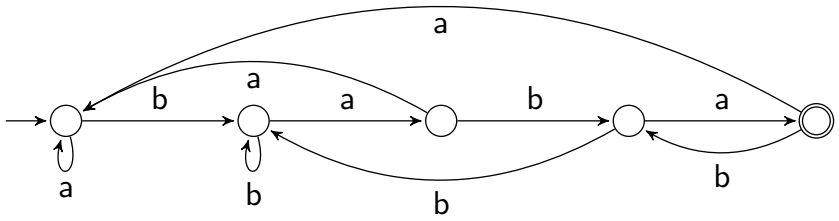
Příklady regulárních jazyků

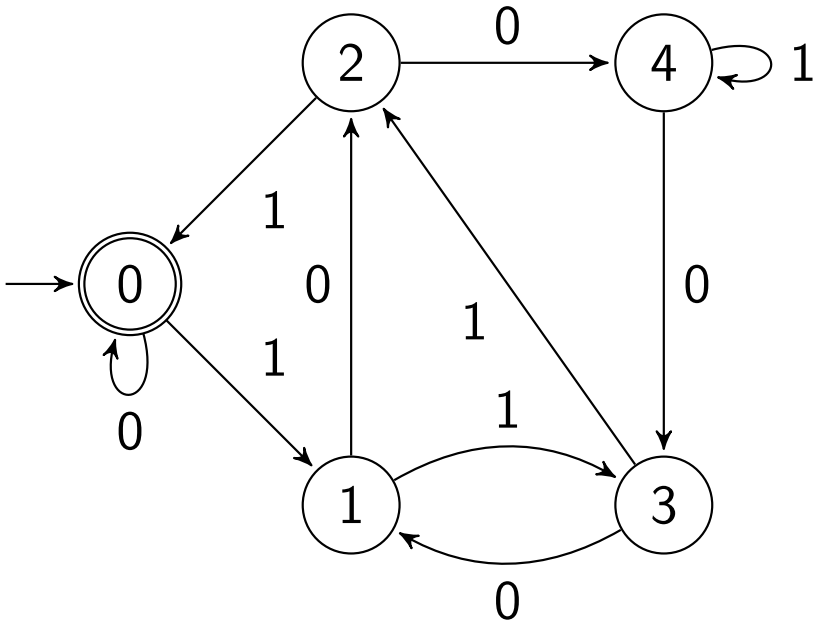
Example 1.3 (regulární jazyk)

- $L = \{w \mid w = ubaba, w \in \{a,b\}^*, u \in \{a,b\}^*\}$.



Example 1.4 (regulární jazyk)

- $L = \{w | w \in \{0,1\}^* \& w$ je binární zápis čísla dělitelného $5\}$.



Example 1.5 (!NEregulární jazyk)

- $L = \{0^n 1^n | w \in \{0,1\}^*, n \in \mathbb{N}\}$ NENÍ regulání jazyk.

Iterační (pumping) lemma pro regulární jazyky

Theorem 1.1 (!Iterační (pumping) lemma pro regulární jazyky)

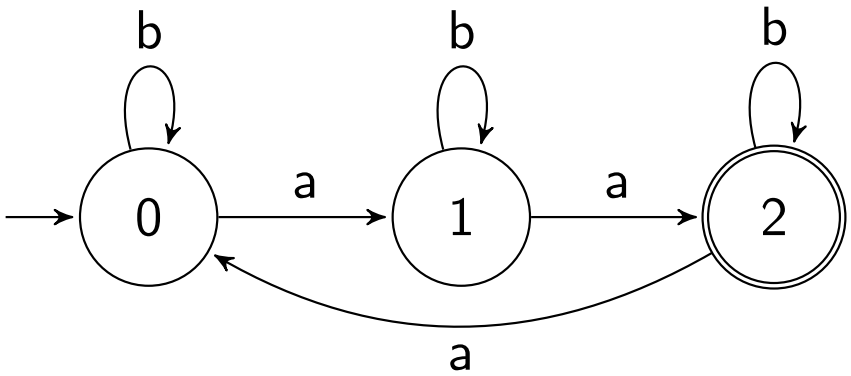
Mějme regulární jazyk L. Pak existuje konstanta $n \in \mathbb{N}$ (závislá na L) tak že každé $w \in L; |w| \geq n$ můžeme rozdělit na tři části, $w = xyz$, že:

- $y \neq \lambda$
- $|xy| \leq n$
- $\forall k \in \mathbb{N}_0$, *slovo* $xy^k z$ *je také v* L.

Example 1.6

- Lemma řeklo: $n = 3$.
- $abbbba = a(b)bbba$;
  $\forall i \geq 0; a(b)^i bbba \in L(A)$.
- $aaaaba = (aaa)aba$;
  $\forall i \geq 0; (aaa)^i aba \in L(A)$.
- aa nelze pumpovat, ale $|aa| < n$.

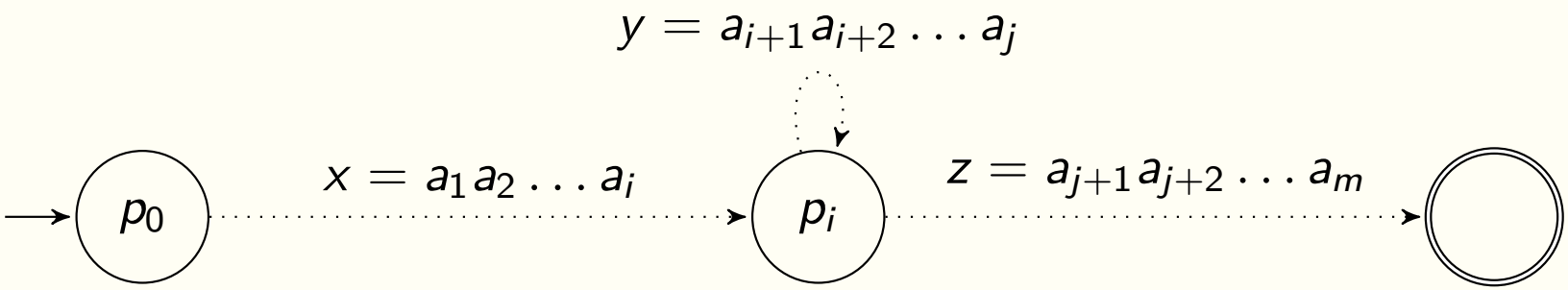
Automat A



Důkaz iteračního lematu pro regulární jazyky

Proof: iteračního lematu pro regulární jazyky

- Mějme regulární jazyk L, pak existuje DFA A s n stavy, že $L = L(A)$.
- Vezměme libovolné slovo $a_1 a_2 \dots a_m = w \in L$ délky $m \geq n$, $a_i \in \Sigma$.
- Definujme: $\forall i \; p_i = \delta^*(q_0, a_1 a_2 \dots a_i)$. Platí $p_0 = q_0$.
- Máme $n+1$ p_i a n stavů, některý se opakuje, vezměme první takový, tj. $(\exists i, j)(0 \leq i < j \leq n \& p_i = p_j)$.
- Definujme: $x = a_1 a_2 \dots a_i$, $y = a_{i+1} a_{i+2} \dots a_j$, $z = a_{j+1} a_{j+2} \dots a_m$, tj. $w = xyz$, $y \neq \lambda$, $|xy| \leq n$.



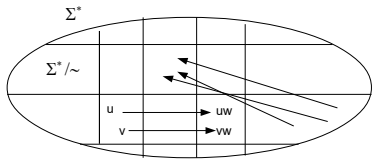
- Smyčka nad p_i se může opakovat libovolně krát a vstup je také akceptovaný.

Kongruence, Myhill–Nerodova věta

Definition 2.1 (kongruence)

Mějme konečnou abecedu Σ a relaci ekvivalence $\sim$ na Σ^* (reflexivní, symetrická, tranzitivní). Potom:

- $\sim$ je **pravá kongruence**, jestliže $(\forall u, v, w \in \Sigma^*) u \sim v \Rightarrow uw \sim vw$.
- je **konečného indexu**, jestliže rozklad $\Sigma^*/\sim$ má konečný počet tříd.
- Třídu kongruence $\sim$ obsahující slovo u značíme $[u]_\sim$, resp. $[u]$.

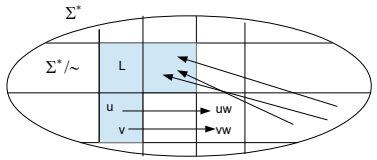


Theorem 2.1 (!Myhill–Nerodova věta)

Nechť L je jazyk nad konečnou abecedou Σ. Potom následující tvrzení jsou ekvivalentní:

a) *L je rozpoznatelný konečným automatem,*

b) *existuje pravá kongruence $\sim$ konečného indexu nad Σ^* tak, že L je sjednocením jistých tříd rozkladu $\Sigma^*/\sim$.*



Proof: Důkaz Myhill–Nerodovy věty

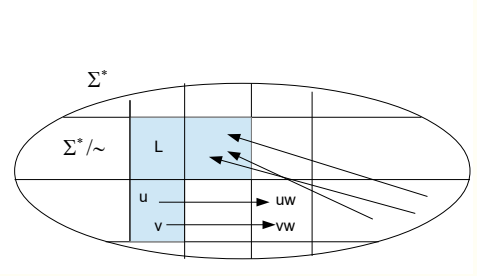
a)⇒b); tj. automat ⇒ pravá kongruence konečného indexu

- definujeme $u \sim v \equiv \delta^*(q_0, u) = \delta^*(q_0, v)$.
- je to ekvivalence (reflexivní, symetrická, transitivní)
- je to pravá kongruence (z definice δ^*)
- má konečný index (konečně mnoho stavů)
- $L = \{w | \delta^*(q_0, w) \in F\} = \bigcup_{q \in F} \{w | \delta^*(q_0, w) = q\} = \bigcup_{q \in F} [w | \delta^*(q_0, w) = q]_\sim$

.

b)⇒a); tj. pravá kongruence konečného indexu ⇒ automat

- abeceda automatu vezmeme Σ
- za stavy Q volíme třídy rozkladu $\Sigma^*/\sim$
- počáteční stav $q_0 \equiv [\lambda]$
- koncové stavy $F = \{c_1, \ldots, c_n\}$, kde $L = \bigcup_{i=1,\ldots,n} c_i$
- přechodová funkce $\delta([u], x) = [ux]$ (je korektní z def. pravé kongruence).
- $L(A) = L$



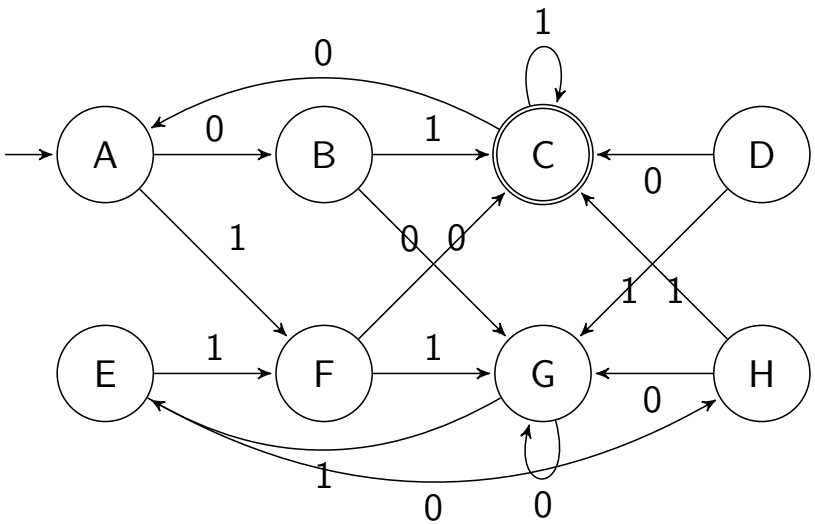
$w \in L \Leftrightarrow w \in \bigcup_{i=1,\ldots,n} c_i \Leftrightarrow w \in c_1 \vee \ldots w \in c_n \Leftrightarrow [w] = c_1 \vee \ldots [w] = c_n \Leftrightarrow [w] \in F \Leftrightarrow w \in L(A)$

$\delta^*([\lambda], w) = [w]$

Algorthm: !Algoritmus hledání rozlišitelných stavů v DFA

Následující algoritmus nalezne rozlišitelné stavy:

- Základ: Pokud $p \in F$ (přijímající) a $q \notin F$, pak je dvojice $\{p, q\}$ rozlišitelná.
- Indukce: Nechť $p, q \in Q$, $a \in \Sigma$ a o dvojici r, s; $r = \delta(p, a)$ a $s = \delta(q, a)$ víme, že jsou rozlišitelné. Pak i $\{p, q\}$ jsou rozlišitelné.
  - opakuj dokud existuje nová trojice $p, q \in Q$, $a \in \Sigma$.



	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x

Křížek značí rozlišitelné dvojice. C je rozlišitelné hned, ostatní kromě $\{A, G\}, \{E, G\}$ také. Vidíme tři ekvivalentní dvojice stavů.

Algoritmus hledání rozlišitelných stavů

Přijímající vs. nepřijímající stavy

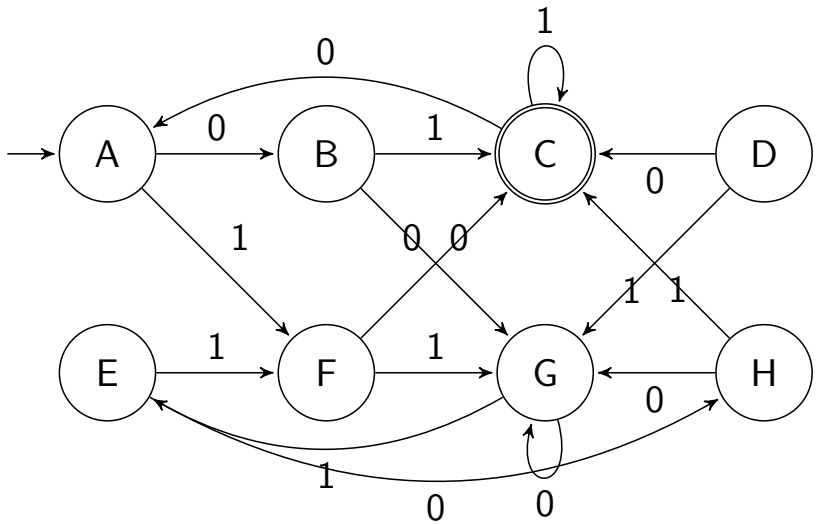
	A	B	C	D	E	F	G
B							
C	x	x					
D			x				
E			x				
F			x				
G			x				
H			x				

1.krok1: $\delta(q, 1) \in F$ pro $q \in \{B, C, H\}$

	A	B	C	D	E	F	G
B	x						
C	x	x					
D		x	x				
E		x	x				
F		x	x				
G		x	x				
H	x		x	x	x	x	x

1.krok0: $\delta(q, 0) \in F$ pro $q \in \{D, F\}$

	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G		x	x	x		x	
H	x		x	x	x	x	x



B a G jsou rozlišitelné, $\delta(A, 0) = B$, $\delta(G, 0) = H$, tj. A,G jsou rozlišitelné.
Obdobně pro E,G vedoucí $\delta(*, 0)$ do rozlišitelných stavů H,G.

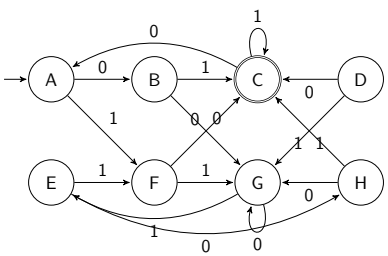
	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x

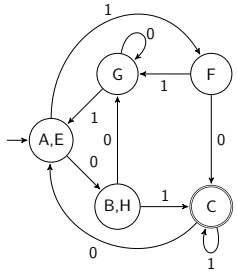
Zůstávají tři ekvivalentní dvojice stavů.

Algoritmus nalezení reduktu DFA A

Algorthm: !Algoritmus nalezení reduktu DFA A

- Ze vstupního DFA A eliminujeme stavy nedosažitelné z počátečního stavu.
- Najdeme rozklad zbylých stavů na třídy ekvivalence.
- Konstruujeme DFA B na třídách ekvivalence jakožto stavech. Přechodovou funkci B označíme γ, mějme $S \in Q_B$. Pro libovolné $q \in S$, označíme T třídu ekvivalence $\delta(q, a)$ a definujeme $\gamma(S, a) = T$. Tato třída musí být stejná pro všechna $a \in S$.
- Počáteční stav B je třída obsahující počáteční stav A.
- Množina přijímajících stavů B jsou bloky odpovídající přijímajícím stavům A.





Ekvivalence nedeterministických a deterministických konečných automatů

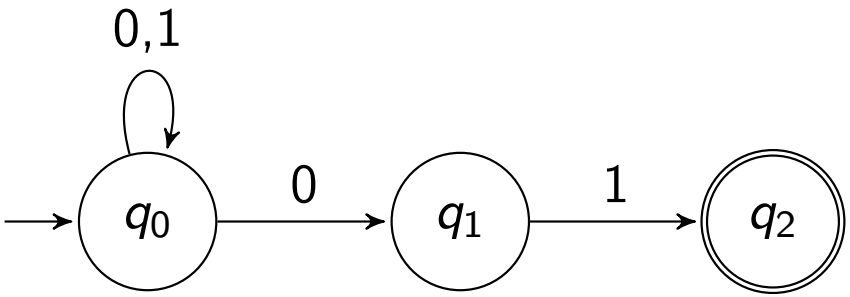
Algorthm: !Podmnožinová konstrukce

Podmnožinová konstrukce začíná s NFA $N = (Q_N, \Sigma, \delta_N, S_0, F_N)$. Cílem je popis deterministického DFA $D = (Q_D, \Sigma, \delta_D, S_0, F_D)$, pro který $L(N) = L(D)$.

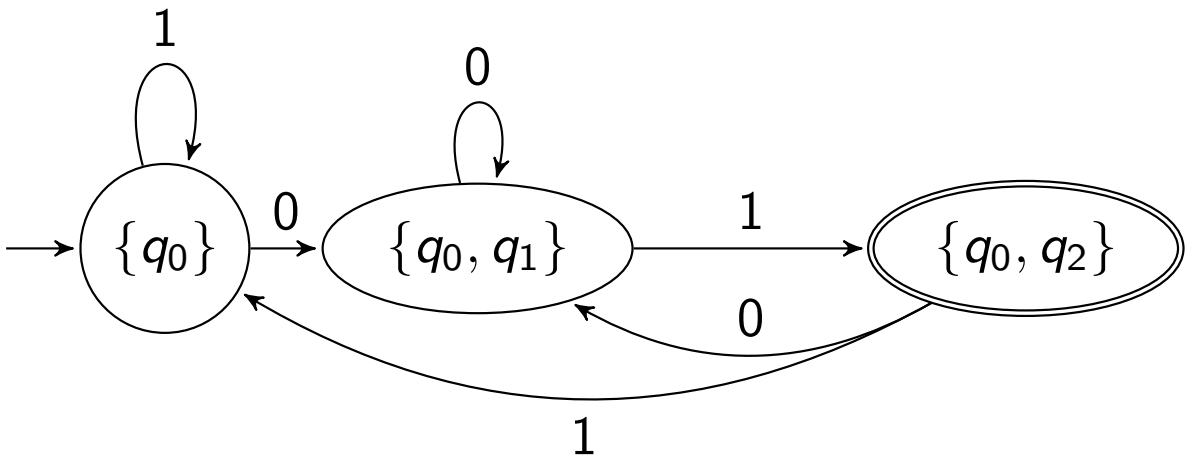
- Q_D je množina podmnožin Q_N, $Q_D = \mathcal{P}(Q_N)$ (potenční množina). Nedosažitelné stavy můžeme vynechat.
- Počáteční stav DFA je stav označený S_0, tj. prvek Q_D.
- $F_D = \{S : S \in \mathcal{P}(Q_N) \ \& \ S \cap F_N \neq \emptyset\}$, tedy S obsahuje alespoň jeden přijímající stav N.
- Pro každé $S \subseteq Q_N$ a každý vstupní symbol $a \in \Sigma$,

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a).$$

Příklad podmnožinové konstrukce pro $\{w.01 | w \in \{0,1\}\}$



	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$



Příklady reguláních výrazů, priorita

Example 4.2 (Regulární výrazy)

Jazyk střídajících se nul a jedniček lze zapsat:

- $(\mathbf{01})^* + (\mathbf{10})^* + \mathbf{1}(\mathbf{01})^* + \mathbf{0}(\mathbf{10})^*$
- $(\lambda + \mathbf{1})(\mathbf{01})^*(\lambda + \mathbf{0})$.

Jazyk $L((\mathbf{0}^*\mathbf{10}^*\mathbf{10}^*\mathbf{1})^*\mathbf{0}^*) = \{w | w \in \{0,1\}^*, |w|_1 = 3k, k \geq 0\}$.

Definition 4.3 (priorita)

Nejvyšší prioritu má iterace $*$, nižší konkatenace (zřetězení), nejnižší sjednocení $+$.

Theorem (4.1a!varianta Kleeneho věty)

Každý jazyk reprezentovaný konečným automatem lze zapsat jako regulární výraz.
Každý jazyk popsaný regulárním výrazem můžeme zapsat jako λ-NFA (a tedy i DFA).

Od DFA k regulárním výrazům

Regulární výraz z DFA

Mějme DFA A, $Q_A = \{1, \ldots, n\}$ o n stavech.

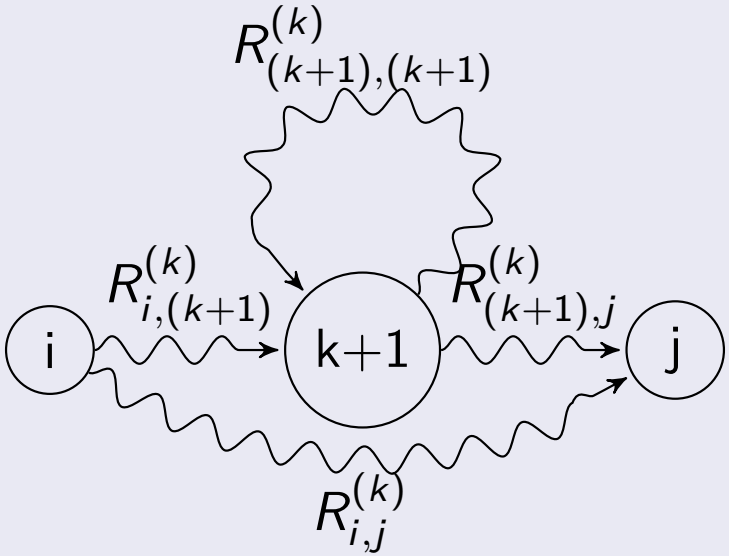
Nechť $R_{ij}^{(k)}$ je regulární výraz, $L(R_{ij}^{(k)}) = \{w | \delta^*_{\leq k}(i, w) = j\}$ množina slov převádějících stav i do stavu j v A cestou, která neobsahuje stav s vyšším indexem než k.

Budeme rekruzivně konstruovat $R_{ij}^{(k)}$ pro $k = 0, \ldots, n$.

$k = 0$, $i \neq j$: $R_{ij}^{(0)} = \mathbf{a_1} + \mathbf{a_2} + \ldots + \mathbf{a_m}$ kde $a_1, a_2, \ldots, a_m$ jsou symboly označující hrany i do j (nebo $R_{ij}^{(0)} = \emptyset$ nebo $R_{ij}^{(0)} = \mathbf{a}$ pro $m = 0, 1$).

$k = 0$, $i = j$: smyčky, $R_{ii}^{(0)} = \lambda + \mathbf{a_1} + \mathbf{a_2} + \ldots + \mathbf{a_m}$ kde $a_1, a_2, \ldots, a_m$ jsou symboly na smyčkách v i.

INDUKCE. Mějme $\forall i, j \in Q\ R_{ij}^{(k)}$. Konstruujeme $R_{ij}^{(k+1)}$.



$$R_{ij}^{(k+1)} = R_{ij}^{(k)} + R_{i(k+1)}^{(k)}(R_{(k+1)(k+1)}^{(k)})^* R_{(k+1)j}^{(k)}$$

- Cesty z i do j neprocházející uzlem $(k+1)$ jsou již v $R_{ij}^{(k)}$.
- Cesty z i do j přes $(k+1)$ s případnými smyčkami můžeme zapsat $R_{i(k+1)}^{(k)}(R_{(k+1)(k+1)}^{(k)})^* R_{(k+1)j}^{(k)}$.
- regulární výrazy jsou uzavřené na sčítání (sjednocení), zřetězení i iteraci, tj. $R_{ij}^{k+1} \in RegE(\Sigma)$

Nakonec, $RegE = \oplus_{j \in F_A} R_{1j}^{(n)}$ sjedncení přes přijímající stavy j.

Chomského hierarchie

Definition 6.2 (Klasifikace gramatik podle tvaru přepisovacích pravidel)

- **gramatiky typu 0** (**rekurzivně spočetné jazyky** $\mathcal{L}_0$)

  pravidla v obecné formě $\alpha \to \omega$, $\alpha, \omega \in (V \cup T)^*, \alpha$ obsahuje neterminál
- **gramatiky typu 1** (**kontextové gramatiky**, jazyky $\mathcal{L}_1$)
  - pouze pravidla ve tvaru $\gamma A \beta \to \gamma \omega \beta$

    $A \in V, \gamma, \beta \in (V \cup T)^*, \omega \in (V \cup T)^+$**!**
  - jedinou výjimkou je pravidlo $S \to \lambda$, potom se ale S nevyskytuje na pravé straně žádného pravidla
- **gramatiky typu 2** (**bezkontextové gramatiky**, jazyky $\mathcal{L}_2$)

  pouze pravidla ve tvaru $A \to \omega, A \in V, \omega \in (V \cup T)^*$
- **gramatiky typu 3** (regulární/**pravé lineární gramatiky**, regulární jazyky $\mathcal{L}_3$)

  pouze pravidla ve tvaru $A \to \omega B, A \to \omega, A, B \in V, \omega \in T^*$

Uspořádanost Chomského hierarchie

- Chomského hierarchie definuje uspořádání tříd jazyků

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

- dokonce vlastní podmnožiny (později)

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

$\mathcal{L}_0 \supseteq \mathcal{L}_1$ rekurzivně spočetné jazyky zahrnují kontextové jazyky
pravidla $\gamma A \beta \to \gamma \omega \beta$ obsahují vlevo neterminál A

$\mathcal{L}_2 \supseteq \mathcal{L}_3$ bezkontextové jazyky zahrnují regulární jazyky
pravidla $A \to \omega B, A \to \omega$ obsahují vpravo řetězec $(V \cup T)^*$

$\mathcal{L}_1 \supseteq \mathcal{L}_2$ kontextové jazyky zahrnují bezkontextové jazyky
problém je s pravidly typu $A \to \lambda$, ale ta umíme eliminovat.

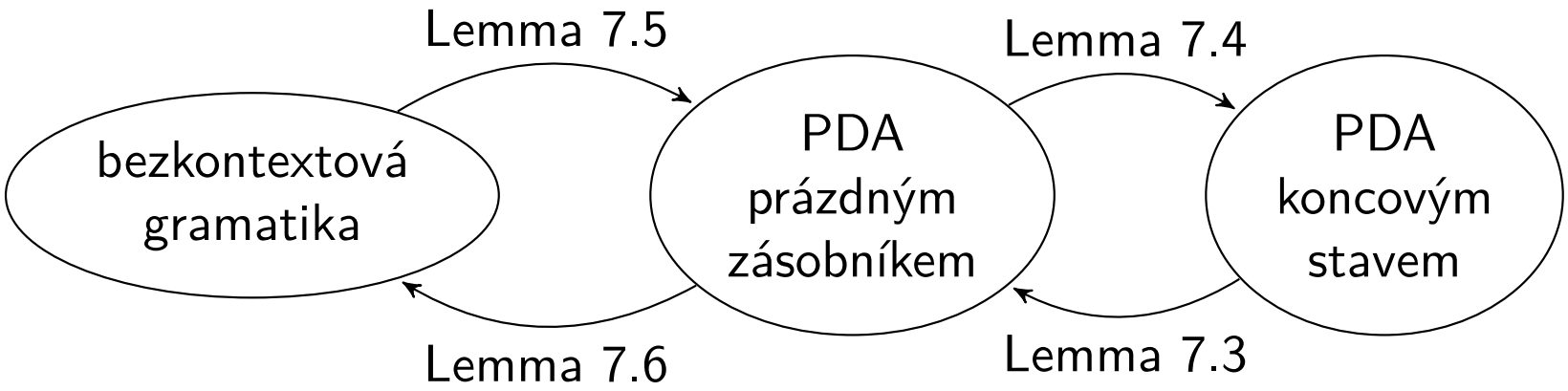
Ekvivalence jazyků rozpoznávaných zásobníkovými automaty a bezkontextových jazyků

Theorem 7.1 (!L(CFG), L(PDA), N(PDA))

Následující tvrzení jsou ekvivalentní

- *Jazyk L je bezkontextový, tj. generovaný CFG*
- *Jazyk L je přijímaný nějakým zásobníkovým automatem koncovým stavem.*
- *Jazyk L je přijímaný nějakým zásobníkovým automatem prázdným zásobníkem.*

Důkaz bude veden směry dle následujícího obrázku.



Theorem 8.1 (!ChNF)

Mějme bezkontextovou gramatiku G, $L(G) - \{\lambda\} \neq \emptyset$. Pak existuje CFG G_1 v Chomského normálním tvaru taková, že $L(G_1) = L(G) - \{\lambda\}$.

Example 8.6

$I \to a|b|IA|IB|IZ|IU$
$F \to LER|a|b|IA|IB|IZ|IU$
$T \to TMF|LER|a|b|IA|IB|IZ|IU$
$E \to EPT|TMF|LER|a|b|IA|IB|IZ|IU$
$A \to a$
$B \to b$
$Z \to 0$
$U \to 1$
$P \to +$
$M \to *$
$L \to ($
$R \to)$

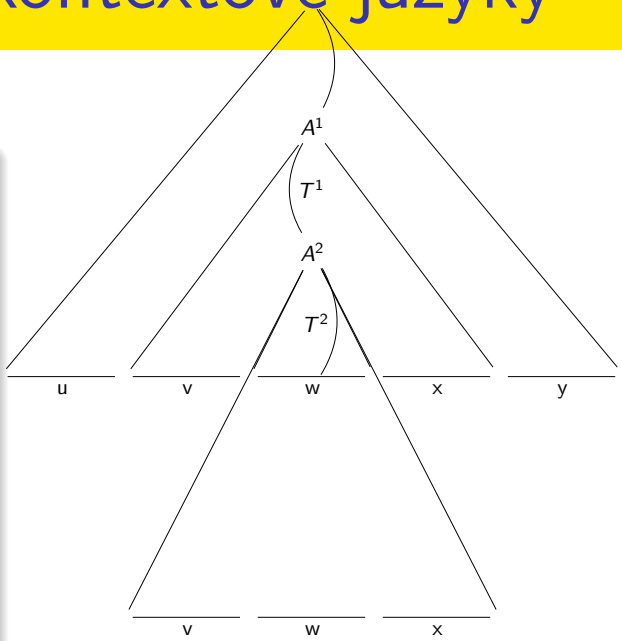
$F \to LC_3|a|b|Ia|IB|IZ|IU$
$T \to TC_2|LC_3|a|b|IA|IB|IZ|IU$
$E \to EC_1|TC_2|LC_3|a|b|IA|IB|IZ|IU$
$C_1 \to PT$
$C_2 \to MF$
$C_3 \to ER$
$I, A, B, Z, U, P, M, L, R$ jako vlevo

Lemma o vkládání (pumping) pro bezkontextové jazyky

Theorem 8.2 (!!Lemma o vkládání (pumping) pro bezkontextové jazyky)

Mějme bezkontextový jazyk L. Pak existují dvě přirozená čísla p, q taková že každé $z \in L, |z| > p$ lze rozložit na $z = uvwxy$ kde:

- $|vwx| \leq q$
- $vx \neq \lambda$
- $\forall i \geq 0$, $uv^i wx^i y \in L$.



Idea důkazu:

- vezmeme derivační strom pro z
- najdeme nejdelší cestu
- na ní dva stejné neterminály
- tyto neterminály určí dva podstromy
- podstromy definují rozklad slova
- nyní můžeme větší podstrom posunout ($i > 1$)
- nebo nahradit menším podstromem ($i = 0$)

Proof: $|z| > p : z = uvwxy, |vwx| \leq q, vx \neq \lambda, \forall i \geq 0 uv^i wx^i y \in L$

- vezmeme gramatiku v Chomského NF (pro $L = \{\lambda\}$ a $\emptyset$ dk jinak).
- Nechť $|V| = n$. Položíme $p = 2^{n-1}, q = 2^n$.
- Pro $z \in L, |z| > p$, má v derivačním stromu z cestu délky $> n$
- vezmeme nejdelší cestu; terminál kam vede označíme t
- Aspoň dva z posledních $(n+1)$ neterminálů na cestě do t jsou stejné
- vezmeme dvojici A^1, A^2 nejblíže k t (určuje podstromy T^1, T^2)
- cesta z A^1 do t je nejdelší v podstromu T^1 a má délku maximálně $(n+1)$

  tedy slovo dané stromem T^1 není delší než 2^n (tedy $|vwx| \leq q$)
- z A^1 vedou dvě cesty (ChNF), jedna do T^2 druhá do zbytku vx

  ChNF je nevypouštějící, tedy $vx \neq \lambda$
- derivace slova $(A^1 \Rightarrow^* vA^2x, A^2 \Rightarrow^* w)$

  $S \Rightarrow^* uA^1y \Rightarrow^* uvA^2xy \Rightarrow^* uvwxy$
- posuneme–li A^2 do A^1 $(i = 0)$

  $S \Rightarrow^* uA^2y \Rightarrow^* uwy$
- posuneme–li A^1 do A^2 $(i = 2, 3, \ldots)$

  $S \Rightarrow^* uA^1y \Rightarrow^* uvA^1xy \Rightarrow^* uvvA^2xxy \Rightarrow^* uvvwxxy.$

□

Cocke-Younger-Kasami algorithm náležení slova do CFL

Exponenciálně k $|w|$: vyzkoušet všechny derivační stromy dostatečné délky pro L.

Algorthm: !CYK algoritmus, v čase $O(n^3)$

- Mějme gramatiku v ChNF $G = (V, T, P, S)$ pro jazyk L a slovo $w = a_1 a_2 \dots a_n \in T^*$.
- Vytvořme trohúhelníkovou tabulku (vpravo),
  - horizontální osa je w
  - X_{ij} jsou množiny neterminálů A takových, že $A \Rightarrow^* a_i a_{i+1} \dots a_j$.

Základ: $X_{ii} = \{A; A \to a_i \in P\}$

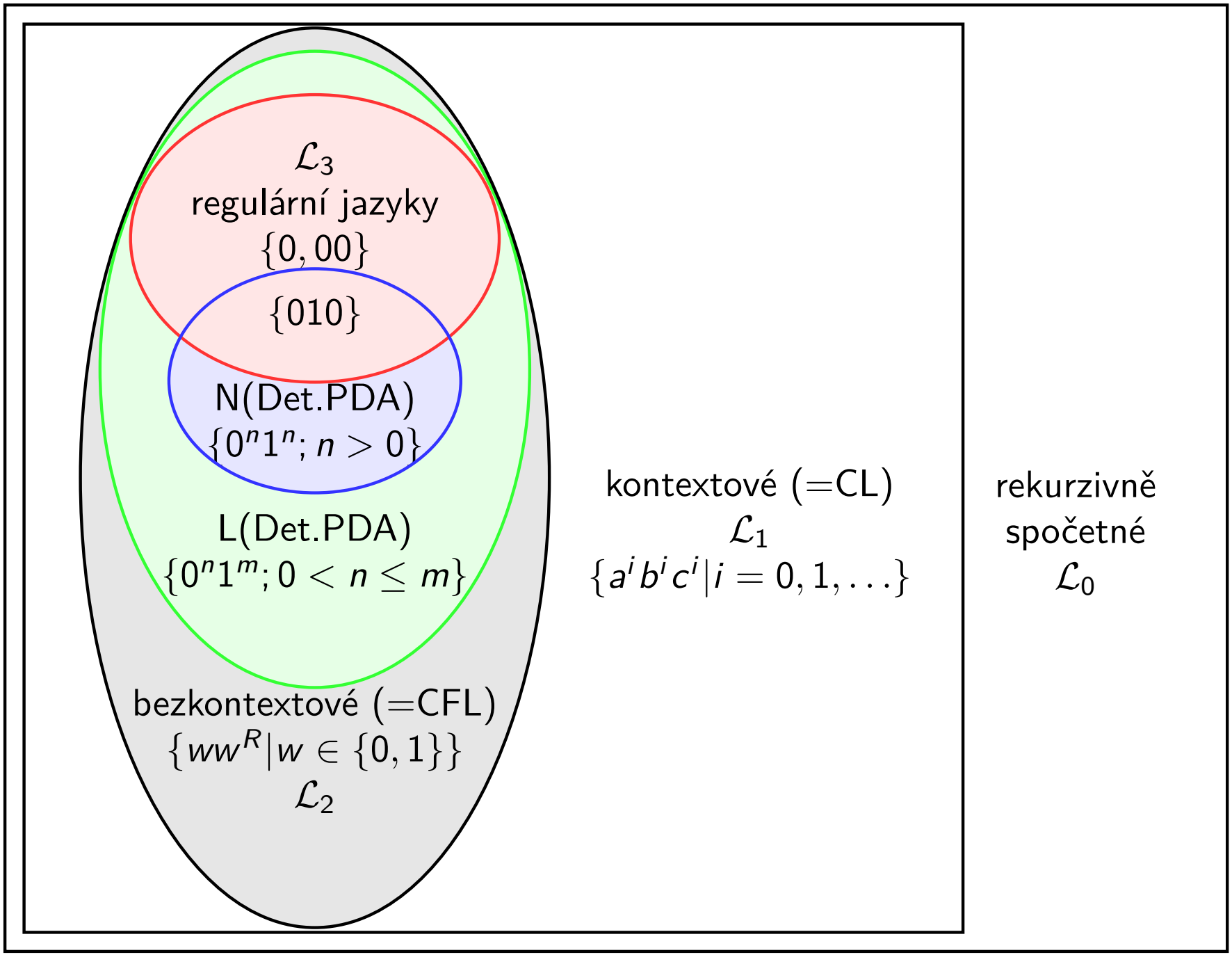
Indukce: $X_{ij} = \{A \to BC; B \in X_{ik}, C \in X_{k+1,j}\}$

- Vyplňujeme tabulku zdola nahoru.
- Pokud $S \in X_{1,n}$, potom $w \in L(G)$.

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

Uzávěrové vlastnosti v kostce

jazyk	regulární (RL)	bezkontextové	deterministické CFL
sjednocení	ANO	ANO	NE
průnik	ANO	NE	NE
$\cap$ s RL	ANO	ANO	ANO
doplněk	ANO	NE	ANO
homomorfismus	ANO	ANO	NE
inverzní hom.	ANO	ANO	ANO


$\mathcal{L}_3$
regulární jazyky
$\{0, 00\}$
$\{010\}$
N(Det.PDA)
$\{0^n1^n; n > 0\}$
L(Det.PDA)
$\{0^n1^m; 0 < n \leq m\}$
bezkontextové (=CFL)
$\{ww^R | w \in \{0, 1\}\}$
$\mathcal{L}_2$
kontextové (=CL)
$\mathcal{L}_1$
$\{a^i b^i c^i | i = 0, 1, \ldots\}$
rekurzivně
spočetné
$\mathcal{L}_0$