. ZNKA rozpoznávají právě regulární jazyky.

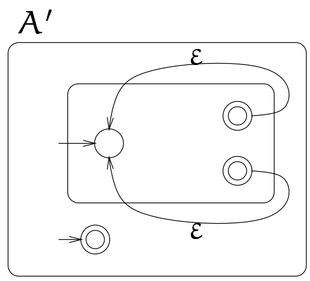


ements ZNKA A



ements





Věta. Jestliže $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak také $L_1 \cdot L_2$ je regulární jazyk.

Důkaz. Nechť $L_1 = L(A_1)$, $L_2 = L(A_2)$ pro KA

$A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$,

$A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$,

kde $Q_1 \cap Q_2 = \emptyset$.

Definujme nyní ZNKA

$A = (Q_1 \cup Q_2, \Sigma, \delta, \{q_{01}\}, F_2)$ tak, že

$\delta(q, a) = \{\delta_1(q, a)\}$ pro $q \in Q_1$ a

$\delta(q, a) = \{\delta_2(q, a)\}$ pro $q \in Q_2$;

navíc pro $q \in F_1$ je $\delta(q, \varepsilon) = \{q_{02}\}$

a pro $q \notin F_1$ je $\delta(q, \varepsilon) = \emptyset$.

Je snadné ověřit, že $L(A) = L_1 \cdot L_2$.

Věta. Jestliže L je regulární, pak také L^* je regulární.

Důkaz. Nechť $L = L(A)$ pro KA

$A = (Q, \Sigma, \delta, q_0, F)$.

Definujme ZNKA

$A' = (Q \cup \{p\}, \Sigma, \delta', \{q_0, p\}, F \cup \{p\})$,

kde $p \notin Q$ a

$\delta'(q, a) = \{\delta(q, a)\}$ (pro $q \in Q$, $a \in \Sigma$);

dále pro $q \in F$ je $\delta'(q, \varepsilon) = \{q_0\}$,

pro $q \notin F$ je $\delta'(q, \varepsilon) = \emptyset$

a navíc $\delta'(p, a) = \emptyset$ také pro vš. $a \in \Sigma$.

Je snadné ověřit, že $L(A') = L^*$.

Věta. Jestliže L je regulární, pak také $\overline{L}$ (doplněk) je regulární. Jestliže L_1, L_2 jsou regulární, pak $L_1 - L_2$ je regulární.

Důkaz. Nechť $L = L(A)$ pro KA

$A = (Q, \Sigma, \delta, q_0, F)$.

Pak $\overline{L} = L(A')$ pro

$A' = (Q, \Sigma, \delta, q_0, Q - F)$.

Dále si všimněme, že $L_1 - L_2 = L_1 \cap \overline{L_2}$.

Zrcadlový obraz slova $u = a_1 a_2 \ldots a_n$ je $u^R = a_n a_{n-1} \ldots a_1$,
zrcadlový obraz jazyka L je
$L^R = \{u \mid \exists v \in L : u = v^R\}$.

Věta. L je regulární právě tehdy, když L^R je regulární.

Důkaz. Nechť $L = L(A)$ pro NKA

$A = (Q, \Sigma, \delta, I, F)$.

NKA A' tž. $L(A') = L^R$ lze definovat takto:

$A' = (Q, \Sigma, \delta', F, I)$, kde

$q_2 \in \delta'(q_1, a) \Leftrightarrow q_1 \in \delta(q_2, a)$.

Dále si všimněme, že $(L^R)^R = L$.

Levý kvocient jazyka L_1 podle jazyka L_2 je jazyk $L_2 \backslash L_1 = \{ u \mid \exists v \in L_2 : vu \in L_1 \}$.

Pravý kvocient jazyka L_1 podle jazyka L_2 je jazyk $L_1 / L_2 = \{ u \mid \exists v \in L_2 : uv \in L_1 \}$.

Věta. Jestliže L_1, L_2 jsou regulární, pak také $L_2 \backslash L_1$ a L_1 / L_2 jsou regulární.

Důkaz. Nechť $L_1 = L(A_1)$, $L_2 = L(A_2)$, kde $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$.

Pro $q \in Q_1$ nechť $B_q = (Q_1, \Sigma, \delta_1, q, F_1)$,
$C_q = (Q_1, \Sigma, \delta_1, q_{01}, \{q\})$.

Dále nechť $U = \{ q \in Q_1 \mid \exists w \in \Sigma^* : w \in L(A_2) \wedge \delta_1(q_{01}, w) = q \}$.

(Tedy $q \in U \Longleftrightarrow L(A_2) \cap L(C_q) \neq \emptyset$.)

Pak $L_2 \backslash L_1 = \bigcup_{q \in U} L(B_q)$.

Dále si všimněme: $L_1 / L_2 = (L_2^R \backslash L_1^R)^R$.

$RV(\Sigma)$ ... množina *regulárních výrazů* nad Σ; induktivní definice:

$$\emptyset \in RV(\Sigma), \quad \varepsilon \in RV(\Sigma), \quad a \in RV(\Sigma)$$
(kde $a \in \Sigma$);

když $\alpha, \beta \in RV(\Sigma)$, pak také

$$(\alpha + \beta) \in RV(\Sigma),$$
$$(\alpha \cdot \beta) \in RV(\Sigma),$$
$$(\alpha^*) \in RV(\Sigma).$$

Reg. výraz α reprezentuje jazyk $[\alpha]$:

$$[\emptyset] = \emptyset, \quad [\varepsilon] = \{\varepsilon\}, \quad [a] = \{a\},$$
$$[(\alpha + \beta)] = [\alpha] \cup [\beta],$$
$$[(\alpha \cdot \beta)] = [\alpha] \cdot [\beta],$$
$$[(\alpha^*)] = [\alpha]^*.$$

Konvence při psaní regulárních výrazů:

$((((0 \cdot 1)^* \cdot 1) \cdot (1 \cdot 1)) + ((0 \cdot 0) + 1)^*)$

lze psát $\quad (01)^*111 + (00+1)^*$

Věta. Regulárními výrazy lze reprezentovat právě regulární jazyky.

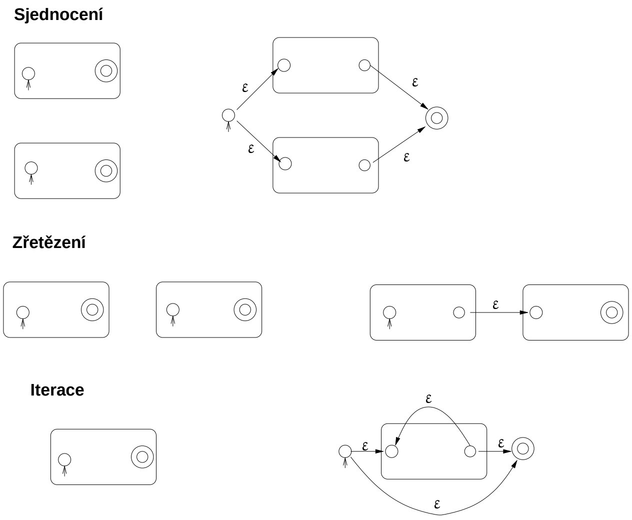
Důkaz.

Jednoduše KA pro $\emptyset$, $\{\varepsilon\}$, $\{a\}$.

Jelikož třída regulárních jazyků je (konstruktivně) uzavřena na regulární operace (sjednocení, zřetězení, iterace), je zřejmé:

> Existuje algoritmus, který k zadanému regul. výrazu α zkonstruuje KA A tž. $L(A) = [\alpha]$.

Konstrukce ZNKA k regulárnímu výrazu



Nechť $L = L(A)$ pro

$$\text{KA } A = (Q, \Sigma, \delta, q_1, F),$$

kde $\quad Q = \{q_1, q_2, \dots, q_n\}$.

Pro $i, j \in \{1, 2, \dots, n\}$ položme

$R_{ij} = \{ w \in \Sigma^* \mid \delta(q_i, w) = q_j \}$

Všimněme si $\quad L = \bigcup_{q_i \in F} R_{1i}$

Pro $k \in \{0, 1, 2, \dots, n\}$ definujme

$R^k_{ij} = \{ w \in R_{ij} \mid \forall u, v : (w = uv \ \wedge \ u \neq \varepsilon \ \wedge \ v \neq \varepsilon \wedge \delta(q_i, u) = q_m) \Longrightarrow m \leq k \}$

Tedy $R_{ij} = R^n_{ij}$

I. $\quad R^0_{ij} \subseteq \Sigma \cup \{\varepsilon\}$

II.

$R^{k+1}_{ij} = R^k_{ij} \ \cup \ (\ R^k_{i,k+1} \ (R^k_{k+1,k+1})^* \ R^k_{k+1,j} \ )$

Třída $RJ(\Sigma)$ *regulárních jazyků* nad abecedou Σ je nejmenší třída jazyků nad abecedou Σ, která obsahuje tzv. *elementární jazyky* a je uzavřena na *regulární operace*, tzn.:

- elementární jazyky, tj. $\emptyset$ a $\{a\}$ (pro každé $a \in \Sigma$), patří do $RJ(\Sigma)$,
- jestliže $L_1, L_2 \in RJ(\Sigma)$, pak také $L_1 \cup L_2 \in RJ(\Sigma)$,
- jestliže $L_1, L_2 \in RJ(\Sigma)$, pak také $L_1 \cdot L_2 \in RJ(\Sigma)$,
- jestliže $L \in RJ(\Sigma)$, pak také $L^* \in RJ(\Sigma)$.

Z uzávěrových vlastností třídy regulárních jazyků:

regulární výrazy by mohly obohatit např. symboly pro průnik a doplněk,

třeba &, ¬, kde

$$[(\alpha\&\beta)] = [\alpha] \cap [\beta],$$
$$[\neg(\alpha)] = \Sigma^* - [\alpha] \text{ (abeceda } \Sigma \text{ z kontextu)}$$

Zkrácení:
např. $[\neg(0^*)] = [(0+1)^*1(0+1)^*]$

Ale "lineární" převod RV $\rightarrow$ ZNKA pak už nelze !

Σ, Δ ... abecedy

nechť pro $a \in \Sigma$ ... $\sigma(a) \subseteq \Delta^*$

Rozšiřme *substituci* $\sigma : \Sigma \rightarrow \Delta^*$ na $\sigma : \Sigma^* \rightarrow \mathcal{P}(\Delta^*)$:

- $\sigma(\varepsilon) = \{\varepsilon\}$
- $\sigma(uv) = \sigma(u) \cdot \sigma(v) \quad (u, v \in \Sigma^*)$

Dále definujme pro $L \subseteq \Sigma^*$

$$\sigma(L) = \bigcup_{w \in L} \sigma(w)$$

(Říkáme: $\sigma(L)$ *vznikl* z L *substitucí* σ.)

Jestliže každé $\sigma(a)$ je regulární jazyk, jedná se o *regulární substituci*.

Jestliže každé $\sigma(a)$ obsahuje právě jedno slovo, σ se nazývá *homomorfismus* ($\sigma : \Sigma^* \rightarrow \Delta^*$).

Tvrzení. Pro regulární jazyk L a regulární substituci σ je $\sigma(L)$ rovněž regulárním jazykem.

Důkaz.
Nechť α a α_a (pro vš. $a \in abec(L)$) jsou regulární výrazy takové, že

- $[\alpha] = L$
- $[\alpha_a] = \sigma(a)$

Dosadíme-li do regulárního výrazu α za každý výskyt symbolu a regulární výraz α_a, dostaneme regulární výraz reprezentující $\sigma(L)$.

KA A_1, A_2 jsou *ekvivalentní* $\Leftrightarrow_{df}$ $L(A_1) = L(A_2)$.

KA A je *minimální* $\Leftrightarrow_{df}$ neexistuje automat ekvivalentní s A, který by měl menší počet stavů než A.

Věta. Ke každému KA A je minimální automat ekvivalentní s A určen jednoznačně (až na pojmenování stavů).

Věta. Existuje algoritmus, který k zadanému konečnému automatu A sestrojí minimální automat ekvivalentní s A.

Pro stav q KA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ definujme:
$L(q) = L(\mathcal{A}_q)$, kde $\mathcal{A}_q = (Q, \Sigma, \delta, q, F)$.

KA $\mathcal{A}$ je *redukovaný* $\Leftrightarrow_{df}$

- $\mathcal{A}$ nemá nedosažitelné stavy a
- v $\mathcal{A}$ pro každé dva různé stavy q_1, q_2 platí $L(q_1) \neq L(q_2)$.

Lemma. Jestliže pro stav q automatu $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ a stav q' automatu $\mathcal{A}' = (Q', \Sigma, \delta', q_0', F')$ platí $L(q) = L(q')$, pak pro každé $a \in \Sigma$ je $L(\,\delta(q, a)\,) = L(\,\delta'(q', a)\,)$.

Tvrzení. Dva *redukované* automaty, které jsou ekvivalentní, jsou izomorfní (stejné, až na pojmenování stavů).

Idea redukce automatu:

Mějme dva různé stavy q_1, q_2 automatu $\mathcal{A}$, kde $L(q_1) = L(q_2)$. Upravme $\mathcal{A}$ takto:

- každou šipku $q \xrightarrow{a} q_2$ nahraďme šipkou $q \xrightarrow{a} q_1$,
- stav q_2 (spolu s vycházejícími šipkami) vypusťme,
- pokud q_2 byl počátečním stavem (automatu $\mathcal{A}$), bude nyní q_1 počátečním.

Vzniklý $\mathcal{A}'$ je ekvivalentní s $\mathcal{A}$, přičemž má méně stavů; navíc nemá nedosažitelné stavy, pokud $\mathcal{A}$ je neměl.

Z toho (a z dříve uvedeného) plyne

minimální KA = redukovaný KA

Relace ρ na množině M je *ekvivalence* $\Leftrightarrow_{df}$ ρ je:

- reflexivní ($\forall x \in M : x\rho x$),
- symetrická ($\forall x, y \in M : x\rho y \Rightarrow y\rho x$),
- tranzitivní ($\forall x, y, z \in M : (x\rho y \wedge y\rho z) \Longrightarrow x\rho z$).

Ekvivalence ρ definuje na M *rozklad*

$$\{\, [x]_\rho \mid x \in M \,\}$$

tj. systém vzájemně disjunktních množin, zvaných třídy ekvivalence, jejichž sjednocením je M, přičemž

$$[x]_\rho = \{\, y \mid x\rho y \,\}.$$

Lemma. K libovolnému KA $\mathcal{A}$ existuje redukovaný automat ekvivalentní s $\mathcal{A}$.

Důkaz. Rovnou předpokládejme, že $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ nemá nedosažitelné stavy. Na Q definujme ekvivalenci $\sim$:

$$q \sim q' \quad \Leftrightarrow_{df} \; L(q) = L(q').$$

K $\mathcal{A}$ definujme tzv. podílový automat podle ekvivalence $\sim$, označený $\mathcal{A}_\sim$, takto (píšeme stručněji $[q]$ místo $[q]_\sim$):

$\mathcal{A}_\sim = (Q_\sim, \Sigma, \delta_\sim, [q_0], F_\sim)$, kde $Q_\sim = \{\, [q] \mid q \in Q \,\}$, $F_\sim = \{\, [q] \mid q \in F \,\}$ a $\delta_\sim([q], a) = [\delta(q, a)]$.

Korektnost plyne z $(p \sim q \Rightarrow (p \in F \Leftrightarrow q \in F))$ a z $(p \sim q \Rightarrow \delta(p, a) \sim \delta(q, a))$.

Platí: $\delta(q, w) = q' \Leftrightarrow \delta_\sim([q], w) = [q']$

Rozhodování otázky $L(q_1) = L(q_2)$:

$L^{\leq i}(q) =_{df} \{w \mid w \in L(q) \wedge |w| \leq i\}$

Na Q zaveďme ekvivalence $\overset{0}{\sim}, \overset{1}{\sim}, \overset{2}{\sim}, \ldots$:

$$q_1 \overset{i}{\sim} q_2 \Leftrightarrow_{df} L^{\leq i}(q_1) = L^{\leq i}(q_2)$$

Všimněme si:

$\overset{0}{\sim} \supseteq \overset{1}{\sim} \supseteq \overset{2}{\sim} \supseteq \ \ldots \ \supseteq \sim$

$q_1 \overset{0}{\sim} q_2 \iff q_1 \in F, q_2 \in F$ nebo

$q_1 \notin F, q_2 \notin F$

$q_1 \overset{i+1}{\sim} q_2 \iff q_1 \overset{i}{\sim} q_2 \wedge$

$(\forall a \in \Sigma : \delta(q_1, a) \overset{i}{\sim} \delta(q_2, a))$

Jestliže $\overset{i}{\sim} = \overset{i+1}{\sim}$, pak $\overset{i}{\sim} = \overset{i+1}{\sim} = \ \ldots \ = \sim$.

Když $|Q| = n$, pak nutně $\overset{n-1}{\sim} = \overset{n}{\sim}$
(pro $n \geq 2$ dokonce $\overset{n-2}{\sim} = \overset{n-1}{\sim}$).

Věta. Existuje algoritmus, který pro zadané konečné automaty A_1, A_2 rozhodne, zda $L(A_1) = L(A_2)$.

Důkaz. K A_1, A_2 zkonstruujeme ekvivalentní redukované automaty *v normovaném tvaru* a ty porovnáme.

Jiný důkaz (využívající již známých faktů):

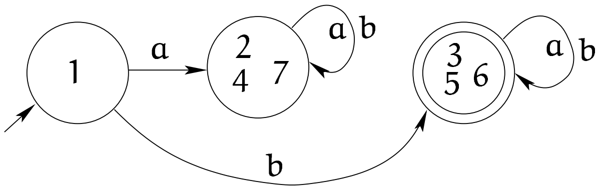
$L(A_1) = L(A_2) \iff$
$(L(A_1) - L(A_2)) \cup (L(A_2) - L(A_1)) = \emptyset$

Poznámka.
Všimněme si: když $|Q| = n$, pak
$L(q) = L(q') \Leftrightarrow L^{\leq n-2}(q) = L^{\leq n-2}(q')$.

	a	b
→ 1	2	3
2	2	4
3	3	5
4	2	7
5	6	3
6	6	6
7	7	4

$I = \{1\}$
$II = \{2, 4, 7\}$
$III = \{3, 5, 6\}$

	a	b
→ I	II	III
II	II	II
III	III	III



Věta. (Pumping lemma.) Nechť L je regulární jazyk. Pak nutně existuje $n \in \mathcal{N}$ tž. každé slovo $z \in L$, $|z| \geq n$, lze psát $z = uvw$, kde $|uv| \leq n$, $|v| \geq 1$ a pro vš. $i \geq 0$ je $uv^iw \in L$.

(($\forall$L tž. L je regulární))

$(\exists n \in \mathcal{N})$

$(\forall z$ tž. $z \in L, |z| \geq n)$

$(\exists\, u, v, w$ tž. $z = uvw, |uv| \leq n, |v| \geq 1)$

$(\forall i \geq 0) \qquad uv^iw \in L$

'Zkratky': $(\forall x$ tž. $A)\, B$ ..... $(\forall x : A \Rightarrow B)$

$(\exists x$ tž $A)\, B$ ..... $(\exists x : A \wedge B)$

Hra dvou hráčů A a B pro zadaný L:

1. A zvolí $n \in \mathcal{N}$
2. B zvolí slovo z tž. $z \in L$ a $|z| \geq n$ (nelze-li, A vyhrál)
3. A zvolí u, v, w tž. $z = uvw$, $|uv| \leq n$ a $|v| \geq 1$
4. B zvolí $i \geq 0$
5. *Výsledek:* je-li $uv^iw \in L$, vyhrál A, v případě $uv^iw \notin L$ vyhrál B.

Tvrzení. Má-li B vítěznou strategii, pak L není regulární.

Pro $L_1 = \{a^jb^j \mid j \geq 0\}$ strategie B např.:

1. A zvolí (libovolné) $n \in \mathcal{N}$
2. B zvolí $z = a^nb^n$
3. A zvolí libovolné u, v, w tž. $z = uvw$, $|uv| \leq n$ a $|v| \geq 1$ (tedy $u = a^j$, $v = a^k$ pro nějaké j, k tž. $j + k \leq n$, $k \geq 1$)
4. B: zvolí $i = 0$ (lze kterékoli $i \neq 1$)
5. Jelikož $a^ja^{n-(j+k)}b^n \notin L$, B vyhrává.

Poznámka. Uvedená strategie funguje i pro jazyk

$$L_2 = \{\, w \in \{a, b\}^* \mid |w|_a = |w|_b \,\}.$$

(Také fakt $L_2 \cap [a^*b^*] = L_1$ spolu s tím, že L_1 je neregulární, implikuje, že L_2 je neregulární.)

Poznámky. Např. jazyk

$$L = \{a, b\}^* \cup (\,\{c\}^+ \cdot \{a^jb^j \mid j \geq 0\}\,)$$

splňuje podmínku v pumping lemmatu (A má vítěznou strategii) a přitom není regulární. (Připomínka: $L^+ = L \cdot L^*$.) (Jinak by totiž i

$$(\{c\}^+ \backslash L\,) \cap \{a, b\}^* = \{\, a^jb^j \mid j \geq 0\,\}$$

byl regulární.)

————————-

Dvoucestné konečné automaty (zde $\delta : Q \times \Sigma \to Q \times \{-1, 0, 1\}$) rozpoznávají také (jen) regulární jazyky (i v nedeterministické verzi).

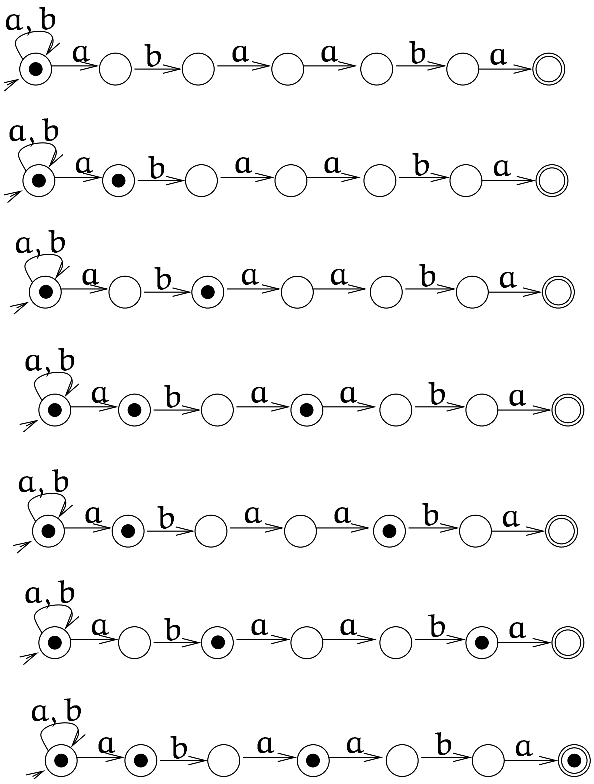
Hledání vzorku p v textu t

Přístup

“posouvající se okénko délky $|p|$”

vede k době běhu úměrné $|p| \cdot |t|$.
Doba běhu “Knuth-Morris-Pratt algoritmu” je úměrná $|p| + |t|$. Jedna z klíčových ideí algoritmu:

K ‘přímočarému’ *nedeterministickému* KA pro jazyk $\Sigma^*\,p$ (který má $|p|+1$ stavů) má standardně zkonstruovaný ekvivalentní DKA také jen $|p| + 1$ stavů.



Zobecněný sekvenční stroj
(GSM, generalized sequential machine)

$M = (Q, \Sigma, \Delta, \delta, \rho, q_0)$:

Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Δ ... konečná *výstupní abeceda*,
$\delta : Q \times \Sigma \to Q$ ... *přechodová funkce*,
$\rho : Q \times \Sigma \to \Delta^*$ ... *výstupní* funkce,
$q_0 \in Q$ ... *počáteční stav*.

Stroj M definuje zobrazení ("překlad") $f_M : \Sigma^* \to \Delta^*$:

1/ $f_M(\varepsilon) = \varepsilon$

2/ $f_M(wa) = f_M(w) \cdot \rho(\,\delta(q_0, w), a\,)$

(Pozn.: nedeterministická verze GSM je 'silnější'.)

Konečně stavový překladač (převaděč)
(finite [state] transducer)

$T = (Q, \Sigma, \Delta, \sigma, q_0, F)$:

Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Δ ... konečná *výstupní abeceda*,
σ ... konečná podmnožina množiny
$Q \times \Sigma^* \times Q \times \Delta^*$
("*přechodová a výstupní*" *relace*),
$q_0 \in Q$ ... *počáteční stav*,
$F \subseteq Q$ ... množina *přijímajících stavů*.

Stroj T definuje relaci $R_T \subseteq \Sigma^* \times \Delta^*$:

$(u, v) \in R_T \quad \Leftrightarrow_{df}$ lze psát $u = u_1 u_2 \ldots u_n$, $v = v_1 v_2 \ldots v_n$ tak, že pro jisté stavy $q_1, q_2, \ldots, q_n$ máme: $q_n \in F$ a $(q_{i-1}, u_i, q_i, v_i) \in \sigma$ pro $i = 1, 2, \ldots, n$.

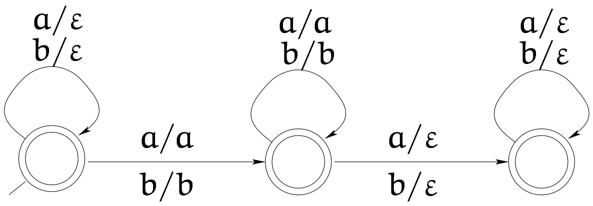
FT-relace .... *racionální relace*

Tvrzení. Je-li R racionální relace, pak také R^{-1} je racionální.

Tvrzení. Je-li R racionální relace a L regulární jazyk, pak $R(L)$ je regulární. (Podobně $R^{-1}(L)$ je regulární.)

($R(L) = \{v \mid \exists u \in L : (u, v) \in R\}$,
$R^{-1}(L) = \{u \mid \exists v \in L : (u, v) \in R\}$)

Příklad: Je-li L regulární, pak
$\mathit{infix}(L) = \{v \mid \exists u, w : uvw \in L\}$
je regulární:



Aritmetické výrazy v abecedě $\{a, +, \times, (,)\}$

Například: $a + a \times a$
$(a + a) \times a$

$$\langle EXPR \rangle \longrightarrow \langle EXPR \rangle + \langle EXPR \rangle$$
$$\langle EXPR \rangle \longrightarrow \langle EXPR \rangle \times \langle EXPR \rangle$$
$$\langle EXPR \rangle \longrightarrow (\langle EXPR \rangle)$$
$$\langle EXPR \rangle \longrightarrow a$$

$$E \longrightarrow E + E \mid E \times E \mid (E) \mid a$$

Odvození (*derivace*), slova $a + a \times a$:

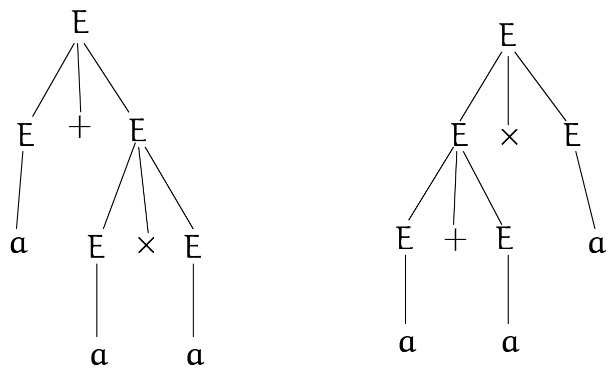
$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

levá derivace, *pravá* derivace ...

$E \Rightarrow E + E \Rightarrow E + E \times E \Rightarrow E + E \times a \Rightarrow E + a \times a \Rightarrow a + a \times a$

$E \Rightarrow E + E \Rightarrow E + E \times E \Rightarrow E + a \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

strom odvození (*derivační strom*)



Jiná *levá* derivace:

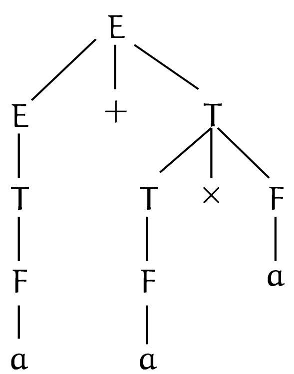
$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

$$E \longrightarrow E + T \mid T$$
$$T \longrightarrow T \times F \mid F$$
$$F \longrightarrow (E) \mid a$$

Jediná levá derivace pro $a + a \times a$:

$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + T \times F \Rightarrow a + F \times F \Rightarrow a + a \times F \Rightarrow a + a \times a$

Jediný derivační strom pro $a + a \times a$:



Bezkontextová gramatika
$G = (\Pi, \Sigma, S, P)$:

Π ... konečná množina *neterminálů*,
Σ ... konečná množina *terminálů* $(\Pi \cap \Sigma = \emptyset)$,
$S \in \Pi$ ... *počáteční neterminál*
P ... konečná množina pravidel typu

$A \to \beta$, kde $A \in \Pi$, $\beta \in (\Pi \cup \Sigma)^*$.

Relace $\Rightarrow$ (resp. $\Rightarrow_G$) na $(\Pi \cup \Sigma)^*$:

jestliže

$\gamma = \mu_1 A \mu_2$, $\delta = \mu_1 \beta \mu_2$, $(A \to \beta) \in P$

pak $\gamma \Rightarrow \delta$

$\Rightarrow^*$ ... refl. a tranz. uzávěr $\Rightarrow$

Tedy $\gamma \Rightarrow^* \delta \quad \Leftrightarrow_{df}$ ex. posloupnost $\mu_0, \mu_1, \ldots, \mu_n$ tak, že

$\gamma = \mu_0 \Rightarrow \mu_1 \Rightarrow \ldots \Rightarrow \mu_n = \delta$.

(*Derivace* (délky n) slova δ ze slova γ.)

Jazyk generovaný gramatikou G:

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

Jazyk L je *bezkontextový* $\Leftrightarrow_{df}$
ex. BG G tak, že $L(G) = L$.

Značení:
$a, b, c, \ldots$ ... terminály
$u, v, w, \ldots$ ... řetězce terminálů
$A, B, C, \ldots, X, Y, Z$ ... neterminály
$\alpha, \beta, \gamma, \ldots$ ... řetezce neterminálů a terminálů

$\alpha \Rightarrow^L \beta$ (*levé přepsání*) $\Leftrightarrow_{df}$
$\alpha = uX\delta$, $\beta = u\gamma\delta$ a $(X \to \gamma) \in P$
($u \in \Sigma^*$, $\delta \in (\Pi \cup \Sigma)^*$).

$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$ je *levou derivací*
$\Leftrightarrow_{df}$ $\alpha_i \Rightarrow^L \alpha_{i+1}$ $(0 \le i \le n-1)$.

Derivační strom (vztahující se ke $G = (\Pi, \Sigma, S, P)$), je uspořádaný kořenový strom, v němž

- vrcholy ohodnoceny prvky $\Pi \cup \Sigma$,

- kořen ohodnocen S,

- vrchol ohodnocený X $(\in \Pi)$ má následníky ohodnocené $Y_1, Y_2, \ldots, Y_n$ ($Y_i \in \Pi \cup \Sigma$), kde $(X \to Y_1 Y_2 \ldots Y_n) \in P$,

- vrchol ohodnocený a $(\in \Sigma)$ je listem.

Derivační strom pro $w = a_1 a_2 \ldots a_n$ ... listy zleva doprava ohodnoceny $a_1, a_2, \ldots, a_n$.

BG G je *jednoznačná* $\Leftrightarrow_{df}$ každé slovo z $L(G)$ má právě jednu levou derivaci (tj. právě jeden derivační strom).
V opačném případě je G *nejednoznačná* (či *víceznačná*).

Bezkontextový jazyk L je *jednoznačný* $\Leftrightarrow_{df}$ ex. jednoznačná G tž. $L(G) = L$; jinak se L nazývá (*vnitřně*) *nejednoznačný* (*víceznačný*).

Např.:
$L_1 = \{ a^n b^n \mid n \geq 0 \}$: $\quad S \longrightarrow aSb \mid \varepsilon$

$L_2 = \{ a^i b^j c^k \mid (i = j) \vee (j = k) \}$:

$S \longrightarrow S_1 C \mid AS_2$
$S_1 \longrightarrow aS_1 b \mid \varepsilon \qquad S_2 \longrightarrow bS_2 c \mid \varepsilon$
$C \longrightarrow c\,C \mid \varepsilon \qquad A \longrightarrow aA \mid \varepsilon$

Neex. jednoznačná BG G tž. $L(G) = L_2$.

Překlad přiřazovacích příkazů typu
$V := E$

$$S \longrightarrow \langle id \rangle := E$$
$$E \longrightarrow E + E \mid E * E \mid (E) \mid \langle id \rangle$$

do asembleru

Instrukce	Efekt
LOAD m	$c(m) \to ACC$
STORE m	$c(ACC) \to m$
ADD m	$c(ACC) + c(m) \to ACC$
MPY m	$c(ACC) * c(m) \to ACC$
LOAD $= m$	$m \to ACC$
ADD $= m$	$c(ACC) + m \to ACC$
MPY $= m$	$c(ACC) * m \to ACC$

Např. $\quad Zisk := (Cena + Dan) * 0.12$

LOAD $= 0.12$;
STORE \$2 ;
LOAD Dan ;
STORE \$1 ;
LOAD $Cena$;
ADD \$1 ;
MPY \$2 ;
STORE $Zisk$

Práce překladače:

- lexikální analýza
- syntaktická analýza
- generování kódu

Pro

$$Zisk := (Cena + Dan) * 0.12$$

je výsledkem lexikální analýzy

$$\langle id \rangle_1 := (\langle id \rangle_2 + \langle id \rangle_3) * \langle id \rangle_4$$

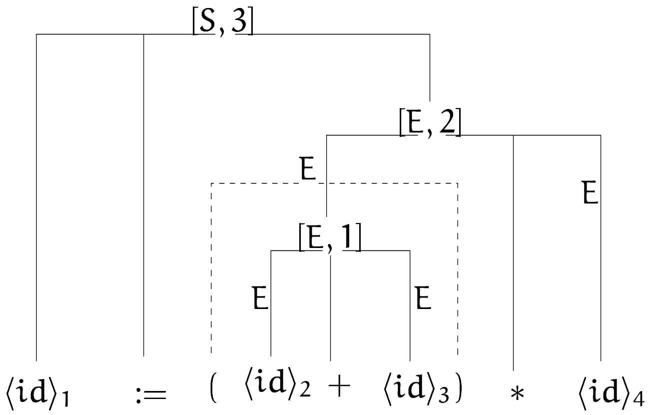
zároveň s tabulkou TAB:

Poř. číslo	Identifikátor	Informace
1	Zisk	prom. real
2	Cena	prom. real
3	Dan	prom. real
4	0.12	konst. real

Pro

$\langle id \rangle_1 := (\langle id \rangle_2 + \langle id \rangle_3) * \langle id \rangle_4$

je výsledkem syntaktické analýzy



- u ohodnocen $\langle id \rangle_i$: $\mathrm{Cod}(u)$ z TAB (např. pro $\langle id_1 \rangle$ ‘Zisk’, pro $\langle id \rangle_4$ ‘$= 0.12$’)
- u ohodnocen $:=, +, *$: $\mathrm{Cod}(u)$ prázdný
- u s následníky u_1, u_2, u_3 ohodnocen $[S, n]$: $\mathrm{Cod}(u)$ je

‘LOAD’ $\mathrm{Cod}(u_3)$ ‘; STORE’ $\mathrm{Cod}(u_1)$

- u s následníky u_1, u_2, u_3 ohodnocen $[E, n]$, přičemž u_2 ohodnocen $+$: $\mathrm{Cod}(u)$ je

$\mathrm{Cod}(u_3)$ ‘; STORE \$’n ‘; LOAD’ $\mathrm{Cod}(u_1)$ ‘; ADD \$’n

- podobně pro $*$:

$\mathrm{Cod}(u_3)$ ‘; STORE \$’n ‘; LOAD’ $\mathrm{Cod}(u_1)$ ‘; MPY \$’n

Bezkontextová gramatika
$G = (\Pi, \Sigma, S, P)$ je *redukovaná* $\Leftrightarrow_{df}$ každý $X \in \Pi$ splňuje:

1. $\exists w \in \Sigma^* : X \Rightarrow^* w$,
2. $\exists \alpha, \beta \in (\Pi \cup \Sigma)^* : S \Rightarrow^* \alpha X \beta$.

Konstrukce $\mathcal{T} = \{ X \in \Pi \mid X \text{ splňuje } 1. \}$:

$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \ldots$, kde

$\mathcal{T}_1 = \{ X \mid \exists w \in \Sigma^* : (X \to w) \in P \}$

$\mathcal{T}_{i+1} = \mathcal{T}_i \cup \{ X \mid \exists \alpha \in \mathcal{T}_i^* : (X \to \alpha) \in P \}$

$\mathcal{T}_n = \mathcal{T}_{n+1} \Longrightarrow \mathcal{T}_n = \mathcal{T} \qquad (n \leq |\Pi|)$

Konstrukce $\mathcal{D} = \{ X \in \Pi \mid X \text{ splňuje } 2. \}$:

$\mathcal{D}_1 = \{ S \}$

$$\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{ X \mid \exists Y \in D_i, \alpha_1, \alpha_2 : (Y \to \alpha_1 X \alpha_2) \in P \}$$

Gramatiky G_1, G_2 jsou *ekvivalentní* $\Leftrightarrow_{df}$ $L(G_1) = L(G_2)$.

Věta. Ke každé BG G, pro niž platí $L(G) \neq \emptyset$, lze sestrojit ekvivalentní redukovanou gramatiku.

Důkaz. Odstraníme všechna pravidla obsahující neterminály nesplňující podmínku 1. ($\exists w \in \Sigma^* : X \Rightarrow^* w$).
Pak odstraníme všechna pravidla obsahující neterminály nesplňující podmínku 2. ($\exists \alpha, \beta \in (\Pi \cup \Sigma)^* : S \Rightarrow^* \alpha X \beta$).

Věta. Existuje algoritmus, který pro zadanou BG G rozhodne, zda $L(G) = \emptyset$.

Poznámka. Ekvivalenci dvou BG nelze algoritmicky rozhodovat.
(Není obdoba konstrukce minim. KA.)

Bezkontextová *gramatika* se nazývá *nevypouštějící* $\Leftrightarrow_{df}$ neobsahuje pravidlo typu $X \longrightarrow \varepsilon$.

Věta. Ke každé BG G lze sestrojit nevypouštějící gramatiku G' tž. $L(G') = L(G) - \{\varepsilon\}$.

Důkaz.
Konstrukce $\mathcal{E} = \{X \in \Pi \mid X \Rightarrow^* \varepsilon\}$:

$\mathcal{E}_1 = \{X \mid (X \to \varepsilon) \in P\}$

$\mathcal{E}_{i+1} = \mathcal{E}_i \cup \{X \mid \exists \alpha \in \mathcal{E}_i^* : (X \to \alpha) \in P\}$

Pro každé pravidlo $(X \to \alpha) \in P$ zařadíme do P' všechna pravidla $X \to \beta$, kde $\beta \neq \varepsilon$ a β vznikne z α vypuštěním některých (třeba žádných) výskytů symbolů z $\mathcal{E}$.

Důsledek. Ke každé BG $G = (\Pi, \Sigma, S, P)$ existuje ekvivalentní BG $G_1 = (\Pi_1, \Sigma, S_1, P_1)$, kde ε může být pravou stranou pouze u pravidla $S_1 \to \varepsilon$; v takovém případě se pak S_1 nevyskytuje na pravé straně žádného z pravidel z P_1.

BG je v *Chomského normální formě* (ChNF) $\Leftrightarrow_{df}$ každé pravidlo je tvaru $X \to YZ$ nebo $X \to a$.

Věta. Ke každé BG G lze sestrojit BG G' v CHNF tž. $L(G') = L(G) - \{\varepsilon\}$.

Důkaz.

- převod na nevypouštějící gramatiku

- odstranění pravidel typu $X \to Y$

- pro každé a přidáme A_a a pravidlo $A_a \to a$; na každé pravé straně delší než 1 nahradíme a neterminálem A_a

- každé pravidlo typu $X \to Y_1 Y_2 \ldots Y_n$, kde $n \geq 3$, nahradíme $X \to Y_1 Z_1$,
$Z_1 \to Y_2 Z_2$, ..., $Z_{n-3} \to Y_{n-2} Z_{n-2}$,
$Z_{n-2} \to Y_{n-1} Y_n$
($Z_1, Z_2, \ldots, Z_{n-2}$ nově přidané)

Odstranění pravidel typu $X \to Y$:

- Pro každý neterm. A zkonstruujeme

$$\mathcal{D}_A = \{B \mid A \Rightarrow^* B\}$$

- Pak pro každé $B \to \alpha$, kde $B \in \mathcal{D}_A$ a α není jeden neterminál, přidáme pravidlo $A \to \alpha$.

- Odstraníme všechna pravidla typu $X \to Y$.

BG je v *Greibachové normální formě* (GNF) $\Leftrightarrow_{df}$ každé pravidlo je tvaru $X \to aY_1Y_2 \dots Y_n$ ($n \geq 0$)

Věta. Ke každé BG G lze sestrojit BG G' v GNF tž. $L(G') = L(G) - \{\varepsilon\}$.

Lemma. Nechť v G jsou $B \to \beta_1$, $B \to \beta_2$, $\dots$, $B \to \beta_n$ všechna pravidla s B na levé straně.
Odstraněním pravidla $A \longrightarrow \alpha B \gamma$ a přidáním pravidel $A \longrightarrow \alpha\beta_1\gamma$, $A \longrightarrow \alpha\beta_2\gamma, \dots, A \longrightarrow \alpha\beta_n\gamma$, dostaneme gramatiku ekvivalentní s G.

Lemma. (Odstranění levé rekurze.) Mějme BG $G = (\Pi, \Sigma, S, P)$. Nechť

$A \to A\alpha_1$, $A \to A\alpha_2$, $\dots$, $A \to A\alpha_m$,

$A \to \beta_1$, $A \to \beta_2$, $\dots$, $A \to \beta_n$

jsou všechna pravidla s A na levé straně, přičemž β_i nezačínají A.

$G' = (\Pi \cup \{Z\}, \Sigma, S, P')$ vzniklá z G dodáním nového neterminálu Z a nahrazením uvedených pravidel soustavou

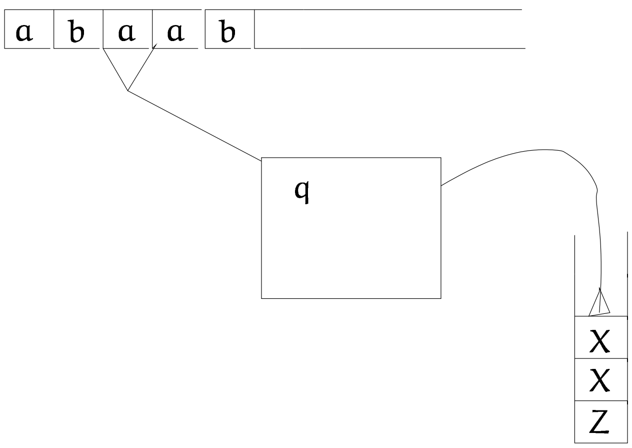
$A \to \beta_1$, $A \to \beta_2$, $\dots$, $A \to \beta_n$,

$A \to \beta_1 Z$, $A \to \beta_2 Z$, $\dots$, $A \to \beta_n Z$,

$Z \to \alpha_1 Z$, $Z \to \alpha_2 Z$, $\dots$, $A \to \alpha_m Z$,

$Z \to \alpha_1$, $Z \to \alpha_2$, $\dots$, $A \to \alpha_m$,

je ekvivalentní gramatice G.



Zásobníkový automat (ZA)
$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$:

Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Γ ... konečná *zásobníková abeceda*,
$q_0 \in Q$ ... *počáteční stav*,
$Z_0 \in \Gamma$ ... *poč. zásobníkový symbol*
$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_{fin}(Q \times \Gamma^*)$.

Konfigurace ZA M je trojice (q, w, α), kde $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.

Na množině konfigurací definujeme relaci $\vdash_M$:

$(q, aw, X\beta) \vdash_M (q', w, \alpha\beta) \Leftrightarrow_{df} \delta(q, a, X) \ni (q', \alpha)$

($a \in (\Sigma \cup \{\varepsilon\})$, $w \in \Sigma^*$, $\beta \in \Gamma^*$)

Relace $\vdash_M^*$ je reflexivním a tranzitivním uzávěrem relace $\vdash_M$.

Slovo $w \in \Sigma^*$ *je přijímáno* ZA M $\Leftrightarrow_{df}$ $(q_0, w, Z_0) \vdash_M^* (q, \varepsilon, \varepsilon)$ pro něj. $q \in Q$.

Jazykem rozpoznávaným ZA M rozumíme jazyk $L(M) = \{w \in \Sigma^* \mid w$ je přijímáno ZA $M\}$.

Poznámka. Jedná se o “přijímání prázdným zásobníkem”.
Přidáme-li k parametrům ZA M množinu přijímajících (neboli ‘koncových’) stavů $F \subseteq Q$, lze definovat jazyk “přijímaný koncovým stavem”

$L_{KS}(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_M^* (q, \varepsilon, \alpha)$ pro něj. $q \in F$ a $\alpha \in \Gamma^*\}$

Lemma. Ke každé BG G lze sestrojit ZA M (s 1 stavem) tž. $L(M) = L(G)$.

Důkaz. Pro BG $G = (\Pi, \Sigma, S, P)$ nechť

$M = (\{q_0\}, \Sigma, \Pi \cup \Sigma, \delta, q_0, S)$, kde

pro $X \in \Pi$:
$\delta(q_0, \varepsilon, X) = \{ (q_0, \alpha) \mid (X \to \alpha) \in P \}$,

pro $a \in \Sigma$: $\delta(q_0, a, a) = \{ (q_0, \varepsilon) \}$

(jinde δ přiřazuje $\emptyset$)

Indukcí se dá ukázat

$S \Rightarrow_G^* u\alpha \;\Leftrightarrow_{df}\; (q_0, u, S) \vdash_M^* (q_0, \varepsilon, \alpha)$

kde $u \in \Sigma^*$, $\alpha \in \{\varepsilon\} \cup \Pi(\Pi \cup \Sigma)^*$;
$\Rightarrow_G^*$ zde označuje *levé* odvození

Uvedený ZA provádí (nedeterministicky) tzv. analýzu “shora dolů”.

Např. ZA sestrojený ke gramatice

1/ $A \longrightarrow A + B$
2/ $A \longrightarrow B$
3/ $B \longrightarrow B * C$
4/ $B \longrightarrow C$
5/ $C \longrightarrow (A)$
6/ $C \longrightarrow a$

simuluje při úspěšném běhu na slově $a * (a + a)$ použití pravidel v pořadí 2,3,4,6,5,1,2,4,6 .

Analýza “zdola nahoru” ... pravá derivace pozpátku (6,4,6,4,2,6,4,1,5,3,2)

Poznámka. LL- či LR-analyzátory ... determ. verze v syntaktické analýze

Lemma. Ke každému ZA M s jedním stavem lze sestrojit bezkontextovou gramatiku G tž. $L(G) = L(M)$.

Důkaz. Mějme

$M = (\{q_0\}, \Sigma, \Gamma, \delta, q_0, Z_0)$, kde $\Sigma \cap \Gamma = \emptyset$.

Sestrojme

$G = (\Gamma, \Sigma, Z_0, P)$ tak, že

$(A \to a\alpha) \in P \Longleftrightarrow \delta(q_0, a, A) \ni (q_0, \alpha)$ ($a \in (\Sigma \cup \{\varepsilon\}$).

Dá se snadno ukázat

$Z_0 \Rightarrow_G^* u\alpha \;\Leftrightarrow_{df}\; (q_0, u, Z_0) \vdash_M^* (q_0, \varepsilon, \alpha)$

kde $u \in \Sigma^*$, $\alpha \in \Gamma^*$;
$\Rightarrow_G^*$ zde označuje *levé* odvození

Lemma. Ke každému ZA M lze sestrojit ZA M' s 1 stavem tž. $L(M) = L(M')$.

Důkaz.
Pro $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ konstruujeme $M' = (\{s\}, \Sigma, \Gamma', \delta', s, R)$,

kde $\Gamma' = (Q \times \Gamma \times Q) \cup \{R\}$,

$\delta'(s, \varepsilon, R) = \{\, (s, \langle q_0, Z_0, q\rangle) \mid q \in Q \,\}$

a δ' je dále dodefinována tak, že platí

$\forall w : (s, w, \langle p, X, q\rangle) \vdash^*_{M'} (s, \varepsilon, \varepsilon) \iff (p, w, X) \vdash^*_M (q, \varepsilon, \varepsilon)$

Je tedy $\forall w : (s, w, R) \vdash^*_{M'} (s, \varepsilon, \varepsilon) \iff (q_0, w, Z_0) \vdash^*_M (q, \varepsilon, \varepsilon)$ pro něj.$q \in Q$

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

δ' je dodefinována takto:

- pro $(q', \varepsilon) \in \delta(q, a, X)$,
kde $a \in (\Sigma \cup \{\varepsilon\})$, zařadíme do $\delta'(s, a, \langle q, X, q'\rangle)$ prvek (s, ε),

- pro $(q', A_1A_2 \ldots A_n) \in \delta(q, a, X)$ $(n \geq 1)$ zařadíme do $\delta'(s, a, \langle q, X, \overline{q}\rangle)$ prvek
$(s, \langle q', A_1, q_1\rangle\langle q_1, A_2, q_2\rangle \ldots \langle q_{n-1}, A_n, \overline{q}\rangle)$
pro každé $\overline{q}, q_1, q_2, \ldots, q_{n-1} \in Q$.

Příklad.
$M = (\{p, q, r\}, \{a, b\}, \{A, B\}, \delta, p, A)$, kde

$\delta(p, a, A) = \{(q, AA), (p, B)\}$,
$\delta(q, b, A) = \{(q, AA)\}$,
$\delta(p, \varepsilon, B) = \{(q, A)\}$, $\delta(q, \varepsilon, A) = \{(r, \varepsilon)\}$,
$\delta(r, a, A) = \{(r, A)\}$, $\delta(r, b, A) = \{(r, \varepsilon)\}$

(jinde je hodnota δ rovna $\emptyset$)

Věta. (Pumping lemma pro bezkontextové jazyky, neboli $uvwxy$-teorém.)
Nechť L je bezkontextový jazyk. Pak existují přirozená čísla p, q tž. každé slovo $z \in L$ delší než p lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq q$, a pro vš. $i \geq 0$ je $uv^iwx^iy \in L$.

Důkaz. Nechť $L = L(G)$ pro BG $G = (\Pi, \Sigma, S, P)$ v ChNF.

Všimněme si: když na větvi deriv. stromu pro $z \in L$ dva výskyty téhož neterminálu, řekněme A, pak

$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvwxy = z$

pro nějaké $u, v, w, x, y \in \Sigma^*$, $vx \neq \varepsilon$.

Nechť $|\Pi| = k$. Všimněme si dále:

- na větvi délky alespoň $k + 1$ jsou jistě aspoň dva výskyty téhož neterminálu;

- má-li derivační strom pro $z \in \Sigma^*$ všechny větve kratší než $k + 1$, pak nutně $|z| \leq 2^{k-1}$ (počet listů binárního stromu hloubky $k-1$);

- v deriv. stromě pro $z \in L$, $|z| > 2^{k-1}$, určitě existují dva různé vrcholy v_1, v_2 na stejné větvi (v_1 blíž ke kořeni) označené stejným neterminálem, přičemž podstrom s kořenem v_1 má hloubku nejvýš $k + 1$ – tedy má nejvýše 2^k listů.

Můžeme proto vzít: $p = 2^{k-1}$, $q = 2^k$.

Lze uvažovat $n = max(p, q) + 1$:

Věta. Nechť L je bezkontextový jazyk. Pak existuje přirozené číslo n tž. každé slovo $z \in L$, $|z| \geq n$, lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq n$ a pro vš. $i \geq 0$ je $uv^iwx^iy \in L$.

Když L bezkontextový, tak:

$(\exists n \in \mathcal{N})$ $(\forall z$ tž. $z \in L, |z| \geq n)$

$(\exists\, u, v, w, x, y$ tž. $z = uvwxy, |vwx| \leq n, vx \neq \varepsilon)$

$(\forall i \geq 0) : uv^iwx^iy \in L$

Má-li B vítěznou strategii v následující hře, pak L není bezkontextový.

1. A zvolí $n \in \mathcal{N}$
2. B zvolí slovo z tž. $z \in L$ a $|z| \geq n$
3. A zvolí u, v, w, x, y tž. $z = uvwxy$, $|vwx| \leq n$ a $vx \neq \varepsilon$
4. B zvolí $i \geq 0$
5. *Výsledek:* je-li $uv^iwx^iy \in L$, vyhrál A, je-li $uv^iwx^iy \notin L$, vyhrál B.

Strategie B pro $L = \{a^nb^nc^n \mid n \geq 0\}$:

1. A zvolí (libovolné) $n \in \mathcal{N}$
2. B zvolí $z = a^nb^nc^n$
3. A zvolí libovolné u, v, w, x, y tž. $z = uvwxy$, $|vwx| \leq n$ a $vx \neq \varepsilon$,
4. B zvolí $i = 0$ (lze kterékoli $i \neq 1$)
5. Jelikož $|vwx| \leq n$, slova v, x neobsahují aspoň jeden ze symbolů a, b, c. Proto uwy nepatří do L; vyhrává B.

Strategie B pro $L = \{ww \mid w \in \{0, 1\}^*\}$

1. A zvolí (libovolné) $n \in \mathcal{N}$
2. B zvolí $z = 0^n1^n0^n1^n$
3. A zvolí libovolné u, v, w, x, y tž. $z = uvwxy$, $|vwx| \leq n$ a $vx \neq \varepsilon$,
4. B zvolí $i = 0$ (lze kterékoli $i \neq 1$)
5. Jelikož $|vwx| \leq n$, slova v, x zasahují do nejvýš jednoho úseku nul a nejvýš jednoho úseku jedniček. Tedy $uwy = 0^{k_1}1^{k_2}0^{\ell_1}1^{\ell_2}$,
kde $k_1 \neq \ell_1$ nebo $k_2 \neq \ell_2$.
Tedy $uwy \notin L$; vyhrává B.

Věta. CFL je uzavřena vůči sjednocení, zřetězení, iteraci, zrcadlovému obrazu, substituci (tedy i homomorfismu).

Důkaz. K BG $G_1 = (\Pi_1, \Sigma, S_1, P_1)$, $G_2 = (\Pi_2, \Sigma, S_2, P_2)$ lze zkonstruovat

$G = (\Pi, \Sigma, S, P)$ tž.$L(G) = L(G_1) \cup L(G_2)$ takto (předp., že $\Pi_1 \cap \Pi_2 = \emptyset$):

$\Pi = \Pi_1 \cup \Pi_2 \cup \{S\}$, kde $S \notin \Pi_1 \cup \Pi_2$,

$P = P_1 \cup P_2 \cup \{ S \longrightarrow S_1, S \longrightarrow S_2 \}$.

Přímočará je i konstrukce BG pro $L(G_1) \cdot L(G_2)$, $L(G_1)^*$, $L(G_1)^R$.

Podobně k BG $G = (\Pi, \Sigma, S, P)$ a gramatikám G_a (pro vš. $a \in \Sigma$) lze snadno zkonstruovat gramatiku, která generuje jazyk vzniklý z $L(G)$ substitucí $L(G_a)$ za každé a.

Věta. CFL je uzavřena vůči průniku s regulárním jazykem, i vůči kvocientu podle regulárního jazyka. (Tj. pro každý bezkontextový L a regulární R, jsou $L \cap R$, $R \setminus L$, L/R bezkontextové.)

Věta. CFL není uzavřena vůči průniku a doplňku.

Důkaz. $L_1 = \{a^i b^j c^k \mid i = j\}$, $L_2 = \{a^i b^j c^k \mid j = k\}$ jsou bezkontextové.

Přitom $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ bezkontextový není.

Z de Morganových pravidel ($L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$) plyne, že kdyby byla CFL uzavřena vůči doplňku, tak by byla uzavřena i vůči průniku (což není).

Deterministický zásobníkový automat (DZA) je ZA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, pro nějž platí:

1. $\delta(q, a, X)$ je vždy nejvýše jednoprvková množina (pro $a \in \Sigma \cup \{\varepsilon\}$) a
2. je-li $\delta(q, \varepsilon, X) \neq \emptyset$, pak $\delta(q, a, X) = \emptyset$ pro vš. $a \in \Sigma$.

Jazyk L je *deterministický bezkontextový jazyk*, jestliže $L = L_{KS}(M)$ pro nějaký DZA M.

Poznámka. Využitím "bottom-symbolu" lze snadno ukázat, že k DZA M lze zkonstruovat DZA M' tž. $L_{PZ}(M) = L_{KS}(M')$. K DZA M lze také sestrojit DZA M' tž. $L_{KS}(M) \cdot \{\$\} = L_{PZ}(M')$, kde $\$$ je nově přidaný znak.

Věta. Třída DCFL je uzavřena vůči doplňku. Na druhé straně není uzavřena vůči průniku ani vůči sjednocení.

Poznámky.

Uzavřenost vůči doplňku ... nestačí prohození přijímajících a nepřijímajících stavů (ale dá se "dotáhnout")

Neuzavřenost vůči průniku:

$L_1 = \{a^i b^j c^k \mid i = j\}$,
$L_2 = \{a^i b^j c^k \mid j = k\}$

jsou *deterministické* bezkontextové.

Přitom $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ není (ani) bezkontextový.

Neuzavřenost vůči sjednocení ... de Morganova pravidla.

Příklad. Bezkontextový jazyk

$L = \{a^i b^j c^k \mid (i \neq j) \vee (j \neq k)\}$

není v DCFL: jinak by i $\overline{L}$ byl v DCFL; pak by ovšem

$\overline{L} \cap [a^* b^* c^*] = \{a^n b^n c^n \mid n \geq 0\}$

byl bezkontextový (což není).

Příklady dalších nedeterministických bezkontextových jazyků:

$\{ww^R \mid w \in \{a, b\}^*\}$,

$\{a^i b^j c^k \mid (i = j) \vee (j = k)\}$

Poznámka. Existuje algoritmus, který rozhoduje ekvivalenci dvou zadaných deterministických zásobníkových automatů (otevř. problém cca 1965 - 1997).

(Generativní) gramatika
$G = (\Pi, \Sigma, S, P)$:

Π ... konečná množina *neterminálů*,
Σ ... konečná množina *terminálů*
($\Pi \cap \Sigma = \emptyset$),
$S \in \Pi$ ... *počáteční neterminál*
P ... konečná množina pravidel typu

$\alpha \rightarrow \beta$

kde

$\alpha \in (\Pi \cup \Sigma)^* \Pi (\Pi \cup \Sigma)^*$

$\beta \in (\Pi \cup \Sigma)^*$

Relace $\Rightarrow$ (resp. $\Rightarrow_G$) na $(\Pi \cup \Sigma)^*$:

jestliže

$\gamma = \mu_1 \alpha \mu_2$, $\delta = \mu_1 \beta \mu_2$, $(\alpha \rightarrow \beta) \in P$

pak $\gamma \Rightarrow \delta$

$\Rightarrow^*$ ... refl. a tranz. uzávěr $\Rightarrow$

Tedy $\gamma \Rightarrow^* \delta \quad \Leftrightarrow_{df}$ ex. posloupnost $\mu_0, \mu_1, \ldots, \mu_n$ tak, že

$\gamma = \mu_0 \Rightarrow \mu_1 \Rightarrow \ldots \Rightarrow \mu_n = \delta$.

Jazyk generovaný gramatikou G:

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

Chomského hierarchie

Gramatika typu 0, neboli *obecná generativní gramatika*: $\alpha \rightarrow \beta$ bez omezení

Gramatika typu 1, neboli *kontextová gramatika*:
$\alpha X \beta \rightarrow \alpha \gamma \beta$, kde $|\gamma| \geq 1$ (jedině lze $S \rightarrow \varepsilon$; pak S není nikde na pravé straně)

Gramatika typu 2, neboli *bezkontextová gramatika*: $X \rightarrow \alpha$

Gramatika typu 3, neboli *regulární gramatika*: $X \rightarrow wY$, $X \rightarrow w$

Jazyk typu i ... generován gram. typu i

$\mathcal{L}_i$ ... třída jazyků typu i

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

Lemma. Ke každé regulární gramatice lze zkonstruovat ekvivalentní gramatiku s pravidly $X \to aY$, $X \to Y$, $X \to \varepsilon$.

Důkaz.

$X \to a_1 a_2 \dots a_n Y$ ($n \geq 2$) nahradíme

$X \to a_1 Z_1$, $Z_1 \to a_2 Z_2$, …, $Z_{n-1} \to a_n Y$

Věta. Jazyk je generován RG právě když je rozpoznáván KA.

Důkaz. Ke KA $A = (Q, \Sigma, \delta, q_0, F)$ sestrojme $G = (Q, \Sigma, q_0, P)$, kde

v P je $q \to aq'$ pro $\delta(q, a) = q'$
a $q \to \varepsilon$ pro $q \in F$.

Indukcí: $\delta(q, w) = q' \iff q \Rightarrow_G^* wq'$.

Odtud snadno plyne $L(G) = L(A)$.

Naopak pro $G = (\Pi, \Sigma, S, P)$ s pravidly typu $X \to aY$, $X \to Y$, $X \to \varepsilon$

sestrojme ZNKA $A = (\Pi, \Sigma, \delta, S, F)$, kde

$Y \in \delta(X, a) \iff (X \to aY) \in P$
$Y \in \delta(X, \varepsilon) \iff (X \to Y) \in P$
$F = \{X \mid (X \to \varepsilon) \in P\}$

Je snadné ověřit $L(A) = L(G)$.

Turingův stroj, zkráceně TS,
$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$:

Q ... konečná množina *stavů*,
Σ ... konečná *vstupní abeceda*,
Γ ... konečná *pracovní abeceda*, $\Gamma \supseteq \Sigma$,
$q_0 \in Q$ ... *počáteční stav*,
$F \subseteq Q$ ... množina *koncových stavů*,
δ ... přechodová funkce:

$\delta : (Q - F) \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\}$

V $\Gamma - \Sigma$ je vždy obsažen
speciální prvek $\square$ ... *prázdný znak*.

Konfigurace TS M ... uqv, kde
$u, v \in \Gamma^*$ a $q \in Q$ (uqv a $\square^i uqv \square^j$ totéž)

Počáteční konfigurace ... $q_0 v$, $v \in \Sigma^*$

Koncová konfigurace ... uqv, $q \in F$

Na množině konfigurací definujeme relaci $\vdash_M$: $K \vdash_M K'$ pro $K = uaqbv$ právě když jedna z možností:

$\delta(q, b) = (q', b', 0)$ a $K' = uaq'b'v$,
$\delta(q, b) = (q', b', +1)$ a $K' = uab'q'v$,
$\delta(q, b) = (q', b', -1)$ a $K' = uq'ab'v$.

Relace $\vdash_M^*$ je reflexivním a tranzitivním uzávěrem relace $\vdash_M$.

Slovo $u \in \Sigma^*$ *je přijímáno* TS M $\Leftrightarrow_{df}$ $q_0 u \vdash^* K$ pro nějakou koncovou K.

Jazykem přijímaným TS M rozumíme jazyk
$L(M) = \{w \in \Sigma^* \mid w \text{ je přijímáno } M\}$.

Nedeterministické Turingovy stroje

$\delta : (Q - F) \times \Gamma \to \mathcal{P}_{fin}(Q \times \Gamma \times \{-1, 0, +1\})$

Věta. Třída jazyků přijímaných (determimistickými) TS se rovná třídě jazyků přijímaných nedetermimistickými TS.

Věta. Jazyky přijímané TS jsou právě jazyky typu 0.

Lineárně omezené automaty, LBA:

TS, kde nelze jít mimo vstupní slovo (levá a pravá zarážka). Záladní verze zde *nedeterministická* (podobně jako u ZA); zde se ale neví, zda DLBA slabší (dlouhodobě otevřený problém).

Věta. L je kontextový $\iff$ L je přijímán nějakým (nedeterministickým) LBA.

$\mathcal{L}_3$ .... konečné automaty (regulární jazyky)

$\mathcal{L}_2$ .... zásobníkové automaty (bezkontextové jazyky)

$\mathcal{L}_1$ .... lineárně omezené automaty (kontextové jazyky)

$\mathcal{L}_0$ .... Turingovy stroje (rekurzivně spočetné jazyky)

Poznámky:

TS ... univerzální výpočetní model.

Tzv. *rekurzivní jazyky* (též rozhodnutelné) ... přijímající TS se vždy zastaví. Mezi $\mathcal{L}_1$ a $\mathcal{L}_0$.

Stroje s 2 zásobníky ... ekvivalentní TS.