Teoretické základy informatiky

Teória množín

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Teória množín

- naivná
  - Intuitívne zavedená
  - Paradoxy – Russellov 1902
  - nekorektná
- axiomatická
  - Buduje sa z axióm
  - ZF: E. Zermelo 1871-1956, A.A.Fraenkel 1891-1956

Naivná teória množín

Pojem množiny definovaný intuitívne:

„Množina je *súhrn* nejakých objektov, jednoznačne rozlíšiteľných, ktoré nazývame prvky."

Množina sa zapisuje dvoma spôsobmi:

- vymenovaním prvkov,
- určením vlastností jej prvkov.

Príklady: naivná teória množín

Príklad 1.2.1 *Zápis množiny vymenovaním prvkov.*

$$A = \{a, c, d, f\}$$

Príklad 1.2.2 *Zápis množiny určením vlastností jej prvkov.*
Nech predikáty $P(x)$, $Q(x)$ a $R(x)$ označujú postupne nasledujúce vlastnosti: "x je párne číslo", "$x \geq 1$", "$x \leq 5$". Nech:

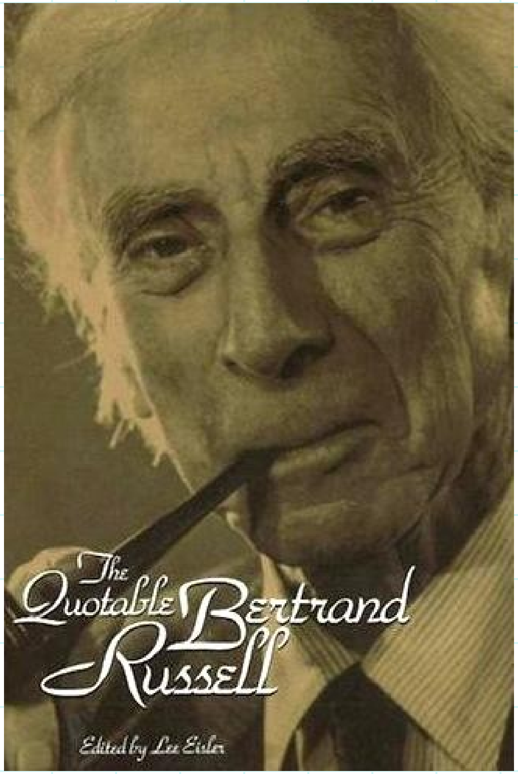
$$B = \{\ x \mid P(x) \wedge Q(x) \wedge R(x)\ \}.$$

Potom platí: $B = \{2, 4\}$.

Paradoxy naivnej teórie množín

Russellov paradox - 1902

$$R = \{A \mid A \notin A\}.$$



Skúmajme platnosť $A \in A$:

•áno $\Rightarrow$ A má spĺňať podmienku P, teda že $A \notin A$

•nie $\Rightarrow$ A nespĺňa podmienku P, teda $A \in A$

Paradox vojenského holiča

Axiomatická teória množín

1. **Axióma existencie množín**
   Existuje aspoň jedna množina.
   $(\exists X)(X = X)$

2. **Axióma extenzionality**
   Množiny, ktoré majú tie isté prvky, sa rovnajú.
   $(\forall U)(U \in X \Leftrightarrow U \in Y) \Rightarrow X = Y$

3. **Schéma axióma vydelenia**
   Z každej množiny je možné vydeliť množinu všetkých prvkov, ktoré spĺňajú danú formulu (vlastnosť).

4. **Axióma dvojice**
   Ľubovoľné dve množiny určujú dvojprvkovú množinu.

5. **Axióma sumy**
   Ku každej množine A je daná množina všetkých prvkov, ktoré patria do nejakého prvku množiny A.

Axiomatická teória množín

6. **Axióma potencie**
   Ku každej množine je daná množina všetkých podm-nožín.

7. **Schéma axióma nahradenia**
   Definovateľné zobrazenie zobrazuje množinu na množinu.

8. **Axióma nekonečna**
   Existuje nekonečná množina.

9. **Axióma fundovanosti**

POZN.: Postačujúce sú axiómy číslo 2,5,6,7,8,9; ostatné sa z nich dajú odvodiť.

Podmnožina

Definícia 1.3.1 Podmnožina
Množina A je podmnožinou *množiny B, píšeme $A \subseteq B$, ak každý prvok množiny A je aj prvkom množiny B.*

Množina A je vlastnou podmnožinou *množiny B, píšeme $A \subset B$, ak platí:* $A \subseteq B \wedge A \neq B$.

Platí, že každá množina je podmnožinou samej seba. Vzťah označovaný symbolom $\subseteq$ sa nazýva *inklúzia*. Vzťah označovaný symbolom $\subset$ sa nazýva *vlastná inklúzia*.

Prázdna množina

Definícia 1.3.2 Prázdna množina
Množina, do ktorej nepatrí žiadny prvok, sa nazýva prázdna, *označuje sa* $\emptyset$. *Teda:*

$$\forall x \quad x \notin \emptyset \, .$$

Existuje jediná prázdna množina.

Lema 1.3.1 *Pre ľubovoľné množiny A, B platí:*

$$A = B \quad \Leftrightarrow \quad A \subseteq B \wedge B \subseteq A \, . \tag{1.1}$$

Zo schémy axióm vydelenia sa odvodzujú množinové operácie prienik a rozdiel.

Prienik a rozdiel množín

Definícia 1.3.3 Prienik množín
Prienikom množín A *a* B *nazývame množinu*

$$A \cap B = \{\ x \mid x \in A\ \wedge\ x \in B\ \}\ . \tag{1.2}$$

Množiny C, D sa nayývajú *disjunktné*, ak $C \cap D = \emptyset$.

Definícia 1.3.4 Rozdiel množín
Rozdielom množín A *a* B *nazývame množinu*

$$A \setminus B = \{\ x \mid x \in A\ \wedge\ x \notin B\ \}\ . \tag{1.3}$$

Päť princípov tvorby množín

P1. Ak x, y sú nejaké objekty, tak:

$$\exists X \;\; \forall z \;\; z \in X \quad \Leftrightarrow \quad (z = x \;\vee\; z = y)\ .$$

P2. Ak A, B sú nejaké množiny, tak:

$$\exists C \;\; \forall z \;\; z \in C \quad \Leftrightarrow \quad (z \in A \;\vee\; z \in B)\ .$$

P3. Ak A, B sú nejaké množiny, tak:

$$\exists D \;\; \forall z \;\; z \in D \quad \Leftrightarrow \quad (\exists x \in A)(\exists y \in B)\; z = (x, y)\ .$$

P4. Ak A je množina, tak:

$$\exists E \;\; \forall z \;\; z \in E \quad \Leftrightarrow \quad z \subseteq A\ .$$

P5. Ak A je množina a P je vlastnosť (predikát), tak:

$$\exists F \;\; \forall z \;\; z \in F \quad \Leftrightarrow \quad (z \in A \;\wedge\; P(z))\ .$$

Zjednotenie množín

Definícia 1.3.5 Zjednotenie množín
Nech A a B sú množiny. Zjednotenie množín *A a B je množina*

$$A \cup B = \{\ x \mid x \in A \ \vee \ x \in B\ \}\ . \tag{1.4}$$

Operáciu zjednotenia je možné aplikovať aj viackrát. V takom prípade budeme používať nasledujúce označenie. Nech k je kladné celé číslo a nech $A_1, \ldots, A_k$ sú množiny. Potom budeme používať nasledujúce označenie:

$$A_1 \cup \cdots \cup A_k = \bigcup_{i=1}^{k} A_i\ .$$

Karteziánsky súčin množín, binárna relácia

Definícia 1.3.6 Karteziánsky súčin množín
Nech A a B sú množiny. Karteziánskym súčinom množín *A a B nazývame množinu*

$$A \times B = \{\ (a,b) \mid a \in A\ \wedge\ b \in B\ \}\ . \tag{1.5}$$

Prvkami množiny $A \times B$ sú *usporiadané dvojice* prvkov, ktoré budeme označovať symbolom (a,b). Ak $a \neq b$, tak $(a,b) \neq (b,a)$. Neusporiadané dvojice prvkov a,b budeme označovať symbolom $\{a,b\}$.

Analogickým spôsobom možno definovať karteziánsky súčin k množín - jeho prvkami sú usporiadané k-tice.

Definícia 1.3.7 Binárna relácia
Nech A a B sú množiny. Ľubovoľnú podmnožinu R karteziánskeho súčinu $A \times B$ budeme nazývať binárnou reláciou, *teda:*

$$R \subseteq A \times B\ .$$

Prvkami relácií sú usporiadané dvojice. Ak $(a,b) \in R$, tak používame tiež zápis aRb. Najdôležitejšou operáciou na binárnych reláciach je kompozícia relácií.

Kompozícia relácií

Definícia 1.3.8 Kompozícia relácií
Nech R_1 *a* R_2 *sú binárne relácie.* Kompozícia (binárnych) relácií, *označuje sa* $R_1 \circ R_2$*, skrátene* R_1R_2*, je definovaná nasledovne:*

$$R_1 \circ R_2 = R_1R_2 = \{ \ (x,z) \mid \exists y \ \ (x,y) \in R_1 \ \wedge \ (y,z) \in R_2 \ \} \ . \tag{1.6}$$

Ak $R \subseteq A \times A$ pre nejakú množinu A, tak hovoríme, že R je *relácia na množine* A. *Identická relácia* $I = \{ \ (a,a) \mid a \in A \ \}$ je tiež takou reláciou. Pre reláciu R na množine A definujeme operácie *mocniny, reflexívneho a tranzitívneho uzáveru.*

Zobrazenie

Definícia 1.3.9 Zobrazenie
Nech A a B sú množiny. Binárna relácia $\varphi \subseteq A \times B$ sa nazýva zobrazenie, *ak $\forall x \in A$ a $\forall u \in B$, $\forall v \in B$ platí:*

$$[\ (x,u) \in \varphi\ \wedge\ (x,v) \in \varphi\ ] \Rightarrow u = v\ .$$

Ak $\varphi \subseteq X \times Y$ je zobrazenie, tak píšeme $\varphi : X \to Y$ a $\varphi(x) = y$, ak $(x,y) \in \varphi$ pre $x \in X$ a $y \in Y$.

Zobrazenie $\varphi : X \to Y$ sa nazýva:

- *injektívne*, ak $\forall x_1, x_2 \in X$, $x_1 \neq x_2$ platí: $\varphi(x_1) \neq \varphi(x_2)$,
- *surjektívne*, ak $\forall y \in Y\ \ \exists x \in X$ tak, že $\varphi(x) = y$,
- *bijektívne*, ak je injektívne a súčasne surjektívne.

Analogicky ako pri reláciách, aj pre zobrazenia je možné definovať ich kompozíciu.

Kompozícia zobrazení

Definícia 1.3.10 Kompozícia zobrazení
Nech $\varphi_1 : X \to Y$ *a* $\varphi_2 : Y \to Z$ *sú zobrazenia.* Kompozícia zobrazení φ_1 *a* φ_2 *je zobrazenie* $(\varphi_2 \circ \varphi_1) = \psi : X \to Z$*, ak:*

$$\forall x \in X \quad \psi(x) = (\varphi_2 \circ \varphi_1)(x) = \varphi_2(\varphi_1(x)) \ . \tag{1.7}$$

Platí, že kompozícia dvoch injektívnych zobrazení je tiež injektívne zobrazenie, dvoch surjektívnych zobrazení je surjektívne zobrazenie a dvoch bijektívnych zobrazení je bijektívne zobrazenie.

Z princípu P4 sa odvodzuje vytvorenie potenčnej množiny.

Potenčná množina

Definícia 1.3.11 Potenčná množina
Nech A je množina. Potenčná množina $\mathcal{P}(A)$ je množina všetkých podmnožín množiny A, teda:

$$\mathcal{P}(A) = \{ X \mid X \subseteq A \} . \tag{1.8}$$

Konečné a nekonečné množiny

Definícia 1.4.1 Množina prirodzených čísel
Množina prirodzených čísel $\mathbb{N}$ *je najmenšia množina (vzhľadom na inklúziu), pre ktorú platí:*

1. $0 \in \mathbb{N}$,

2. *ak* $x \in \mathbb{N}$, *tak* $(x+1) \in \mathbb{N}$.

Z definície množiny prirodzených čísel je možné odvodiť korektnosť dôkazu matematickou indukciou.

O matematickej indukcii

Veta 1.4.1 O matematickej indukcii
Nech $P(x)$ je predikát vyjadrujúci nejakú vlastnosť čísel. Nech $k \in \mathbb{N}$. Predpokladajme, že platí:

1. $P(0)$,
2. *ak* $P(k)$, *tak* $P(k+1)$.

Potom pre všetky $n \in \mathbb{N}$ platí $P(n)$, teda vlastnosť P majú všetky prirodzené čísla.

Nekonečné množiny - príklady

Množina kladných celých čísel $\mathbb{N}^+$

$$\mathbb{N}^+ = \mathbb{N} \setminus \{0\} \tag{1.11}$$

Množina (nezáporných) párnych čísel $\mathbb{E}v$ *(Even)*

$$\mathbb{E}v = \{\ 2 \cdot k \mid k \in \mathbb{N}\ \} \tag{1.12}$$

Množina (kladných) nepárnych čísel $\mathbb{O}dd$

$$\mathbb{O}dd = \{\ 2 \cdot k + 1 \mid k \in \mathbb{N}\ \} \tag{1.13}$$

Množina celých čísel $\mathbb{Z}$

$$\mathbb{Z} = \mathbb{N} \cup \{\ -k \mid k \in \mathbb{N}\ \} \tag{1.14}$$

Neformálny popis nekonečných množín - príklady

Množina všetkých prvočísel $\mathbb{P}r$
Je to množina všetkých takých čísel z $\mathbb{N}^+$, ktoré majú iba dva rôzne kladné delitele.

Množina racionálnych čísel $\mathbb{Q}$
Každé racionálne číslo sa dá zapísať v tvare *zlomku*, teda v tvare $\frac{p}{q}$, pričom $p \in \mathbb{Z}, q \in \mathbb{N}^+$. Avšak rôzne zlomky môžu vyjadrovať to isté číslo (napr. $\frac{1}{5} = \frac{3}{15}$). Preto sa množina $\mathbb{Q}$ definuje ako množina všetkých zlomkov v tzv. vykrátenom (resp. kanonickom) tvare.

Množina reálnych čísel $\mathbb{R}$
Množina reálnych čísel vznikne tzv. zúplnením množiny racionálnych čísel. Okrem všetkých čísel z $\mathbb{Q}$ obsahuje aj ďalšie čísla, ako napr. $\sqrt{2}, \sqrt{3}, \pi$ a iné.

Mohutnosť množiny, počet prvkov množiny

Namiesto pojmu ”počet prvkov množiny” budeme používať termín ”mohutnosť množiny”. Pri definovaní mohutnosti budeme vychádzať z pojmu zobrazenia.

Mohutnosť množiny
Symbolom $|A|$ budeme označovať *mohutnosť (kardinalitu) množiny* A.

Definícia 1.4.2 Porovnávanie mohutností
Nech A, B *sú množiny. Budeme hovoriť, že* množiny A *a* B majú rovnakú mohutnosť, *ak existuje bijektívne zobrazenie* $\varphi : A \to B$. *Túto skutočnosť budeme zapisovať označením* $|A| = |B|$ *alebo* $A \approx B$.

Aby sme mohli mohutnosti množín porovnávať efektívnejšie, zavedieme pre mohutnosti ešte aj relácie $\leq$ a $<$.

Definícia 1.4.3 Porovnávanie mohutností
Nech A, B *sú množiny. Budeme hovoriť, že množina* A má mohutnosť menšiu alebo rovnakú ako množina B, *píšeme* $|A| \leq |B|$, *ak existuje injektívne zobrazenie* $\psi : A \to B$.
Množina A má mohutnosť menšiu ako množina B, *píšeme* $|A| < |B|$, *ak* $|A| \leq |B|$ *a neplatí* $|A| = |B|$.

Pomocou relácie $<$ definujeme pojem konečnej množiny.

Konečná množina

Definícia 1.4.4 Konečná množina
Množina A sa nazýva konečná, *ak* $|A| < |\mathbb{N}|$.

Mohutnosť konečných množín je možné vyjadriť prirodzeným číslom. Toto číslo sa rovná tomu, čo intuitívne chápeme ako počet prvkov danej množiny. Exaktne sa tento pojem zavádza nasledujúcim spôsobom.

Nech $k \in \mathbb{N}$. Mohutnosť konečnej množiny A označíme číslom k, píšeme $|A| = k$, ak $|A| = |\{1, 2, \ldots, k\}|$. Mohutnosť prázdnej množiny sa označuje nulou.

Nasledujúca veta vyčísľuje mohutnosť rôznych konečných množín vo vzťahu k niektorým množinovým operáciam.

Veta 1.4.2 *Pre konečné množiny A, B platí:*

$$|A \times B| = |A| \cdot |B| \tag{1.17}$$

$$|A \cup B| \leq |A| + |B| \tag{1.18}$$

$$|A \cap B| \leq \min\{|A|, |B|\} \tag{1.19}$$

$$|A \setminus B| \leq |A| \tag{1.20}$$

$$|\mathcal{P}(A)| = 2^{|A|} \ . \tag{1.21}$$

Príklady

Príklad 1.4.1 *Dokážeme, že platí* $|\mathbb{O}dd| = |\mathbb{N}|$.

Definujme zobrazenie $\varphi : \mathbb{O}dd \to \mathbb{N}$ *tak, že každé nepárne číslo sa zobrazí na dolnú celú časť jeho polovice, t.j.:*

$$\varphi(k) = \lfloor k/2 \rfloor \qquad \textit{pre všetky} \quad k \in \mathbb{O}dd \ .$$

Konkrétne, $\varphi(1) = 0$, $\varphi(3) = 1$, $\varphi(5) = 2$, $\varphi(7) = 3$, *atď.*
Uvedené zobrazenie je bijektívne. *Z toho vyplýva, že platí* $|\mathbb{O}dd| = |\mathbb{N}|$. $\square$

Napriek tomu, že množiny $\mathbb{O}dd$ a $\mathbb{E}v$ sú vlastnými podmnožinami množiny $\mathbb{N}$, všetky tri majú rovnakú mohutnosť. Podobne $\mathbb{N}^+ \subset \mathbb{N}$, pričom obidve množiny majú rovnakú mohutnosť.

Príklad 1.4.2 *Dokážeme, že platí* $|\mathbb{N}^+| = |\mathbb{N}|$.

Nech $\varphi : \mathbb{N}^+ \to \mathbb{N}$ *je zobrazenie, definované ako*

$$\varphi(k) = k - 1 \qquad \textit{pre všetky} \quad k \in \mathbb{N}^+ \ .$$

Zobrazenie φ *je bijektívne.* *Z toho vyplýva platnosť rovnosti* $|\mathbb{N}^+| = |\mathbb{N}|$. $\square$

Veta 1.4.3 *Platí:* $|\mathbb{N}| = |\mathbb{Z}|$.

Dôkaz Definujme zobrazenie $\varphi : \mathbb{N} \to \mathbb{Z}$ tak, že sa všetky párne čísla z $\mathbb{N}$ zobrazia na nezáporné zo $\mathbb{Z}$ a všetky nepárne čísla z $\mathbb{N}$ sa zobrazia na záporné zo $\mathbb{Z}$. Formálne:

$$\varphi(k) = \begin{cases} k/2 \,, & \text{ak} \quad k \in \mathbb{E}v \\ -\lceil k/2 \rceil \,, & \text{ak} \quad k \in \mathbb{O}dd. \end{cases}$$

Konkrétne: $\varphi(0) = 0$, $\varphi(2) = 1$, $\varphi(4) = 2$, $\ldots$ a tiež: $\varphi(1) = -1$, $\varphi(3) = -2$, $\varphi(5) = -3$, $\ldots$
Nie je ťažké dokázať, že zobrazenie φ je bijektívne. Z toho vyplýva platnosť dokazovanej rovnosti. $\square$

Lema 1.4.2 *Nech A, B sú disjunktné nekonečné množiny, pre ktoré platí: $|\mathbb{N}| = |A| = |B|$. Potom platí:*

$$|\mathbb{N}| = |A \cup B|.$$

Dôkaz Z predpokladu $|\mathbb{N}| = |A|$ vyplýva, že existuje bijektívne zobrazenie $\varphi_1 : \mathbb{N} \to A$, ktoré každé prirodzené číslo zobrazí na nejaký prvok z množiny A. Konkrétne, prirodzené číslo $n \in \mathbb{N}$ sa zobrazí na prvok $\varphi_1(n) \in A$. Podobne, z rovnosti $|\mathbb{N}| = |B|$ vyplýva, že existuje bijektívne zobrazenie $\varphi_2 : \mathbb{N} \to B$, ktoré prirodzené číslo $m \in \mathbb{N}$ zobrazí na prvok $\varphi_2(m) \in B$. Keďže množiny A a B sú disjunktné, môžeme definovať nové zobrazenie $\psi : \mathbb{N} \to (A \cup B)$ nasledujúcim spôsobom:

$$\psi(k) = \begin{cases} \varphi_1(k/2), & \text{ak } k \in \mathbb{E}v \\ \varphi_2(\lfloor k/2 \rfloor), & \text{ak } k \in \mathbb{O}dd. \end{cases}$$

Teda platí: $\psi(0) = \varphi_1(0)$, $\psi(2) = \varphi_1(1)$, $\psi(4) = \varphi_1(2)$, $\psi(6) = \varphi_1(3)$, atď.
To znamená, že v zobrazení ψ sa napr. číslo 4 zobrazí na ten prvok z množiny A, na ktorý bolo v zobrazení φ_1 zobrazené číslo 2.
Tiež platí, že $\psi(1) = \varphi_2(0)$, $\psi(3) = \varphi_2(1)$, $\psi(5) = \varphi_2(2)$, $\psi(7) = \varphi_2(3)$, atď.
Zobrazenie ψ je bijektívne. Tým sme dokázali, že platí $|\mathbb{N}| = |A \cup B|$. $\square$

Poznámka 1.4.1 *Tvrdenie z lemy 1.4.2 platí aj vtedy, ak v predpokladoch vynecháme podmienku disjunktnosti množín A a B.*

Veta 1.4.4 *Platí:* $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$.

Dôkaz Myšlienka dôkazu je založená na nasledujúcej úvahe. Prvkami množiny $\mathbb{N} \times \mathbb{N}$ sú usporiadané dvojice prirodzených čísel, ktoré usporiadame do dvojrozmernej tabuľky (viď Tabuľka 1.1), ohraničenej ľavým a horným okrajom.

$\mathbb{N} \times \mathbb{N}$	0	1	2	3	$\cdots$
0	$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$\cdots$
1	$(1,0)$	$(1,1)$	$(1,2)$	$(1,3)$	$\cdots$
2	$(2,0)$	$(2,1)$	$(2,2)$	$(2,3)$	$\cdots$
3	$(3,0)$	$(3,1)$	$(3,2)$	$(3,3)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Tabuľka 1.1: Usporiadané dvojice z $\mathbb{N} \times \mathbb{N}$.

Prvky v tejto tabuľke očíslujeme prirodzenými číslami tak, aby uvedené očíslovanie reprezentovalo bijektívne zobrazenie $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Dvojiciam v tabuľke budeme priraďovať prirodzené čísla v smere diagonály, zhora-dole a sprava-doľava tak, ako je to naznačené v tabuľke (viď Tabuľka 1.2).

$\mathbb{N}$	0	1	2	3	$\cdots$
0	0	1	3	6	$\cdots$
1	2	4	7	11	$\cdots$
2	5	8	12	17	$\cdots$
3	9	13	18	24	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Tabuľka 1.2: Spôsob očíslovania prvkov z $\mathbb{N} \times \mathbb{N}$.

Platí: $f(0,0) = 0$, $f(0,1) = 1$, $f(1,0) = 2$, $f(0,2) = 3$, $f(1,1) = 4$, $f(2,0) = 5$, atď. Funkčný predpis potom možno vyjadriť nasledovne:

$$f(m,n) = \frac{1}{2}(m+n)(m+n+1) + m \; .$$

Takto definované zobrazenie je bijektívne, a teda požadovaná rovnosť je dokázaná.

Veta:
$|\mathbb{N}| < |\mathbb{R}|$

Dôkaz sporom:
Nech $|\mathbb{N}| = |\mathbb{R}|$, potom aj $|(0,1)| = |\mathbb{N}|$ lebo $|\mathbb{R}| = |(0,1)| \Rightarrow$ čísla z (0, 1) je možné zoradiť do nekonečnej postupnosti:

$(0,\ 1) = \{a_n | n \in \mathbb{N}^+\}$

z vlastností postupností nad $\mathbb{R}$ vyplýva, že $\forall a_n$ má dekadický zápis:

$a_n = \sum_{k=1}^{\infty} (a_{n,k}.10^{-k})$

Ak by a_n malo dva rôzne dekadické zápisy, tak vyberieme nekonečný zápis; t.j. taký, že pre nekonečne veľa k je $a_{n,k} \neq 0$.
Čísla a_n a ich dekadické zápisy zoradíme do tabuľky a diagonalizáciou zostrojíme číslo

$b = 0.b_1 b_2 ... = \sum_{k=1}^{\infty} (b_k.10^{-k})$

Spočítateľná a nespočítateľná množina

Definícia 1.4.5 Spočítateľná a nespočítateľná množina
Množina A sa nazýva spočítateľná, *ak* $|A| \leq |\mathbb{N}|$. *(T. j.* $|A| \leq \aleph_0$*)*
Množina, ktorá nie je spočítateľná, sa nazýva nespočítateľná.

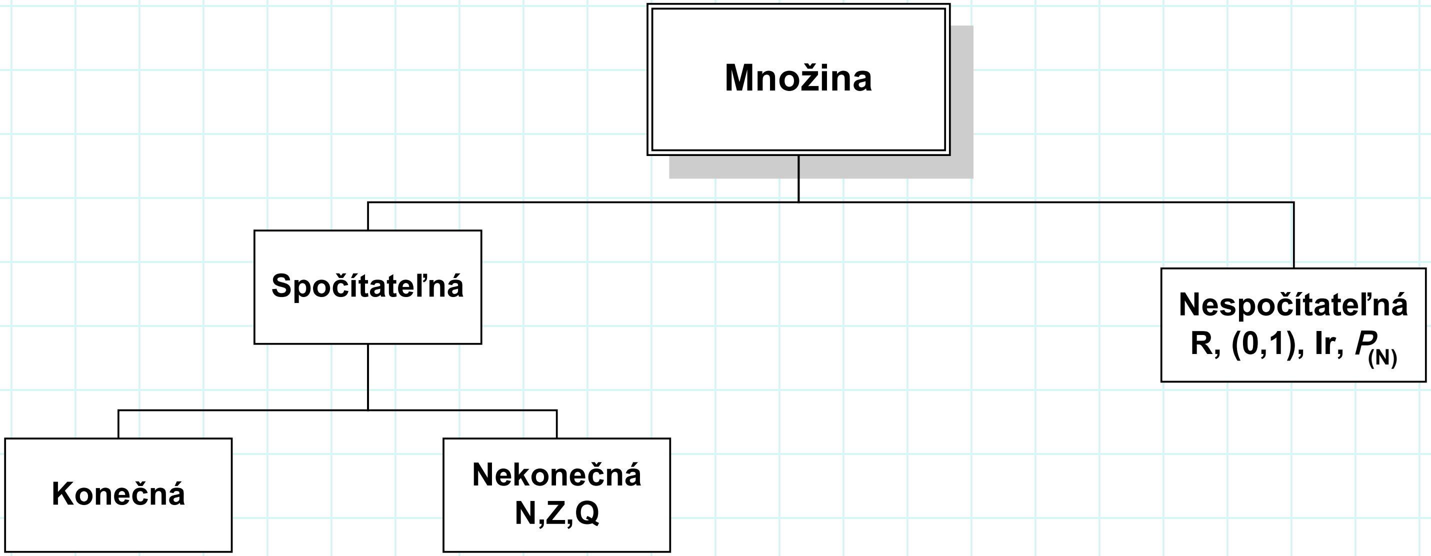
V nasledujúcom tvrdení dokážeme, že aj množina iracionálnych čísel je nespočítateľná.

Veta 1.4.7 *Množina* $\mathbb{R} \setminus \mathbb{Q}$ *iracionálnych čísel je nespočítateľná.*

Dôkaz Vetu dokážeme sporom. Predpokladajme, že množina $\mathbb{R} \setminus \mathbb{Q}$ je spočítateľná. V tom prípade je buď konečná alebo nekonečná spočítateľná.
Predpokladajme najprv, že množina $\mathbb{R} \setminus \mathbb{Q}$ je konečná. Keďže podľa vety 1.4.5 platí, že $|\mathbb{Q}| = \aleph_0$, dalo by sa zostrojiť bijektívne zobrazenie $(\mathbb{R} \setminus \mathbb{Q}) \cup \mathbb{Q} \to \mathbb{N}$. (Pozri cvičenie 1.5.6.) To by ale znamenalo, že množina $(\mathbb{R} \setminus \mathbb{Q}) \cup \mathbb{Q} = \mathbb{R}$ je nekonečná spočítateľná, čo je spor s vetou 1.4.6!

V prípade, keby bola množina $\mathbb{R} \setminus \mathbb{Q}$ nekonečná spočítateľná, dostaneme taktiež spor. Množina $\mathbb{Q}$ je nekonečná spočítateľná, pričom množiny $\mathbb{Q}$ a $\mathbb{R} \setminus \mathbb{Q}$ sú disjunktné. Podľa lemy 1.4.2 potom ich zjednotenie, teda množina $\mathbb{R}$, musí byť spočítateľná. Opäť sme dostali spor s vetou 1.4.6! Tým je tvrdenie dokázané. $\square$

Klasifikácia množín vzhľadom na ich mohutnosť



Ďakujem za pozornosť.

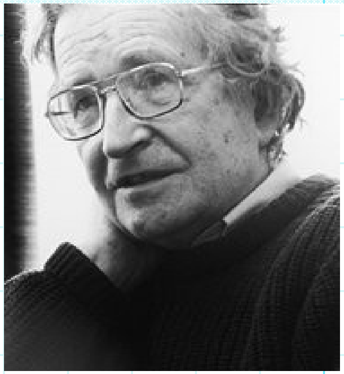
chuda@fiit.stuba.sk

Teoretické základy informatiky

Gramatiky a Jazyky

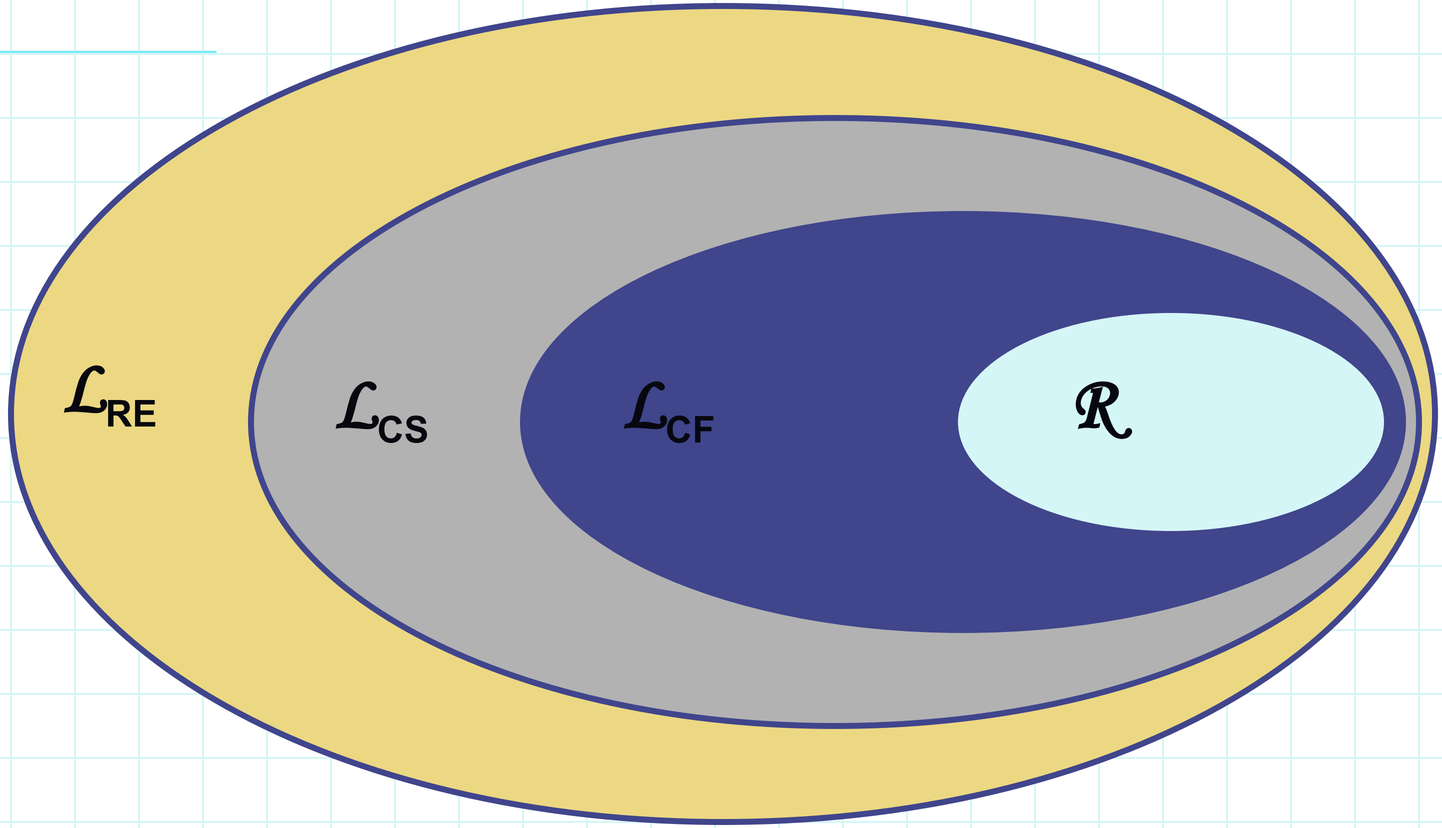
Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Úvod



- americký lingvista Noam Chomsky: iniciovanie teórie formálnych jazykov (TFJ) – 50-te roky 20.storočia
- TFJ – súhrn poznatkov, ktoré sa v informatike uplatňujú pri tvorbe prekladačov a textoých procesorov
- abstraktná úroveň – 3 typy jazykov:
  - prirodzené (lingvistické),
  - umelé,
  - formálne.
- Každý jazyk je možné skúmať z dvoch hľadísk:
  - syntax (štruktúra a stavba),
  - sémantika (význam a obsah).

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)

$\mathcal{L}_{CS}$ Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)

$\mathcal{L}_{CF}$ Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)

$\mathcal{R}$ Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)

Jazyky a ich reprezentácie

Definícia jazyka podľa webster dictionary:

„Jazyk ako súbor slov a metód skladania slov, ktorý používa a chápe dostatočne veľká ľudská spoločnosť.“

(nepresná definícia pre vybudovanie matematickej teórie jazykov)

Formálny jazyk – abstraktne definovaný ako matematický systém.

Reprezentácia jazyka (špecfikácia slov jazyka) **:**

- Zostrojiť algoritmus, ktorý určí, či dané slovo patrí do jazyka. Rozpoznanie jazyka. Automat.
- Systematicky generovať slová z jazyka. Generovanie jazyka. Gramatika.
- Matematickým opisom jazyka. Množina.

Abeceda

Abeceda, označuje sa Σ, je konečná množina prvkov, ktoré nazývame *symboly* alebo *znaky*. *Slovo* (resp. *reťazec*) nad abecedou Σ je ľubovoľná (môže byť aj opakujúca sa) postupnosť symbolov zo Σ. Slová budeme označovať najčastejšie písmenami u, v, w, x, y alebo ako postupnosti znakov, teda napr.:

$$w = a_1 a_2 \dots a_k \ ,$$

kde w je slovo, $k \in \mathbb{N}^+$ a a_i sú symboly z abecedy Σ, kde $i = 1, \dots, k$.

Slovo, podslovo, prázdne slovo

Slovo (reťazec) je ľubovoľná postupnosť symbolov z abecedy.

Podslovo slova u je nejaká jeho časť po sebe nasledujúcich symbolov.

Prázdne slovo je slovo, ktoré neobsahuje žiaden znak, budeme ho označovať $\boldsymbol{\varepsilon}$.

Príklad: abeceda, slovo, podslovo

abeceda: $\Sigma = \{a,b,c\}$

slová: abba, aaab=a^3b , aab≠aba

Prefix – počiatočné podslovo **<u>ca</u>**a

Postfix – koncové podslovo ab**<u>ca</u>**

infix ab**<u>bc</u>**a

Rovnosť slov

Rovnosť slov sa definuje nasledovne. Nech $k, l \in \mathbb{N}^+$ a nech $u = a_1 \ldots a_k$, $v = b_1 \ldots b_l$ sú slová. Platí, že $u = v$, ak $k = l$ a súčasne $a_i = b_i$ pre všetky $i = 1, \ldots, k$.

Naopak, slová u, v *sa nerovnajú* (píšeme $u \neq v$), ak $k \neq l$ alebo existuje také j $(1 \leq j \leq k)$, že $a_j \neq b_j$.

Príklady: rovnosť slov

$aaab=a^3b$

$aab.\varepsilon=\varepsilon.aab$

$aab=aab$

Zreťazenie slov

Zreťazenie slov $u = a_1 \dots a_k$, $v = b_1 \dots b_l$ je operácia (označuje sa $\cdot$), ktorej výsledkom je slovo $u \cdot v = a_1 \dots a_k b_1 \dots b_l$. Ak u, v, w sú ľubovoľné slová nad danou abecedou Σ, tak operácia zreťazenia spĺňa nasledujúce vlastnosti:

1. (*uzavretosť*) $u \cdot v$ je tiež slovo nad abecedou Σ,
2. (*asociativita*) $(u \cdot v) \cdot w = u \cdot (v \cdot w)$,
3. (*existencia neutrálneho prvku*) $u \cdot \epsilon = \epsilon \cdot u = u$.

$$u \cdot v = uv$$

Príklad: zreťazenia

u=aa

v=bb

x=aab

y=ε

u.v=aabb

u.x=aaaab

u.y.x=aaaab

Príklad: abeceda, podslovo, zreťazenia

$A=\{0,1\}$

$x_1=0101$

$x_2=111$

$x_3=101$

$x_1 . x_2 = 0101111$

$A=\{0,1\}$

Slovo $x_3=101$ môže mať:

prefix: ε, 1, 10, 101

postfix: ε, 1, 01, 101

podslovo: ε, 1, 10, 101, 01, 0

$A=\{0,1\}$

$A^*=\{\varepsilon, 0,1,00,01,10,11,000,\ldots.\}$

$A^+=\{0,1,00,01,10,11,000,\ldots.\}$

Vlastnosti zreťazenia

	Sčítanie	Násobenie	Zreťazenie
použitie	3+5=8	3*5=15	3·5=35
komutatívnosť	áno 3+5=5+3	áno 3*5=5*3	nie 3·5≠5·3
asociatívnosť	áno (3+5)+7=3+(5+7)	áno (3*5)*7=3*(5*7)	áno (3·5)·7=3·(5·7)
neutrálny prvok	0 3+0=0+3=3	1 3*1=1*3=3	ε w·ε=ε·w=w
inverzný prvok	opačné číslo 3+(-3)=0	obrátené číslo 3*(1/3)=1	neexistuje

Zreťazenie je binárna operácia, je asociatívna, nie je komutatívna, neutrálny prvok je ε a pri jej aplikovaní neexistuje inverzný prvok.

Operácie nad slovami

- Mocnina slova
- Zrkadlový obraz
- Dĺžka slova
- Počet výskytov symbolu
- Homomorfizmus

Mocnina slova

Mocnina slova u, označuje sa u^i, ak $i \in \mathbb{N}$.

1. $u^0 = \epsilon$,
2. $u^i = u^{i-1} \cdot u$ pre $i \in \mathbb{N}^+$.

Špeciálne, ak $b \in \Sigma$ a $k \in \mathbb{N}^+$, tak:

$$\underbrace{b \ldots b}_{k} = b^k \ .$$

Príklady: mocnina slova

$u=a$

$u^0=\varepsilon$
$u^1=a$
$u^2=aa$
$u^3=aaa$
$u^4=aaaa$
$u^5=aaaaa$

Zrkadlový obraz

Zrkadlový obraz slova, označuje sa ^{R}.

1. $\epsilon^R = \epsilon$,
2. $(a \cdot u)^R = u^R \cdot a$.

Slovo, pre ktoré platí $u^R = u$, sa nazýva *palindróm*.

Príklad: zrkadlový obraz

$(aa)^R=aa$

$(abaccb)^R=bccaba$

$(a^Nb^N)^R=b^Na^N$

$(aa)^R=aa$

$(abc)^R=cba$

$(radar)^R=radar$

Slová, ktoré majú rovnaký tvar po aplikovaní zrkadlového obrazu – **palindróm**.

Dĺžka slova

Dĺžka slova, označuje sa $|\_|$.

1. $|\epsilon| = 0$,
2. $|a \cdot u| = 1 + |u|$.

Dĺžka slova spĺňa nasledujúce vlastnosti.

Lema 2.1.1 *Pre všetky slová u, v platí:*

- $|u^R| = u$,
- $|u \cdot v| = |u| + |v|$.

STU FIIT

Príklady: dĺžka slova

$|aabc|=4$

$|\ a^n b^n\ | =2n$

$|\ \varepsilon\ | =0$

$|aaa.bb| =|aaa|+|bb|=3+2=5$

$|\ a^3b^2\ | =5$

$|(a^3b^2c)^R| = |cb^2a^3| =6$

Zrkadlový obraz zachováva dĺžku slova.

Počet výskytov symbolu

Počet výskytov daného symbolu v slove, označuje sa $\#$.

1. $\#_z\, \epsilon = 0$,
2. $\#_z\, (a \cdot u) = \begin{cases} \#_z\, u, & ak \;\; z \neq a \\ 1 + \#_z\, u, & ak \;\; z = a \;. \end{cases}$

Príklad: počet výskytov slova

$\#_a(aa)=2$

$\#_a(aba)=2$

$\#_b(aba)=1$

$\#_a(a^ib^ja^k)=i+k$

Homomorfizmus

Homomorfizmus nad slovami, označuje sa h, definujeme nasledovne. Ak Σ_1 a Σ_2 sú abecedy, tak homomorfizmus h je zobrazenie z množiny všetkých slov nad abecedou Σ_1 do množiny všetkých slov nad abecedou Σ_2, pričom pre ľubovoľné slová u, v nad abecedou Σ_1 platí:

$$h(u \cdot v) = h(u) \cdot h(v) \ .$$

Každý homomorfizmus spĺňa nasledujúcu vlastnosť.

Lema 2.1.2 *Platí:*

$$h(\epsilon) = \epsilon \ .$$

Príklad: homomorfizmus

$\Sigma_1 = \{a,b,c\}$ $\Sigma_2 = \{0,1\}$

$h(abc)=101$

$h(a)=1$
$h(b)=0$
$h(c)=1$

Definícia: Jazyk

Def.: Jazyk je *množina* slov.

Mohutnosť:

- jazyky konečné
- jazyky nekonečné (väčšinou spočítateľné)

Množinové operácie:

- zjednotenie, prienik, rozdiel, rovnosť, nerovnosť, podmnožina,
- zreťazenie jazykov, mocnina jazyka, kleeneho iterácia - uzáver, kladný uzáver, doplnok jazyka.

Zreťazenie jazykov

Zret'azenie jazykov, označuje sa ”·”. Nech L_1 a L_2 sú jazyky, potom ich zret'azenie je definované nasledovne:

$$L_1 \cdot L_2 = L_1 L_2 = \{ \; uv \mid u \in L_1, v \in L_2 \; \} \; .$$

Ani pre zret'azenie jazykov neplatí komutatívny zákon, preto je potrebné rozlišovat' jazyky $L_1 \cdot L_2$ a $L_2 \cdot L_1$.

Mocnina jazyka

Mocnina jazyka L, označuje sa L^i, ak $i \in \mathbb{N}$. Nech L je jazyk, potom:

1. $L^0 = \{\epsilon\}$,
2. $L^i = L^{i-1} \cdot L$ pre $i \in \mathbb{N}^+$.

Kleeneho iterácia (uzáver jazyka)

Kleeneho iterácia alebo *uzáver jazyka* L, označuje sa L^*. Ak L je jazyk, tak:

$$L^* = \bigcup_{i=0}^{\infty} L^i \,.$$

Kladný uzáver jazyka

Kladný uzáver jazyka L, označuje sa L^+. Ak L je jazyk, tak:

$$L^+ = \bigcup_{i=1}^{\infty} L^i \,.$$

Doplnok (komplement) jazyka

Doplnok (komplement) jazyka L, označuje sa L^C. Ak L je jazyk, tak:

$$L^C = \Sigma_L^* \setminus L \ ,$$

kde Σ_L označuje abecedu, nad ktoru sú vytvorené slová z jazyka L.

Lema 2.1.3 *Nech Σ je abeceda a nech Q je predikát určujúci nejakú vlastnosť slov. Ak je možné jazyk L napísať v tvare*

$$L = \{ \ w \in \Sigma^* \mid Q(w) \ \} \ ,$$

tak platí, že:

$$L^C = \{ \ w \in \Sigma^* \mid \neg Q(w) \ \} \ .$$

S T U · ·
F I I T ·

Homomorfizmus jazyka

Homomorfizmus jazyka L, označuje sa $h(L)$. Nech Σ_1 a Σ_2 sú abecedy, nech L je jazyk nad abecedou Σ_1 a nech $h : \Sigma_1^* \to \Sigma_2^*$ je homomorfizmus. Potom:

$$h(L) = \{ \, h(w) \mid w \in L \, \} \, .$$

STU FIIT

Gramatiky

- Gramatika – možnosť ako konečným spôsobom určiť nekonečný jazyk.
- Opis jazyka generatívnou paradigmou.
- Generovanie: slová jazyka sa postupne vytvárajú z kratších/jednoduchších na dlhšie/zložitejšie.
- Široké uplatnenie pri definovaní umelých (najmä počítačových) jazykov.

Frázová gramatika

Definícia 2.2.1 Frázová gramatika

Frázová gramatika *je štvorica* $G = (N, T, P, S)$, *kde* N *a* T *sú abecedy* terminálnych *resp.* neterminálnych symbolov, *pričom platí:* $(N \cap T) = \emptyset$, *ďalej*

$$P \subseteq_{KON} \ (N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$$

je konečná množina pravidiel *a* $S \in N$ *je* počiatočný (štartovací) neterminálny symbol.

Krok odvodenia *v gramatike* G *je binárna relácia* $\Rightarrow_G$ *na množine* $(N \cup T)^* \times (N \cup T)^*$ *definovaná nasledovne:*

$$x \Rightarrow_G y \ ,$$

ak existujú $w_1, w_2 \in (N \cup T)^*$ *a pravidlo* $(u \rightarrow v) \in P$, *že platí:*

$$x = w_1 u w_2 \qquad a \qquad y = w_1 v w_2 \ .$$

Jazyk definovaný gramatikou G *je množina* $L(G)$ *daná nasledovne:*

$$L(G) = \{ \ w \in T^* \mid S \Rightarrow_G^* w \ \} \ ,$$

pričom $\Rightarrow_G^*$ *je reflexívny a tranzitívny uzáver relácie* $\Rightarrow_G$.

Reťazec $x \in (N \cup T)^*$ sa nazýva *vetná forma* v gramatike $G = (N, T, P, S)$, ak $S \Rightarrow_G^* x$.

Chomského hierarchia gramatík

Definícia 2.2.2 Chomského hierarchia gramatík

Frázová gramatika *je gramatika v zmysle definície 2.2.1.*

Kontextová gramatika *je taká frázová gramatika, v ktorej všetky pravidlá majú tvar:*

$$u \to v \ , \quad kde \ \ |u| \leq |v| \ .$$

Bezkontextová gramatika *je taká frázová gramatika, v ktorej všetky pravidlá majú tvar:*

$$A \to w \ , \quad kde \ \ A \in N \ , \quad w \in (N \cup T)^* \ .$$

Regulárna gramatika *je taká frázová gramatika, v ktorej všetky pravidlá majú tvar:*

$$A \to wB \ , \quad alebo \quad A \to w \ ,$$

$$kde \quad A, B \in N \ , \quad w \in T^* \ .$$

Chomského hierarchia gramatík indukuje hierarchiu jazykov.

Chomského hierarchia jazykov

Definícia 2.2.3 Chomského hierarchia jazykov

Jazyk sa nazýva kontextový (bezkontextový, *resp.* regulárny), *ak je generovaný kontextovou (bezkontextovou, resp. regulárnou) gramatikou.*

Jazyk sa nazýva rekurzívne vyčísliteľný, *ak je generovaný frázovou gramatikou.*

Pre označenie tried jazkov patriacich do Chomského hierarchie sa používajú nasledujúce symboly:

- $\mathcal{L}_{RE}$ - trieda všetkých rekurzívne vyčísliteľných jazykov,
- $\mathcal{L}_{CS}$ - trieda všetkých kontextových jazykov,
- $\mathcal{L}_{CF}$ - trieda všetkých bezkontextových jazykov,
- $\mathcal{R}$ - trieda všetkých regulárnych jazykov.

Vzťahy medzi triedami jazykov

Veta 2.2.1 *Medzi triedami jazkov Chomského hierarchie platia nasledujúce vzťahy:*

$$\mathcal{R} \subset \mathcal{L}_{CF} \subset \mathcal{L}_{CSE} \subset \mathcal{L}_{RE} \,. \tag{2.1}$$

Poznámka 2.2.1 *Označenia jednotlivých tried sa odvodzujú z príslušných anglických názvov:*

RE - recursive enumerable,

CS - context sensitive,

CF - context free,

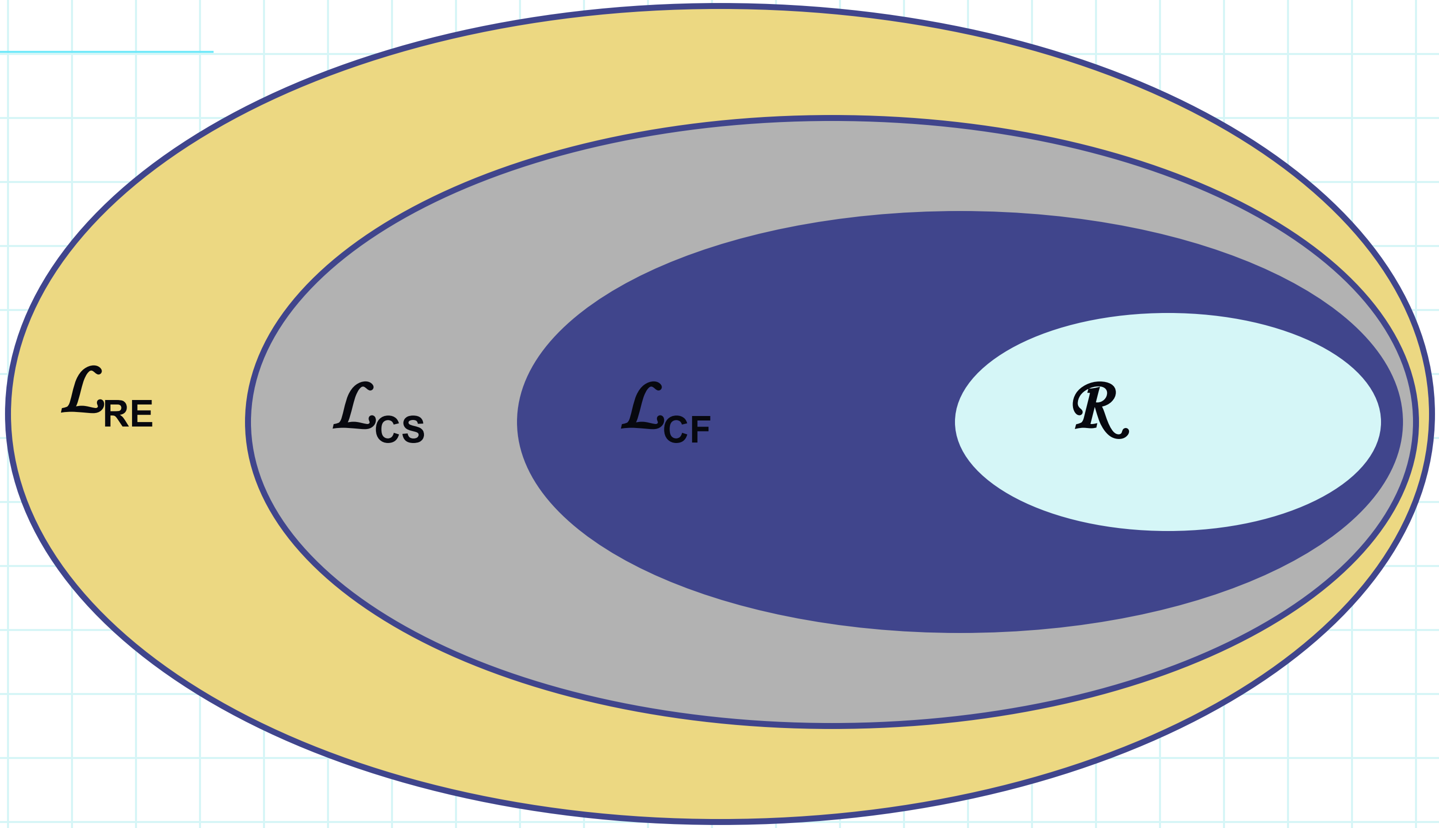
R - regular.

Trieda jazykov L_{CSE}

Uvedomme si, že žiadne pravidlo kontextovej gramatiky nemôže generovať jazyk, ktorý by obsahoval prázdne slovo. Naprotitomu všetky ostatné typy gramatík takýto jazyk generovať môžu. Preto zavedieme ešte jednu triedu jazykov, $\mathcal{L}_{CSE}$, ktorá obsahuje spolu so všetkými kontextovými jazyky aj všetky kontextové jazyky obohatené o prázdne slovo.

$$\mathcal{L}_{CSE} = \mathcal{L}_{CS} \cup \bigcup_{L \in \mathcal{L}_{CS}} (L \cup \{\epsilon\}) \ .$$

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ **Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)**

$\mathcal{L}_{CS}$ **Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)**

$\mathcal{L}_{CF}$ **Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)**

$\mathcal{R}$ **Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)**

Vzťah automatov a jazykov z Chomského hierarchie

Automat	Determin. verzia		Nedetermin. verzia		Trieda jazykov
konečný	$\mathcal{L}(DFA)$	$=$	$\mathcal{L}(NFA)$	$=$	$\mathcal{R}$
zásobníkový	$\mathcal{L}(DPDA)$	$\subsetneq$	$\mathcal{L}(NPDA)$	$=$	$\mathcal{L}_{CF}$
lin. ohraničený	$\mathcal{L}(DLBA)$	$\subseteq^{\star}$	$\mathcal{L}(NLBA)$	$=$	$\mathcal{L}_{CS}$
Turingov stroj	$\mathcal{L}(DTM)$	$=$	$\mathcal{L}(NTM)$	$=$	$\mathcal{L}_{RE}$

* 1. LBA problém; platí jedna z dvoch možností: buď $\subsetneq$ alebo $=$.

Uzáverové vlastnosti tried jazykov z Chomského hierarchie

Trieda jazykov	$\cup$	$\cdot$	$\cap$	$*$	$+$	c	R	h	h bez ϵ
$\mathcal{R}$	A	A	A	A	A	A	A	A	A
$\mathcal{L}_{CF}$	A	A	N	A	A	N	A	A	A
$\mathcal{L}_{CS}$	A	A	A		A	A	A	N	A
$\mathcal{L}_{RE}$	A	A	A	A	A	N	A	A	A

S T U
F I I T

Ďakujem za pozornosť.

chuda@fiit.stuba.sk

Príklad 1a. $\mathbf{c^{4n}}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ c^{4n} \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

ε, $cccc$, $cccccccc$, $cccccccccccc$, $\ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S\}$
- $T = \{c\}$
- $P = \{$

$$S \to ccccS$$
$$S \to \varepsilon$$
$$\}$$

Odvodenie slov:

- ε

$S \Rightarrow \varepsilon$

- $cccc$

$S \Rightarrow ccccS \Rightarrow cccc$

- $cccccccc$

$S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow cccccccc$

- $cccccccccccc$

$S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow ccccccccccccS \Rightarrow cccccccccccc$

Príklad 1b. $\mathbf{c^{4n}} \mid \mathbf{n} \in \mathbb{N}^{+}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ c^{4n} \mid n \in \mathbb{N}^{+}\}$

Príklady slov z jazyka:

$cccc$, $cccccccc$, $cccccccccccc$, ...

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S\}$
- $T = \{c\}$
- $P = \{$

  $S \to cccc \mid ccccS$

  $\}$

Odvodenie slov:

- $cccc$

  $S \Rightarrow cccc$

- $cccccccc$

  $S \Rightarrow ccccS \Rightarrow cccccccc$

- $cccccccccccc$

  $S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow cccccccccccc$

Príklad 2. $a^i b^j$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^i b^j \mid i, j \in \mathbb{N}\}$

Príklady slov z jazyka:

ε, a, b, ab, aab, abb, $aaab$, $\ldots$

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{a, b\}$
- $P = \{$

$$A \to aA \mid B$$
$$B \to bB \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- ε

$A \Rightarrow B \Rightarrow \varepsilon$

- ab

$A \Rightarrow aA \Rightarrow aB \Rightarrow abB \Rightarrow ab$

- bb

$A \Rightarrow B \Rightarrow bB \Rightarrow bbB \Rightarrow bb$

- $aabbb$

$A \Rightarrow aA \Rightarrow aaA \Rightarrow aaB \Rightarrow aabB \Rightarrow aabbB \Rightarrow aabbbB \Rightarrow aabbb$

Riešenie 2:

- ◇ $G = (N, T, P, S)$
- ◇ $N = \{S\}$
- ◇ $T = \{a, b\}$
- ◇ $P = \{$

$$S \rightarrow \varepsilon$$
$$S \rightarrow aS$$
$$S \rightarrow Sb$$
$$\}$$

Odvodenie slov:

▶ ε

$S \Rightarrow \varepsilon$

▶ ab

$S \Rightarrow aS \Rightarrow aSb \Rightarrow ab$

▶ bb

$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow bb$

▶ $aabbb$

$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaSb \Rightarrow aaSbb \Rightarrow aaSbbb \Rightarrow aabbb$

Príklad 3. Prvé dve písmená rovnaké

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\, w \mid \text{ prvé dve písmená rovnaké } \mid w \in \{a,b\}^*\}$

Príklady slov z jazyka:

$aa,\ bb,\ aab,\ aaab,\ bba,\ bbba,\ bbaab,\ \ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S, X\}$
- $T = \{a, b\}$
- $P = \{$

$$S \to aaX \mid bbX$$
$$X \to aX \mid bX \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- $aaba$

$S \Rightarrow aaX \Rightarrow aabX \Rightarrow aabaX \Rightarrow aaba$

- $bbbba$

$S \Rightarrow bbX \Rightarrow bbbbX \Rightarrow bbbba$

Príklad 4. Práve jedna 0 na posledných dvoch miestach

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ w \mid$ práve jedna 0 na posledných dvoch miestach $\mid w \in \{0,1\}^*\}$

Príklady slov z jazyka:

01, 10, 001, 101, 010, 110, 0001, 1101, 0010, 1110, ...

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{0, 1\}$
- $P = \{$

$$A \to B01 \mid B10$$
$$B \to 0B \mid 1B \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- 0110

$A \Rightarrow B10 \Rightarrow B0110 \Rightarrow 0110$

- 1101

$A \Rightarrow B01 \Rightarrow 1B01 \Rightarrow 11B01 \Rightarrow 1101$

Riešenie 2:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{0, 1\}$
- $P = \{$

$$A \to B01 \mid B10$$
$$B \to B1 \mid B0 \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- ▶ 0110

  $A \Rightarrow B10 \Rightarrow B0110 \Rightarrow 0110$

- ▶ 1101

  $A \Rightarrow B01 \Rightarrow B101 \Rightarrow B1101 \Rightarrow 1101$

Riešenie 3:

- ◇ $G = (N, T, P, A)$
- ◇ $N = \{A\}$
- ◇ $T = \{0, 1\}$
- ◇ $P = \{$

  $A \rightarrow 1A \mid 0A \mid 10 \mid 01$

  $\}$

Odvodenie slov:

- ▶ 0110

  $A \Rightarrow 0A \Rightarrow 01A \Rightarrow 0110$

- ▶ 1101

  $A \Rightarrow 1A \Rightarrow 11A \Rightarrow 1101$

Príklad 1a. $\mathbf{c^{4n}}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ c^{4n} \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

ε, $cccc$, $cccccccc$, $cccccccccccc$, $\ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S\}$
- $T = \{c\}$
- $P = \{$

  $S \to ccccS$

  $S \to \varepsilon$

  $\}$

Odvodenie slov:

- ε

  $S \Rightarrow \varepsilon$

- $cccc$

  $S \Rightarrow ccccS \Rightarrow cccc$

- $cccccccc$

  $S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow cccccccc$

- $cccccccccccc$

  $S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow ccccccccccccS \Rightarrow cccccccccccc$

Príklad 1b. $\mathbf{c^{4n}} \mid \mathbf{n} \in \mathbb{N}^+$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ c^{4n} \mid n \in \mathbb{N}^+\}$

Príklady slov z jazyka:

$cccc,\ cccccccc,\ cccccccccccc,\ \ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S\}$
- $T = \{c\}$
- $P = \{$

  $S \rightarrow cccc \mid ccccS$

  $\}$

Odvodenie slov:

- $cccc$

  $S \Rightarrow cccc$

- $cccccccc$

  $S \Rightarrow ccccS \Rightarrow cccccccc$

- $cccccccccccc$

  $S \Rightarrow ccccS \Rightarrow ccccccccS \Rightarrow cccccccccccc$

Príklad 2. $\mathbf{a^i b^j}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^i b^j \mid i, j \in \mathbb{N}\}$

Príklady slov z jazyka:

$\varepsilon,\ a,\ b,\ ab,\ aab,\ abb,\ aaab,\ \ldots$

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{a, b\}$
- $P = \{$

$$A \to aA \mid B$$
$$B \to bB \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- ε

$A \Rightarrow B \Rightarrow \varepsilon$

- ab

$A \Rightarrow aA \Rightarrow aB \Rightarrow abB \Rightarrow ab$

- bb

$A \Rightarrow B \Rightarrow bB \Rightarrow bbB \Rightarrow bb$

- $aabbb$

$A \Rightarrow aA \Rightarrow aaA \Rightarrow aaB \Rightarrow aabB \Rightarrow aabbB \Rightarrow aabbbB \Rightarrow aabbb$

Riešenie 2:

- ◇ $G = (N, T, P, S)$
- ◇ $N = \{S\}$
- ◇ $T = \{a, b\}$
- ◇ $P = \{$

  $S \rightarrow \varepsilon$

  $S \rightarrow aS$

  $S \rightarrow Sb$

  $\}$

Odvodenie slov:

- ▶ ε

  $S \Rightarrow \varepsilon$

- ▶ ab

  $S \Rightarrow aS \Rightarrow aSb \Rightarrow ab$

- ▶ bb

  $S \Rightarrow Sb \Rightarrow Sbb \Rightarrow bb$

- ▶ $aabbb$

  $S \Rightarrow aS \Rightarrow aaS \Rightarrow aaSb \Rightarrow aaSbb \Rightarrow aaSbbb \Rightarrow aabbb$

Príklad 3. **Prvé dve písmená rovnaké**

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ w \mid \text{ prvé dve písmená rovnaké } \mid w \in \{a, b\}^*\}$

Príklady slov z jazyka:

$aa,\ bb,\ aab,\ aaab,\ bba,\ bbba,\ bbaab,\ \dots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S, X\}$
- $T = \{a, b\}$
- $P = \{$

$$S \to aaX \mid bbX$$
$$X \to aX \mid bX \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- $aaba$

$$S \Rightarrow aaX \Rightarrow aabX \Rightarrow aabaX \Rightarrow aaba$$

- $bbbba$

$$S \Rightarrow bbX \Rightarrow bbbbX \Rightarrow bbbba$$

Príklad 4. Práve jedna 0 na posledných dvoch miestach

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ w \mid$ práve jedna 0 na posledných dvoch miestach $\mid w \in \{0,1\}^*\}$

Príklady slov z jazyka:

01, 10, 001, 101, 010, 110, 0001, 1101, 0010, 1110, ...

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{0, 1\}$
- $P = \{$

$$A \to B01 \mid B10$$
$$B \to 0B \mid 1B \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- 0110

$A \Rightarrow B10 \Rightarrow B0110 \Rightarrow 0110$

- 1101

$A \Rightarrow B01 \Rightarrow 1B01 \Rightarrow 11B01 \Rightarrow 1101$

Riešenie 2:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{0, 1\}$
- $P = \{$

$$A \to B01 \mid B10$$
$$B \to B1 \mid B0 \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- ▶ 0110

  $A \Rightarrow B10 \Rightarrow B0110 \Rightarrow 0110$

- ▶ 1101

  $A \Rightarrow B01 \Rightarrow B101 \Rightarrow B1101 \Rightarrow 1101$

Riešenie 3:

- ◇ $G = (N, T, P, A)$
- ◇ $N = \{A\}$
- ◇ $T = \{0, 1\}$
- ◇ $P = \{$

  $A \rightarrow 1A \mid 0A \mid 10 \mid 01$

  $\}$

Odvodenie slov:

- ▶ 0110

  $A \Rightarrow 0A \Rightarrow 01A \Rightarrow 0110$

- ▶ 1101

  $A \Rightarrow 1A \Rightarrow 11A \Rightarrow 1101$

Príklad 5. Obsahuje podslovo *ab*

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{ \ w = uabv \mid u, v \in \{a, b\}^* \}$

Príklady slov z jazyka:

ab, aab, bab, aba, $bbaba$, $aaabbb$, $\ldots$

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{a, b\}$
- $P = \{$

$$A \to BabB$$
$$B \to aB \mid bB \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- $bbaba$

$$A \Rightarrow BabB \Rightarrow bBabB \Rightarrow bbBabB \Rightarrow bbabB \Rightarrow bbabaB \Rightarrow bbaba$$

- $aaabbb$

$$A \Rightarrow BabB \Rightarrow BabbB \Rightarrow BabbbB \Rightarrow Babbb \Rightarrow aBabbb \Rightarrow aaBabbb \Rightarrow aaabbb$$

Riešenie 2:

- $G = (N, T, P, S)$
- $N = \{S, T\}$
- $T = \{a, b\}$
- $P = \{$

$$S \to aS \mid bS \mid abT$$
$$T \to aT \mid bT \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- $bbaba$

  $S \Rightarrow bS \Rightarrow bbS \Rightarrow bbabT \Rightarrow bbabaT \Rightarrow bbaba$

- $aaabbb$

  $S \Rightarrow aS \Rightarrow aaS \Rightarrow aaabT \Rightarrow aaabbT \Rightarrow aaabbbT \Rightarrow aaabbb$

Príklad 6. **Záporné čísla deliteľné 5**

Zadanie:

- Napíš gramatiku, ktorá generuje jazyk $L = \{\ w \mid\ \text{záporné čísla deliteľné 5}\}$

Príklady slov z jazyka:

$-5,\ -10,\ -15,\ -20,\ -25,\ -30,\ -125,\ \dots$

Riešenie 1:

- $G = (N, T, P, A)$
- $N = \{A, B, C\}$
- $T = \{-, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $P = \{$

$$A \to -B0 \mid -B5 \mid -5$$
$$B \to 1C \mid 2C \mid 3C \mid \dots \mid 9C$$
$$C \to \varepsilon \mid 0C \mid 1C \mid 2C \mid 3C \mid \dots \mid 9C$$
$$\}$$

Odvodenie slov:

- -35

$$A \Rightarrow -B5 \Rightarrow -3C5 \Rightarrow -35$$

- -2940

$$A \Rightarrow -B0 \Rightarrow -2C0 \Rightarrow -29C0 \Rightarrow -294C0 \Rightarrow -2940$$

Riešenie 2:

- $G = (N, T, P, S)$
- $N = \{S, X\}$
- $T = \{-, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $P = \{$

$$S \to -1X \mid -2X \mid -3X \mid \dots \mid -9X \mid -5$$
$$X \to 0X \mid 1X \mid 2X \mid \dots \mid 9X \mid 0 \mid 5$$
$$\}$$

Odvodenie slov:

- -35

  $S \Rightarrow -3X \Rightarrow -35$

- -2940

  $S \Rightarrow -2X \Rightarrow -29X \Rightarrow -294X \Rightarrow -2940$

Príklad 7. $\mathbf{a^n b^n}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^n b^n \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

ε, ab, $aabb$, $aaabbb$, $aaaabbbb$, $\ldots$

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A\}$
- $T = \{a, b\}$
- $P = \{$

  $A \rightarrow \varepsilon \mid aAb$

  $\}$

Odvodenie slov:

- ab

  $A \Rightarrow aAb \Rightarrow ab$

- $aabb$

  $A \Rightarrow aAb \Rightarrow aaAbb \Rightarrow aabb$

Príklad 8a. $\mathbf{a^{n+1}b^n}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^{n+1}b^n \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

$a,\ aab,\ aaabb,\ aaaabbb,\ \ldots$

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A\}$
- $T = \{a, b\}$
- $P = \{$

  $A \to aAb \mid a$

  $\}$

Odvodenie slov:

- aab

  $A \Rightarrow aAb \Rightarrow aab$

- $aaabb$

  $A \Rightarrow aAb \Rightarrow aaAbb \Rightarrow aaabb$

<u>Príklad 8b.</u> $\mathbf{a^{3n+1}b^n}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^{3n+1}b^n \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

$a,\ aaaab,\ aaaaaaabb,\ \ldots$

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A\}$
- $T = \{a, b\}$
- $P = \{$

  $A \to aaaAb \mid a$

  $\}$

Odvodenie slov:

- $aaaab$

  $A \Rightarrow aaaAb \Rightarrow aaaab$

- $aaaaaaabb$

  $A \Rightarrow aaaAb \Rightarrow aaaaaaAbb \Rightarrow aaaaaaabb$

Príklad 8c. $\mathbf{a^{3n+1}b^{n+3}}$

Zadanie:

■ Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^{3n+1}b^{n+3} \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

$abbb$, $aaaabbbb$, $aaaaaaabbbbb$, $\ldots$

Riešenie:

◇ $G = (N, T, P, A)$

◇ $N = \{A\}$

◇ $T = \{a, b\}$

◇ $P = \{$

$A \to aaaAb \mid abbb$

$\}$

Odvodenie slov:

▶ $aaaabbbb$

$A \Rightarrow aaaAb \Rightarrow aaaabbbb$

▶ $aaaaaaabbbbb$

$A \Rightarrow aaaAb \Rightarrow aaaaaaAbb \Rightarrow aaaaaaabbbbb$

Príklad 8d. $\mathbf{a^{3n+2}cb^{n+3}}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^{3n+2}cb^{n+3} \mid n \in \mathbb{N}\}$

Príklady slov z jazyka:

$aacbbb$, $aaaaacbbbb$, $aaaaaaaacbbbbb$, $\ldots$

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A\}$
- $T = \{a, b, c\}$
- $P = \{$

  $A \to aacbbb \mid aaaAb$

  $\}$

Odvodenie slov:

- $aaaaacbbbb$

  $A \Rightarrow aaaAb \Rightarrow aaaaacbbbb$

- $aaaaaaaacbbbbb$

  $A \Rightarrow aaaAb \Rightarrow aaaaaaAbb \Rightarrow aaaaaaaacbbbbb$

Príklad 9a. $\mathbf{a^x b^y a^y b^x}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^x b^y a^y b^x \mid x, y \in \mathbb{N}\}$

Príklady slov z jazyka:

$\varepsilon,\ ba,\ ab,\ abab,\ abbaab,\ aabbaabb,\ \dots$

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A, B\}$
- $T = \{a, b\}$
- $P = \{$

$$A \to aAb \mid B$$
$$B \to bBa \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- *abbaab*

$A \Rightarrow aAb \Rightarrow aBb \Rightarrow abBab \Rightarrow abab$

- *aabbaabb*

$A \Rightarrow aAb \Rightarrow aaAbb \Rightarrow aaBbb \Rightarrow aabBabb \Rightarrow aabbBaabb \Rightarrow aabbaabb$

Príklad 9b. $\mathbf{a^{x+3}b^{2y}a^{y}b^{x}}$

Zadanie:

■ Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^{x+3}b^{2y}a^{y}b^{x} \mid x, y \in \mathbb{N}\}$

Príklady slov z jazyka:

$\varepsilon,\ aaa,\ aaaab,\ aaabba,\ aaaabbab,\ \ldots$

Riešenie 1:

◇ $G = (N, T, P, A)$

◇ $N = \{A, B\}$

◇ $T = \{a, b\}$

◇ $P = \{$

$$A \rightarrow aAb \mid aaaB$$
$$B \rightarrow bbBa \mid \varepsilon$$
$$\}$$

Odvodenie slov:

▶ $aaabba$

$A \Rightarrow aaaB \Rightarrow aaabbBa \Rightarrow aaabba$

▶ $aaaabbab$

$A \Rightarrow aAb \Rightarrow aaaaBb \Rightarrow aaaabbBab \Rightarrow aaaabbab$

Riešenie 2:

◇ $G = (N, T, P, S)$

◇ $N = \{S, T, U\}$

◇ $T = \{a, b\}$

◇ $P = \{$

$$S \rightarrow aaaT$$
$$T \rightarrow aTb \mid U \mid \varepsilon$$
$$U \rightarrow bbUa \mid \varepsilon$$
$$\}$$

Odvodenie slov:

- $aaabba$

  $S \Rightarrow aaaT \Rightarrow aaaU \Rightarrow aaabbUa \Rightarrow aaabba$

- $aaaabbab$

  $S \Rightarrow aaaT \Rightarrow aaaaTb \Rightarrow aaaaUb \Rightarrow aaaabbUab \Rightarrow aaaabbab$

Príklad 10. $0^x 1^{3x} 0^y 1^{y+5}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ 0^x 1^{3x} 0^y 1^{y+5} \mid x, y \in \mathbb{N}\}$

Príklady slov z jazyka:

11111, 01110111111, 00111111001111111, ...

Riešenie:

- $G = (N, T, P, A)$
- $N = \{A, B, C\}$
- $T = \{0, 1\}$
- $P = \{$

$$A \to BC$$
$$B \to 0B111 \mid \varepsilon$$
$$C \to 0C1 \mid 11111$$
$$\}$$

Odvodenie slov:

- 01110111111

$A \Rightarrow BC \Rightarrow 0B111C \Rightarrow 0111C \Rightarrow 01110C1 \Rightarrow 01110111111$

- 00111111001111111

$A \Rightarrow BC \Rightarrow 0B111C \Rightarrow 00B111111C \Rightarrow 00111111C \Rightarrow 001111110C1 \Rightarrow 0011111100C11$
$\Rightarrow 00111111001111111$

<u>Príklad 11.</u> $\mathbf{ww^R}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ ww^R\}$

Príklady slov z jazyka:

ε, aa, bb, $abba$, $baab$, $abaaba$, $bbabbabb$, ...

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S\}$
- $T = \{a, b\}$
- $P = \{$

  $S \to aSa \mid bSb \mid \varepsilon$

  $\}$

Odvodenie slov:

- $abba$

  $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba$

- $bbabbabb$

  $S \Rightarrow bSb \Rightarrow bbSbb \Rightarrow bbaSabb \Rightarrow bbabSbabb \Rightarrow bbabbabb$

Príklad 1. $\#_0\mathbf{w} = \#_1\mathbf{w}$

Zadanie:

■ Napíšte gramatiku, ktorá generuje jazyk $L = \{\ w \mid \#_0 w = \#_1 w,\ w \in \{0,1\}^*\}$

Príklady slov z jazyka:

$\varepsilon,\ 0,\ 1,\ 01,\ 10,\ 0101,\ 1100,\ 001011,\ \ldots$

Riešenie 1:

◇ $G = (N, T, P, S)$

◇ $N = \{S, A, B\}$

◇ $T = \{0, 1\}$

◇ $P = \{$

$$S \to 0A \mid 1B \mid \varepsilon$$
$$A \to 1S \mid 0AA$$
$$B \to 0S \mid 1BB$$
$$\}$$

Odvodenie slov:

▶ 0011

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 001SA \Rightarrow 001A \Rightarrow 0011S \Rightarrow 0011$$

▶ 1001

$$S \Rightarrow 1B \Rightarrow 10S \Rightarrow 100A \Rightarrow 1001S \Rightarrow 1001$$

▶ 001011

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 001SA \Rightarrow 0010AA \Rightarrow 00101SA \Rightarrow 00101A \Rightarrow 001011S \Rightarrow 001011$$

Riešenie 2:

◇ $G = (N, T, P, S)$

◇ $N = \{S\}$

◇ $T = \{0, 1\}$

◇ $P = \{$

$$S \to \varepsilon \mid 01 \mid 10 \mid 1S0 \mid 0S1$$
$$S \to S01 \mid S10 \mid 10S \mid 01S$$
$$\}$$

Odvodenie slov:

- 0011

  $S \Rightarrow 0A \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0011$

- 1001

  $S \Rightarrow 10S \Rightarrow 1001S \Rightarrow 1001$

- 001011

  $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 001011$

Príklad 2. wcw

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ wcw \mid w \in \{a, b\}^*\}$

Príklady slov z jazyka:

c, aca, $abcab$, $baacbaa$, $aabcaab$, $ababcabab$, ...

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S, A, B, C\}$
- $T = \{a, b, c\}$
- $P = \{$

$$
\begin{aligned}
&S \rightarrow aAS \mid bBS \mid C \\
&Aa \rightarrow aA \\
&Bb \rightarrow bB \\
&Ba \rightarrow aB \\
&Ab \rightarrow bA \\
&BC \rightarrow Cb \\
&AC \rightarrow Ca \\
&C \rightarrow c \\
&\}
\end{aligned}
$$

Vysvetlenie:

- Generujem vetnú formu tak, že obsahuje všetko 2x a C je nakonci a potom tú druhú polovicu prenesiem za C tak, aby sa nepomiešala.

Odvodenie slov:

- aca

$S \Rightarrow aAS \Rightarrow aAC \Rightarrow aCa \Rightarrow aca$

- $aabcaab$

$S \Rightarrow aAS \Rightarrow aAaAS \Rightarrow aaAAS \Rightarrow aaAAbBS \Rightarrow aaAbABS \Rightarrow aabAABS \Rightarrow aabAABC$
$\Rightarrow aabAACb \Rightarrow aabACab \Rightarrow aabCaab \Rightarrow aabcaab$

Príklad 3. $\mathbf{a^n b^n c^n}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^n b^n c^n \mid n \in \mathbb{N}^+\}$

Príklady slov z jazyka:

$abc,\ aabbcc, aaabbbccc,\ aaaabbbbcccc,\ \dots$

Riešenie:

- $G = \{N, T, P, S\}$
- $N = \{S, B, C\}$
- $T = \{a, b, c\}$
- $P = \{$

$$
\begin{aligned}
&S \to aSBC \mid abC \\
&CB \to BC \\
&bC \to bc \\
&bB \to bb \\
&cC \to cc \\
&\}
\end{aligned}
$$

Vysvetlenie:

▷ Nagenerujem vetnú formu so správnymi počtami abc - ABC a potom ich prepisujem na konečné slovo, aby boli v správnom poradí.

Odvodenie slov:

▶ *abc*

$S \Rightarrow abC \Rightarrow abc$

▶ *aabbcc*

$S \Rightarrow aSBC \Rightarrow aabCBC \Rightarrow aabBCC \Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcc$

Príklad 4. $\mathbf{a^n b^n c^n d^n}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^n b^n c^n d^n \mid n \in \mathbb{N}^+\}$

Príklady slov z jazyka:

$abcd,\ aabbccdd, aaabbbcccddd,\ aaaabbbbccccdddd,\ \ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S, B, C, D\}$
- $T = \{a, b, c, d\}$
- $P = \{$

$$
\begin{aligned}
&S \rightarrow aSBCD \mid abCD \\
&DB \rightarrow BD \\
&DC \rightarrow CD \\
&CB \rightarrow BC \\
&bC \rightarrow bc \\
&bB \rightarrow bb \\
&cC \rightarrow cc \\
&cD \rightarrow cd \\
&dD \rightarrow dd \\
&\}
\end{aligned}
$$

Odvodenie slov:

- $abcd$

  $S \Rightarrow abCD \Rightarrow abcD \Rightarrow abcd$

- $aabbccdd$

  $S \Rightarrow aSBCD \Rightarrow aabCDBCD \Rightarrow aabCBDCD \Rightarrow aabBCDCD \Rightarrow aabBCCDD \Rightarrow aabbCCDD \Rightarrow aabbcCDD \Rightarrow aabbccDD \Rightarrow aabbccdD \Rightarrow aabbccdd$

Príklad 5. $\mathbf{a^i b^j c^i d^j}$

Zadanie:

- Napíšte gramatiku, ktorá generuje jazyk $L = \{\ a^i b^j c^i d^j \mid i, j \in \mathbb{N}^+\}$

Príklady slov z jazyka:

$abcd,\ aabccd, abbbcddd,\ aabbccdd,\ \ldots$

Riešenie:

- $G = (N, T, P, S)$
- $N = \{S, L, P, B, C\}$
- $T = \{a, b, c, d\}$
- $P = \{$

$$
\begin{aligned}
&S \to LP\\
&L \to aLC \mid aC\\
&P \to BPd \mid Bd\\
&CB \to BC\\
&aB \to ab\\
&DC \to CD\\
&bB \to bb\\
&bC \to bc\\
&cC \to cc\\
&\}
\end{aligned}
$$

Vysvetlenie:

▷ Rozdeliť na dve časti, nechať generovať časti slova umocnené na i a časti slova umocnené na j a potom preusporiadať.

Odvodenie slov:

▶ $abcd$

$S \Rightarrow LP \Rightarrow aCP \Rightarrow aCBd \Rightarrow aBCd \Rightarrow abCd \Rightarrow abcd$

▶ $aabccd$

$S \Rightarrow LP \Rightarrow aLCP \Rightarrow aaCCP \Rightarrow aaCCBd \Rightarrow aaCBCd \Rightarrow aaBCCd \Rightarrow aabCCd \Rightarrow aabcCd \Rightarrow aabccd$

▶ $aabccd$

$S \Rightarrow LP \Rightarrow aLCP \Rightarrow aaCCP \Rightarrow aaCCBPd \Rightarrow aaCCBBdd \Rightarrow aaCBCBdd \Rightarrow aaBCCBdd \Rightarrow aaBCBCdd \Rightarrow aaBBCCdd \Rightarrow aabBCCdd \Rightarrow aabbCCdd \Rightarrow aabbcCdd \Rightarrow aabbccdd$

Príklad 6a. Viacznačnosť gramatiky G [1]

Zadanie:

■ Dokážte, že gramatika G je viacznačná.

$G = (N, T, P, S)$

$N = \{S\}$

$T = \{a\}$

$P = \{$

$S \rightarrow aaSaa \mid a$

$S \rightarrow aS$

$\}$

Riešenie:

► $aaaaaa$

$S \Rightarrow aS \Rightarrow aaaSaa \Rightarrow aaaaaa$

► $aaaaaa$

$S \Rightarrow aaSaa \Rightarrow aaaSaa \Rightarrow aaaaaa$

Vysvetlenie:

▷ Našli sme dva rôzne stromy odvodenia pre slovo $aaaaaa$ a preto je gramatika G viacznačná.

Príklad 6b. Viacznačnosť gramatiky G [2]

Zadanie:

- Dokážte, že gramatika G je viacznačná.

  $G = (N, T, P, S)$

  $N = \{S\}$

  $T = \{if, (1), else, a\}$

  $P = \{$

  $S \rightarrow if\ (1)\ S \mid a$
  $S \rightarrow if\ (1)\ S\ else\ S$
  $S \rightarrow a$
  $\}$

Riešenie:

- $if\ (1)\ if\ (1)\ a\ else\ a$

  $S \Rightarrow if\ (1)\ S\ else\ S \Rightarrow if(1)\ if\ (1)\ S\ else\ S \Rightarrow if\ (1)\ if\ (1)\ a\ else\ S \Rightarrow if\ (1)\ if\ (1)\ a\ else\ a$

- $if\ (1)\ if\ (1)\ a\ else\ a$

  $S \Rightarrow if\ (1)\ S \Rightarrow if\ (1)\ if\ (1)\ S\ else\ S \Rightarrow if\ (1)\ if\ (1)\ a\ else\ S \Rightarrow if\ (1)\ if\ (1)\ a\ else\ a$

Vysvetlenie:

- Našli sme dva rôzne stromy odvodenia pre slovo *if (1) if (1) a else a* a preto je gramatika G viacznačná.

Príklad 7a. Homomorfizmus - viacznačnosť gramatiky G [1]

Zadanie:

- Nájdite homomorfizmus h, tak aby gramatika $G' = \{N, h(T), P', S\}$ bola viacznačná. Množina P' vznikne z P aplikovaním h na každý terminál.

  $G = (N, T, P, S)$

  $N = \{S, A, B\}$

  $T = \{a, b, c, d\}$

  $P = \{$

  $S \rightarrow abA \mid a$

  $A \rightarrow cS \mid dbB \mid b$

  $B \rightarrow cBd \mid bA \mid cA$

  $\}$

Riešenie:

- $G' = (N, h(T), P', S)$
- $N = \{S, A, B\}$
- $h(T) = \{4\}$
  resp.
  $h(a) = 4$
  $h(b) = 4$
  $h(c) = 4$
  $h(d) = 4$
- $P' = \{$

  $S \rightarrow 44A$

  $A \rightarrow 4S \mid 44B \mid 4$

  $B \rightarrow 4B4 \mid 4A$

  $\}$

- 444444

  $S \Rightarrow 44A \Rightarrow 444S \Rightarrow 44444A \Rightarrow 444444$

- 444444

  $S \Rightarrow 44A \Rightarrow 4444B \Rightarrow 44444A \Rightarrow 444444$

Vysvetlenie:

- Našli sme dva rôzne stromy odvodenia pre slovo 444444 a preto je gramatika G' viacznačná.

Príklad 7b. Homomorfizmus - viacznačnosť gramatiky G [2]

Zadanie:

- Nájdite homomorfizmus h, tak aby gramatika $G' = \{N, h(T), P', A\}$ bola viacznačná. Množina P' vznikne z P aplikovaním h na každý terminál.

  $G = (N, T, A, S)$

  $N = \{A, B\}$

  $T = \{a, b, c, d\}$

  $P = \{$

  $A \rightarrow bcBdA \mid c \mid bA$
  $B \rightarrow cBd \mid b$
  $\}$

Riešenie:

- $G' = (N, h(T), P', A)$
- $N = \{A, B\}$
- $h(T) = \{4\}$
  $resp.$
  $h(a) = 4$
  $h(b) = 4$
  $h(c) = 4$
  $h(d) = 4$
- $P' = \{$

  $A \rightarrow 44B4A \mid 4 \mid 4A$
  $B \rightarrow 4B4 \mid 4$
  $\}$

- 4444444

  $S \Rightarrow 44B4A \Rightarrow 4444A \Rightarrow 44444A \Rightarrow 444444A \Rightarrow 4444444$

- 4444444

  $S \Rightarrow 44B4A \Rightarrow 444B44A \Rightarrow 444444A \Rightarrow 4444444$

Vysvetlenie:

- Našli sme dva rôzne stromy odvodenia pre slovo 4444444 a preto je gramatika G' viacznačná.

Teoretické základy informatiky

Konečné automaty

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Chomského hierarchia jazykov

Klasifikácia:

- regulárne,
- bezkontextové,
- kontextové,
- rekurzívne vyčísliteľné.

Regulárne jazyky – konečná gramatika.

Bezkontextové jazyky – nekonečná gramatika.

Jazyky a ich reprezentácie

Definícia jazyka podľa webster dictionary:

„Jazyk ako súbor slov a metód skladania slov, ktorý používa a chápe dostatočne veľká ľudská spoločnosť."

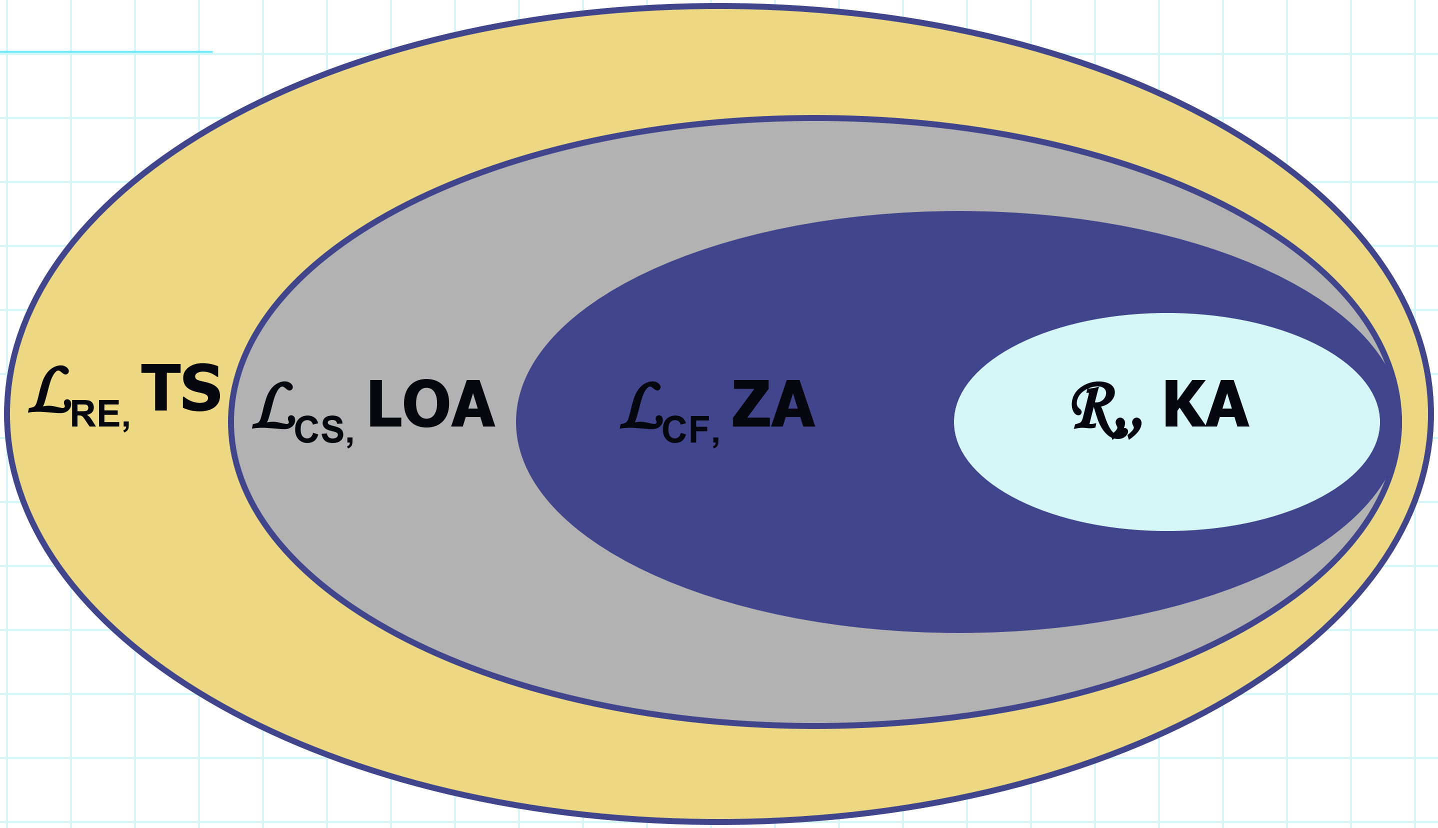
(nepresná definícia pre vybudovanie matematickej teórie jazykov)

Formálny jazyk – abstraktne definovaný ako matematický systém.

Reprezentácia jazyka (špecfikácia slov jazyka) :

- Zostrojiť algoritmus, ktorý určí, či dané slovo patrí do jazyka. Rozpoznanie jazyka. **Automat.** **Výpočet.**
- Systematicky generovať slová z jazyka. Generovanie jazyka. **Gramatika.**
- Matematickým opisom jazyka. **Množina.**

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ **Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)**

$\mathcal{L}_{CS}$ **Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)**

$\mathcal{L}_{CF}$ **Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)**

$\mathcal{R}$ **Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)**

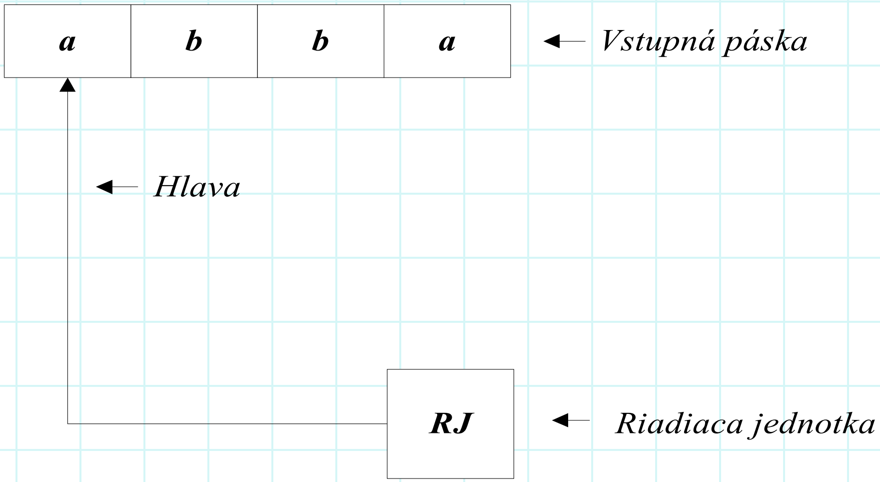
Vzťah automatov a jazykov z Chomského hierarchie

Automat	Determin. verzia		Nedetermin. verzia		Trieda jazykov
konečný	$\mathcal{L}(DFA)$	$=$	$\mathcal{L}(NFA)$	$=$	$\mathcal{R}$
zásobníkový	$\mathcal{L}(DPDA)$	$\subsetneq$	$\mathcal{L}(NPDA)$	$=$	$\mathcal{L}_{CF}$
lin. ohraničený	$\mathcal{L}(DLBA)$	$\subseteq^\star$	$\mathcal{L}(NLBA)$	$=$	$\mathcal{L}_{CS}$
Turingov stroj	$\mathcal{L}(DTM)$	$=$	$\mathcal{L}(NTM)$	$=$	$\mathcal{L}_{RE}$

* 1. LBA problém; platí jedna z dvoch možností: buď $\subsetneq$ alebo $=$.

Reprezentácia jazyka výpočtom

Výpočtový model reprezentovaný konečným automatom:



- riadiaca jednotka,
- čítacia hlava, čítajúca symboly, každý symbol raz,
- vstupná páska.

Deterministický konečný automat DKA -definícia

Definícia 3.1.1 Deterministický konečný automat

Deterministický konečný automat *(DFA - deterministic finite automaton) je pätica* $A = (K, \Sigma, \delta, q_0, F)$, *kde:*

K je konečná množina stavov,

Σ je vstupná abeceda,

δ je prechodová funkcia, pričom platí

$$\delta : K \times \Sigma \to K$$

$q_0 \in K$ je počiatočný stav,

$F \subseteq K$ je množina koncových (akceptačných) stavov.

Konfigurácia deterministického konečného automatu *je dvojica*

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

kde:

q je momentálny stav,

w je zvyšok vstupu.

Deterministický konečný automat – krok výpočtu, jazyk rozpoznávaný DKA

Krok výpočtu deterministického konečného automatu *je relácia* $\vdash_A$ *na konfigurácii definovaná nasledovne:*

$$(q, au) \vdash_A (p, u), \qquad ak \qquad \delta(q, a) = p,$$

pričom $p,\ q \in K,\ \ a \in \Sigma \cup \{\epsilon\},\ \ u \in \Sigma^*.$

Jazyk rozpoznávaný deterministickým konečným automatom *(koncovým stavom) je množina*

$$L(A) = \{\ w \in \Sigma^* \mid (q_0, w) \vdash_A^* (q, \epsilon),\ \ q \in F\}.$$

Konečný automat, reprezentácia

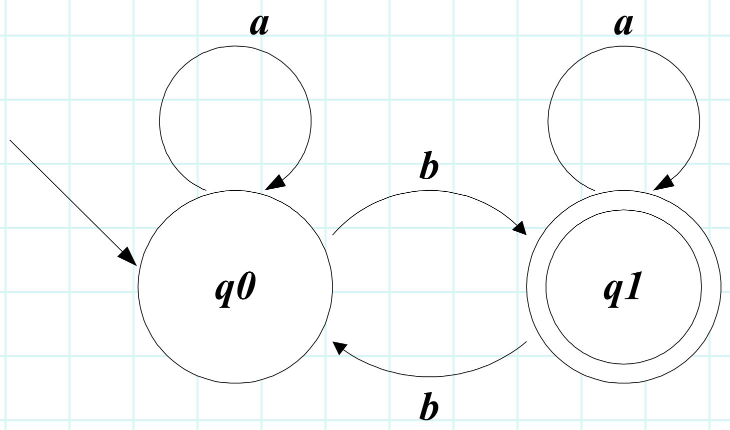
Rozlišujeme dva druhy konečných automatov:

- deterministické,
- nedeterministické.

Konečný automat môžeme reprezentovať:

- stavovým diagramom,
- formálnym zápisom,
- maticou prechodových funkcií.

DKA - Stavový diagram



$$A(L_1) = \{\ w \in \{a,b\}^* \mid \#_b w = 2k+1,\ k \in \mathbb{N}\ \}$$

DKA - Formálny zápis

Konečný automat A_1 *môžeme opísať nasledujúcim formálnym zápisom.*
Nech $A_1 = (K, \Sigma, \delta, q_0, F)$, *kde:*
$K = \{q_0, q_1\}$,
$\Sigma = \{a, b\}$,
$\delta(q_0, a) = q_0$
$\delta(q_0, b) = q_1$
$\delta(q_1, a) = q_1$
$\delta(q_1, b) = q_0$
$F = \{q_1\}$.

$$A(L_1) = \{\ w \in \{a, b\}^* \mid \#_b w = 2k + 1,\ k \in \mathbb{N}\ \}$$

DKA - Matica prechodových funkcií

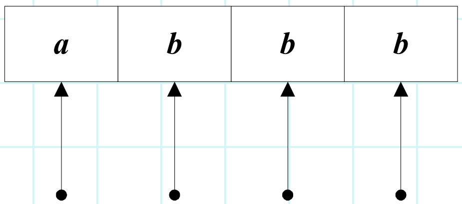
Konečný automat A_1 *môžeme opísať maticou prechodových funkcií.*

	a	b
q_0	q_0	q_1
q_1	q_1	q_0

$$A(L_1) = \{\ w \in \{a,b\}^* \mid \#_b w = 2k+1,\ k \in \mathbb{N}\ \}$$

Simulovanie výpočtu DKA – postupnosť krokov

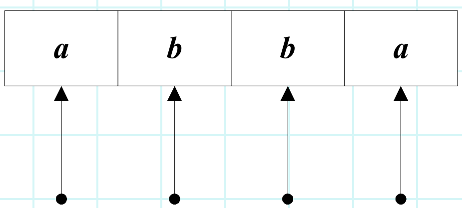
$$A(L_1) = \{\ w \in \{a,b\}^* \mid \#_b w = 2k+1,\ k \in \mathbb{N}\ \}$$



$$F = \{q_1\}.$$

$$(q_0, abbb) \vdash (q_0, bbb) \vdash (q_1, bb) \vdash (q_0, b) \vdash \underline{(q_1, \epsilon)}.$$

slovo abbb patrí do jazyka L_1



$$F = \{q_1\}.$$

$$(q_0, abba) \vdash (q_0, bba) \vdash (q_1, ba) \vdash (q_0, a) \vdash \underline{(q_0, \epsilon)}.$$

slovo abba nepatrí do jazyka L_1

Konštrukcia konečného automatu z jazyka - postup

V popise jazyka vystupuje obmedzujúca podmienka $f(a_1, a_2, ..., a_s) = \alpha k + \beta$.

- určíme počet stavov, ak konštruujeme konečný automat rozpoznávajúci takýto jazyk, tak počet jeho stavov sa rovná koeficientu α.

- určia sa koncové stavy, ak stavy označujeme q_0, q_1, ..., $q_\alpha - 1$, tak koncový stav je určený koeficientom β, tak, že $q_\beta \in F$.

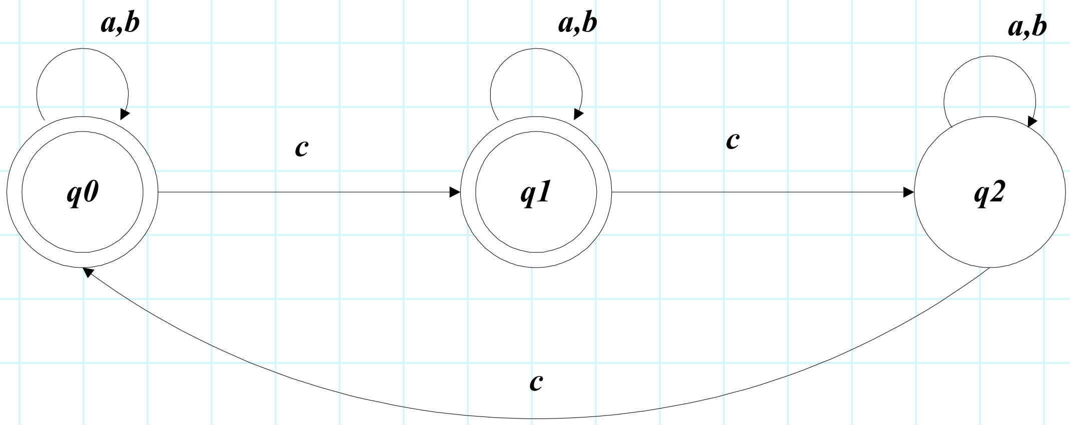
- konštruujeme prechodovú funkciu, realizujú sa prechody pre jednotlivé znaky. V prípade, že v ľavej strane obmedzujúcej podmienky sa daný znak vyskytuje s koeficientom nula, prechodová funkcia zostáva v tom istom stave, ak sa daný znak vyskytuje s koeficientom +1 prechodová funkcia sa preklápa do nasledovného stavu, ak sa ak sa daný znak vyskytuje s koeficientom -1 prechodová funkcia sa preklápa do predošlého stavu, ak sa daný znak vyskytuje s koeficientom +2 prechodová funkcia sa preklápa ob 2 do nasledovného stavu, s výskytmi ďalších znakov konštruujeme prechodovú funkciu analogicky.

Príklad – konštrukcia DKA

$$L_2 = \{\ w \in \{a,b,c\}^* \mid \#_c w = 3k \vee \#_c w = 3k+1,\ k \in \mathbb{N}\ \}.$$

Príklad – konštrukcia DKA

$$L_2 = \{\ w \in \{a, b, c\}^* \mid \#_c w = 3k \vee \#_c w = 3k + 1,\ k \in \mathbb{N}\ \}.$$

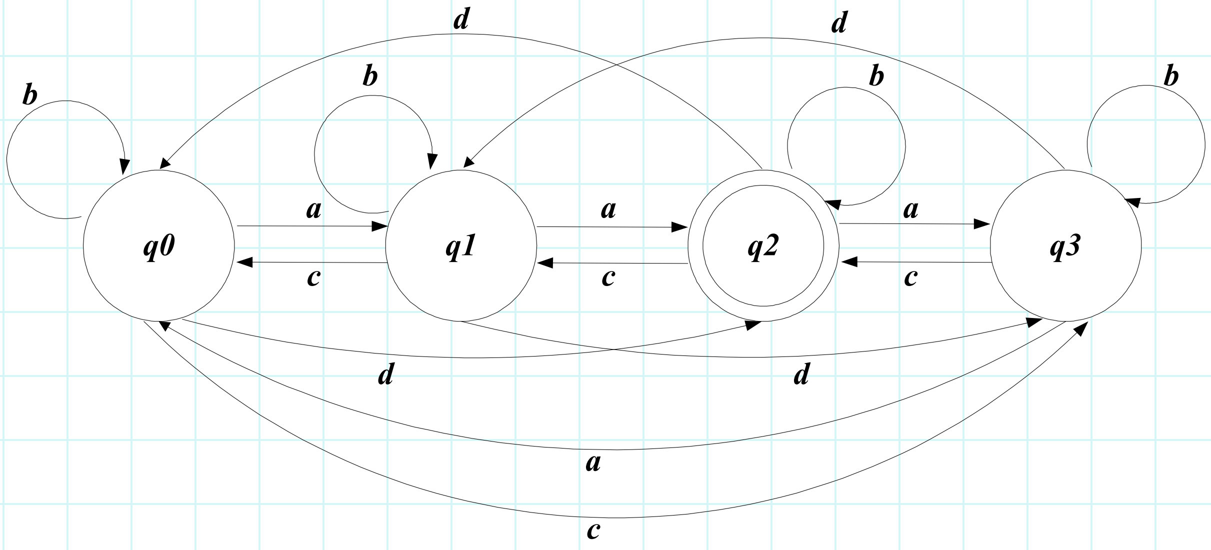


Príklad – konštrukcia DKA

$$L_3 = \{\ w \in \{a,b,c,d\}^* \mid \#_a w - \#_c w + 2\#_d w = 4k+2,\ k \in \mathbb{Z}\ \}$$

Príklad – konštrukcia DKA

$$L_3 = \{\ w \in \{a,b,c,d\}^* \mid \#_a w - \#_c w + 2\#_d w = 4k+2,\ k \in \mathbb{Z}\ \}$$



Konštrukcia konečného automatu z jazyka - formálne

Lema 3.1.1 *Formálna formulácia postupu konštrukcie konečného automatu z jazyka. Nech* Σ *je vstupná abeceda,* $\alpha,\ \beta \in \mathbb{N}$,

$$L = \{\ w \in \Sigma^* \mid f(a_1, a_2, ..., a_s) = \alpha k + \beta,\ k \in \mathbb{Z}\ \}, s \leq |\Sigma|, 0 \leq \beta < \alpha,$$

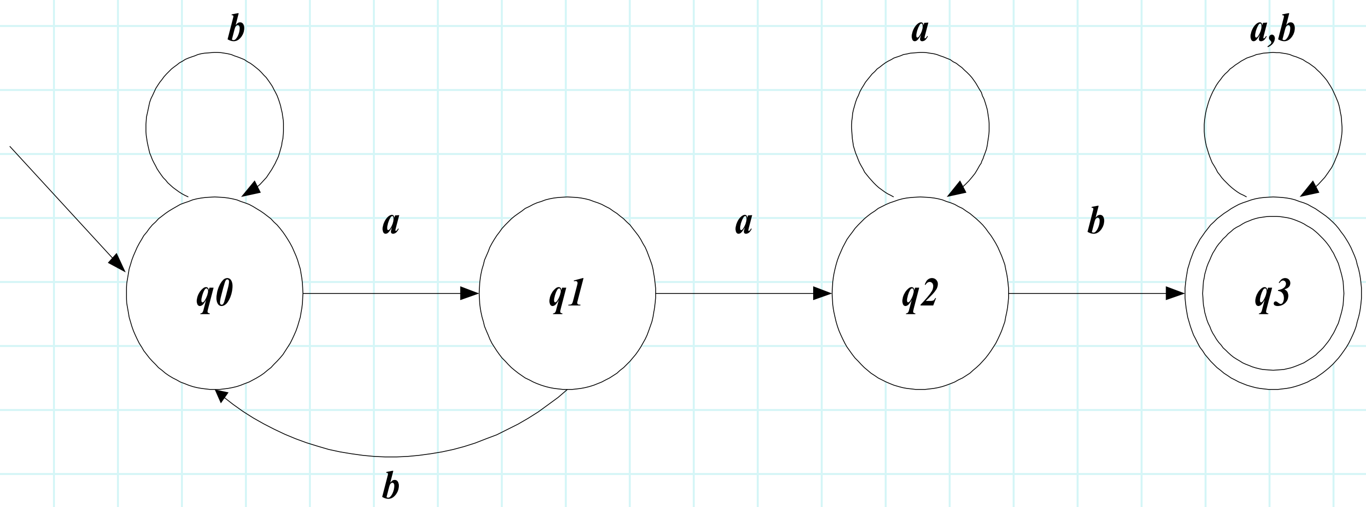
potom $A = (K, \Sigma, \delta, q_0, F)$, $|K| = \alpha,\ F = \{q_\beta\},\ |\delta| = \alpha.|\Sigma|$.

Rozpoznávanie reťazcov

rozpoznávanie L – obsahuje podslovo aab – deterministická verzia

Skonštruujeme konečný automat, ktorý rozpoznáva všetky slová, v ktorých sa nachádza podslovo aab. Takýto automat bude rozpoznávať jazyk:

$L_4 = \{xaaby \mid x, y \in \{a, b\}^*\}$



Program rozpoznávajúci L – obsahuje podslovo aab

```
int q=0;
  while ((c=fgetc()) != EOF)        {
    switch (c) {
     case 'a':       if((q>0)&&(q<2)) q++;
                     break;
     case 'b':       if((q==0)||(q==1)) q=0;
                     if((q==2) q++;
                     break;
             }
                                  }
  if (q==3) printf("abb sa nachadza v subore");
```

Program rozpoznavajúci L – obsahuje podslovo aab

```
int q=0;
while ((c=fgetc()) != EOF)
    q=M[q][c];
if (q==3) printf("abb sa nachadza v subore");
```

	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
q_3	q_3	q_3

Von Neumannov princíp:

Dáta a program sú zameniteľné. V tej istej pamäti, kde mám inštrukcie, môžem uchovávať dáta. To čo viem naprogramovať v C -> pretransformujem na údaje – matica prechodovej funkcie.

Zovšeobecnené konečné automaty – nedeterministické - NKA

Nedeterminizmus - formálna abstrakcia takých výpočtov, v ktorých nie je jednoznačne určený nasledujúci krok. Ako príklady môžu slúžiť konkurentné procesy, distribuované výpočty.

2 zdroje nedeterminizmu pri KA:

- epsilonové kroky,
- prechodová relácia, nie je prechodová funkcia.

Nedeterministický konečný automat NKA - definícia

Definícia 3.2.1 Nedeterministický konečný automat

Nedeterministický konečný automat *(NFA - nondeterministic finite automaton) je pätica* $A = (K, \Sigma, \delta, q_0, F)$, *kde:*

K *je konečná množina stavov,*

Σ *je vstupná abeceda,*

δ *je prechodové zobrazenie, pričom platí*

$$\delta : K \times (\Sigma \cup \{\epsilon\}) \to 2^K,$$

$q_0 \in K$ *je počiatočný stav,*

$F \subseteq K$ *je množina koncových (akceptačných) stavov.*

Konfigurácia nedeterministického konečného automatu *je dvojica*

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

kde:

q *je momentálny stav,*

w *je zvyšok vstupu.*

Nedeterministický konečný automat – krok výpočtu, jazyk rozpoznávaný NKA

Krok výpočtu nedeterministického konečného automatu *je relácia* $\vdash_A$ *na konfigurácii definovaná nasledovne:*

$$(q, au) \vdash_A (p, u), \qquad ak \qquad p \in \delta(q, a),$$

kde $a \in \Sigma \cup \{\epsilon\}$, $u \in \Sigma^*$.

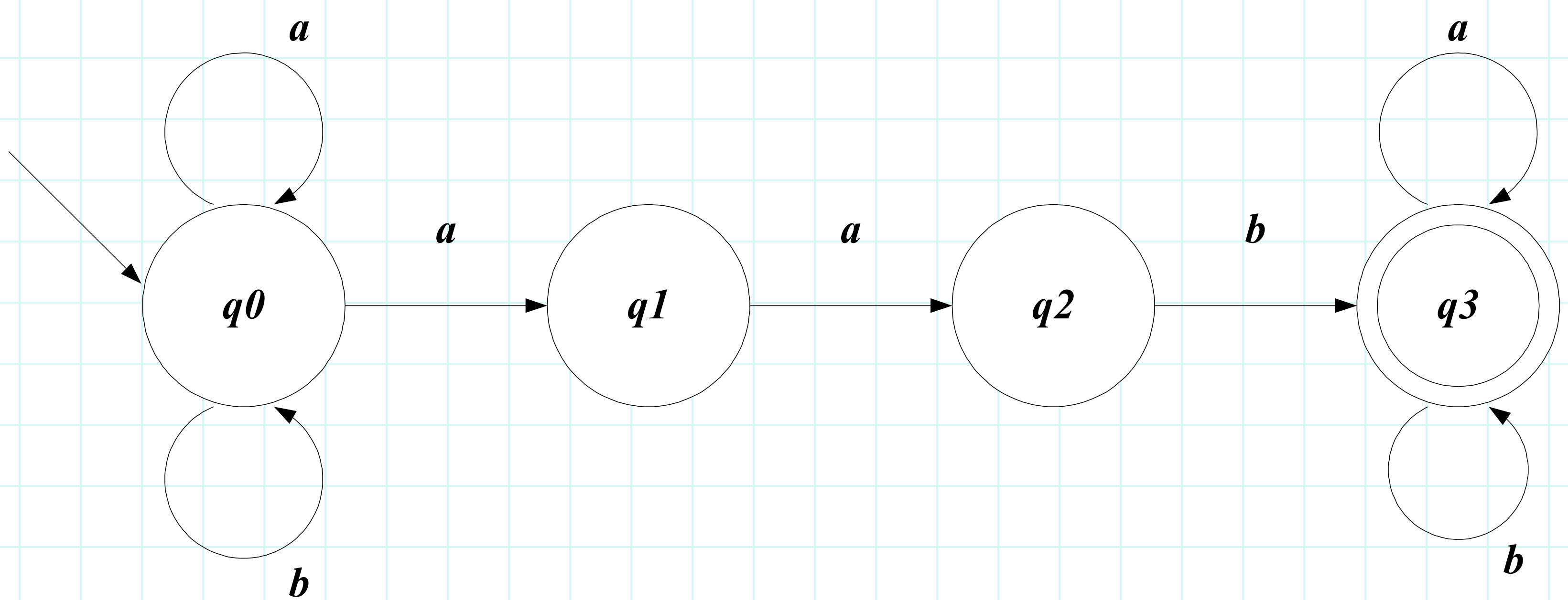
Jazyk rozpoznávaný deterministickým konečným automatom *(koncovým stavom) je množina*

$$L(A) = \{\, w \in \Sigma^* \mid (q_0, w) \vdash_A^* (q, \epsilon),\ q \in F \}.$$

Rozpoznávanie reťazcov

rozpoznávanie L – obsahuje podslovo aab – nedeterministická verzia

$$L_5 = \{xaaby \mid x, y \in \{a,b\}^*\}$$



Princíp nedeterminizmu

- Ak existuje akceptujúci výpočet, tak nedeterministické rozhodnutie mi zabezpečí, že konečný automat nájde tento výpočet.
- Ak pre dané slovo akceptujúci výpočet neexistuje, nesmie existovať spôsob, ako sa do koncového stavu dostať (žiadnou kombináciou prechodov).
- V skutočnosti sa to rieši prehľadávaním všetkých možností – backtracking.

• zdroj nedeterminizmu - ε kroky

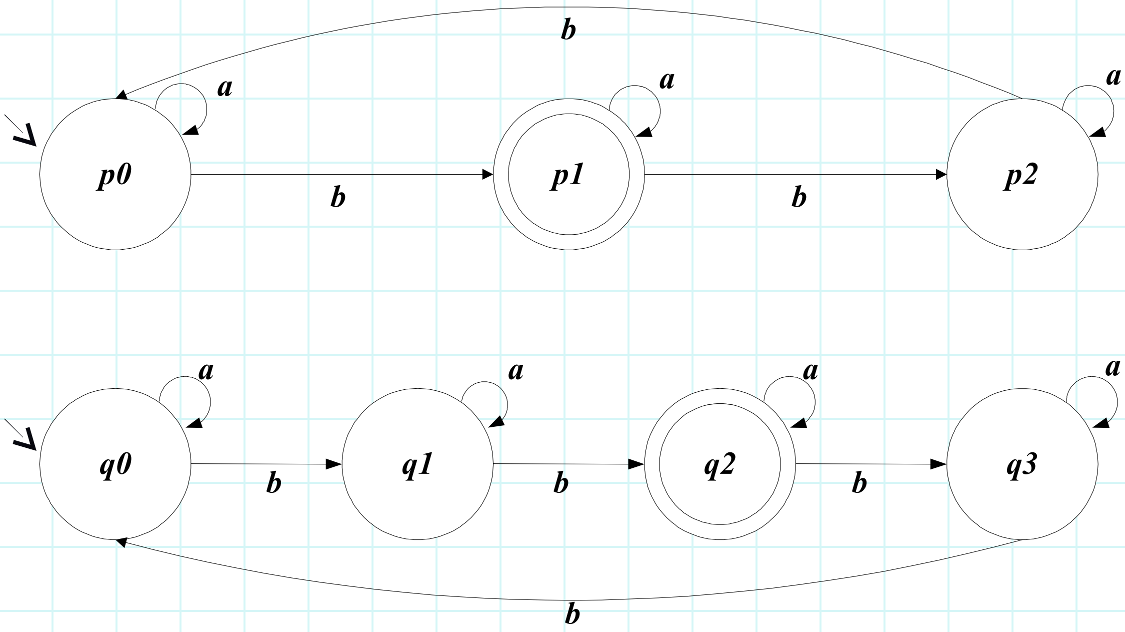
$$L_6 = L_{2b} \cup L_{3b},$$

pričom $L_{2b} = \{\ w \in \{a,b\}^* \mid \#_b w = 3k+1,\ k \in \mathbb{N}\ \}$, $L_{3b} = \{\ w \in \{a,b\}^* \mid \#_b w = 4k+2,\ k \in \mathbb{N}\ \}$.

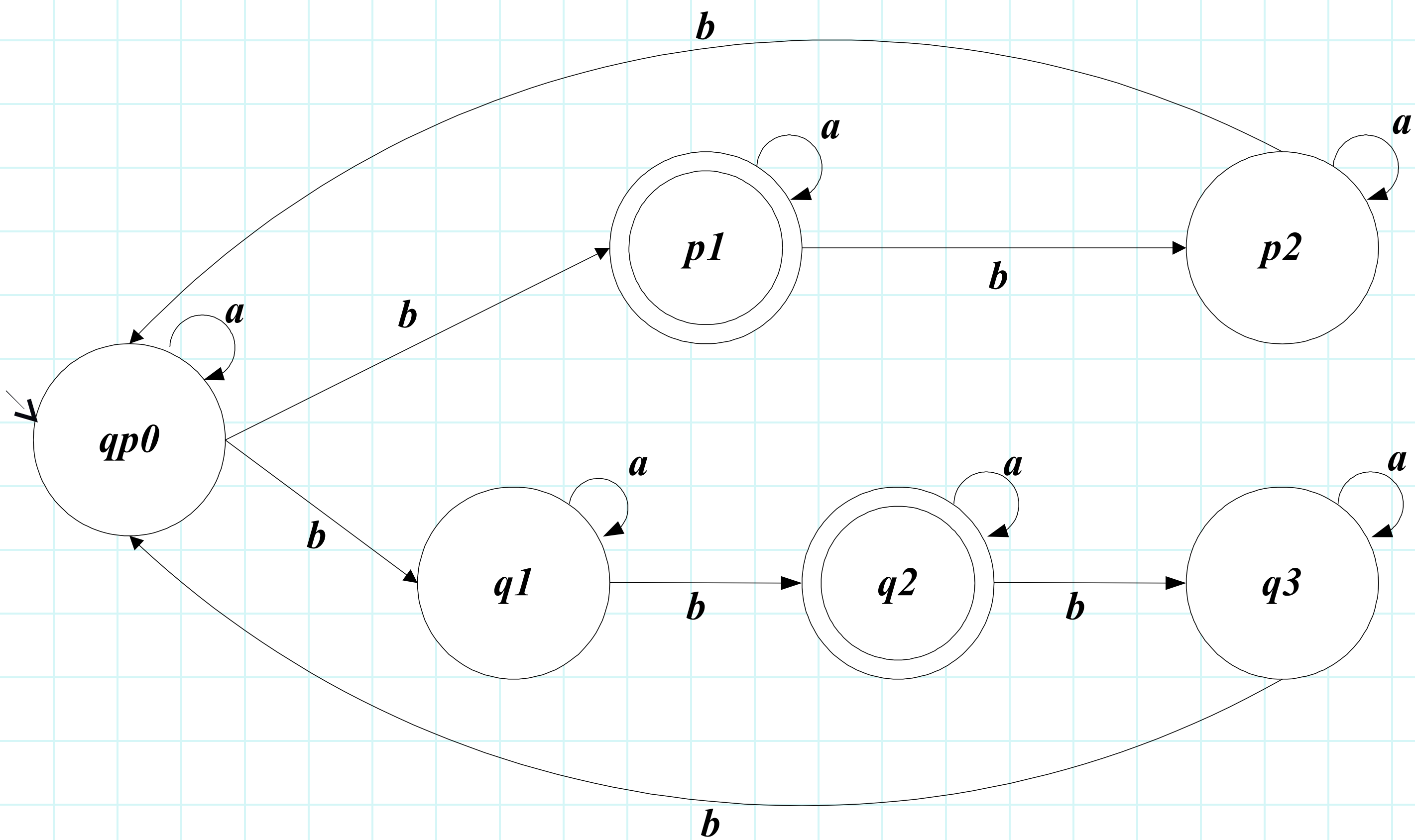
?

$L_{2b} \cup L_{3b}$

?

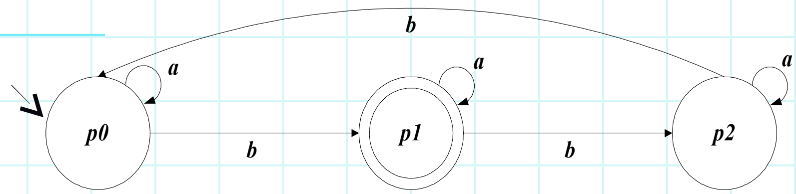


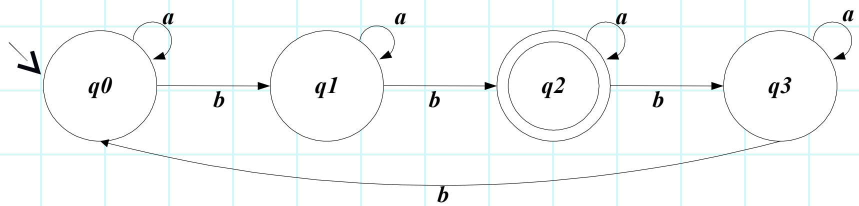
Nesprávne riešenie !!!

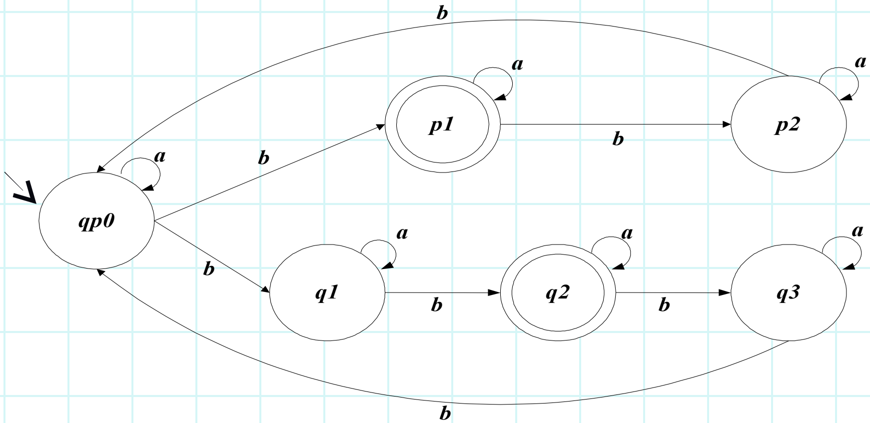


$$L_6 = L_{2b} \cup L_{3b},$$

pričom $L_{2b} = \{\ w \in \{a,b\}^* \mid \#_b w = 3k+1,\ k \in \mathbb{N}\ \}$, $L_{3b} = \{\ w \in \{a,b\}^* \mid \#_b w = 4k+2,\ k \in \mathbb{N}\ \}$.







Nesprávne riešenie !!!

1. zdroj nedeterminizmu - ε kroky

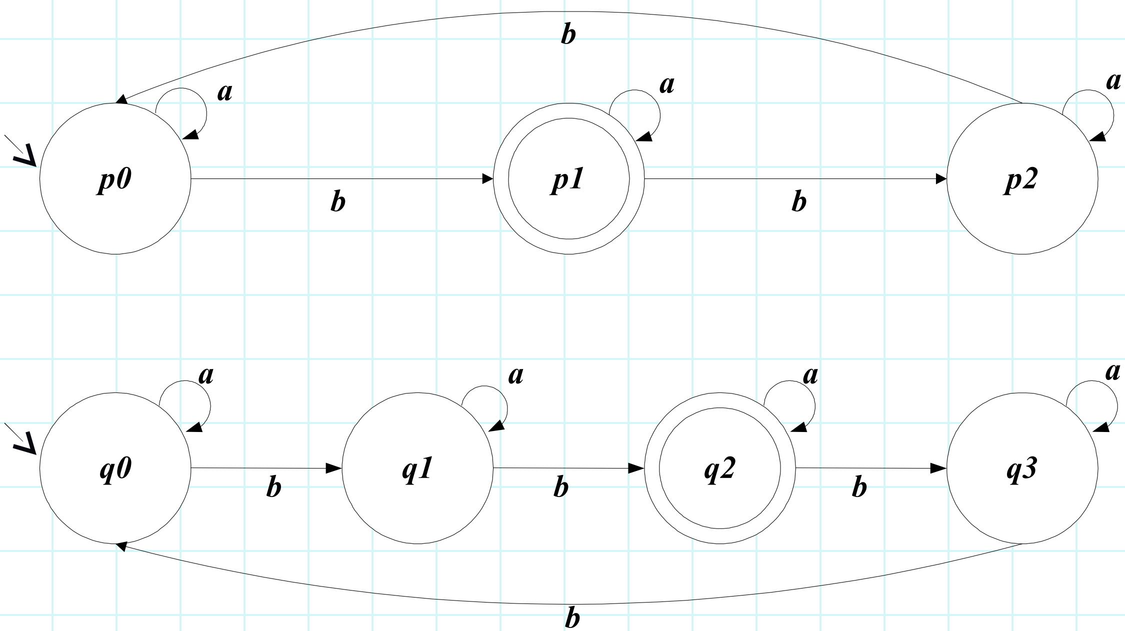
$$L_6 = L_{2b} \cup L_{3b},$$

pričom $L_{2b} = \{\ w \in \{a,b\}^* \mid \#_b w = 3k+1,\ k \in \mathbb{N}\ \}$, $L_{3b} = \{\ w \in \{a,b\}^* \mid \#_b w = 4k+2,\ k \in \mathbb{N}\ \}$.

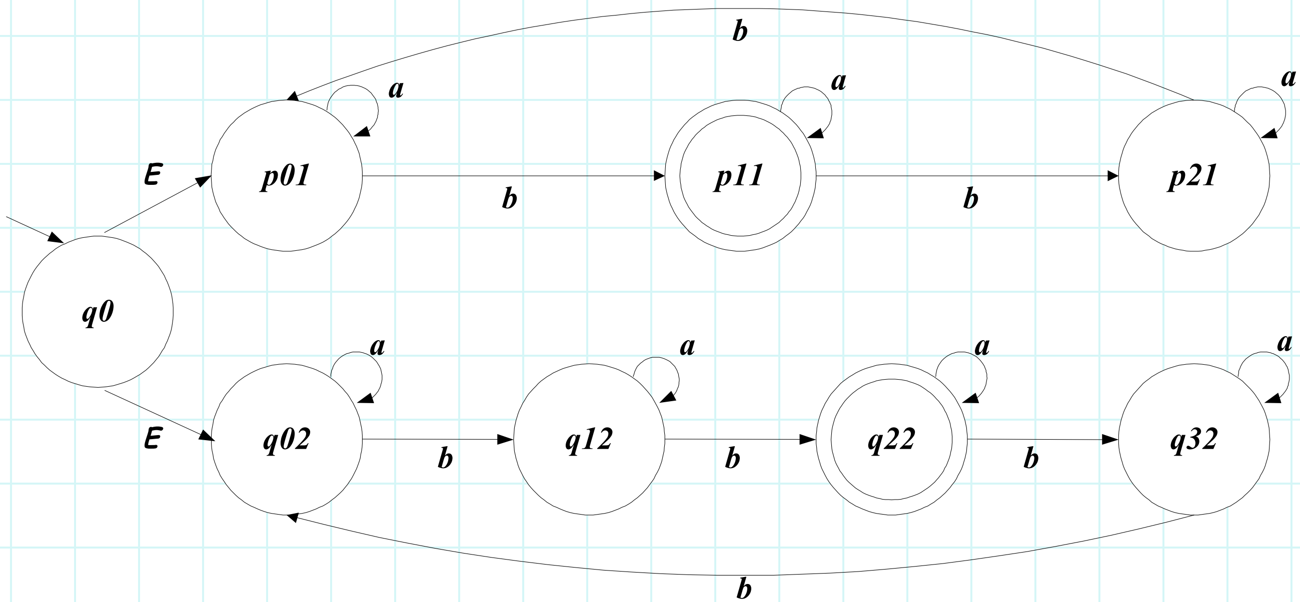
?

$L_{2b} \cup L_{3b}$

?

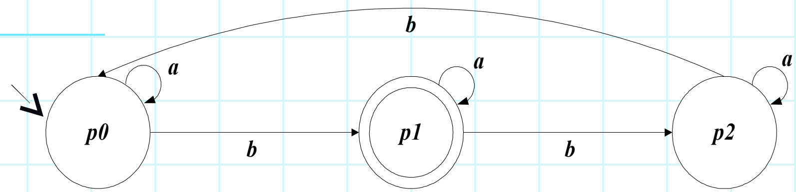


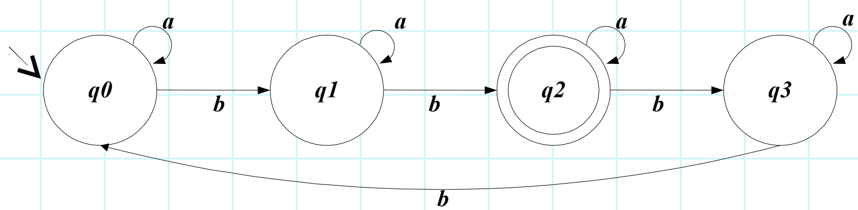
Správne riešenie



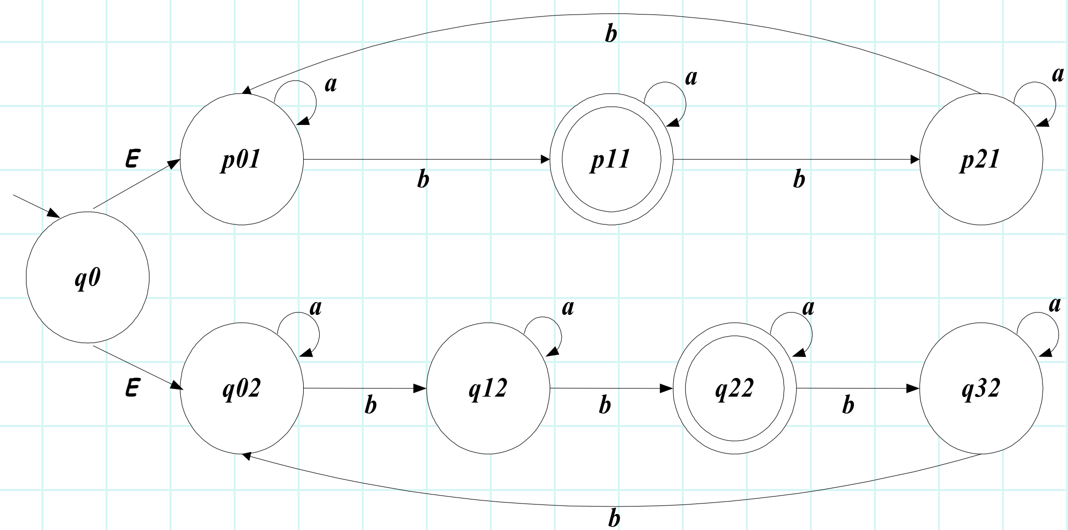
$$L_6 = L_{2b} \cup L_{3b},$$

pričom $L_{2b} = \{\ w \in \{a,b\}^* \mid \#_b w = 3k+1,\ k \in \mathbb{N}\ \}$, $L_{3b} = \{\ w \in \{a,b\}^* \mid \#_b w = 4k+2,\ k \in \mathbb{N}\ \}$.





Správne riešenie !!!



Trieda jazykov rozpoznávaných konečnými automatmi

Pre skúmanie triedy jazykov rozpoznávaných konečnými automatmi zaveďme nasledovné označenia:

- trieda jazykov rozpoznávaných deterministickými konečnými automatmi **L(DFA)**,
- trieda jazykov rozpoznávaných nedeterministickými konečnými automatmi **L(NFA)**,
- trieda jazykov rozpoznávaných konečnými automatmi **L(FA)**.

Trieda jazykov rozpoznávaných konečnými automatmi

Veta 3.3.1 *Trieda jazykov rozpoznávaných deterministickými konečnými automatmi je ekvivalentná s triedou jazykov rozpoznávaných nedeterministickými konečnými automatmi. Formálne:*

$$\mathcal{L}(DFA) = \mathcal{L}(NFA)\ . \tag{3.1}$$

Poznámka 3.3.1 *Ku každému nedeterministickému konečnému automatu môžeme zostrojiť deterministický konečný automat.*

Veta 3.3.2 *Trieda jazykov rozpoznávaných konečnými automatmi je zhodná s triedou všetkých regulárnych jazykov. Formálne:*

$$\mathcal{L}(FA) = \mathcal{R}\ . \tag{3.2}$$

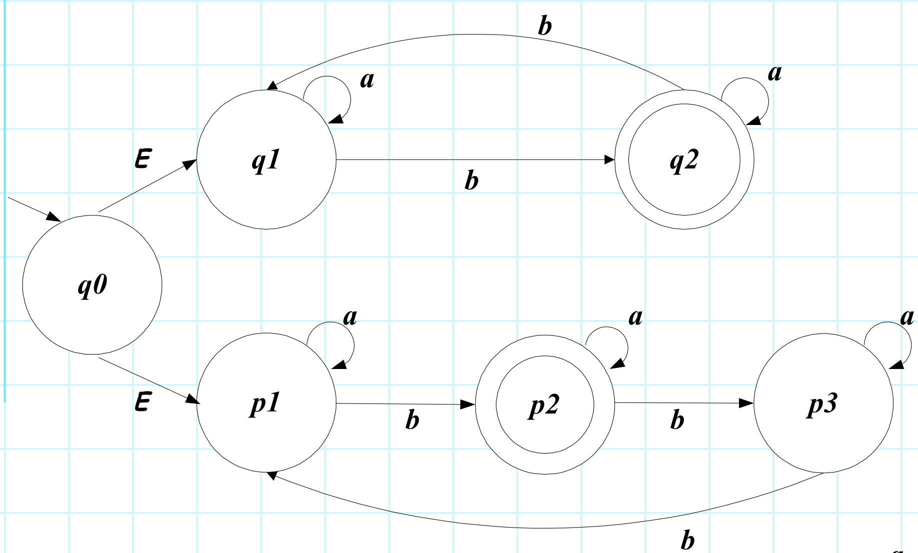
Prevod NKA ->DKA

Ku každému nedeterministickému KA viem zostrojiť nedeterministický KA.

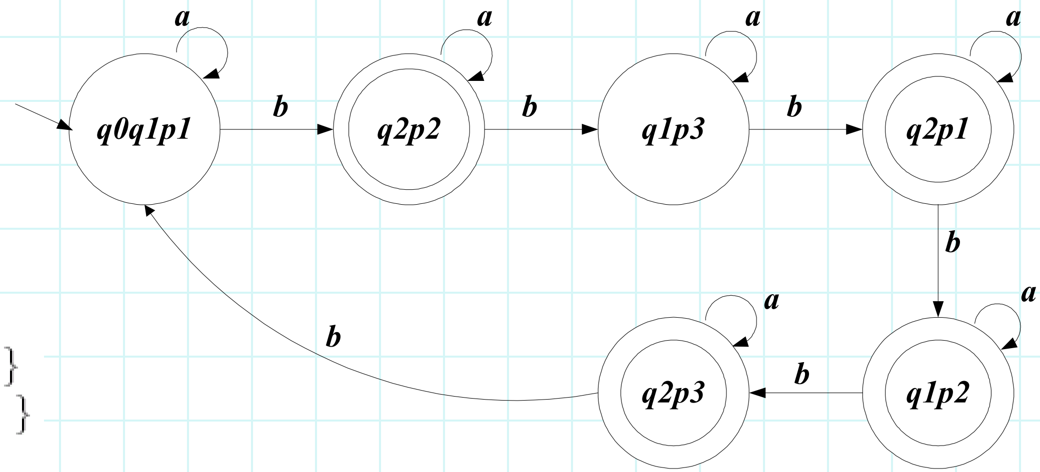
Algoritmus transformácie:

- Odstránenie ε krokov, konštrukcia ε uzáveru, zrušenie prvého zdroja nedeterminizmu
- Zrušenie druhého zdroja nedeterminizmu, „transformácia prechodovej relácie na prechodovú funkciu“

Prevod NKA ->DKA - príklad



Prevod
NKA ->DKA



$$A_{7N} = A_1 \cup A_{2b}.$$

$$A(L_1) = \{\ w \in \{a,b\}^* \mid \#_b w = 2k+1,\ \ k \in \mathbb{N}\ \}$$
$$A(L_{2b}) = \{\ w \in \{a,b\}^* \mid \#_b w = 3k+1,\ \ k \in \mathbb{N}\ \}$$

ϵ-uzáver konečného automatu

Vstup:
Množina stavov, $T \subseteq K$
Výstup:
$\epsilon - U(T)$
Premenné:
Z - zásobník
begin

```
    Z := T // ulož všetky stavy z T do Z
    ε − U(T) := T
    while Z nie je prázdny
        A := Z.Pop()
        for ∀ stav t do ktorého existuje ε − krok z s
            if t ∉ ε − U(T)
            {
                ε − U(T) := ε − U(T) ⋃ {t}
                Z.Push(t)
            }
        endfor
    endwhile
end
```

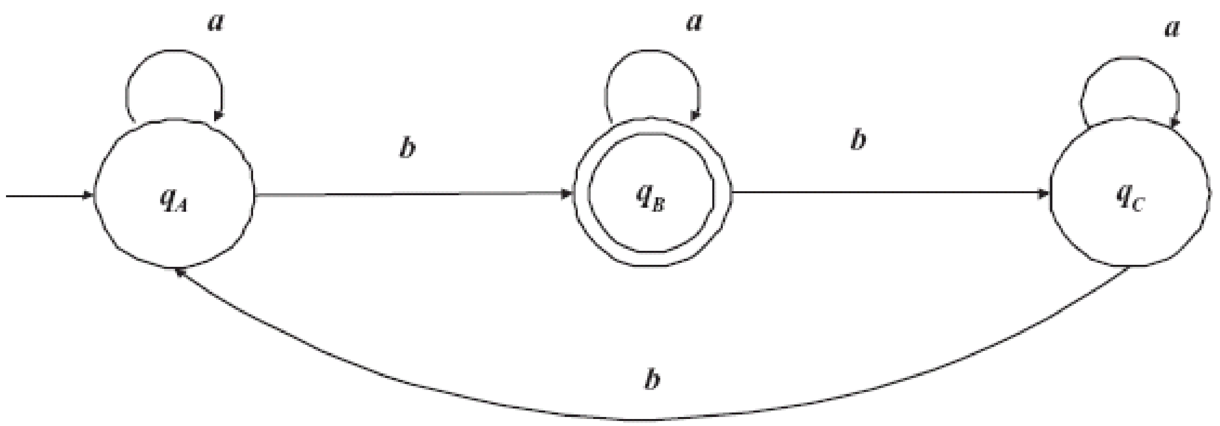
Konštrukcia DFA

Vstup:
NFA, $A = (K, \Sigma, \delta, q_0, F)$
Výstup:
DFA, $A_D = (K_D, \Sigma, \delta_D, q_{0D}, F_D)$
Poznámka:
dva typy stavov:
•neoznačené q
•označené $\overline{q}$
begin
$K_D := \epsilon - U(q_0)$ // všetky stavy v K_D zatiaľ neoznačené
while $\exists$ neoznačený stav $t \in K_D$
označ t v K_D ako $\overline{t}$
for $\forall a \in \Sigma$
$u := \epsilon - U(\delta(t, a))$
if $u \notin K_D$
$K_D := K_D \bigcup \{u\}$ // neoznačené
endif
$\delta_D(t, a) := u$ // nová prechodová funkcia
endfor
endwhile
Označ q_{0D} a F_D
end

Príklad 3.3.2 *Vzťah konečných automatov a regulárnych gramatík.*
Majme regulárnu gramatiku G_8.
Nech $G_8 = (N, T, P, A)$, *kde* $N = \{A, B, C\}$, $T = \{a, b\}$,
$P = \{$

$$A \to aA \mid bB$$
$$B \to aB \mid bC$$
$$C \to aC \mid bA$$
$$B \to \varepsilon$$

$\}$.
Gramatika G_8 *generuje jazyk* $L(G_8) = \{\ w \in \{a, b\}^* \mid \#_b w = 3k + 1,\ k \in \mathbb{N}\ \}$.
Skonštruujme konečný automat A_8, *taký že* $L(A_8) = L(G_8)$.



Obrázok 3.12: Konečný automat A_8.

Skonštruovaný konečný automat A_8 *vidíme na obrázku 3.12.*

Ďakujem za pozornosť.

chuda@fiit.stuba.sk

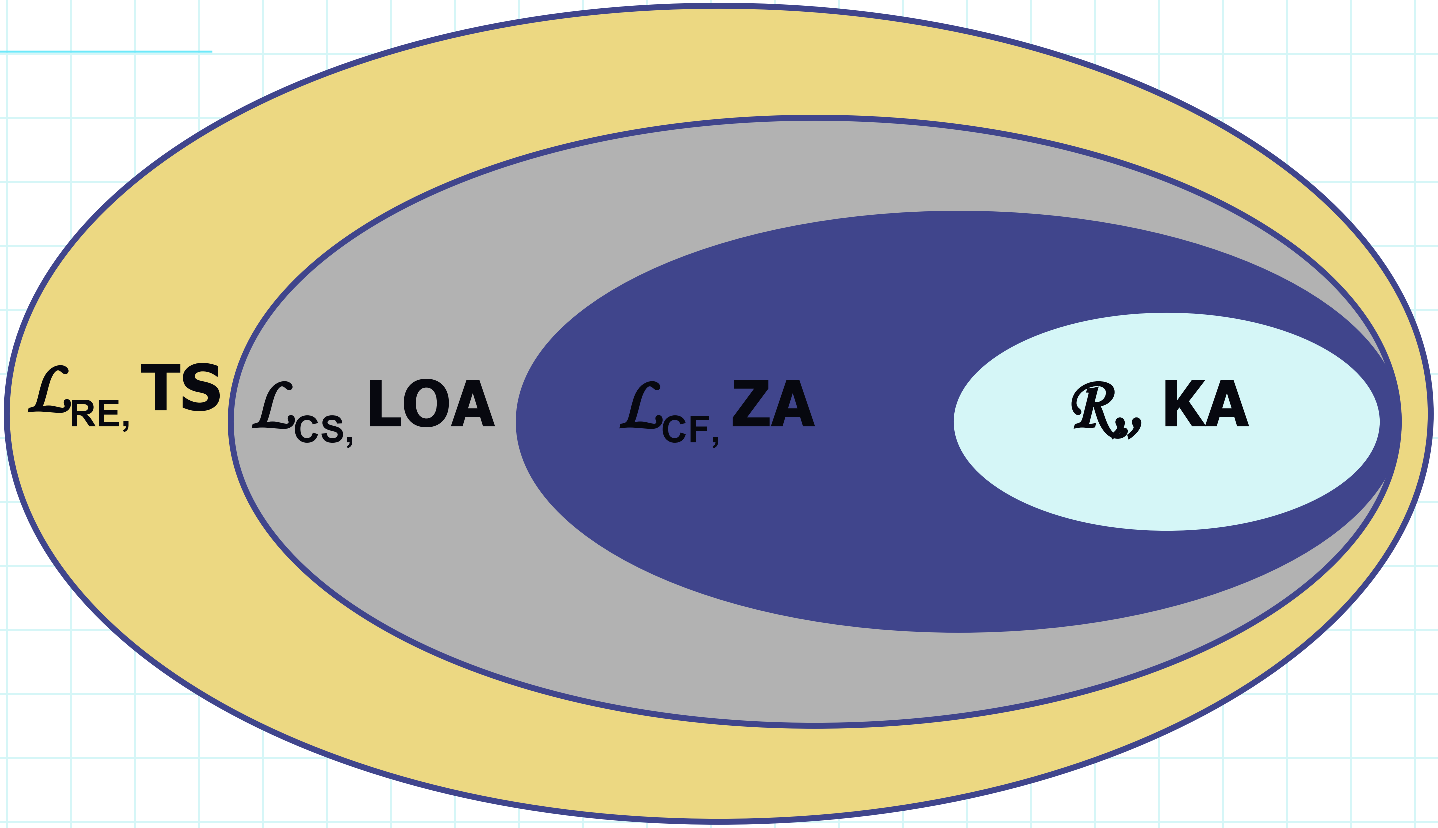
S T U
F I I T

Teoretické základy informatiky

Uzáverové vlastnosti Regulárnych jazykov

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ **Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)**

$\mathcal{L}_{CS}$ **Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)**

$\mathcal{L}_{CF}$ **Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)**

$\mathcal{R}$ **Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)**

Deterministický konečný automat DKA -definícia

Definícia 3.1.1 Deterministický konečný automat

Deterministický konečný automat *(DFA - deterministic finite automaton) je pätica* $A = (K, \Sigma, \delta, q_0, F)$*, kde:*

K je konečná množina stavov,

Σ je vstupná abeceda,

δ je prechodová funkcia, pričom platí

$$\delta : K \times \Sigma \to K$$

$q_0 \in K$ je počiatočný stav,

$F \subseteq K$ je množina koncových (akceptačných) stavov.

Konfigurácia deterministického konečného automatu *je dvojica*

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

kde:

q je momentálny stav,

w je zvyšok vstupu.

Nedeterministický konečný automat NKA - definícia

Definícia 3.2.1 Nedeterministický konečný automat

Nedeterministický konečný automat *(NFA - nondeterministic finite automaton) je pätica* $A = (K, \Sigma, \delta, q_0, F)$*, kde:*

K *je konečná množina stavov,*

Σ *je vstupná abeceda,*

δ *je prechodové zobrazenie, pričom platí*

$$\delta : K \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^K,$$

$q_0 \in K$ *je počiatočný stav,*

$F \subseteq K$ *je množina koncových (akceptačných) stavov.*

Konfigurácia nedeterministického konečného automatu *je dvojica*

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

kde:

q *je momentálny stav,*

w *je zvyšok vstupu.*

Vzťah automatov a jazykov z Chomského hierarchie

Veta 3.3.1 *Trieda jazykov rozpoznávaných deterministickými konečnými automatmi je ekvivalentná s triedou jazykov rozpoznávaných nedeterministickými konečnými automatmi. Formálne:*

$$\mathcal{L}(DFA) = \mathcal{L}(NFA)\ . \tag{3.1}$$

Poznámka 3.3.1 *Ku každému nedeterministickému konečnému automatu môžeme zostrojiť deterministický konečný automat.*

Veta 3.3.2 *Trieda jazykov rozpoznávaných konečnými automatmi je zhodná s triedou všetkých regulárnych jazykov. Formálne:*

$$\mathcal{L}(FA) = \mathcal{R}\ . \tag{3.2}$$

Automat	Determin. verzia		Nedetermin. verzia		Trieda jazykov
konečný	$\mathcal{L}(DFA)$	$=$	$\mathcal{L}(NFA)$	$=$	$\mathcal{R}$

Uzavretosť

Definícia 3.4.1 Uzavretosť
Trieda jazykov $\mathcal{L}$ je uzavretá vzhľadom na operáciu $\Box : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ *ak platí:*

$$\forall\ L_1, L_2 \in \mathcal{L} \quad (L_1 \Box L_2) \in \mathcal{L}\ .$$

- Pri dokazovaní platnosti – dôkaz.
- Pri dokazovaní neplatnosti – kontrapríklad.

Uzavretosť $\mathcal{R}$ vzhľadom na operácie.

Veta 3.4.1 *Trieda $\mathcal{R}$ je uzavretá vzhľadom nasledujúce na operácie:*

1. *zjednotenie* $(\cup)$,
2. *zreťazenia* $(\cdot)$,
3. *prieniku* $(\cap)$,
4. *Kleeneho iterácie* $(^*)$,
5. *doplnku* $(^C)$,
6. *kladný uzáver* $(^+)$,
7. *zrkadlový obraz* $(^R)$.

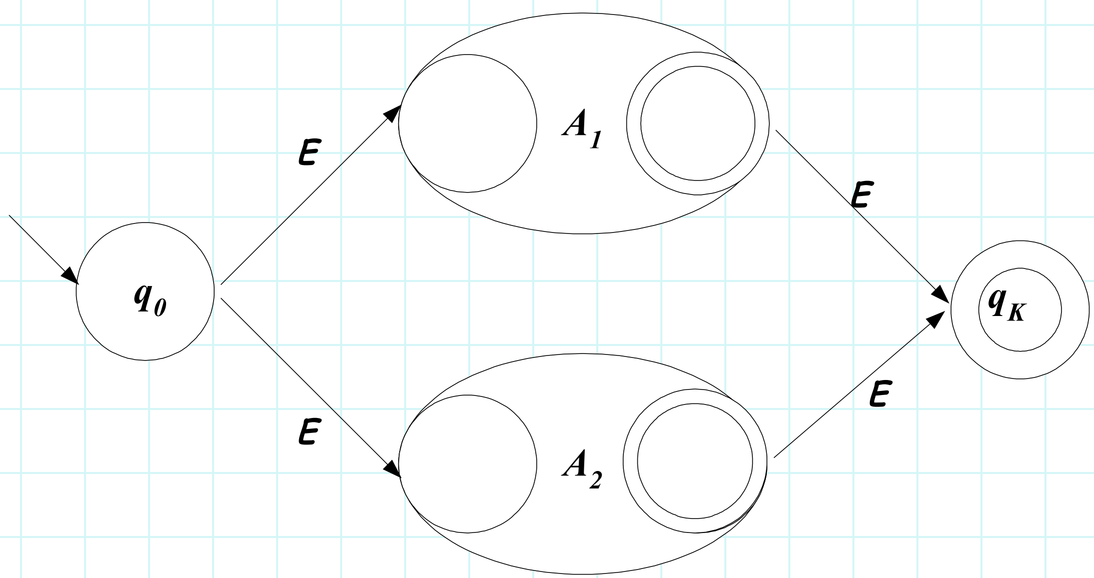
Jediný koncový stav KA

Lema 3.4.1 *Jediný koncový stav konečného automatu.*
Ku každému konečnému automatu A existuje konečný automat A' taký, ktorý má jediný koncový stav a platí:

$$L(A) = L(A')$$

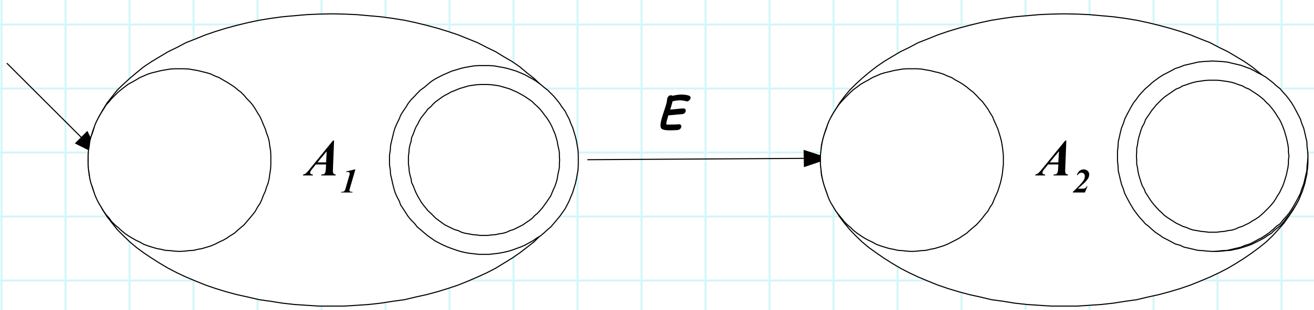
Uzavretosť vzhľadom na zjednotenie

- L_1, L_2 ϵ$\mathcal{R}$ chcem ukázať L_1 u L_2 ϵ $\mathcal{R}$
- Keď L_1, L_2 ϵ$\mathcal{R}$ => musia exist. A_1,A_2 také že L(A_1)= L_1 a L(A_2)= L_2 , také čo majú jeden koncový stav
- Nový automat zostrojím pomocou ε krokov s novým počiatočným a koncovým stavom.



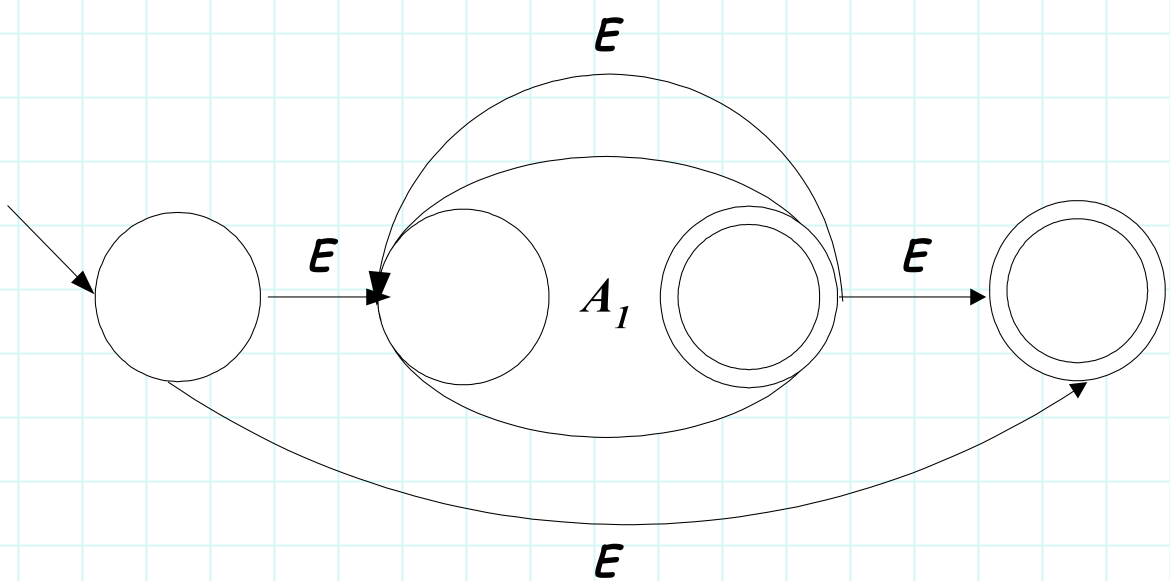
Uzavretosť vzhľadom na zreťazenie

- L_1, $L_2 \in \mathcal{R}$ chcem ukázať $L_1 . L_2 \in \mathcal{R}$
- Keď L_1, $L_2 \in \mathcal{R}$ => musia exist. A_1,A_2 také že $L(A_1) = L_1$ a $L(A_2) = L_2$, také čo majú jeden koncový stav
- Nový automat zostrojím pomocou ε kroku.



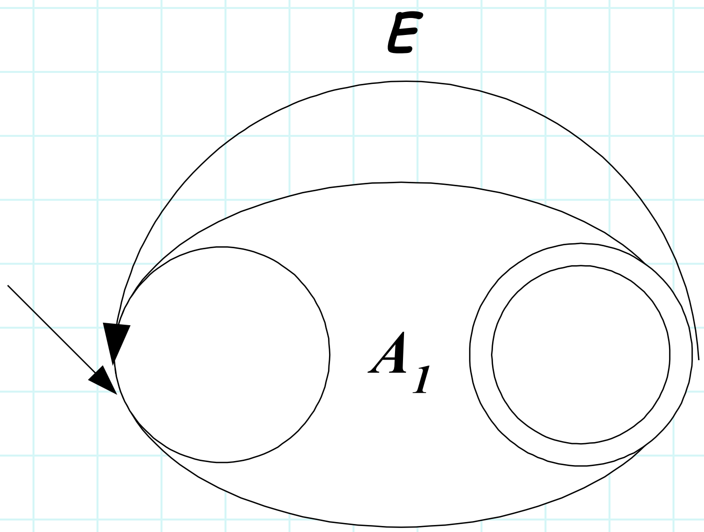
Uzavretosť vzhľadom na Kleeneho iteráciu-unárnu

- $L_1 \in \mathcal{R}$ chcem ukázať $L_1^* \in \mathcal{R}$
- Keď $L_1 \in \mathcal{R}$ => musí exist. A_1 taký že $L(A_1) = L_1$, taký čo má jeden koncový stav
- Nový automat zostrojím pomocou ε krokov.



Uzavretosť vzhľadom na kladný uzáver

- $L_1 \in \mathcal{R}$ chcem ukázať $L_1^+ \in \mathcal{R}$
- Keď $L_1 \in \mathcal{R}$ => musí exist. A_1 taký že $L(A_1) = L_1$, taký čo má jeden koncový stav
- Nový automat zostrojím pomocou ε kroku.

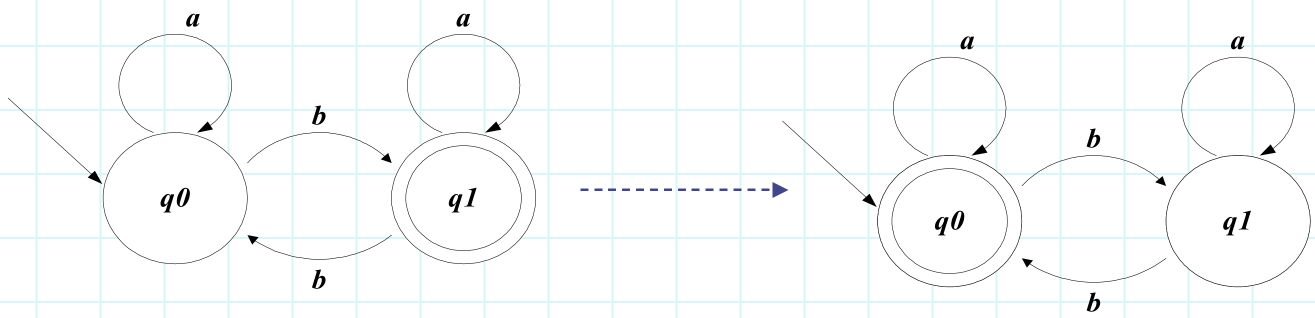


Uzavretosť vzhľadom na doplnok

- $L_1 \in \mathcal{R}$ chcem ukázať $L_1^c \in \mathcal{R}$
- Keď $L_1 \in \mathcal{R}$ => musí exist. A_1 také že $L(A_1) = L_1$
- Nový automat zostrojím „zámenou koncových a nekoncových stavov".

$L=\{w \in \Sigma^* \mid P(w)\}$ ⇢ $L^c=\{w \in \Sigma^* \mid \text{not } P(w)\}$

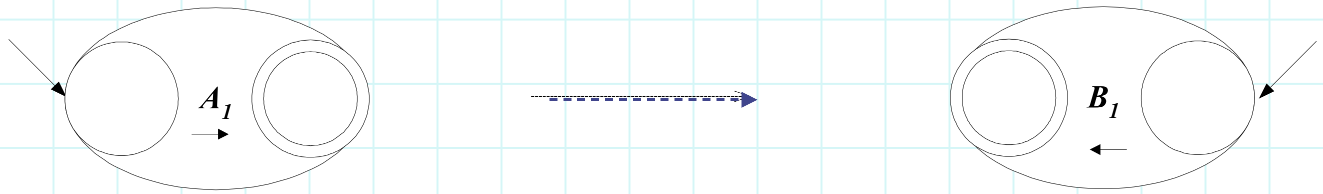
Príklad:



$L=\{w \in \{a,b\}^* \mid \#_b w = 2k+1\}$ ⇢ $L=\{w \in \{a,b\}^* \mid \#_b w = 2k\}$

Uzavretosť vzhľadom na zrkadlový obraz

- $L_1 \in \mathcal{R}$ chcem ukázať $L_1^R \in \mathcal{R}$
- Keď $L_1 \in \mathcal{R}$ => musí exist. A_1 také že $L(A_1) = L_1$, taký čo má jeden koncový stav
- Nový automat zostrojím „výmenou smeru".



Uzavretosť vzhľadom na prienik

- $L_1, L_2 \in \mathcal{R}$ chcem ukázať $L_1 \cap L_2 \in \mathcal{R}$

- **Nekonštruktívnym spôsobom**

využijem demorganové pravidlá, trieda $\mathcal{R}$ je uzavretá vzhľadom na $\cup$, ^{c} ,keď ich použijem viacnásobne, je uzavretá aj na $\cap$

$$\mathbf{L_1 \cap L_2 = (L_1^c \cup L_2^c)^c}$$

- **Konštruktívnym spôsobom**

skonštruujem takýto automat $\mathbf{L_1 \cap L_2 = (L_1^c \cup L_2^c)^c}$

Úplný algebraický systém

- pomocou operácií u, ., * zadefinujem akýkoľvek jazyk.

Príklad na uzavretosť

Nech $\triangle$ je binárna relácia nad jazykmi definovaná nasledovne:

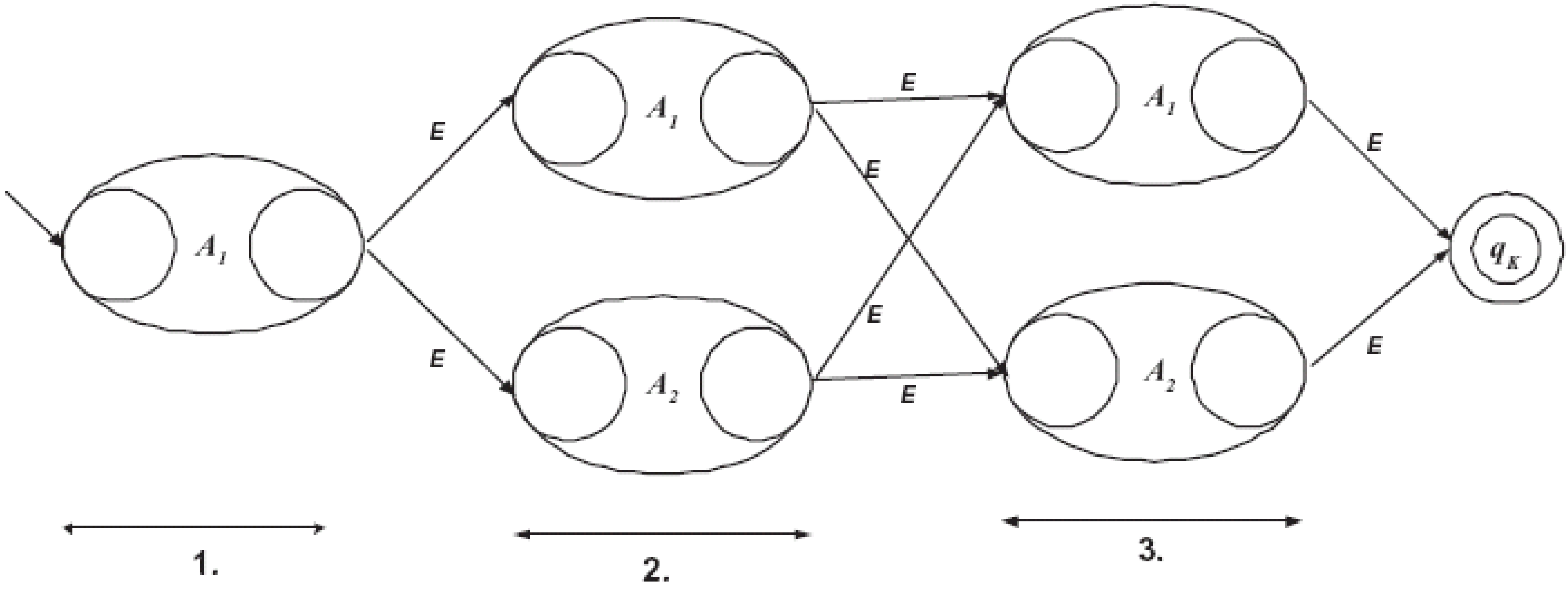
$$L_1 \triangle L_2 = L_1.(L_1 \cup L_2)^2$$

Zistite, či je trieda $\mathcal{R}$ uzavretá vzhľadom na $\triangle$?

Príklad na uzavretosť

Príklad 3.4.1 *Nech* $\triangle$ *je binárna operácia nad jazykmi definovaná nasledovne:* $L_1 \triangle L_2 = L_1.(L_1 \cup L_2)^2$. *Zistite, či je trieda regulárnych jazykov* $\mathcal{R}$ *uzavretá vzhľadom na operáciu* $\triangle$. *Odpoveď zdôvodnite.*

Dokážme, že trieda regulárnych jazykov $\mathcal{R}$ *je uzavretá vzhľadom na operáciu* $\triangle$.
Využijeme vetu 3.3.2 $\mathcal{L}(FA) = \mathcal{R}$. *Stačí skonštruovať konečný automat, ktorý reprezentuje jazyk* $L_1 \triangle L_2$.
Rozpíšme jazyk
$L_1 \triangle L_2 = L_1.(L_1 \cup L_2)^2 = L_1.(L_1 \cup L_2).(L_1 \cup L_2)$.
Keď využijeme fakt
$L(A_1) = L_1$
$L(A_2) = L_2$
potom sa dá jazyk reprezentovať automatom B skonštruovaným na obrázku 3.13. Vidíme, že celý obrázok 3.13 reprezentuje konečný automat B, ktorý rozpoznáva nejaký jazyk a po aplikácii vety 3.3.2 $\mathcal{L}(FA) = \mathcal{R}$ *vyplýva, že tento jazyk je regulárny.*



Obrázok 3.13: Konečný automat B.

Uzáverové vlastnosti triedy $\mathcal{R}$

Trieda jazykov	$\cup$	$\cdot$	$\cap$	$*$	$+$	c	R	h	h bez ϵ
$\mathcal{R}$	A	A	A	A	A	A	A	A	A

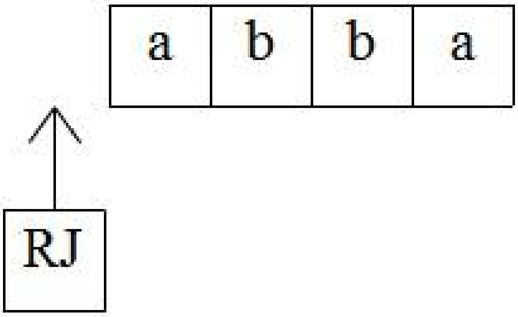
Aplikácie konečných automatov

- V praxi:
  - automat na mince
  - semafor, spínače, človeče nehnevaj sa
- V informatike:
  - lexikálna analýza
    - rozpoznávanie identifikátorov, kľúčových slov
    - kontrola tej časti programu, ktorá je definovaná $\mathcal{R}$
  - stavové diagramy
    - UML, OMT
    - jeden z nástrojov, pomocou ktorého sa generuje dynamika v informačnom systéme

Ďakujem za pozornosť.

chuda@fiit.stuba.sk

$$L_1=\{w\in\{a,b\}^* \mid |w|=3k+1;\ k\in N\}$$



Vstupné slovo: abba NASTAV

$DFA_1=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2\}$

$\Sigma=\{a,b\}$

$\delta\ (q_0,a)=q_1$

$\delta\ (q_0,b)=q_1$

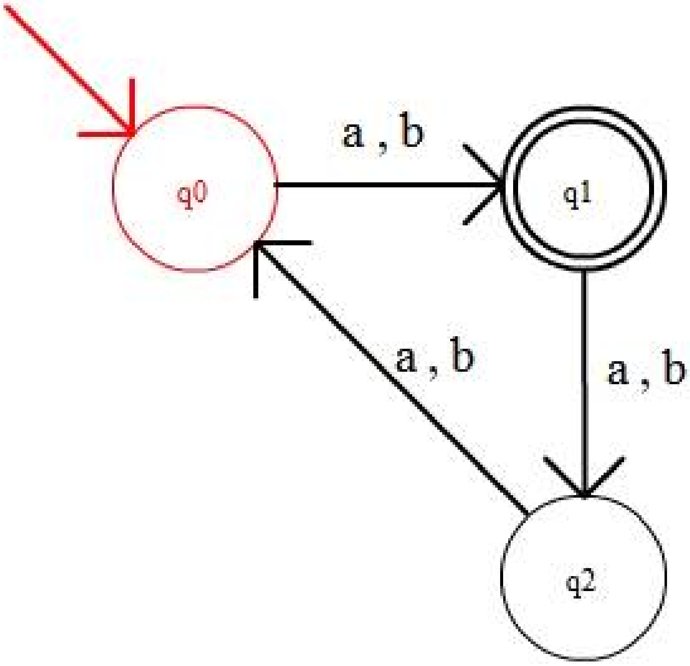
$\delta\ (q_1,a)=q_2$

$\delta\ (q_1,b)=q_2$

$\delta\ (q_2,a)=q_0$

$\delta\ (q_2,b)=q_0$

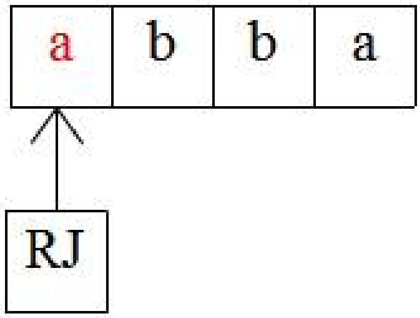
$F=\{q_1\}$



POMOC KROK ANIMACIA

Frame 2

$L_1 = \{ w \in \{a,b\}^* \mid |w| = 3k+1;\ k \in N \}$



Vstupné slovo: abba NASTAV

$DFA_1 = (K, \Sigma, \delta, q_0, F)$

$K = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_1$

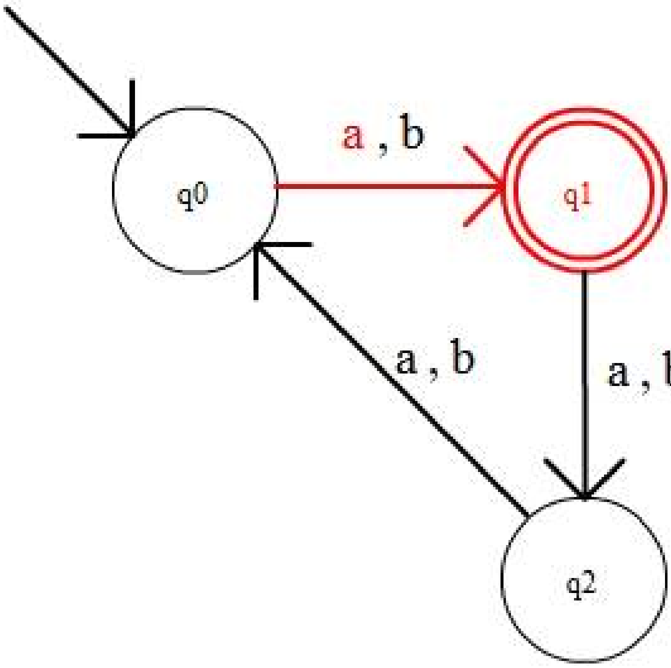
$\delta(q_1, a) = q_2$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_0$

$\delta(q_2, b) = q_0$

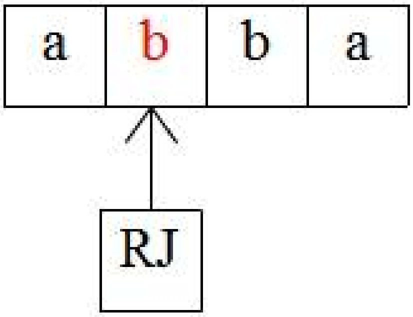
$F = \{q_1\}$



POMOC KROK ANIMACIA

Frame 4

$$L_1=\{w\in\{a,b\}^* \mid |w|=3k+1;\ k\in N\}$$



Vstupné slovo: abba NASTAV

$DFA_1=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2\}$

$\Sigma=\{a,b\}$

$\delta\ (q_0,a)=q_1$

$\delta\ (q_0,b)=q_1$

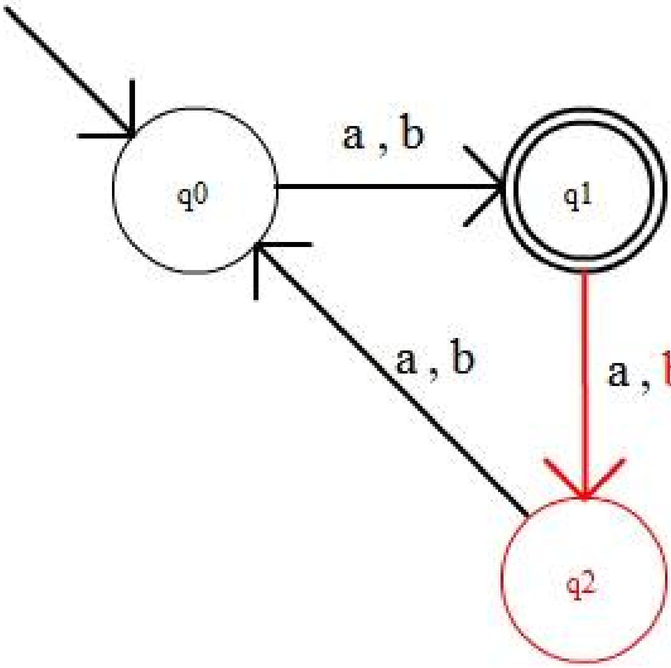
$\delta\ (q_1,a)=q_2$

$\delta\ (q_1,b)=q_2$

$\delta\ (q_2,a)=q_0$

$\delta\ (q_2,b)=q_0$

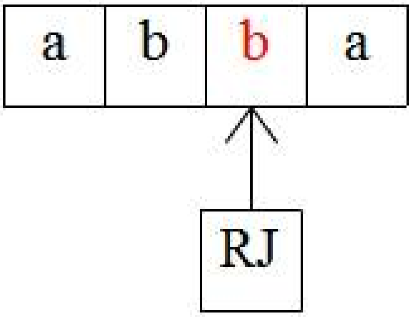
$F=\{q_1\}$



POMOC KROK ANIMACIA

Frame 7

$$L_1=\{w\in\{a,b\}^* \mid |w|=3k+1;\ k\in N\}$$



Vstupné slovo: abba NASTAV

$DFA_1=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2\}$

$\Sigma=\{a,b\}$

$\delta\,(q_0,a)=q_1$

$\delta\,(q_0,b)=q_1$

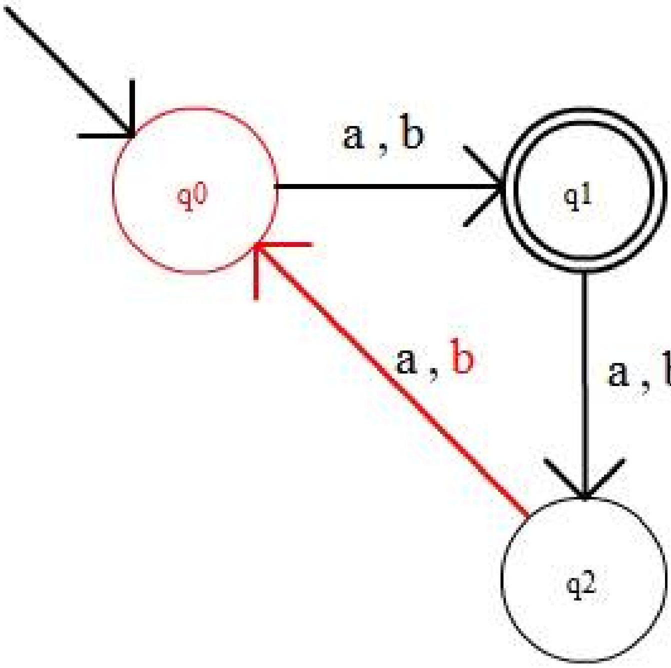
$\delta\,(q_1,a)=q_2$

$\delta\,(q_1,b)=q_2$

$\delta\,(q_2,a)=q_0$

$\delta\,(q_2,b)=q_0$

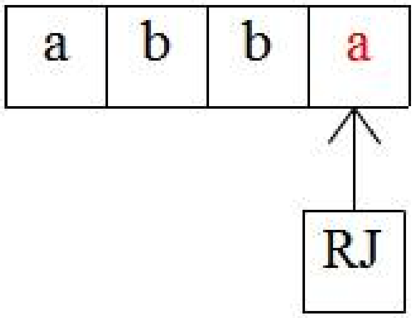
$F=\{q_1\}$



POMOC KROK ANIMACIA

Frame 10

$$L_1 = \{ w \in \{a,b\}^* \mid |w| = 3k+1;\ k \in N \}$$



Vstupné slovo: abba NASTAV

$DFA_1 = (K, \Sigma, \delta, q_0, F)$

$K = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

$\delta\ (q_0, a) = q_1$

$\delta\ (q_0, b) = q_1$

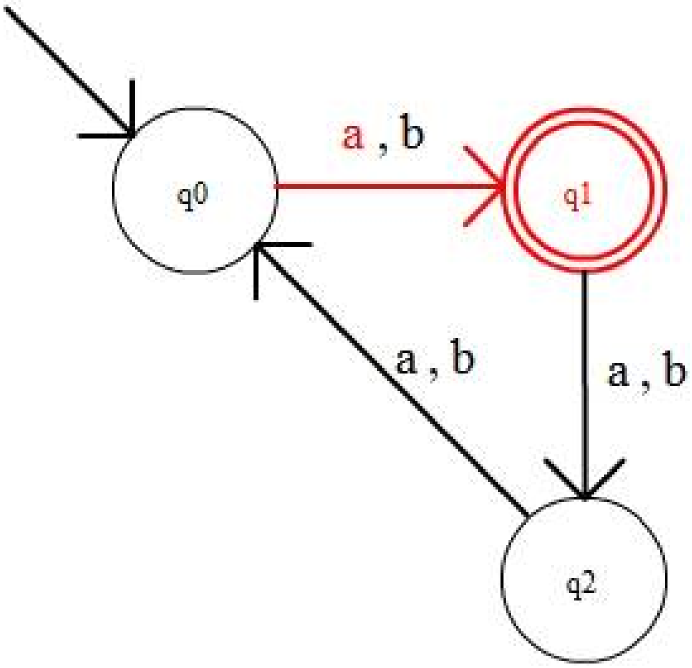
$\delta\ (q_1, a) = q_2$

$\delta\ (q_1, b) = q_2$

$\delta\ (q_2, a) = q_0$

$\delta\ (q_2, b) = q_0$

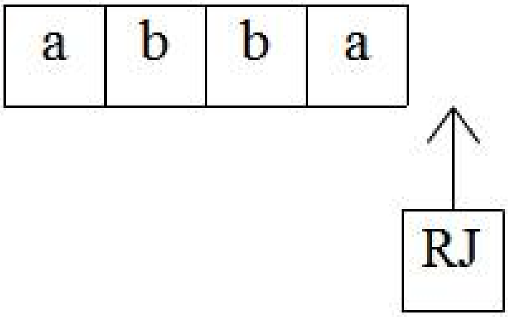
$F = \{q_1\}$



POMOC KROK ANIMACIA

Frame 13

$L_1 = \{ w \in \{a,b\}^* \mid |w| = 3k+1;\ k \in N \}$



Vstupné slovo: abba NASTAV

$DFA_1 = (K, \Sigma, \delta, q_0, F)$

$K = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

$\delta\,(q_0, a) = q_1$

$\delta\,(q_0, b) = q_1$

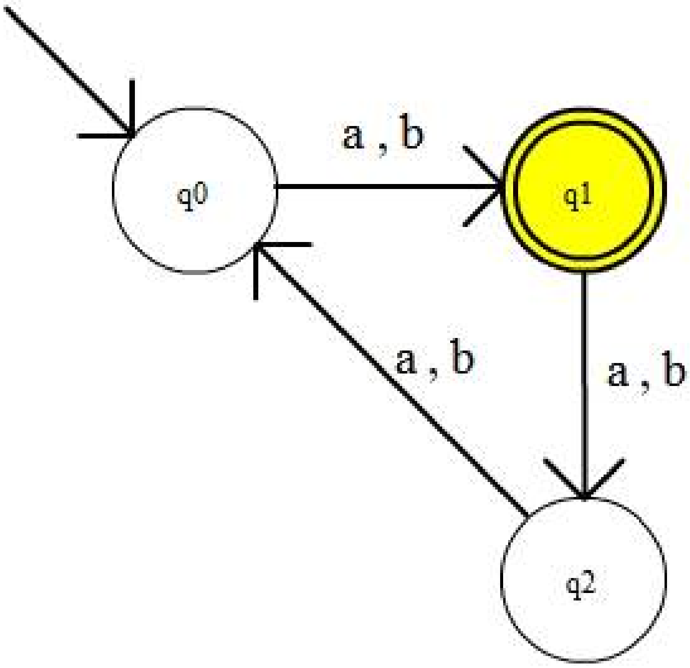
$\delta\,(q_1, a) = q_2$

$\delta\,(q_1, b) = q_2$

$\delta\,(q_2, a) = q_0$

$\delta\,(q_2, b) = q_0$

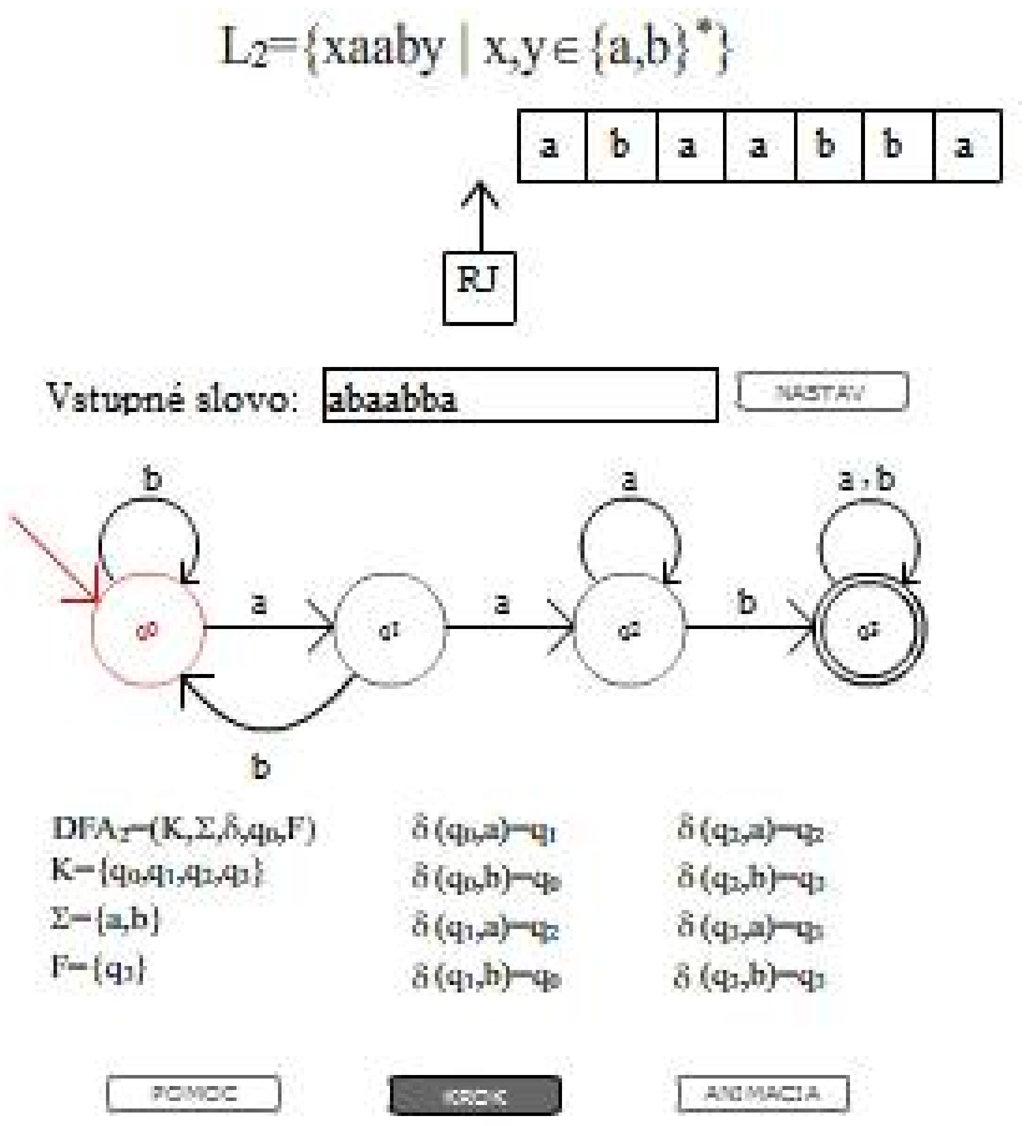
$F = \{q_1\}$



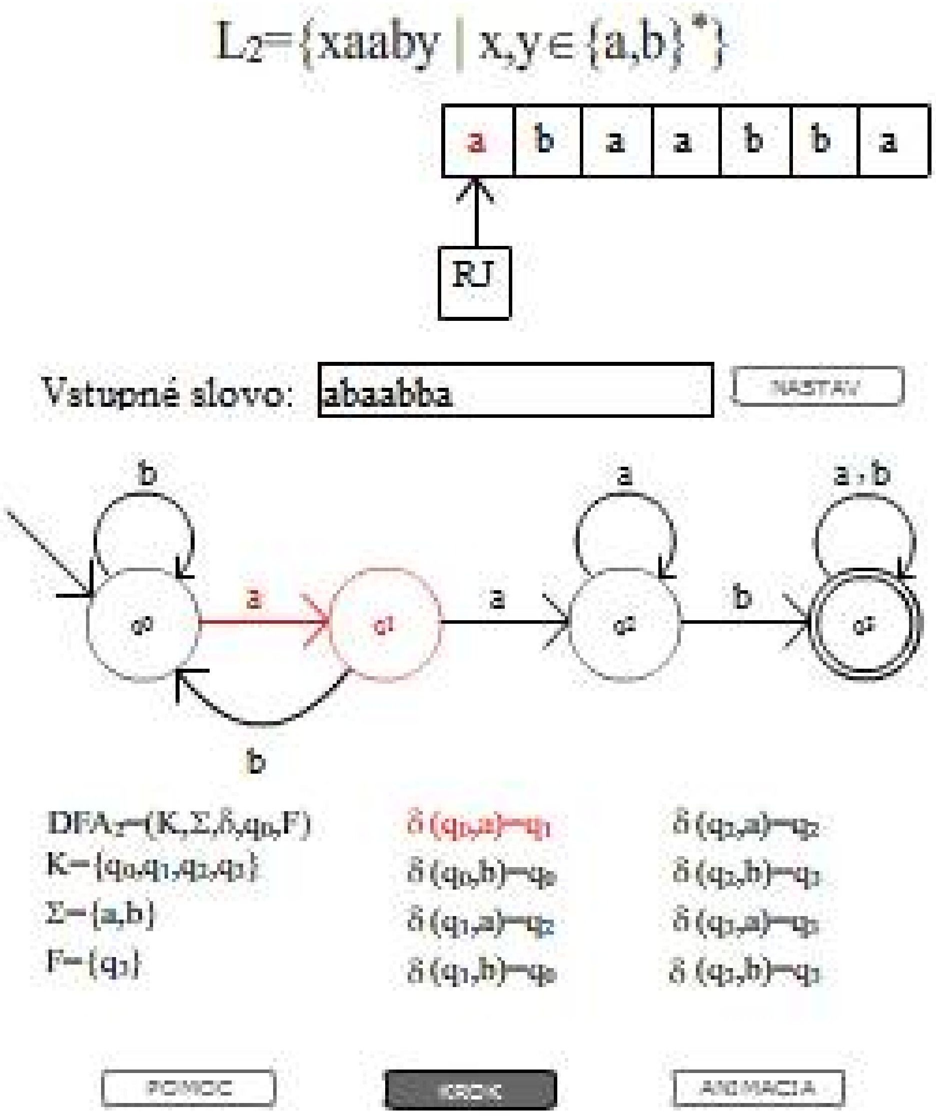
Automat skončil v stave q1 => AKCEPTUJEM

POMOC KROK ANIMACIA

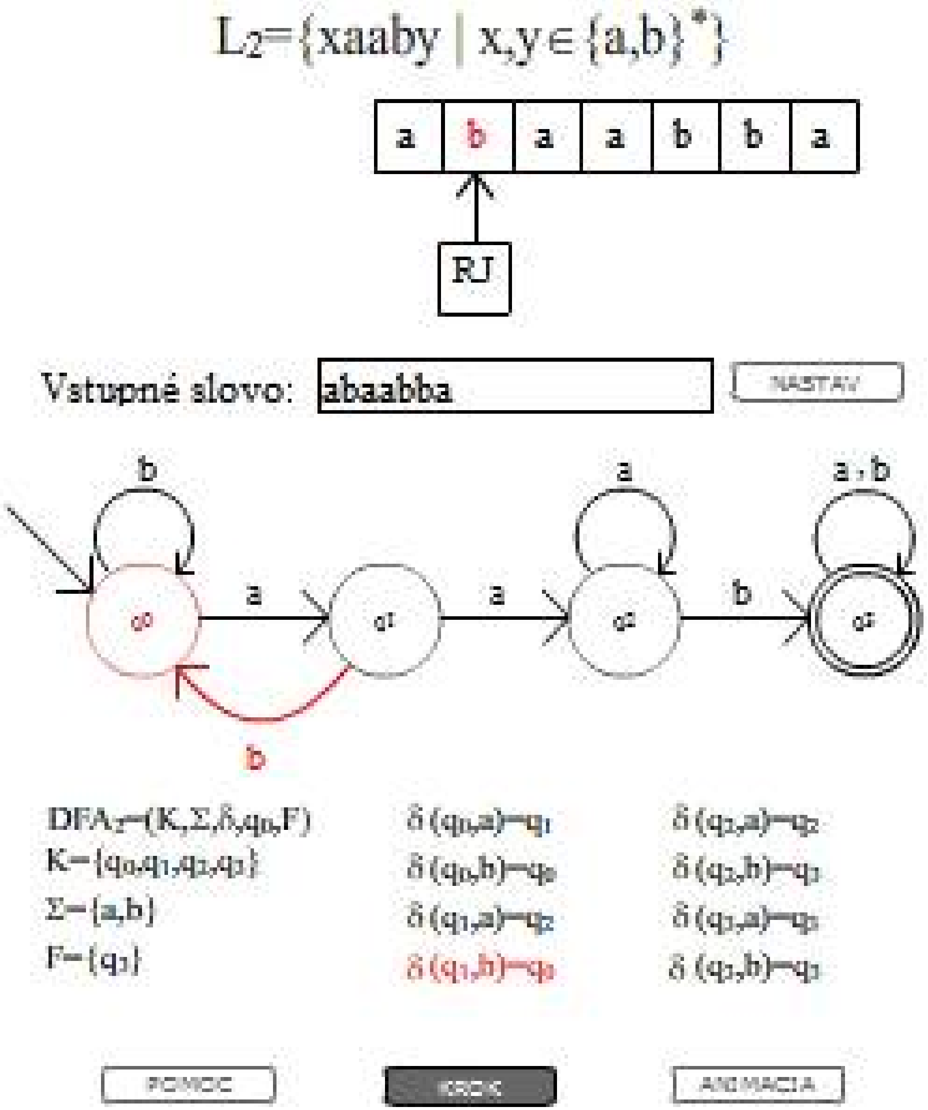
Frame 16


L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo:
abaabba
NASTAV
b
a
a
a
b
a , b
q0
q1
q2
q3
b
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ (q0,a)=q1
δ (q0,b)=q0
δ (q1,a)=q2
δ (q1,b)=q0
δ (q2,a)=q2
δ (q2,b)=q3
δ (q3,a)=q3
δ (q3,b)=q3
POMOC
KROK
ANIMACIA

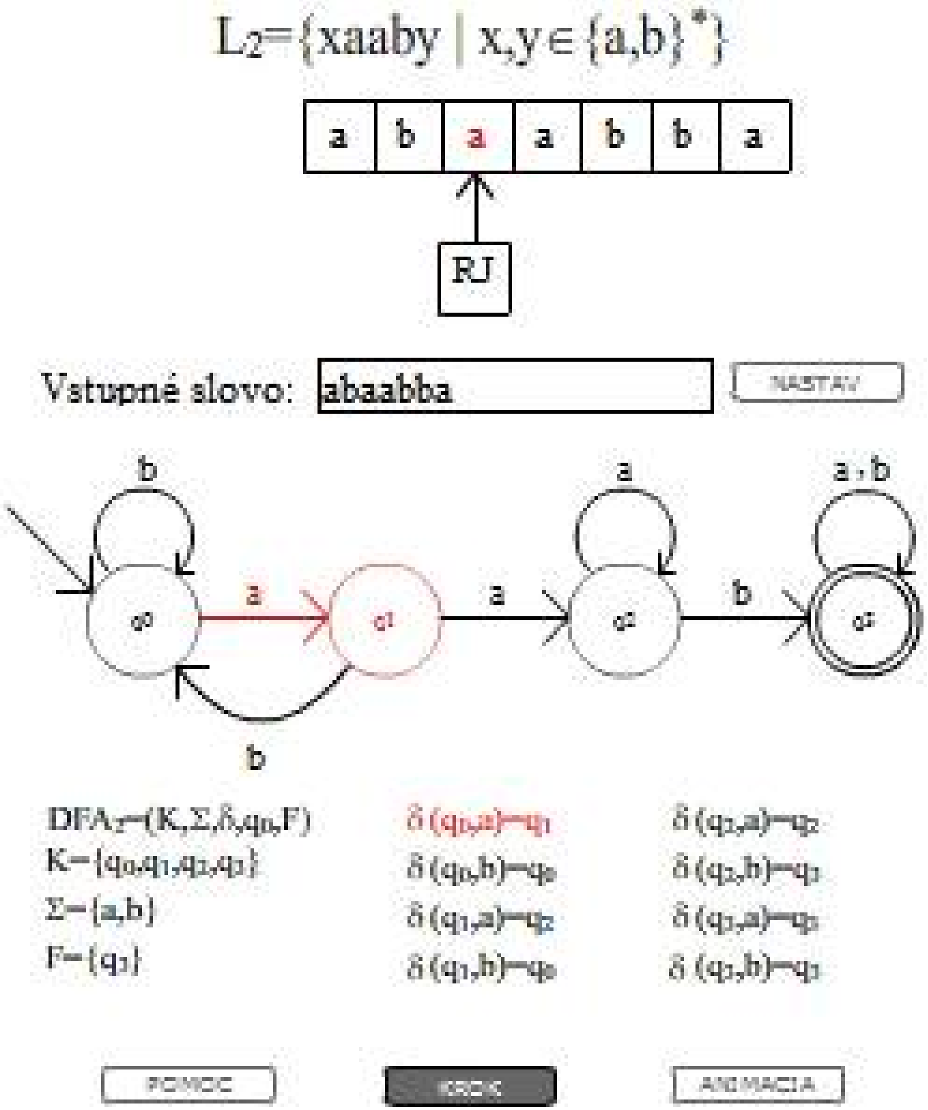
Frame 2


$L_2=\{xaaby \mid x,y \in \{a,b\}^*\}$
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a
a, b
q0
q1
q2
q3
a
b
b
$DFA_2=(K,\Sigma,\delta,q_0,F)$
$K=\{q_0,q_1,q_2,q_3\}$
$\Sigma=\{a,b\}$
$F=\{q_3\}$
$\delta(q_0,a)=q_1$
$\delta(q_0,b)=q_0$
$\delta(q_1,a)=q_2$
$\delta(q_1,b)=q_0$
$\delta(q_2,a)=q_2$
$\delta(q_2,b)=q_3$
$\delta(q_3,a)=q_3$
$\delta(q_3,b)=q_3$
POMOC
KROK
ANIMACIA

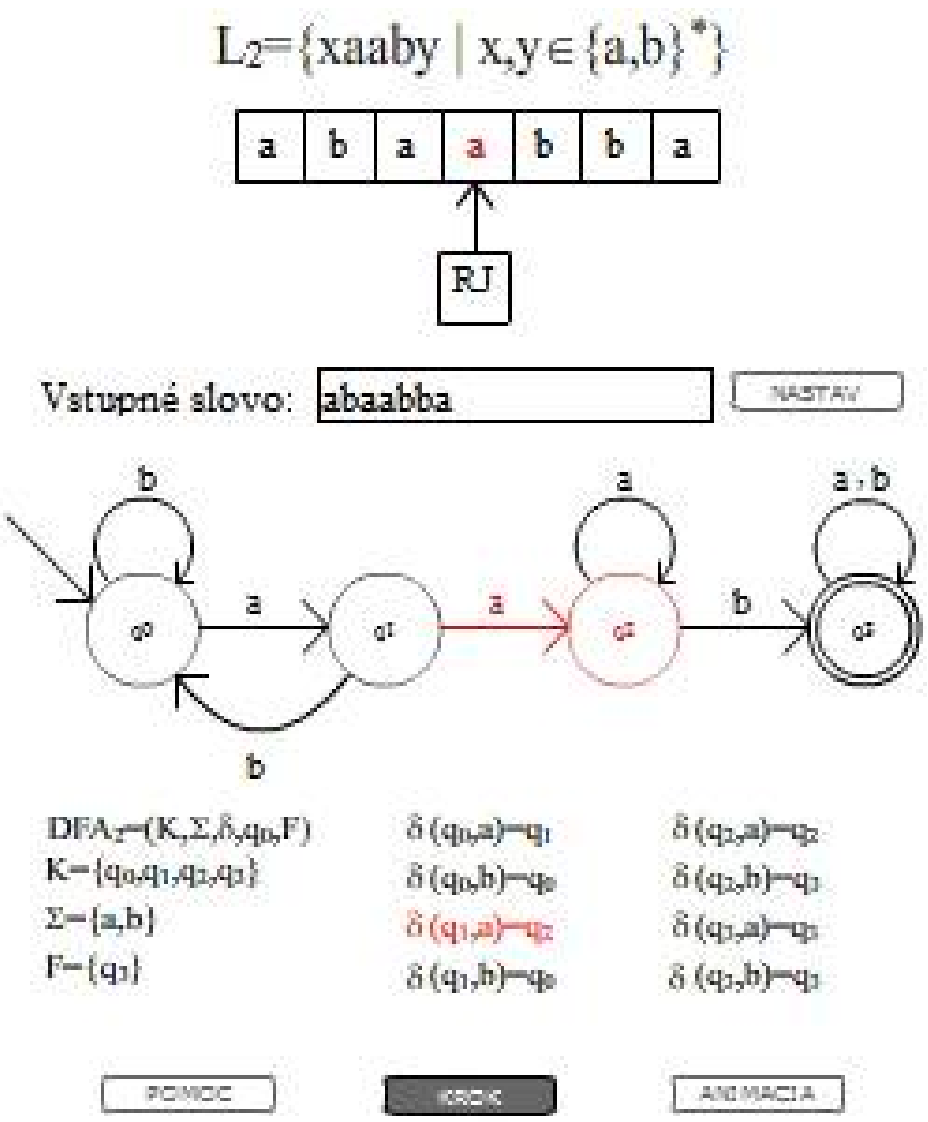
Frame 3


$L_2=\{xaaby \mid x,y \in \{a,b\}^*\}$
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a
a , b
a
a
b
q_0
q_1
q_2
q_3
b
$DFA_2=(K,\Sigma,\delta,q_0,F)$
$K=\{q_0,q_1,q_2,q_3\}$
$\Sigma=\{a,b\}$
$F=\{q_3\}$
$\delta(q_0,a)=q_1$
$\delta(q_0,b)=q_0$
$\delta(q_1,a)=q_2$
$\delta(q_1,b)=q_0$
$\delta(q_2,a)=q_2$
$\delta(q_2,b)=q_3$
$\delta(q_3,a)=q_3$
$\delta(q_3,b)=q_3$
POMOC
KROK
ANIMACIA

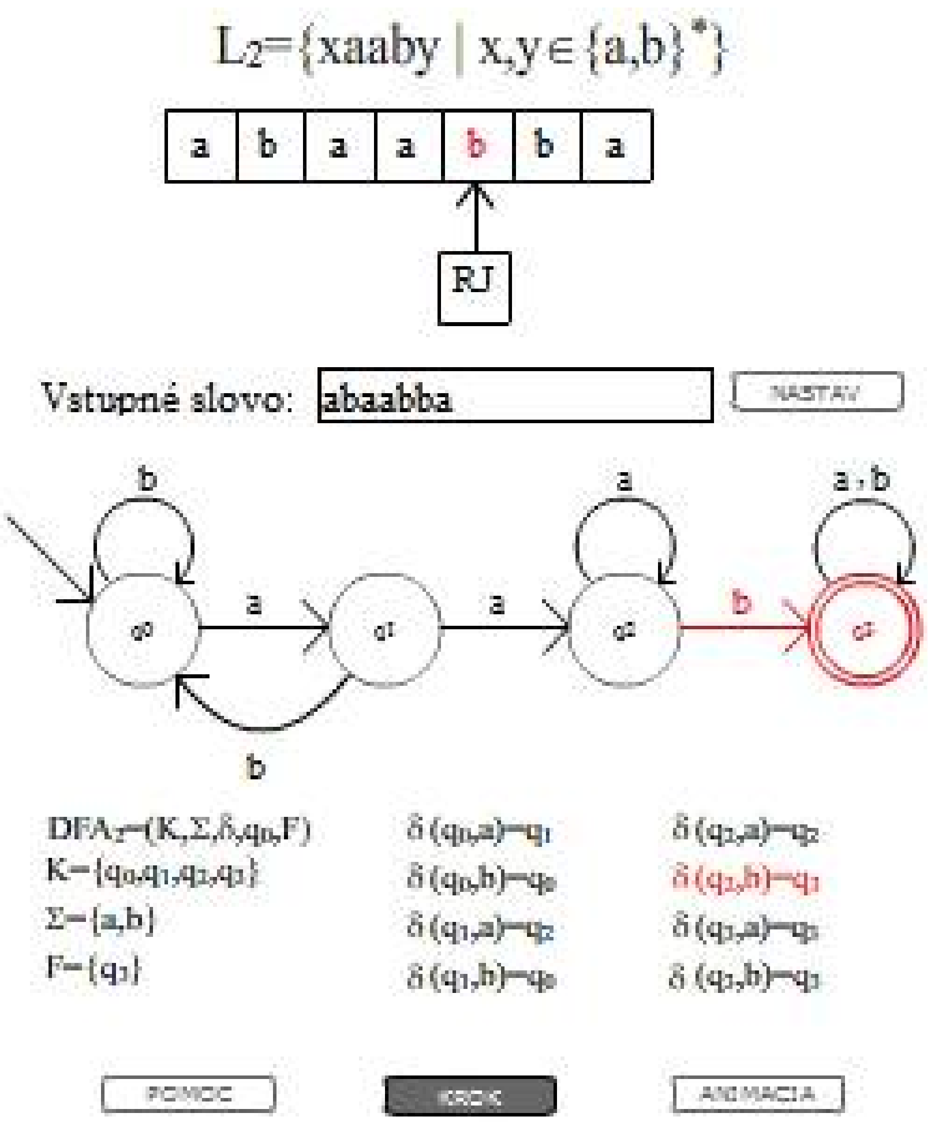
Frame 6


$L_2=\{xaaby \mid x,y \in \{a,b\}^*\}$
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a
b
a , b
b
$DFA_2=(K,\Sigma,\delta,q_0,F)$
$K=\{q_0,q_1,q_2,q_3\}$
$\Sigma=\{a,b\}$
$F=\{q_3\}$
$\delta(q_0,a)=q_1$
$\delta(q_0,b)=q_0$
$\delta(q_1,a)=q_2$
$\delta(q_1,b)=q_0$
$\delta(q_2,a)=q_2$
$\delta(q_2,b)=q_3$
$\delta(q_3,a)=q_3$
$\delta(q_3,b)=q_3$
POMOC
KROK
ANIMACIA

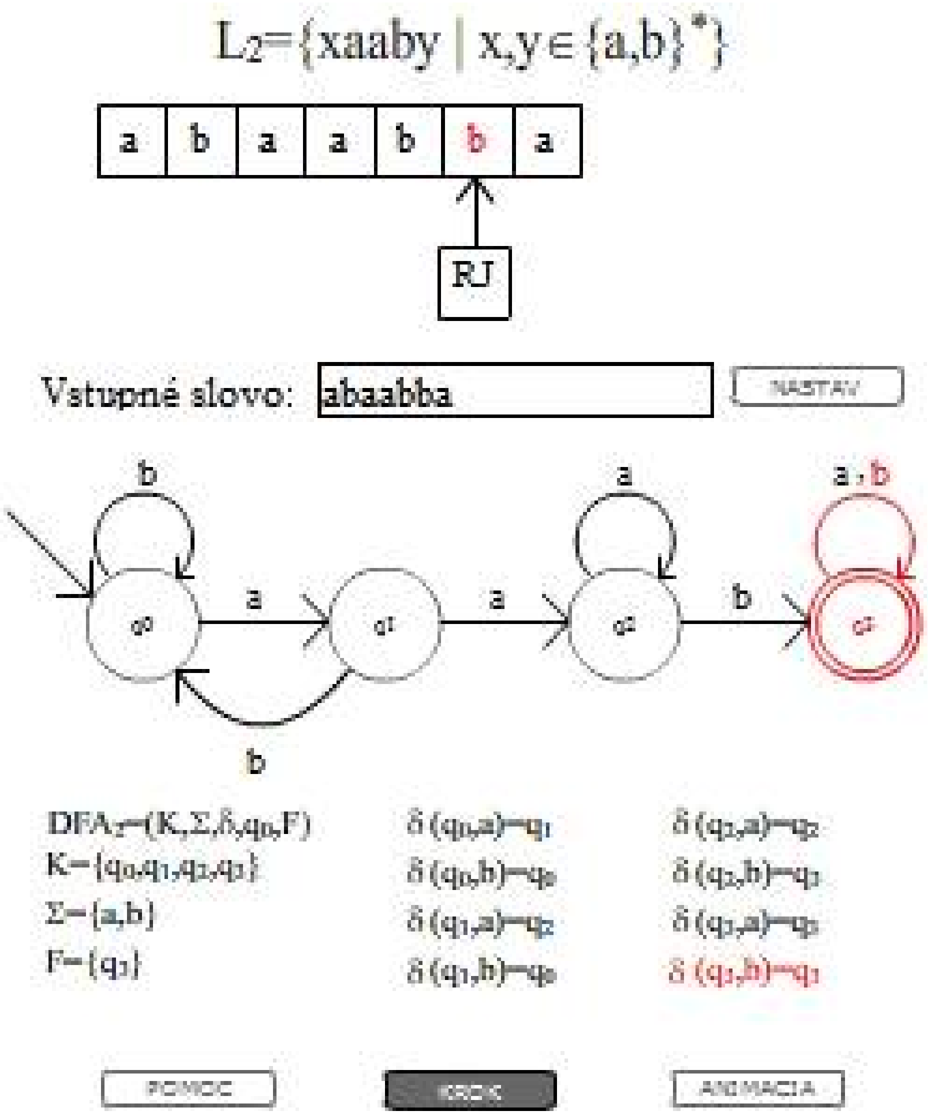
Frame 9


L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a
a , b
a
b
b
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ(q0,a)=q1
δ(q0,b)=q0
δ(q1,a)=q2
δ(q1,b)=q0
δ(q2,a)=q2
δ(q2,b)=q3
δ(q3,a)=q3
δ(q3,b)=q3
POMOC
KROK
ANIMACIA

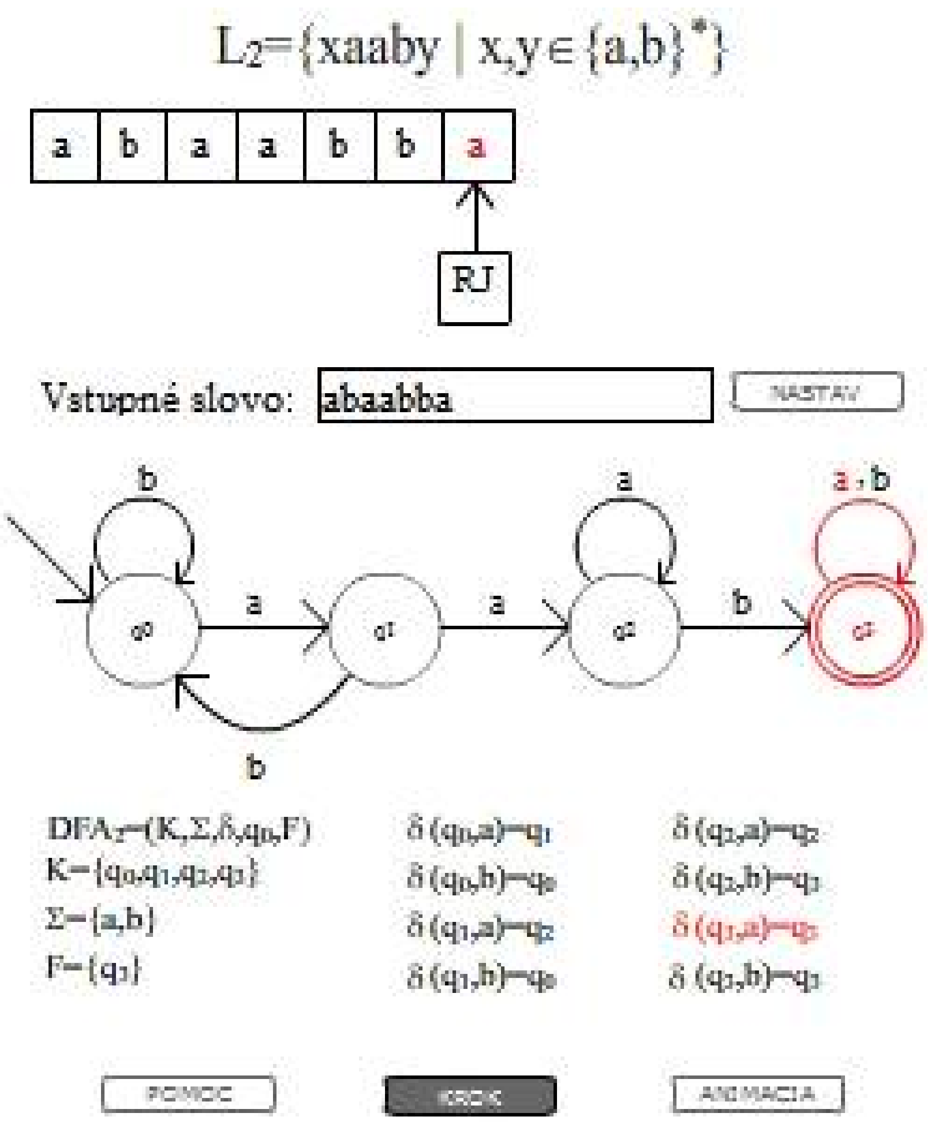
Frame 12


L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo:
abaabba
NASTAV
b
a
a, b
q0
q1
q2
q3
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ (q0,a)=q1
δ (q0,b)=q0
δ (q1,a)=q2
δ (q1,b)=q0
δ (q2,a)=q2
δ (q2,b)=q3
δ (q3,a)=q3
δ (q3,b)=q3
POMOC
KROK
ANIMACIA

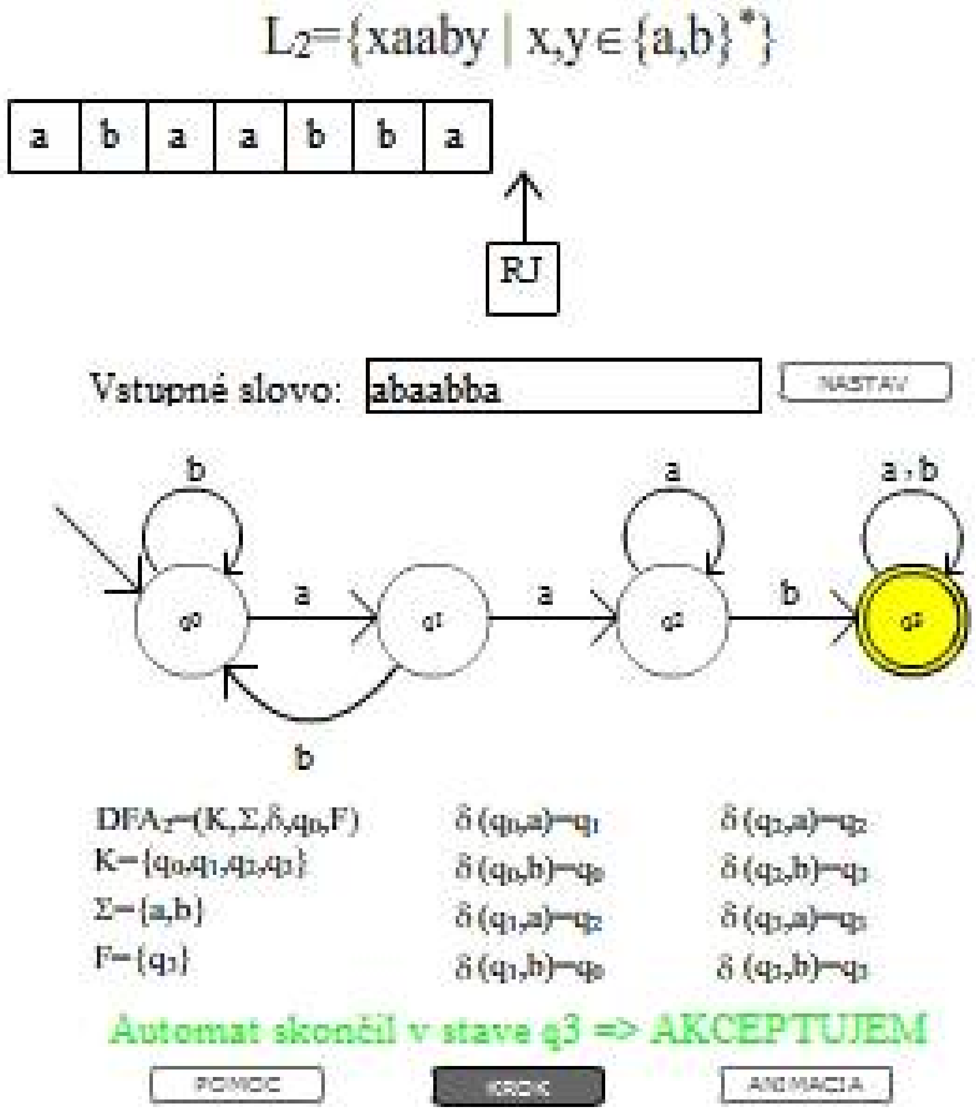
Frame 15


L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo:
abaabba
NASTAV
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ(q0,a)=q1
δ(q0,b)=q0
δ(q1,a)=q2
δ(q1,b)=q0
δ(q2,a)=q2
δ(q2,b)=q3
δ(q3,a)=q3
δ(q3,b)=q3
POMOC
KROK
ANIMACIA

Frame 18

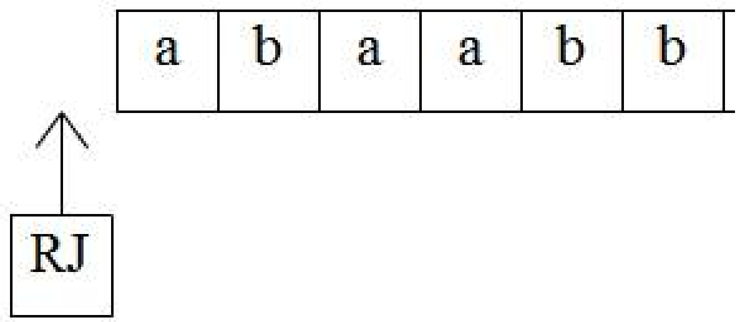

L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a
a
b
a , b
b
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ (q0,a)=q1
δ (q0,b)=q0
δ (q1,a)=q2
δ (q1,b)=q0
δ (q2,a)=q2
δ (q2,b)=q3
δ (q3,a)=q3
δ (q3,b)=q3
POMOC
KROK
ANIMACIA

Frame 21


L2={xaaby | x,y∈{a,b}*}
a b a a b b a
RJ
Vstupné slovo: abaabba
NASTAV
b
a
a, b
a
a
b
b
DFA2=(K,Σ,δ,q0,F)
K={q0,q1,q2,q3}
Σ={a,b}
F={q3}
δ(q0,a)=q1
δ(q0,b)=q0
δ(q1,a)=q2
δ(q1,b)=q0
δ(q2,a)=q2
δ(q2,b)=q3
δ(q3,a)=q3
δ(q3,b)=q3
Automat skončil v stave q3 => AKCEPTUJEM
POMOC
KROK
ANIMACIA

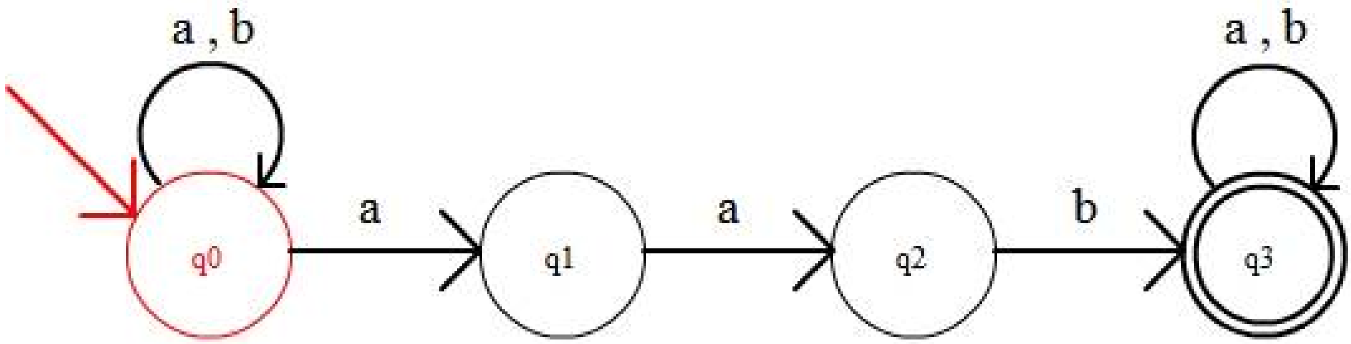
Frame 24

$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$



Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

Frame 2

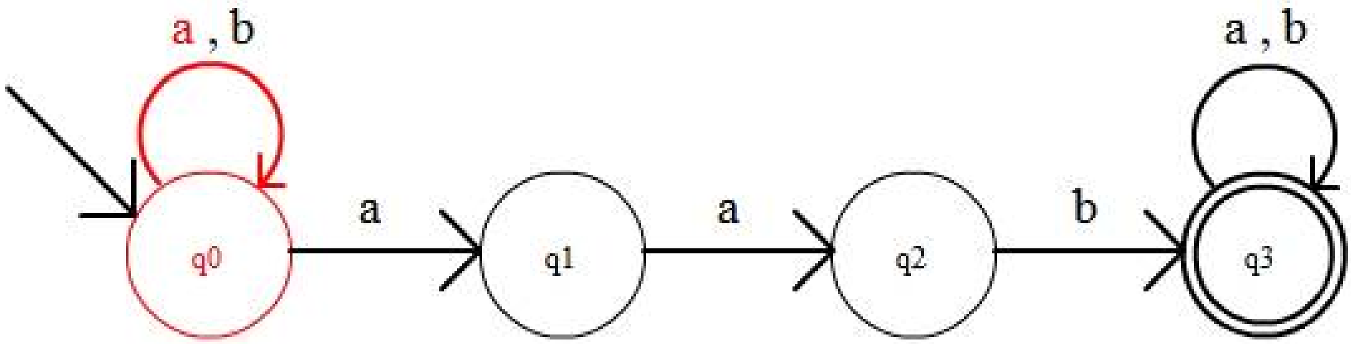
$$L_3=\{xaaby \mid x,y\in\{a,b\}^*\}$$

a	b	a	a	b	b	a

RJ

Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

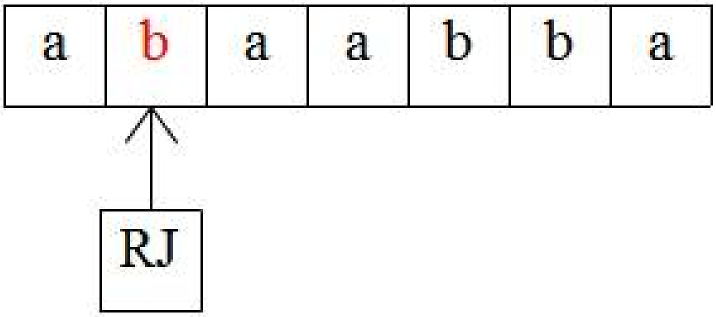
$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

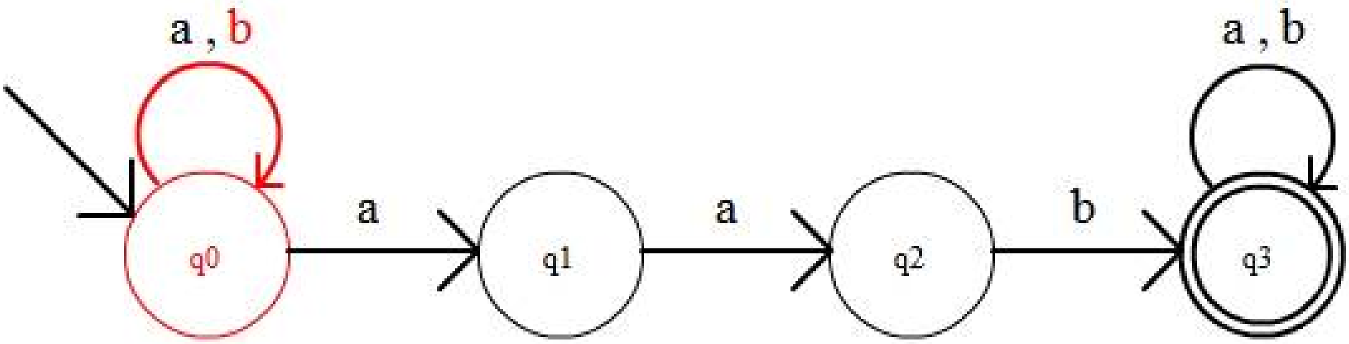
Frame 3

$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$



Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

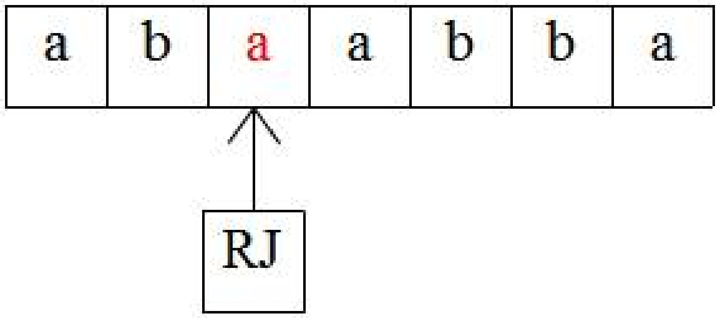
$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

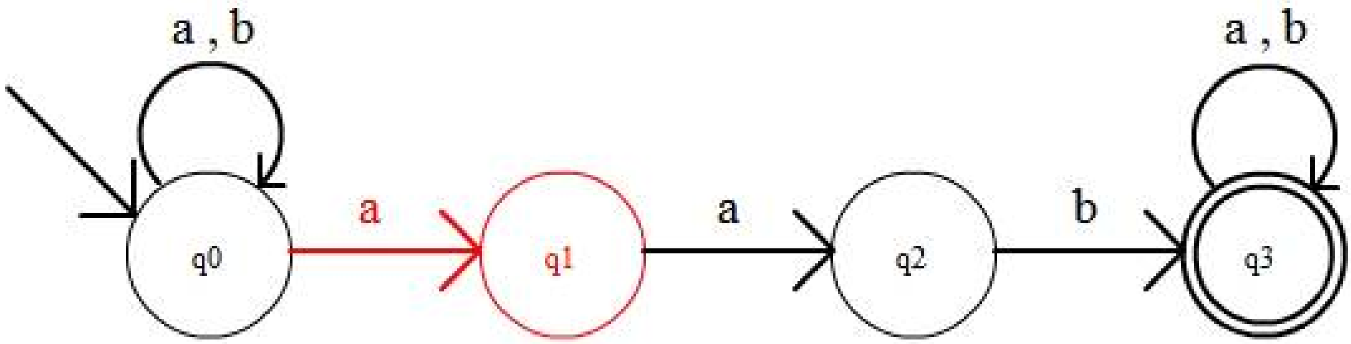
POMOC KROK ANIMACIA

Frame 6

$$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$$



Vstupné slovo: abaabba NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$
$K=\{q_0,q_1,q_2,q_3\}$
$\Sigma=\{a,b\}$
$F=\{q_3\}$

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

$\delta(q_2,b)=q_3$
$\delta(q_3,a)=q_3$
$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

Frame 9

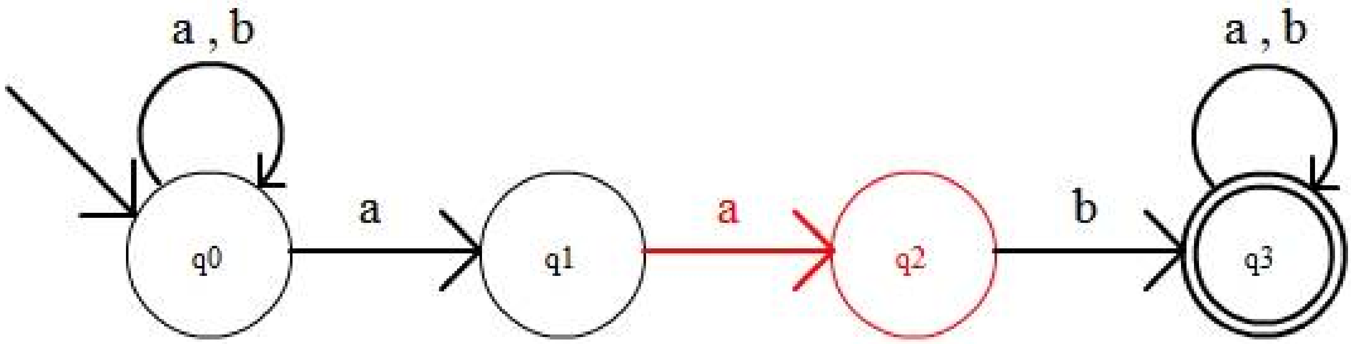
$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$

a	b	a	a	b	b	a

RJ

Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

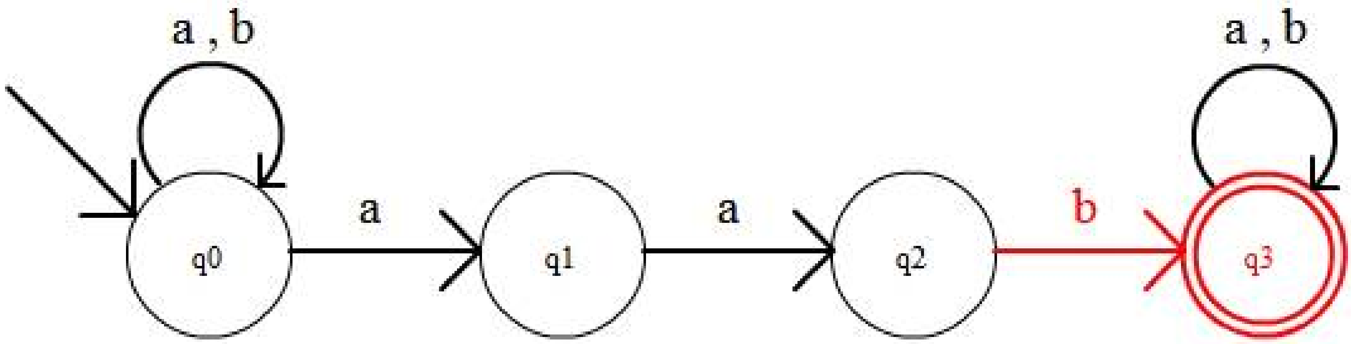
Frame 12

$$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$$

a	b	a	a	b	b	a

RJ

Vstupné slovo: abaabba NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

Frame 15

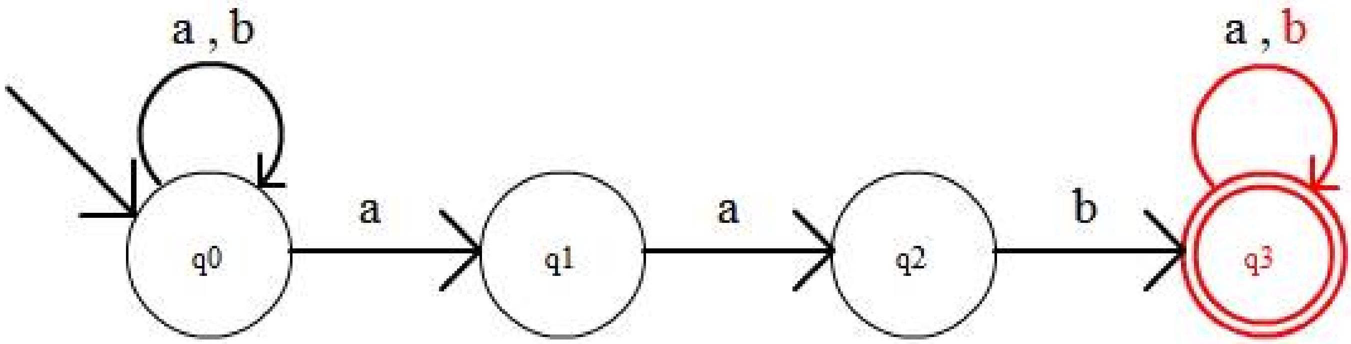
$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$

a	b	a	a	b	b	a

RJ

Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

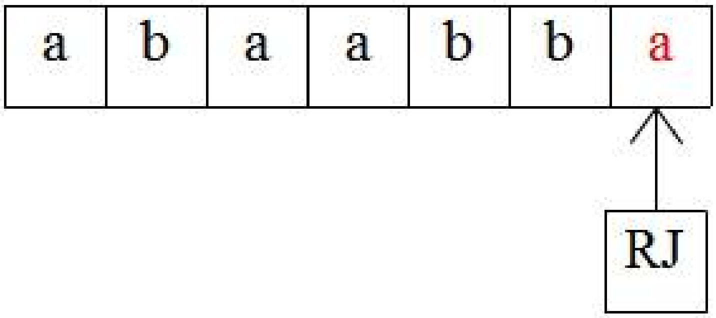
$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

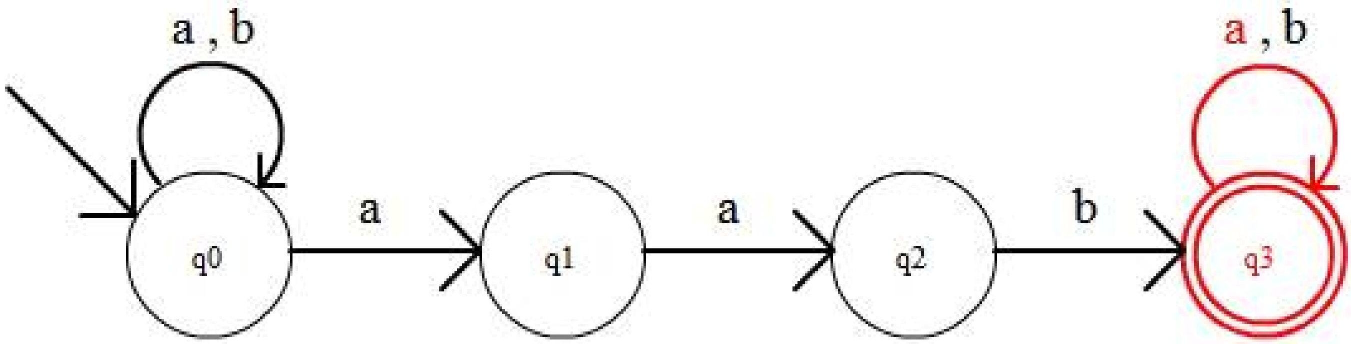
POMOC KROK ANIMACIA

Frame 18

$$L_3=\{xaaby \mid x,y \in \{a,b\}^*\}$$



Vstupné slovo: abaabba NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

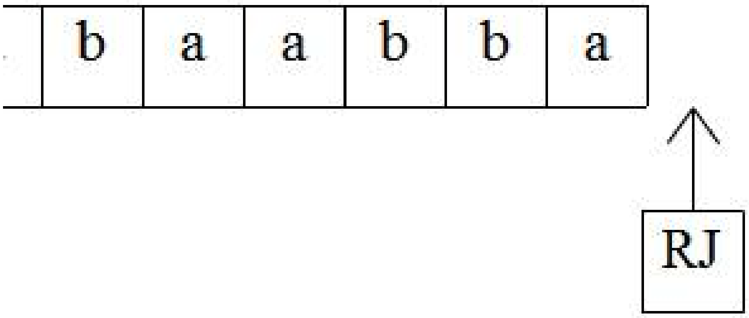
$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

POMOC KROK ANIMACIA

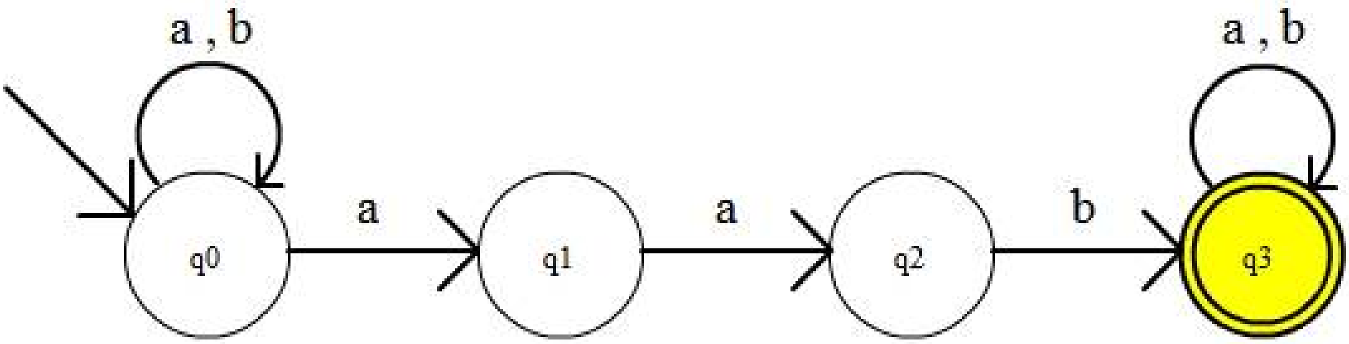
Frame 21

$$L_3=\{xaaby \mid x,y\in\{a,b\}^*\}$$



Vstupné slovo: abaabba

NASTAV



$NFA_3=(K,\Sigma,\delta,q_0,F)$

$K=\{q_0,q_1,q_2,q_3\}$

$\Sigma=\{a,b\}$

$F=\{q_3\}$

$\delta(q_0,a)=\{q_0,q_1\}$

$\delta(q_0,b)=q_0$

$\delta(q_1,a)=q_2$

$\delta(q_2,b)=q_3$

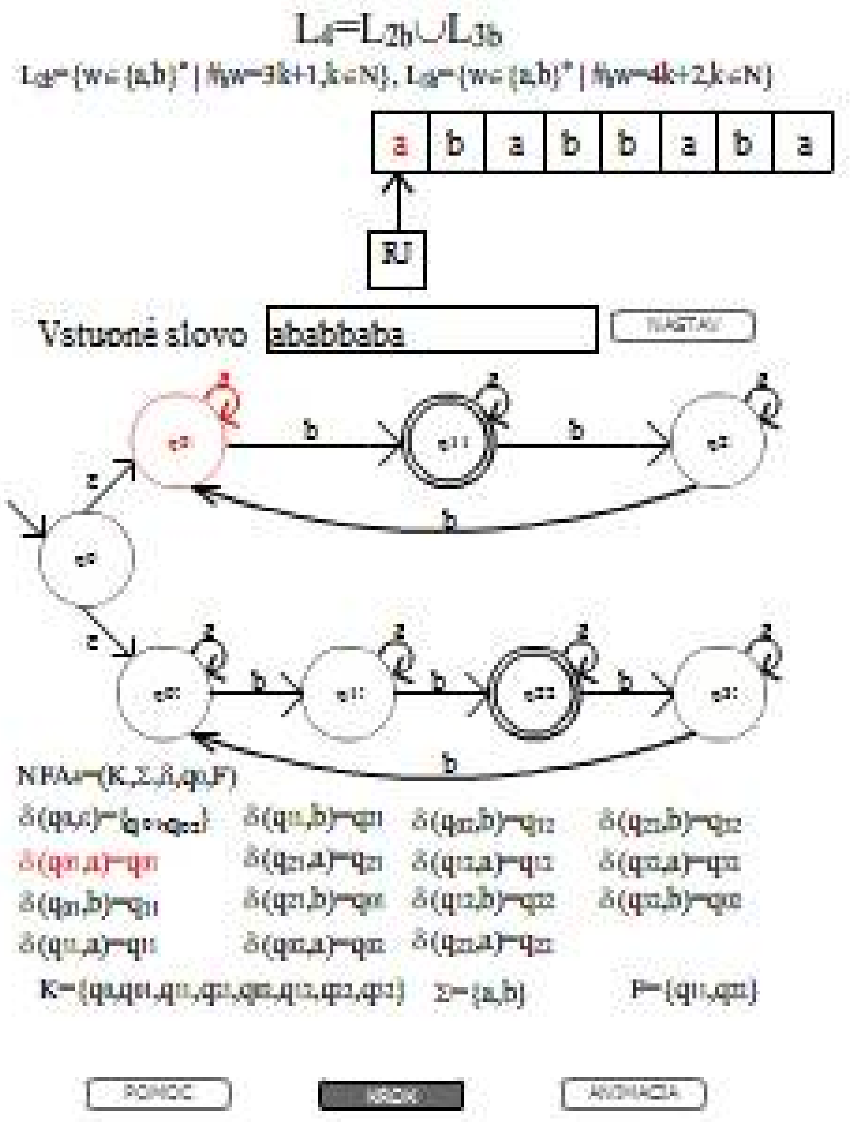
$\delta(q_3,a)=q_3$

$\delta(q_3,b)=q_3$

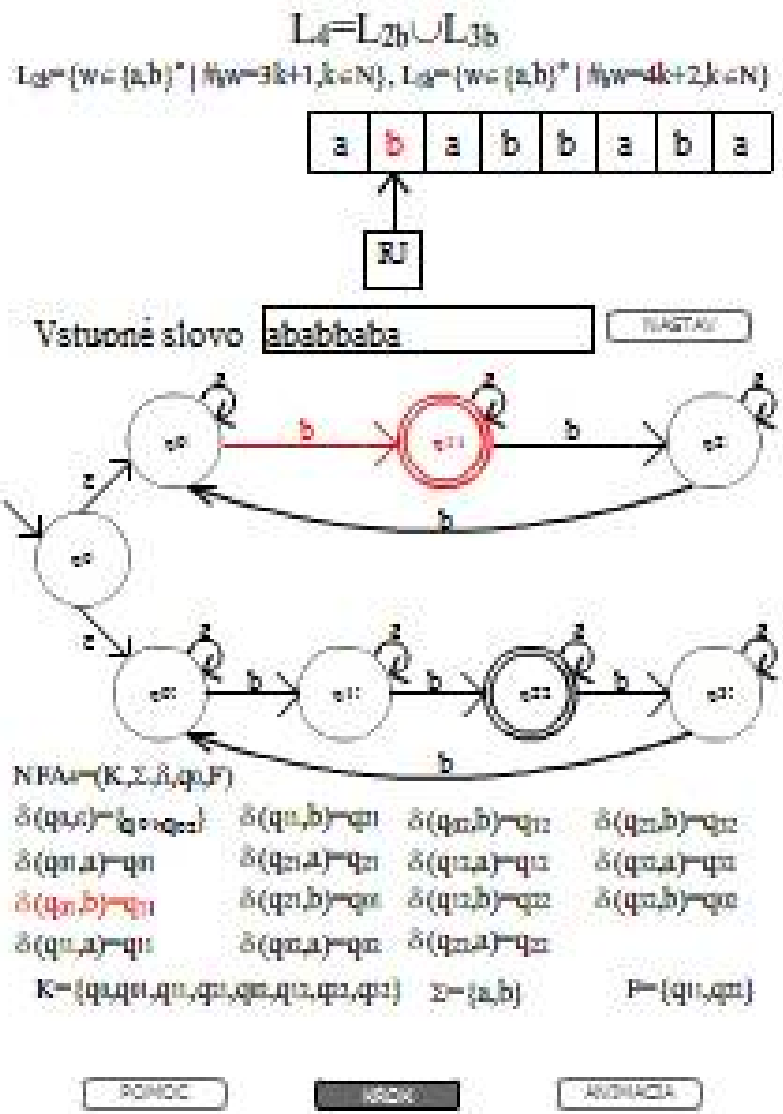
Automat skončil v stave q3 => AKCEPTUJEM

POMOC KROK ANIMACIA

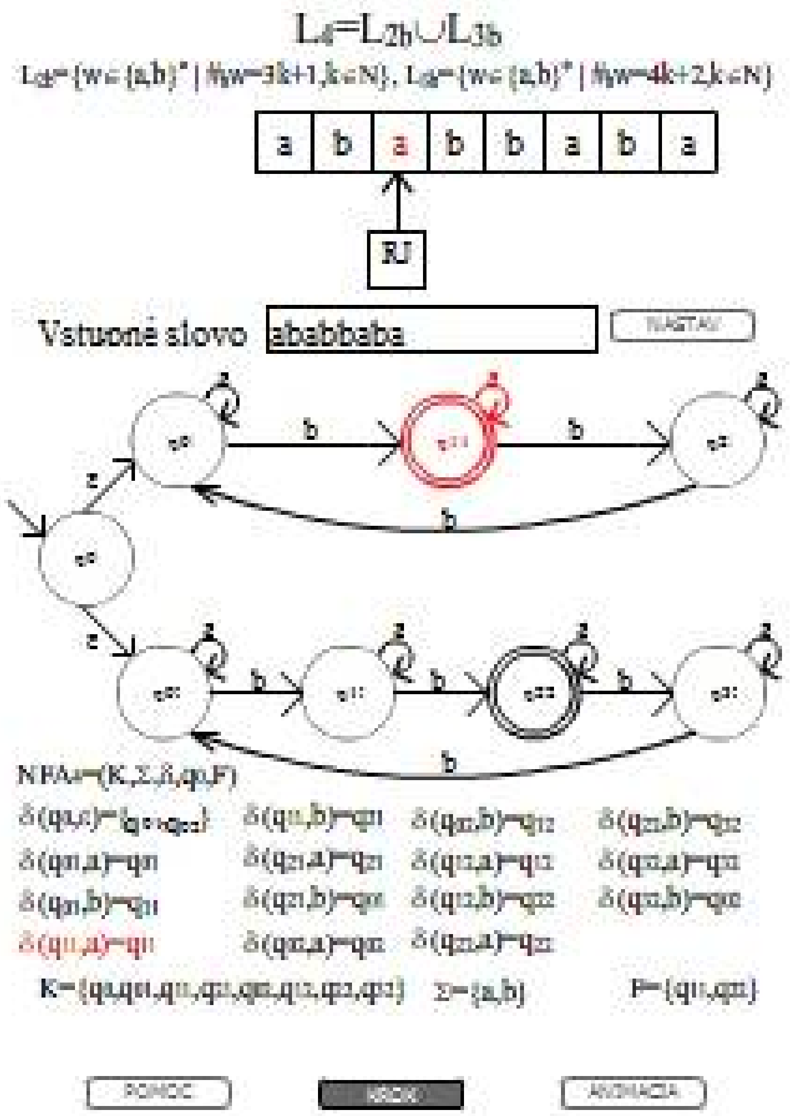
Frame 24



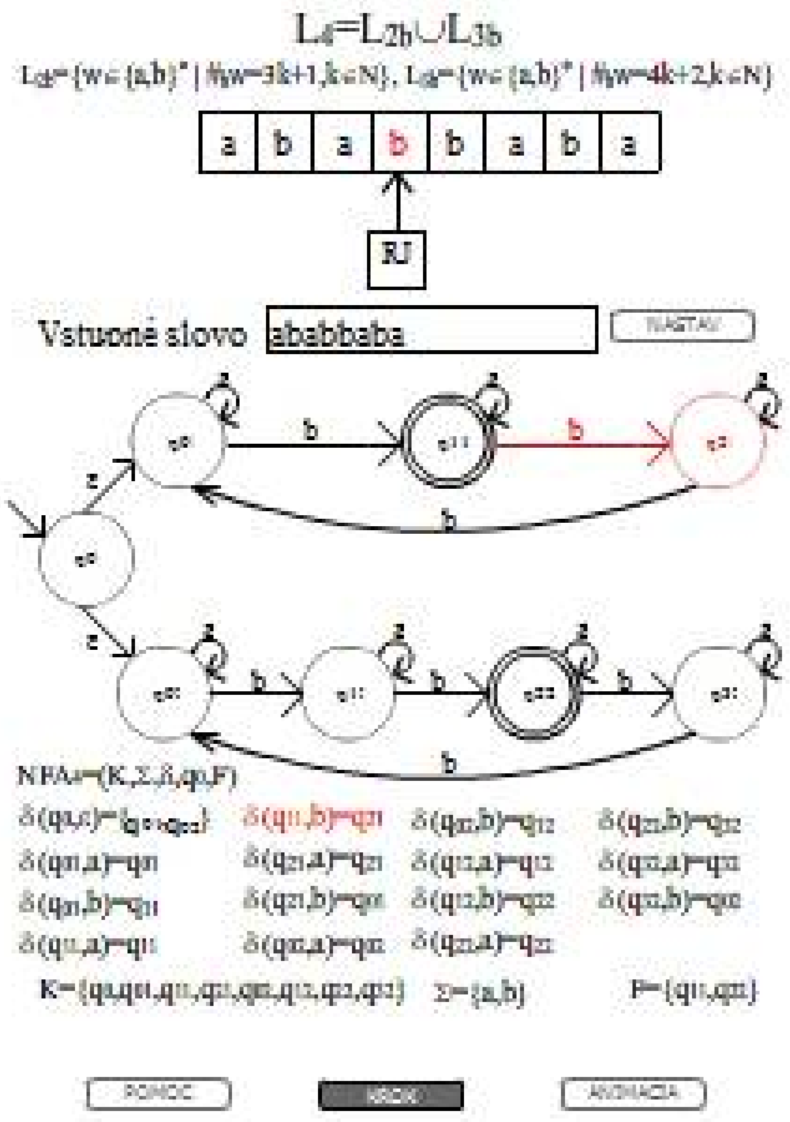
Frame 2



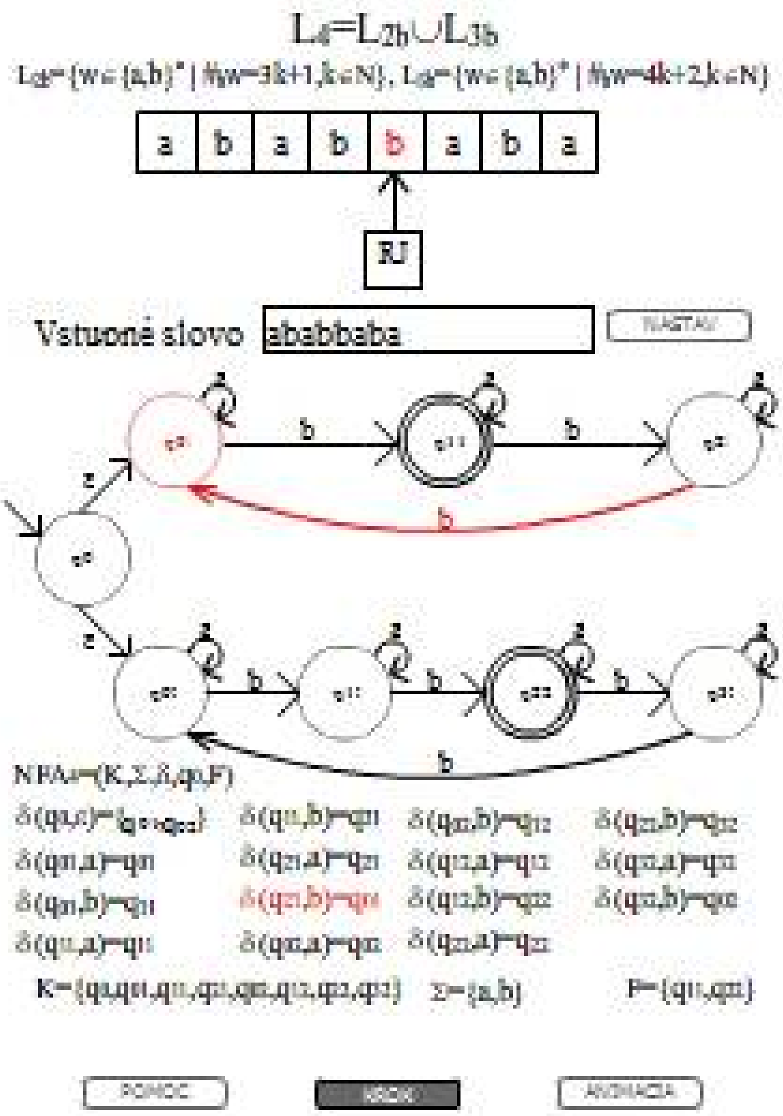
Frame 3



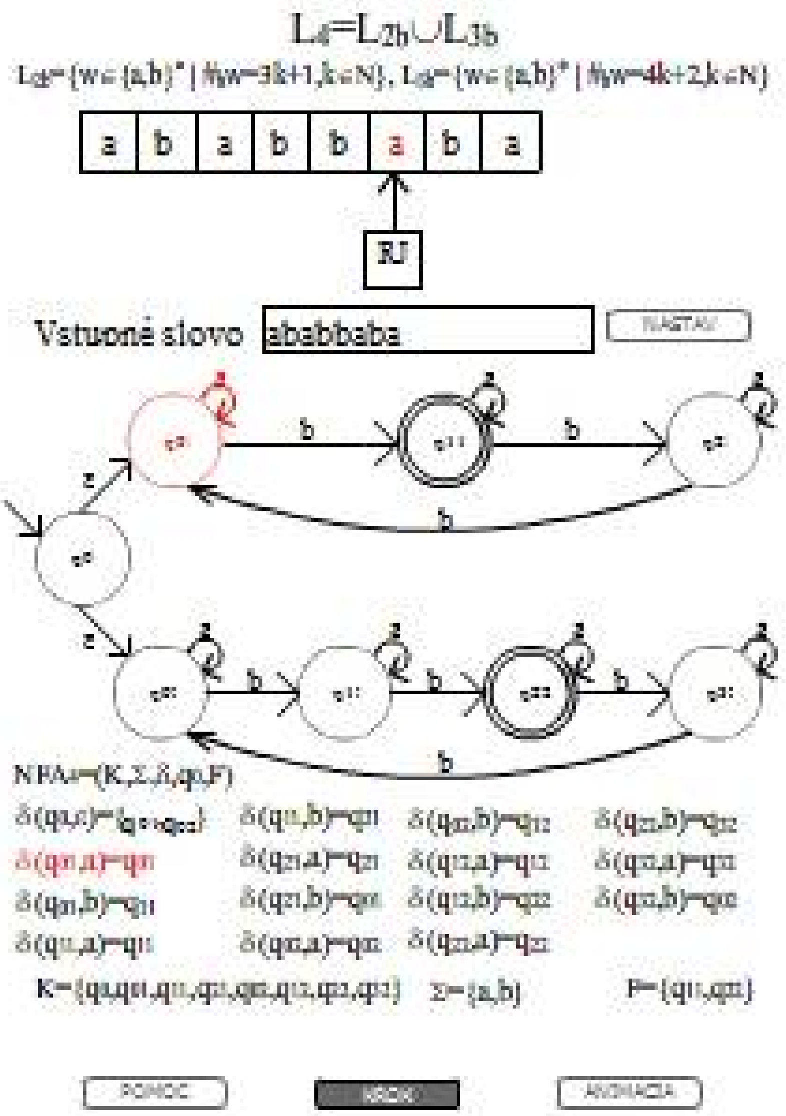
Frame 5



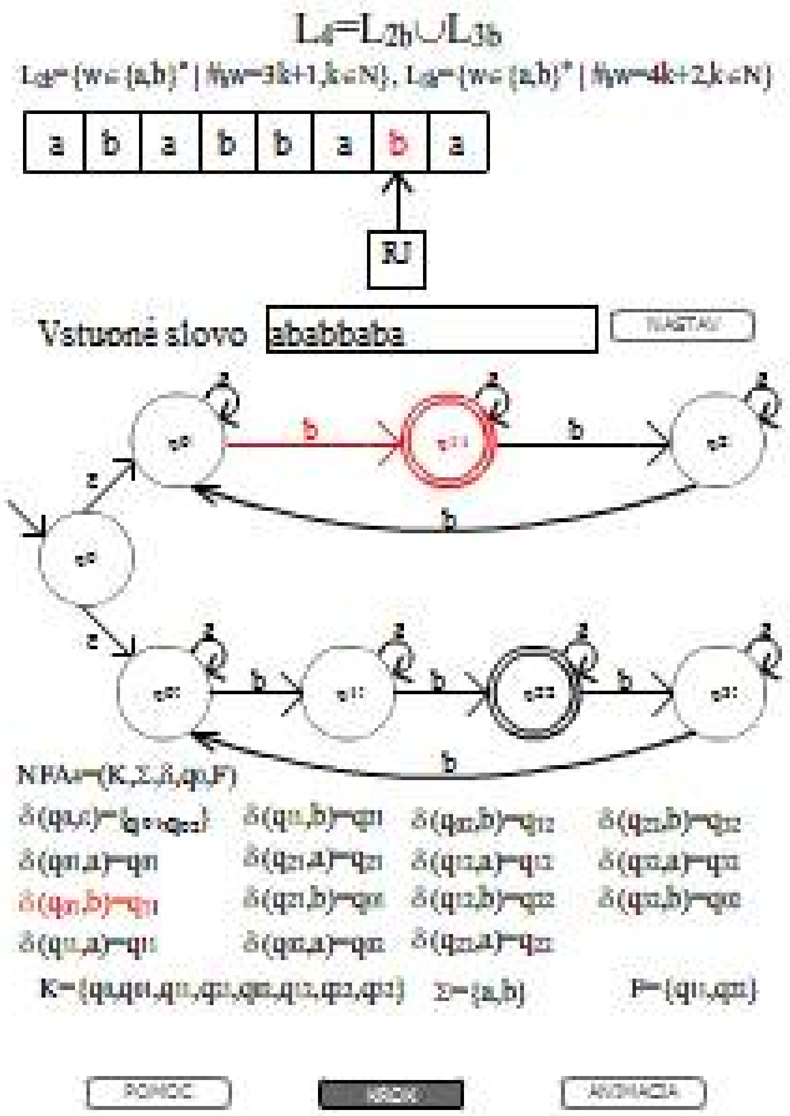
Frame 7



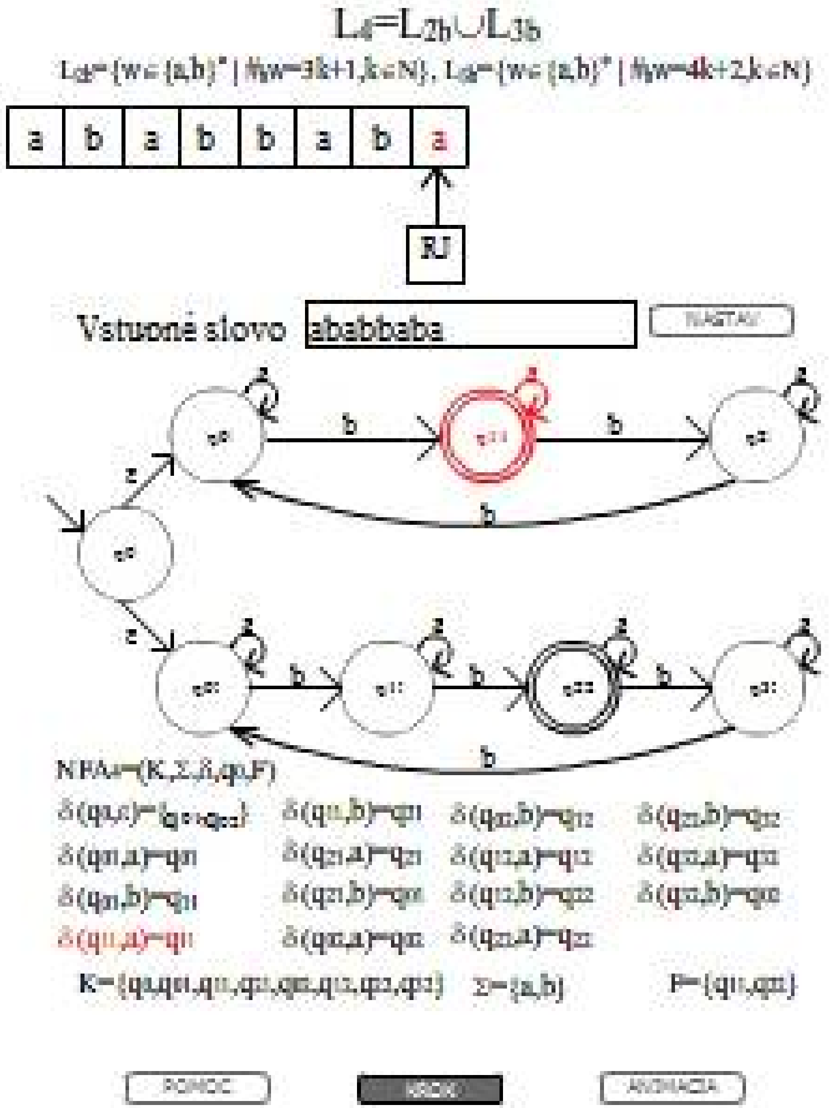
Frame 9



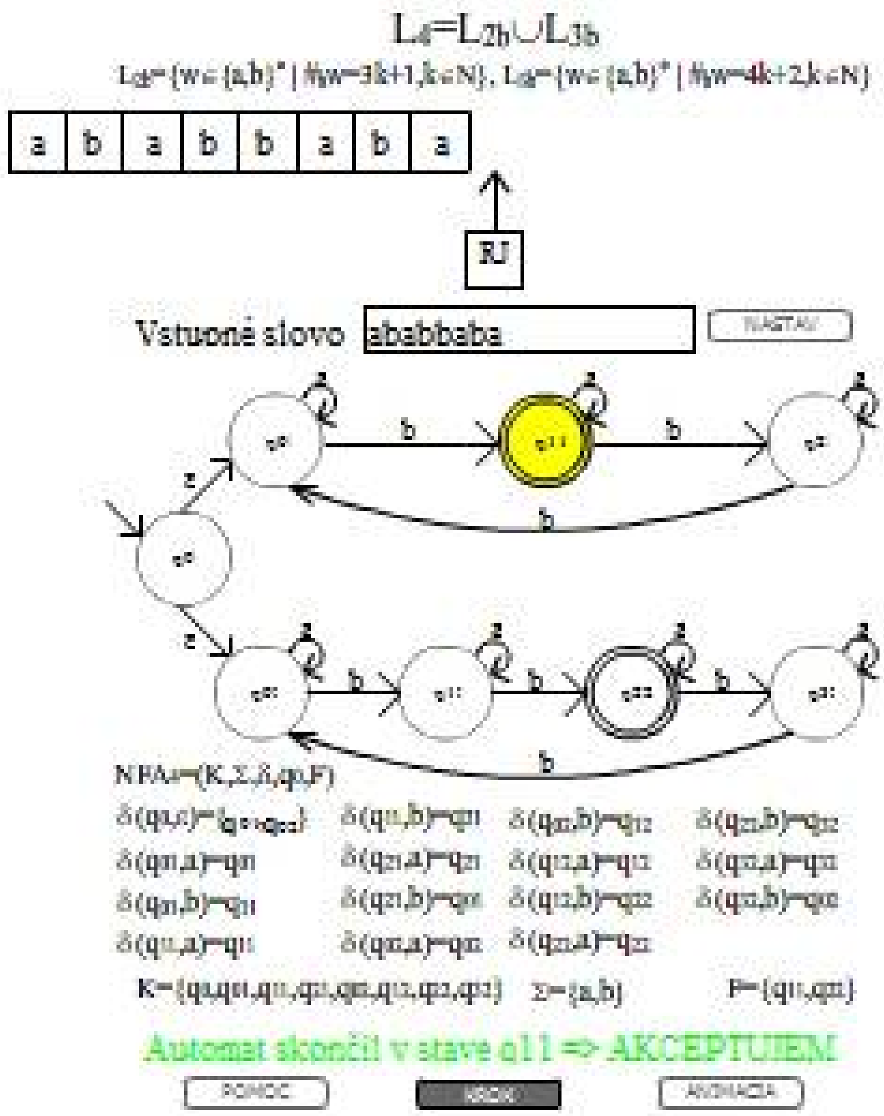
Frame 11



Frame 13



Frame 15



a b a b b a b a
RJ
Vstupné slovo
ababbaba
NFA=(K,Σ,δ,q0,F)
Automat skončil v stave q11 => AKCEPTUJEM

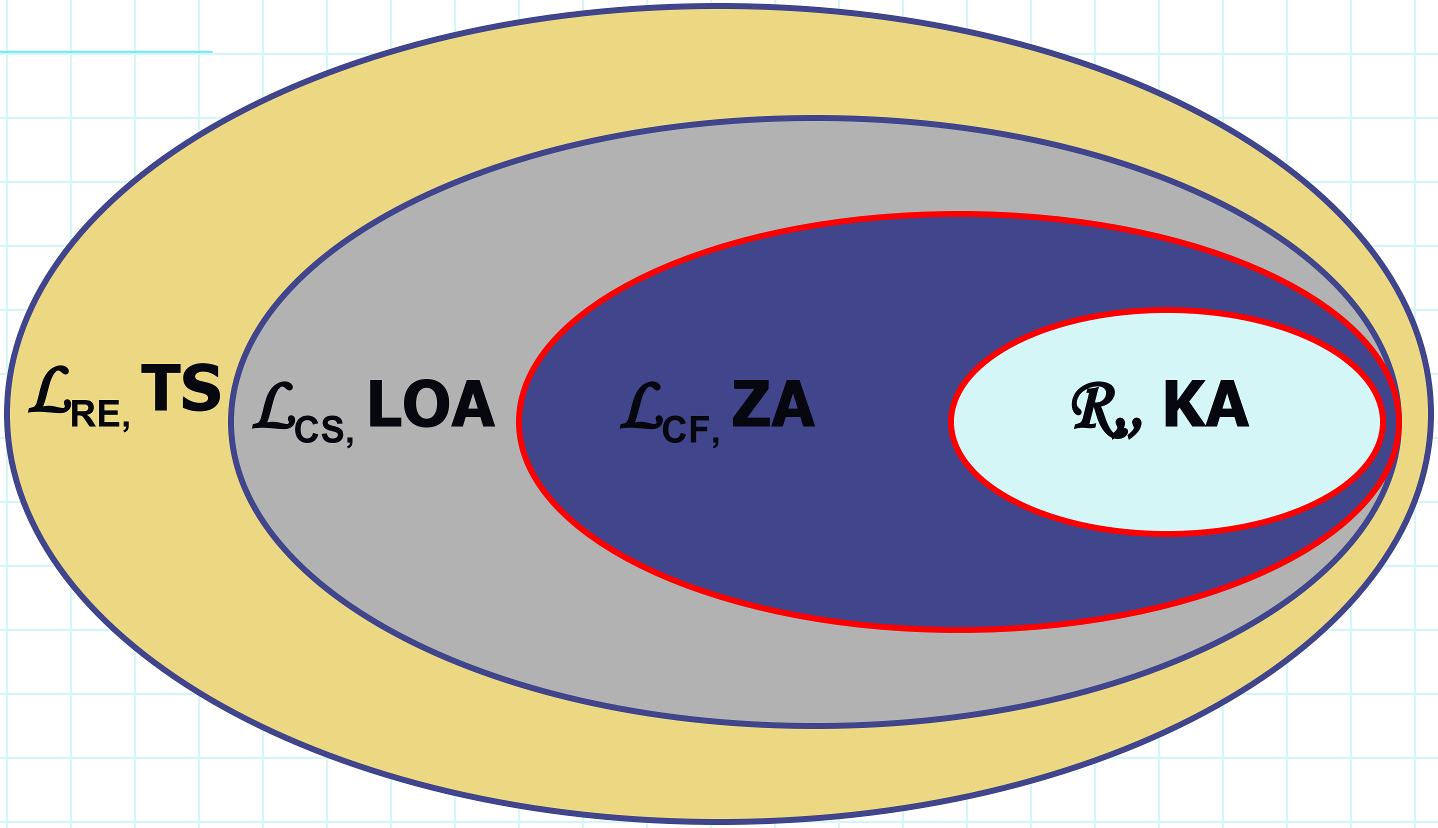
Frame 17

Teoretické základy informatiky

Zásobníkové automaty

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ **Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)**

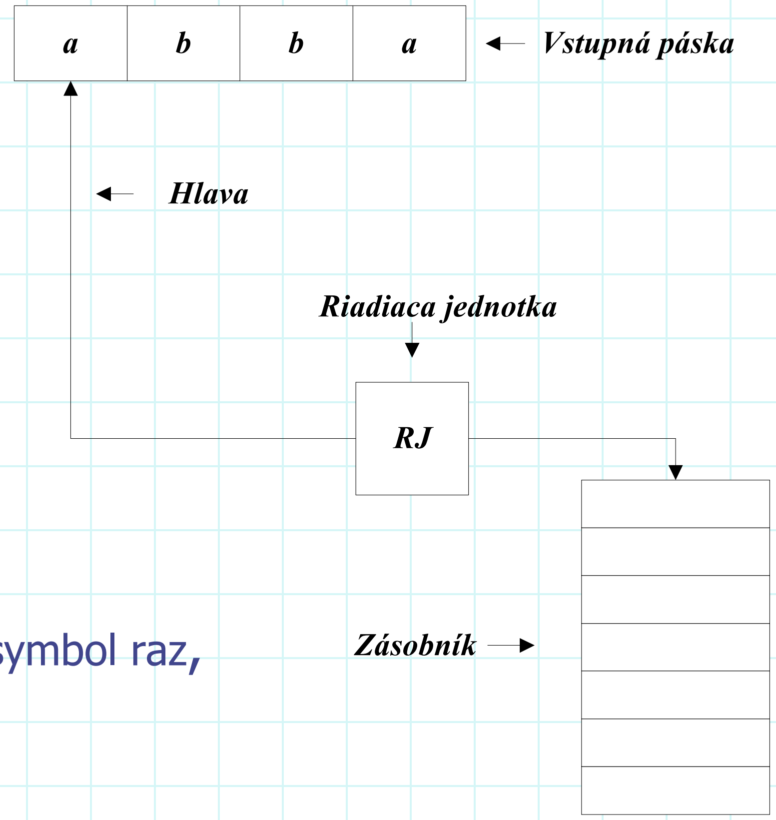
$\mathcal{L}_{CS}$ **Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)**

$\mathcal{L}_{CF}$ **Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)**

$\mathcal{R}$ **Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)**

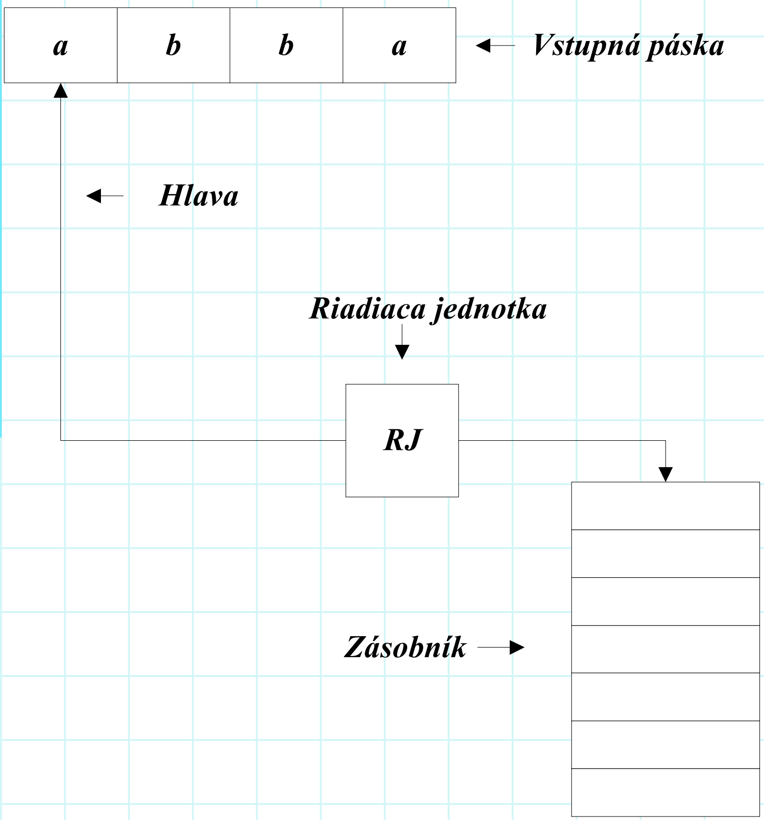
Reprezentácia jazyka výpočtom

Výpočtový model reprezentovaný zásobníkovým automatom:



- riadiaca jednotka,
- čítacia hlava, čítajúca symboly, každý symbol raz,
- vstupná páska,
- zásobník.

Výpočtový model – zásobníkový automat



Operácie so zásobníkom:

push – pridanie prvku na vrchol zásobníka

pop – odobratie prvku na vrchol zásobníka

skip – prázdna operácia, bez zmeny zásobníka

top – vráti znak z vrcholu zásobníka

empty – test na prázdnosť zásobníka

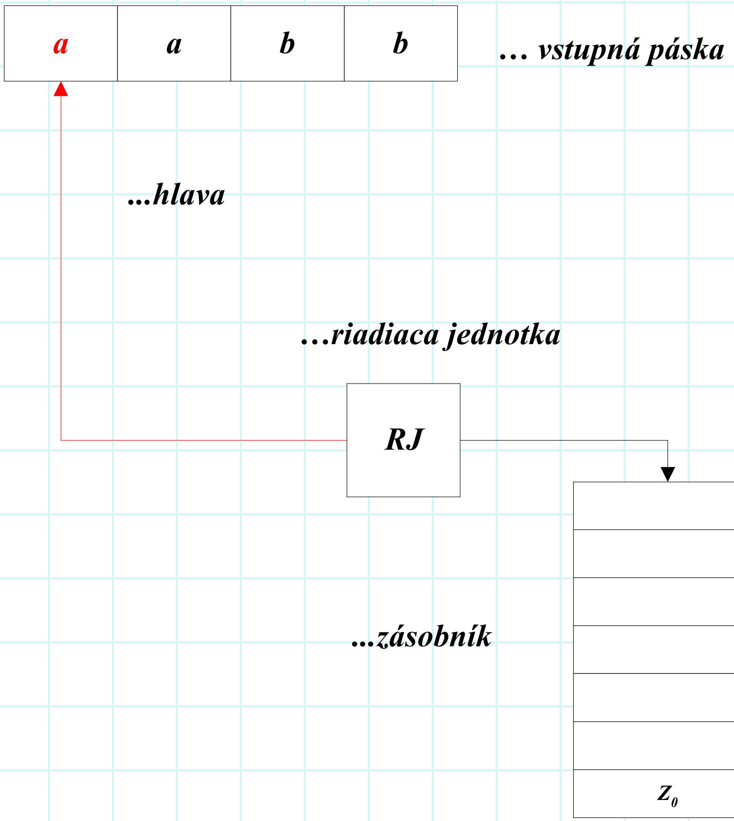
Abstraktná údajová štruktúra **zásobník**

Operácie pri manipulovaní so zásobníkom:

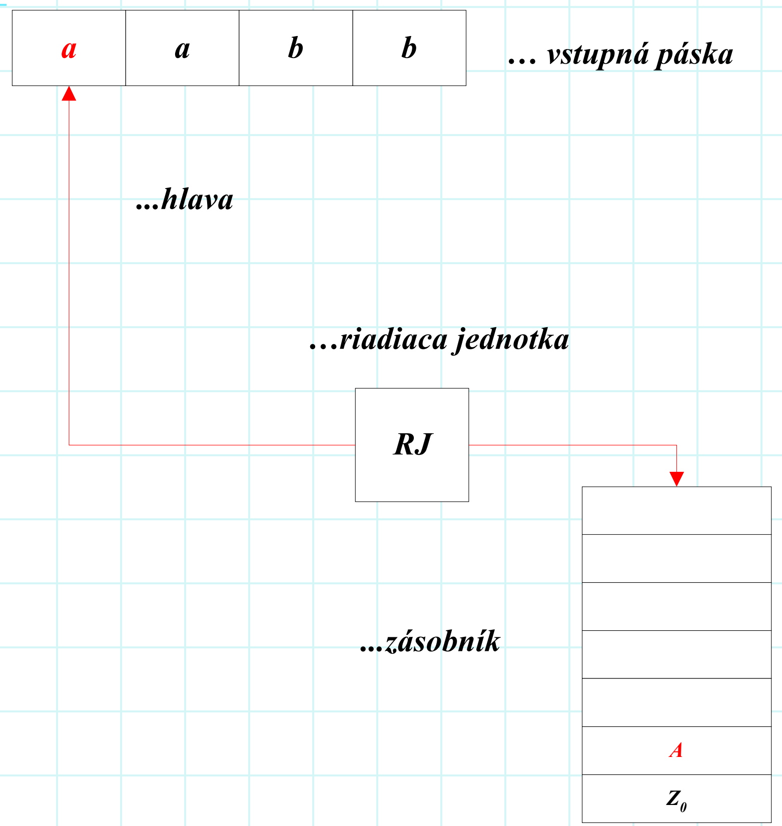
- `push` - pridanie prvku na vrchol zásobníka,
- `pop` - odobratie prvku z vrchola zásobníka,
- `top` - vráti znak, ktorý sa nachádza na vrchole zásobníka (prečítaný znak ostáva na svojom mieste),
- `skip` - prázdna operácia (obsah zásobníka ostáva nezmenený)
- `empty` - test na prázdnosť.

Operácie `top` a `empty` sú funkcie (t.j. ich výsledkom je vrátená hodnota).

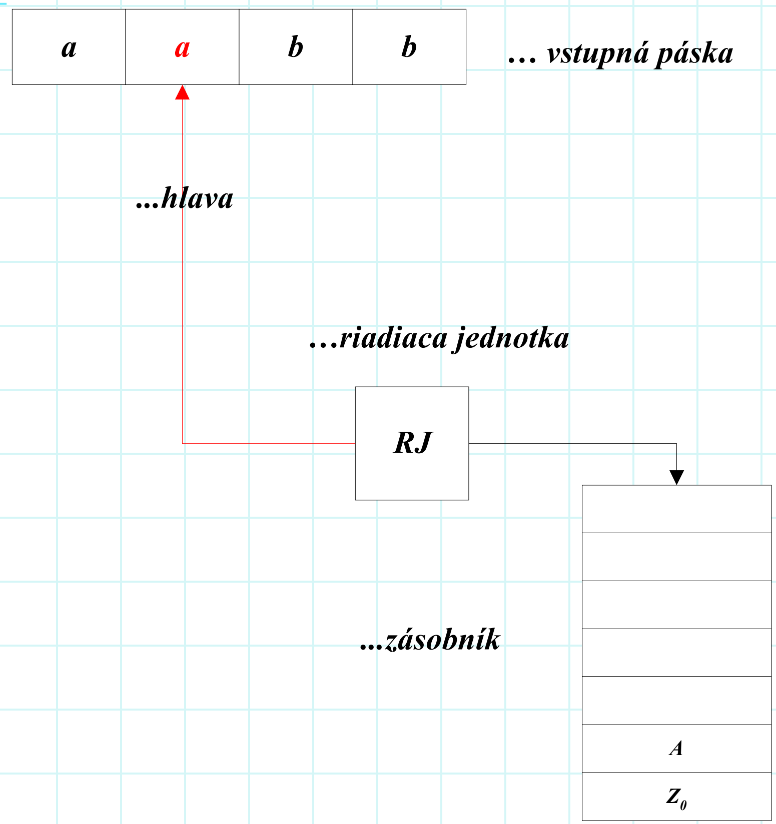
a^2b^2



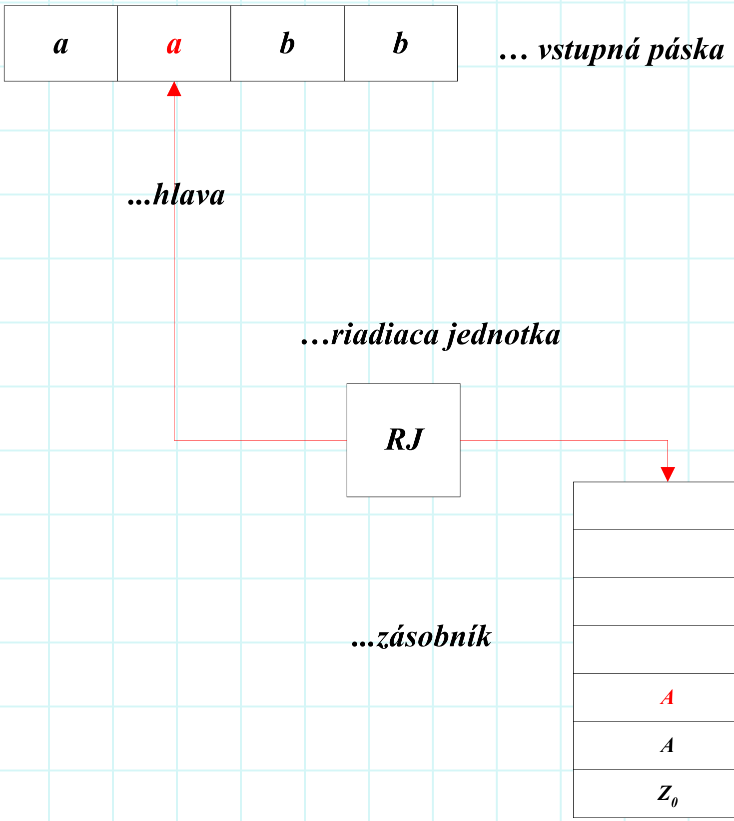
a^2b^2


a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
Z_0

a^2b^2

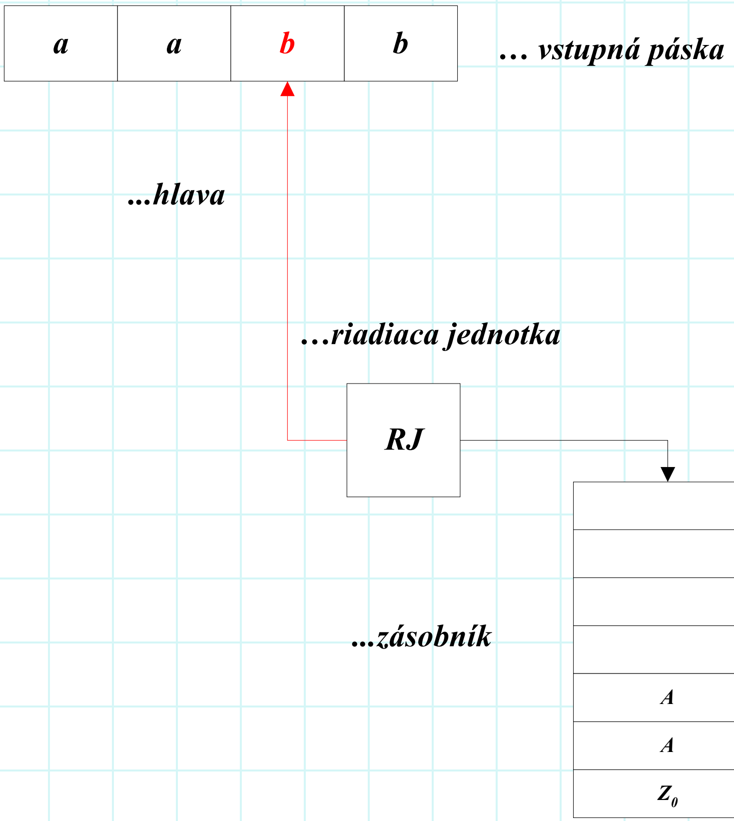

a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
Z0

a^2b^2



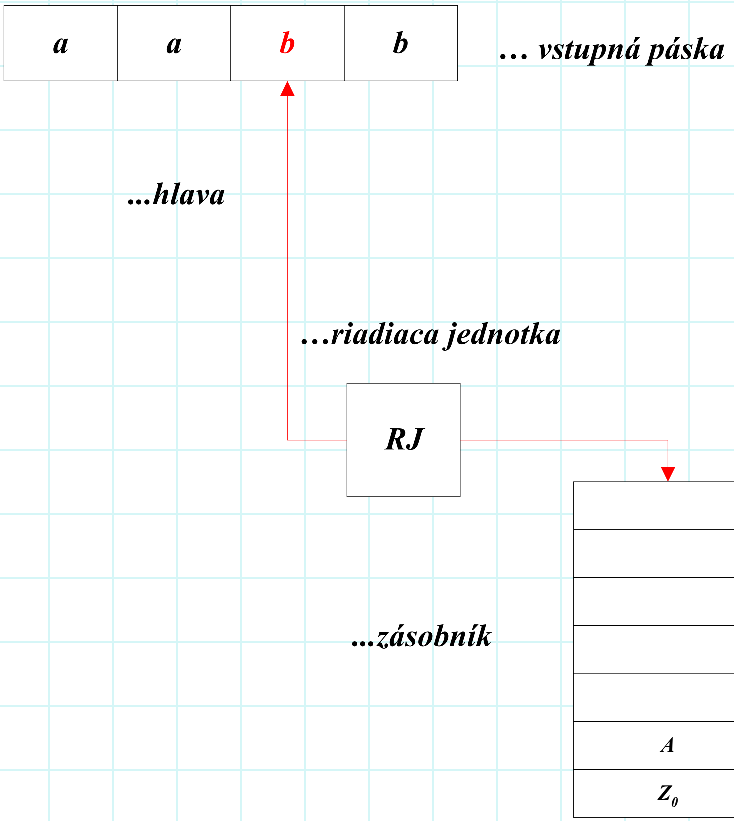
a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
A
Z_0

a^2b^2

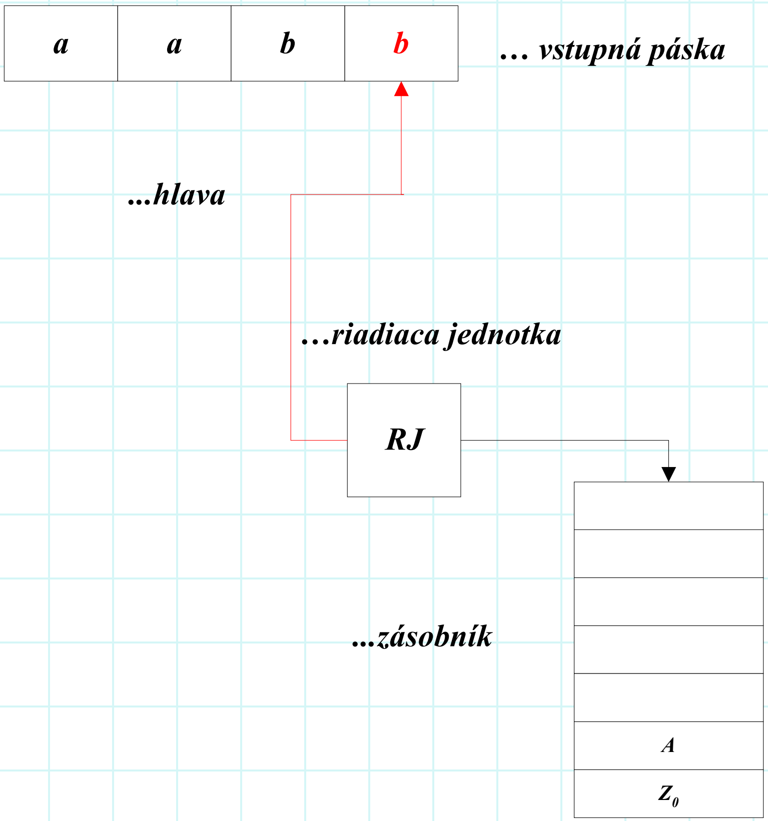


a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
A
Z0

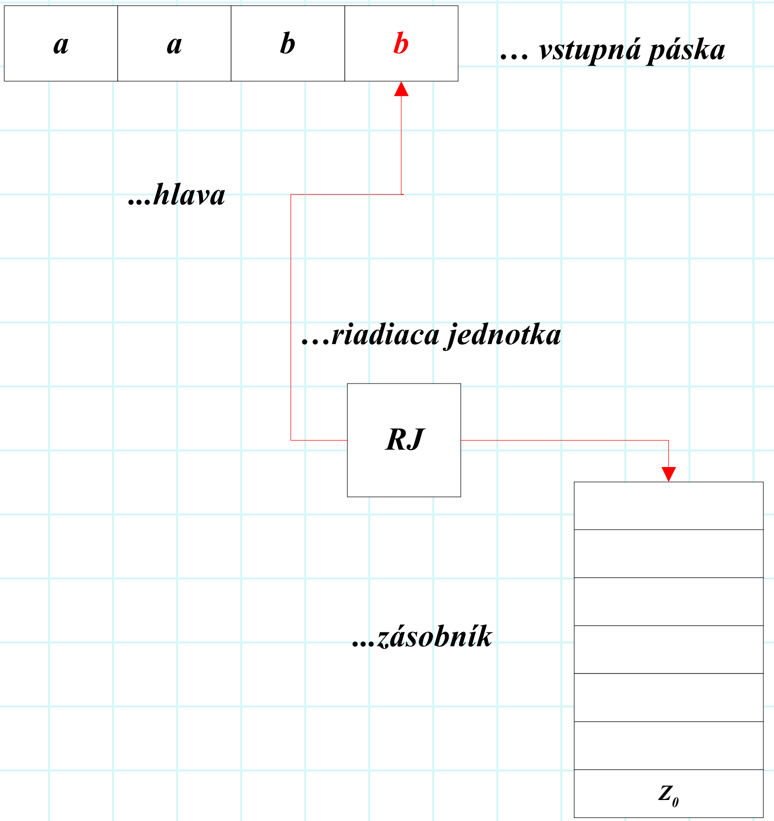
a^2b^2


a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
Z0

a^2b^2


a
a
b
b
… vstupná páska
...hlava
…riadiaca jednotka
RJ
...zásobník
A
Z0

a^2b^2



Nedeterministický zásobníkový automat NZA – definícia 4.1.1

Nedeterministický zásobníkový automat *(NPDA - nondeterministic push-down automaton) je sedemtica* $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, *kde:*

K *je konečná množina stavov,*

Σ *je vstupná abeceda,*

Γ *je zásobníková abeceda,*

δ *je prechodov, zobrazenie, pričom platí*

$$\delta : K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{K \times \Gamma^*}_{KON}$$

(Symbol $2^{K \times \Gamma^*}_{KON}$ *označuje konečné podmnožiny množiny* $K \times \Gamma^*$*.)*

$q_0 \in K$ *je počiatočný stav,*

$Z_0 \in \Gamma$ *je počiatočný zásobníkový symbol,*

$F \subseteq K$ *je množina koncových stavov.*

Konfigurácia nedeterministického zásobníkového automatu *je trojica*

$$(q, w, \gamma) \ \in \ K \times \Sigma^* \times \Gamma^*,$$

kde:

q *je momentálny stav,*

w *je zvyšok vstupu,*

γ *je aktuálny obsah zásobníka.*

Nedeterministický zásobníkový automat– krok výpočtu, jazyk rozpoznávaný NZA

Krok výpočtu nedeterministického zásobníkového automatu *je relácia* $\vdash_A$ *na množine konfigurácií* $K \times \Sigma^* \times \Gamma^*$ *definovaná nasledovne:*

$$(q, aw, Z\gamma) \vdash_A (p, w, \beta\gamma), \qquad ak \qquad (p, \beta) \in \delta(q, a, Z),$$

pričom $p,\ q \in K,\ \ a \in \Sigma \cup \{\varepsilon\},\ \ Z \in \Gamma,\ \ w \in \Sigma^*,\ \ \beta,\ \gamma \in \Gamma^*.$

Jazyk rozpoznávaný nedeterministickým zásobníkovým automatom *(koncovým stavom) je množina*

$$L(A) = \{\ w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \gamma),\ q \in F,\ \gamma \in \Gamma^*\ \}.$$

Výpočet nedeterministického zásobníkového automatu A na slove w je postupnosť konfigurácií $(q_0, w, Z_0) \vdash_A \ldots$.

Metódy akceptovania ZA

- koncovým stavom,
- prázdnym zásobníkom.

Jazyk rozpoznávaný nedeterministickým zásobníkovým automatom *(koncovým stavom) je množina*

$$L(A) = \{ \ w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \gamma), \ q \in F, \ \gamma \in \Gamma^* \ \}.$$

Poznámka 4.1.1 *Definícia 4.1.1 zaručuje, že vstupné slovo je akceptované koncovým stavom, pričom obsah zásobníka môže byť* ľubovoľný, *teda aj neprázdny. Ekvivalentnú verziu nedeterministického zásobníkového automatu predstavuje automat, ktorý akceptuje* prázdnym zásobníkom. *(Jeho definíciu uvádzať nebudeme.)*

Deterministický zásobníkový automat DZA - definícia

Definícia 4.2.1 Deterministický zásobníkový automat

Nech $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ *je zásobníkový automat v zmysle definície 4.1.1. Automat A sa nazýva* deterministický zásobníkový automat, *(DPDA - deterministic push-down automaton) ak* $\forall\ q \in K,\ \forall\ a \in \Sigma,\ \forall\ Z \in \Gamma$ *platí:*

1. $\delta(q, a, Z)$ *obsahuje najviac jeden prvok,*
2. $\delta(q, \varepsilon, Z)$ *obsahuje najviac jeden prvok,*
3. *ak množina* $\delta(q, \varepsilon, Z)$ *nie je prázdna, potom množina* $\delta(q, a, Z)$ *je prázdna* $\forall\ a \in \Sigma$.

Namiesto slovného spojenia "deterministický zásobníkový automat" budeme používať aj skrátené označenie $DPDA$. Symbolom $\mathcal{L}(DPDA)$ budeme označovať triedu jazykov rozpoznávaných deterministickými zásobníkovými automatmi.

Riešené príklady

Animácie deterministických ZA pre jazyky:

- $L=\{a^n b^n \mid n \in N\}$
- $L=\{wcw^R \mid w \in \{a,b\}^*\}$

Trieda jazykov $\mathcal{L}_{CF}$ Veta o vzťahu PDA a $\mathcal{L}_{CF}$

Veta 4.1.1 *Trieda jazykov rozpoznávaných nedeterministickými zásobníkovými automatmi je zhodná s triedou všetkých bezkontextových jazykov. Formálne:*

$$\mathcal{L}(NPDA) = \mathcal{L}_{CF} \ . \tag{4.1}$$

Z uvedenej vety vyplýva, že ku každému jazyku $L_0 \in \mathcal{L}_{CF}$ sa d zostrojiť nedeterministický zásobníkový automat A_0 tak, že $L(A_0) = L_0$. Tiež platí, že každý zásobníkový automat rozpoznáva nejaký bezkontextový jazyk. Dôkaz vety 4.1.1 je vysoko netriviálny.

Dôkaz: (konštrukcia dôkazu časti $L_{(A)=}L_{(G)}$)

G=(N,T,P,S) v Greibachovej norm. tv. $A \rightarrow aX$, $A \in N$, $a \in T$, $X \in N^*$

PDA=$(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$

$\Sigma=T$, $K=\{q_0, q_1, q_F\}$, $\Gamma=N \cup \{Z_0\}$, $F=\{q_F\}$

konštrukcia δ: $\delta(q_0,\varepsilon, Z_0)=(q_1,SZ_0)$ *počiatočný neterminál*

$\delta(q_1,a, A)=(q_1,X) \Leftrightarrow A \rightarrow aX \in P$ *prepis pravidiel*

$\delta(q_1,\varepsilon, Z_0)=(q_F,Z_0)$ *akceptovanie*

Príklad: prevod G → PDA

$G=(N,T,P,S)$ $N=\{S,A\}$ $T=\{a,b\}$

P: $S \rightarrow aAA \mid a$

$A \rightarrow bS \mid aSA \mid b$

$PDA=(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ $\Sigma=T=\{a,b\}$ $\Gamma=\{S,A,Z_0\}$ $F=\{q_F\}$

$\delta(q_0,\varepsilon,Z_0)=(q_1,SZ_0)$

$\delta(q_1,a,S)=\{(q_1,AA),(q_1,\varepsilon)\}$

$\delta(q_1,a,A)=(q_1,SA)$

$\delta(q_1,b,A)=\{(q_1,S),(q_1,\varepsilon)\}$

$\delta(q_1,\varepsilon,Z_0)=(q_F,Z_0)$

Príklad: odvodenie slova v G a PDA

$G=(N,T,P,S)$ $N=\{S,A\}$ $T=\{a,b\}$ P: $S \to aAA|a$ $A \to bS|aSA|b$

$PDA=(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ $\Sigma=T=\{a,b\}$ $\Gamma=\{S,A,Z_0\}$ $F=\{q_F\}$

$\delta(q_0,\varepsilon,Z_0)=(q_1,SZ_0)$ $\delta(q_1,a,S)=\{(q_1,AA),(q_1,\varepsilon)\}$

$\delta(q_1,a,A)=(q_1,SA)$ $\delta(q_1,b,A)=\{(q_1,S),(q_1,\varepsilon)\}$

$\delta(q_1,\varepsilon,Z_0)=(q_F,Z_0)$

Pre ľubovoľné slovo $w \in L(A)$ existuje akceptujúci výpočet automatu A na tomto slove. Napríklad pre slovo abba existuje nasledujúce odvodenie v gramatike G.

$$S \Rightarrow aAA \Rightarrow abA \Rightarrow abbS \Rightarrow abba$$

Príslušný (nedeterminisitický) výpočet automatu A pre slovo abba zodpovedá jeho odvodeniu.

$$(q_0,abba,Z_0) \vdash (q_1,abba,SZ_0) \vdash (q_1,bba,AAZ_0) \vdash (q_1,ba,AZ_0) \vdash$$
$$\vdash (q_1,a,SZ_0) \vdash (q_1,\varepsilon,Z_0) \vdash (q_F,\varepsilon,Z_0)$$

Dôsledok – nedeterministický automat

Keď mám v gramatike aspoň 2 pravidlá začínajúce tým istým deterministickým symbolom – mám nedeterministický krok v konštrukcii automatu PDA.

v G: $S \rightarrow \underline{a}A \mid \underline{a}$

v PDA: $\delta(q_1, a, S) = \{(q_1, A), (q_1, \varepsilon)\}$

Pr.: dokážte, kedy dostaneme deterministický automat

...keď je pravidlo typu $A \rightarrow aX$ len s jedným počiatočným a...

Riešený príklad $L=\{ww^R \mid w\in\{a,b\}^*\}$

Nech $A_N = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, *kde:*

$K = \{q_0, q_1\}$,

$\Sigma = \{a, b\}$,

$\Gamma = \{Z_0, A, B\}$,

$F = \{q_2\}$,

w	$\delta(q_0, a, Z_0) = (q_0, AZ_0)$	push(A)
	$\delta(q_0, b, Z_0) = (q_0, BZ_0)$	push(B)
	$\delta(q_0, a, A) = \{\ (q_0, AA),\ \ (q_1, \varepsilon)\ \}$	*nedeterministický krok*
	$\delta(q_0, b, B) = \{\ (q_0, BB),\ \ (q_1, \varepsilon)\ \}$	*nedeterministický krok*
	$\delta(q_0, a, B) = (q_0, AB)$	push(A)
	$\delta(q_0, b, A) = (q_0, BA)$	push(B)
w^R	$\delta(q_1, a, A) = (q_1, \varepsilon)$	pop
	$\delta(q_1, b, B) = (q_1, \varepsilon)$	pop
Akcept.	$\delta(q_1, \varepsilon, Z_0) = (q_2, Z_0)$	*akceptovanie.*

Uzáverové vlastnosti triedy $\mathcal{L}_{CF}$

Veta 4.1.2 *Trieda $\mathcal{L}_{CF}$ je uzavretá vzhľadom na operácie:*

1. *zjednotenie* $(\cup)$,
2. *zreťazenie* $(\cdot)$,
3. *Kleeneho iterácia* $(^*)$,
4. *prienik s regulárnymi jazykmi.*

Veta 4.1.3 *Trieda $\mathcal{L}_{CF}$ nie je uzavretá vzhľadom na operácie:*

1. *prienik* $(\cap)$,
2. *doplnok* $(^C)$.

Uzáverové vlastnosti triedy $\mathcal{L}_{\text{DPDA}}$

Uzáverové vlastnosti triedy $\mathcal{L}(DPDA)$ sú uvedené v nasledujúcom tvrdení.

Veta 4.2.1 *Trieda $\mathcal{L}(DPDA)$ je uzavretá vzhľadom na operácie:*

1. *prienik s regulárnymi jazykmi,*
2. *doplnok (^{C}).*

$\mathcal{L}_{\text{DPDA}} \neq \mathcal{L}_{\text{NPDA}}$

Rozlišujeme teda dve rôzne triedy jazykov:

- $\mathcal{L}(NPDA)$ - trieda jazykov rozpoznávaných nedeterministickými zásobníkovými automatmi,
- $\mathcal{L}(DPDA)$ - trieda jazykov rozpoznávaných deterministickými zásobníkovými automatmi.

Ich vzájomný vzťah určuje nasledujúce tvrdenie.

Veta 4.2.2 *Trieda jazykov rozpoznávaných deterministickými zásobníkovými automatmi je vlastnou podmnožinou triedy jazykov rozpoznávaných nedeterministickými zásobníkovými automatmi. Formálne:*

$$\mathcal{L}(DPDA) \subsetneq \mathcal{L}(NPDA) \ . \tag{4.2}$$

Jazyk L={ww^R | w∈{a,b}^*}

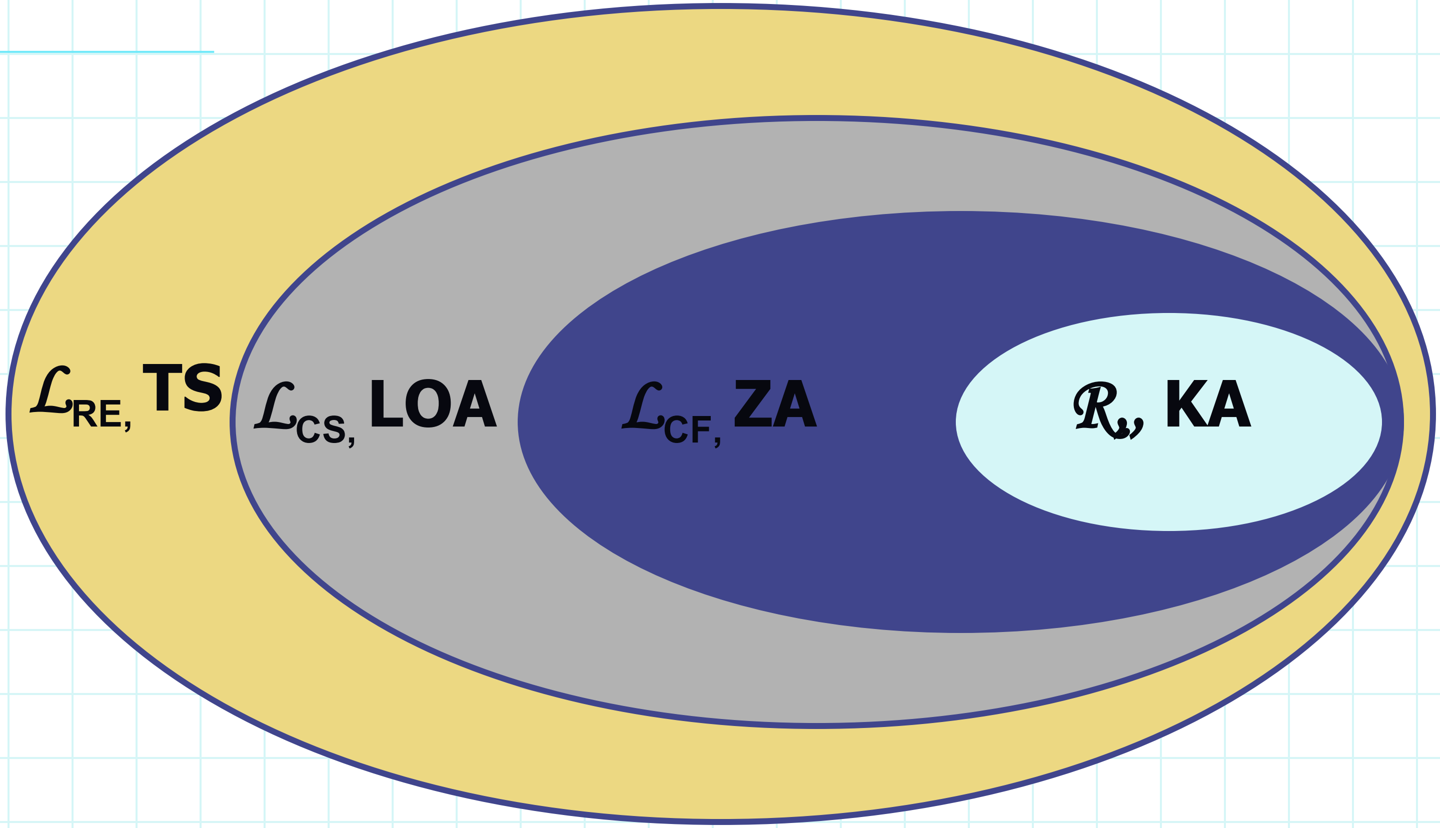
Lema 4.2.1 *Nech* $L_N = \{ww^R \mid w \in \{a,b\}^+\}$.

1. *Neexistuje deterministický zásobníkový automat, ktorý by rozpoznával jazyk* L_N.
2. *Existuje nedeterministický zásobníkový automat* A_N, *pre ktorý platí:*

   $L(A_N) = L_N$.

Z toho vyplýva, že $L_N \in \mathcal{L}(NPDA) \setminus \mathcal{L}(DPDA)$.

Chomského hierarchia jazykov



$\mathcal{L}_{RE}$ **Trieda rekurzívne vyčísliteľných jazykov, generovaných frázovou gramatikou (Turingov Stroj)**

$\mathcal{L}_{CS}$ **Trieda kontextových jazykov, generovaných kontextovou gramatikou (Lineárne ohraničený Automat)**

$\mathcal{L}_{CF}$ **Trieda bezkontextových jazykov, generovaných bezkontextovou gramatikou (Zásobníkový Automat)**

$\mathcal{R}$ **Trieda regulárnych jazykov, generovaných regulárnou gramatikou (Konečný Automat)**

Vzťah automatov a jazykov z Chomského hierarchie

Automat	Determin. verzia		Nedetermin. verzia		Trieda jazykov
konečný	$\mathcal{L}(DFA)$	$=$	$\mathcal{L}(NFA)$	$=$	$\mathcal{R}$
zásobníkový	$\mathcal{L}(DPDA)$	$\subsetneq$	$\mathcal{L}(NPDA)$	$=$	$\mathcal{L}_{CF}$

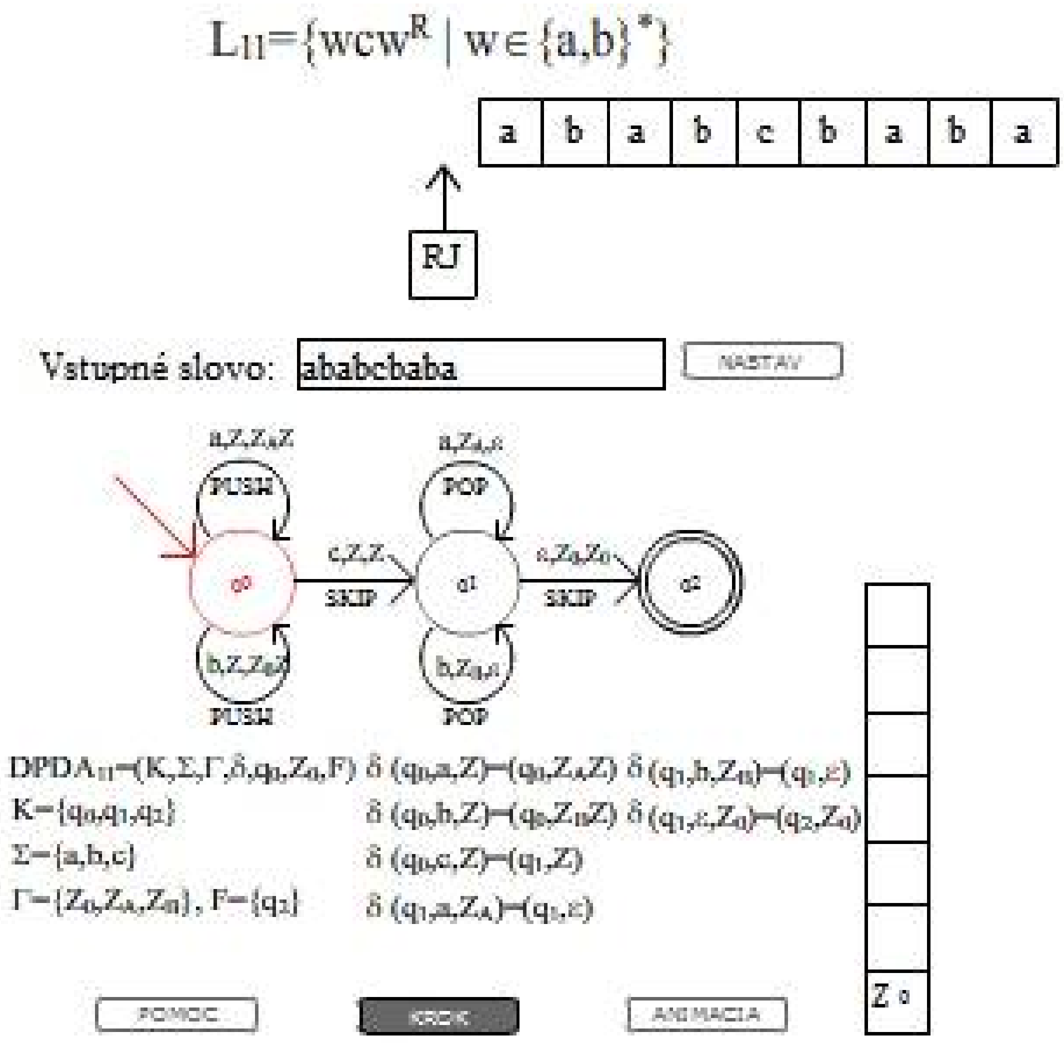
Uzáverové vlastnosti tried jazykov z Chomského hierarchie

Trieda jazykov	$\cup$	$\cdot$	$\cap$	$*$	$+$	c	R	h	h bez ϵ
$\mathcal{R}$	A	A	A	A	A	A	A	A	A
$\mathcal{L}_{CF}$	A	A	N	A	A	N	A	A	A

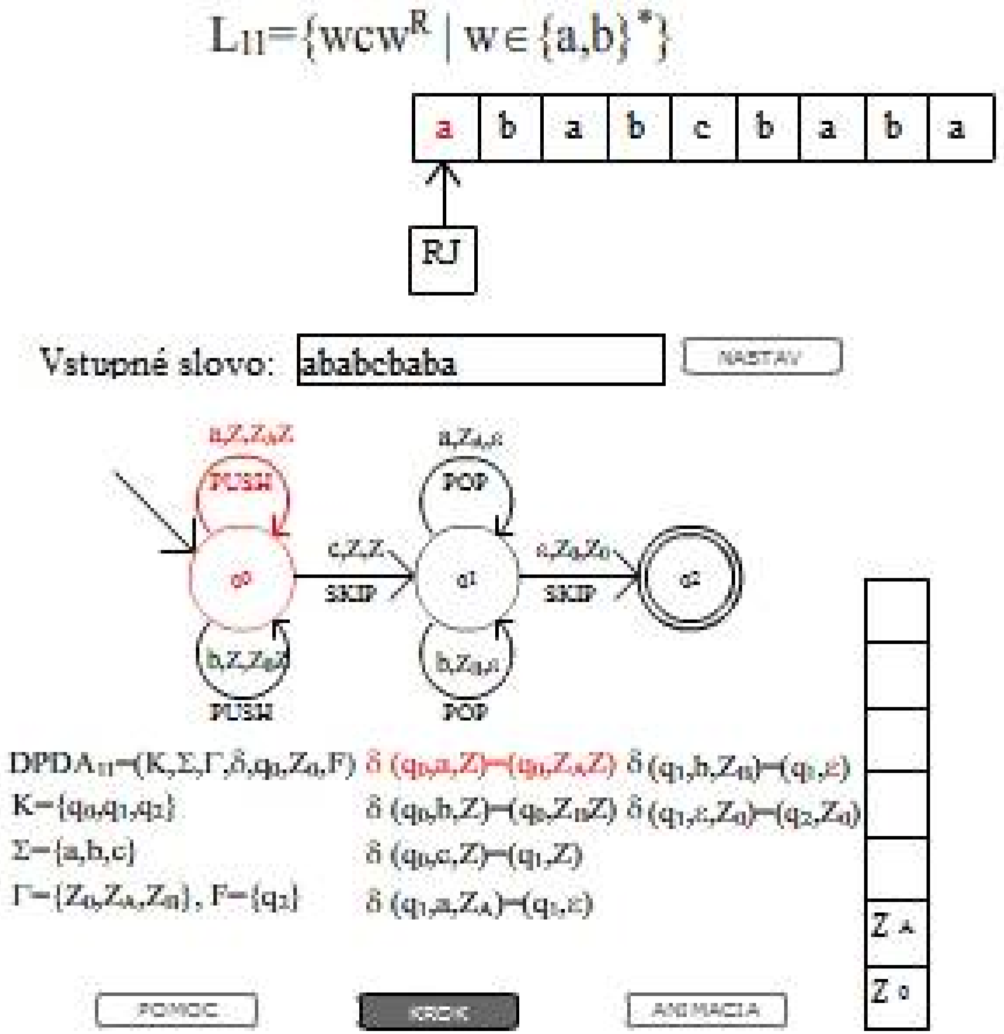
STU FIIT

Ďakujem za pozornosť.

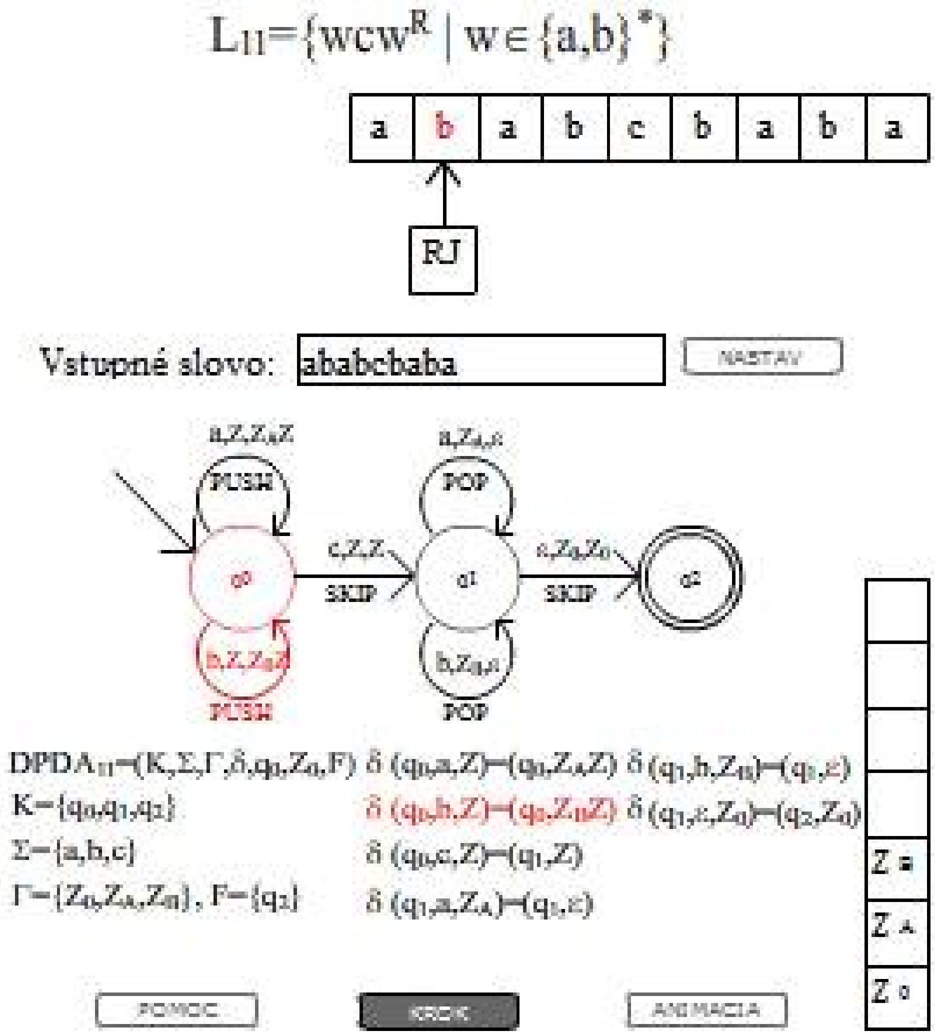
chuda@fiit.stuba.sk


$L_{11}=\{wcw^R \mid w \in \{a,b\}^*\}$
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,Z_aZ
PUSH
a,Z_a,ε
POP
c,Z,Z
SKIP
ε,Z_0,Z_0
SKIP
PUSH
POP
$DPDA_{11}=(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$
$K=\{q_0,q_1,q_2\}$
$\Sigma=\{a,b,c\}$
$\Gamma=\{Z_0,Z_A,Z_B\}$, $F=\{q_2\}$
$\delta(q_0,a,Z)=(q_0,Z_AZ)$
$\delta(q_0,b,Z)=(q_0,Z_BZ)$
$\delta(q_0,c,Z)=(q_1,Z)$
$\delta(q_1,a,Z_A)=(q_1,\varepsilon)$
$\delta(q_1,b,Z_B)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,Z_0)=(q_2,Z_0)$
Z_0

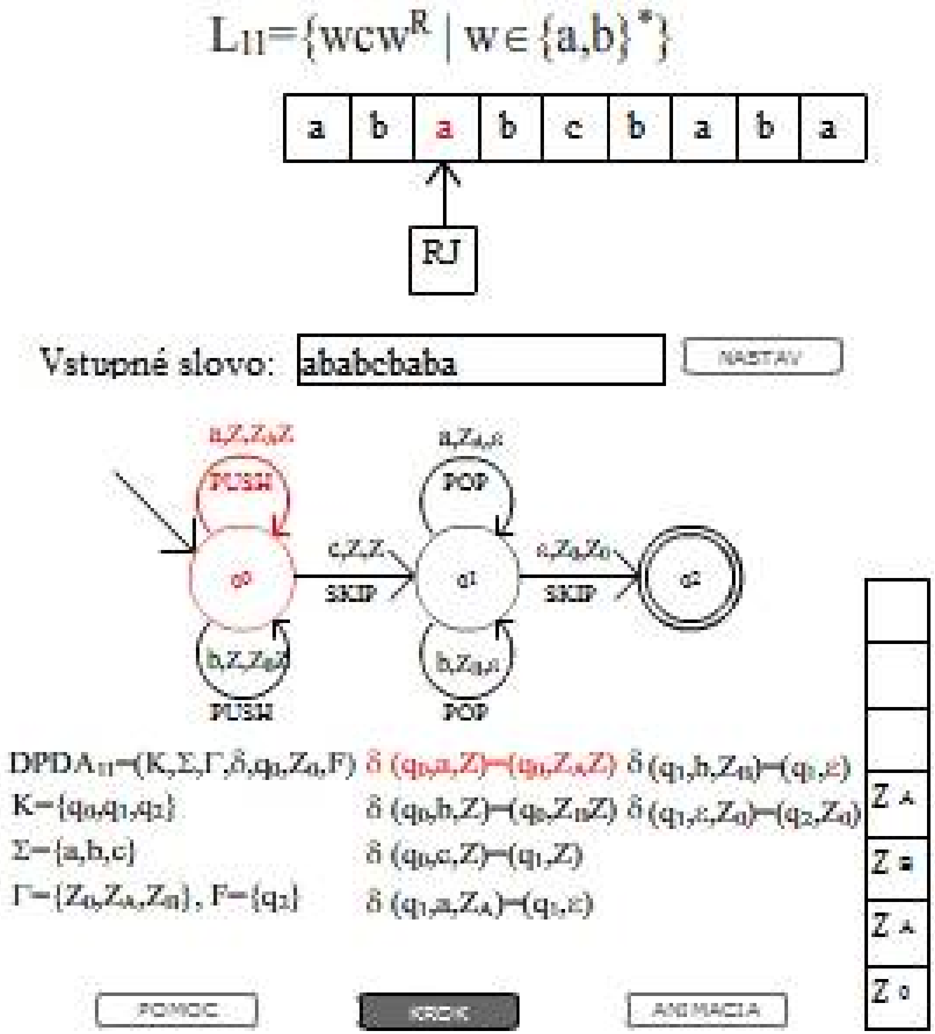
Frame 2


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
PUSH
POP
SKIP
SKIP
PUSH
POP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
δ (q0,c,Z)=(q1,Z)
δ (q1,a,ZA)=(q1,ε)
δ (q1,b,ZB)=(q1,ε)
POMOC
KROK
ANIMACIA

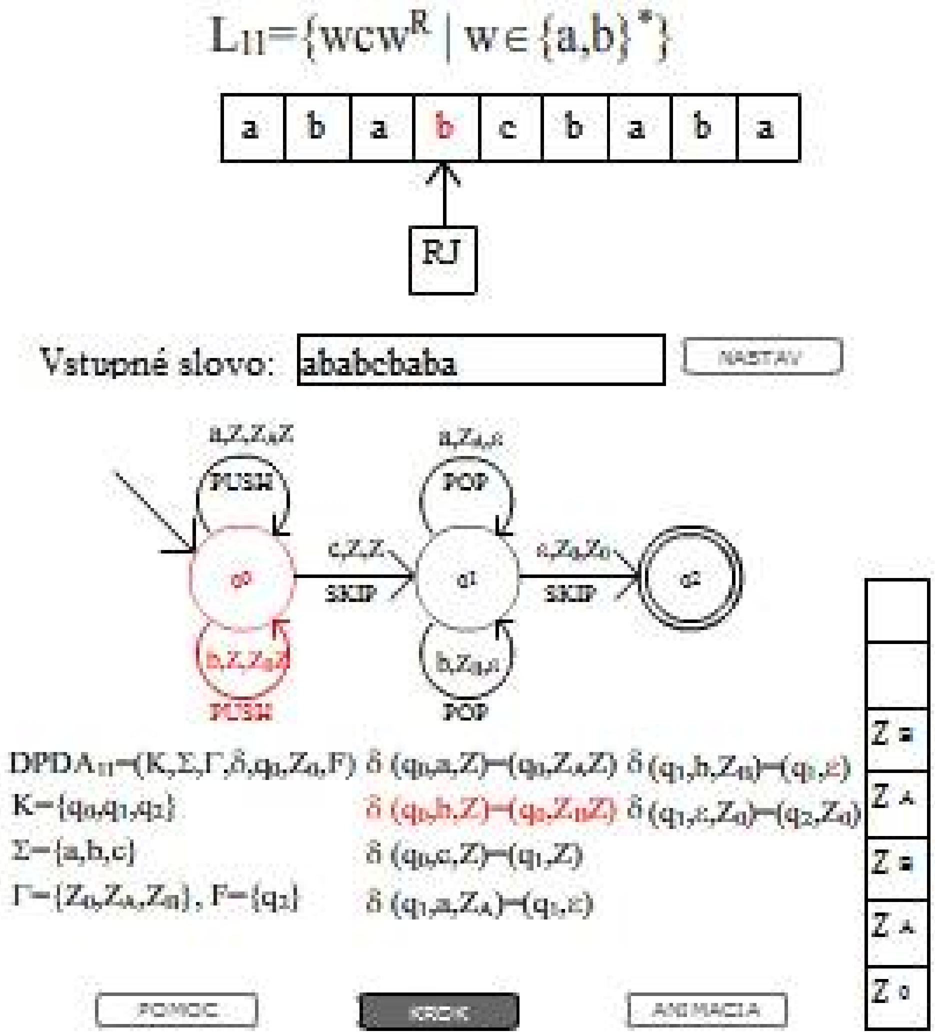
Frame 3


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo:
ababcbaba
NASTAV
a,Z,ZaZ
PUSH
a,Za,ε
POP
c,Z,Z
SKIP
ε,Z0,Z0
SKIP
q0
q1
q2
b,Z,ZbZ
PUSH
b,Zb,ε
POP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,Za,Zb}, F={q2}
δ (q0,a,Z)=(q0,ZaZ)
δ (q0,b,Z)=(q0,ZbZ)
δ (q0,c,Z)=(q1,Z)
δ (q1,a,Za)=(q1,ε)
δ (q1,b,Zb)=(q1,ε)
δ (q1,ε,Z0)=(q2,Z0)
Z B
Z A
Z 0
POMOC
KROK
ANIMACIA

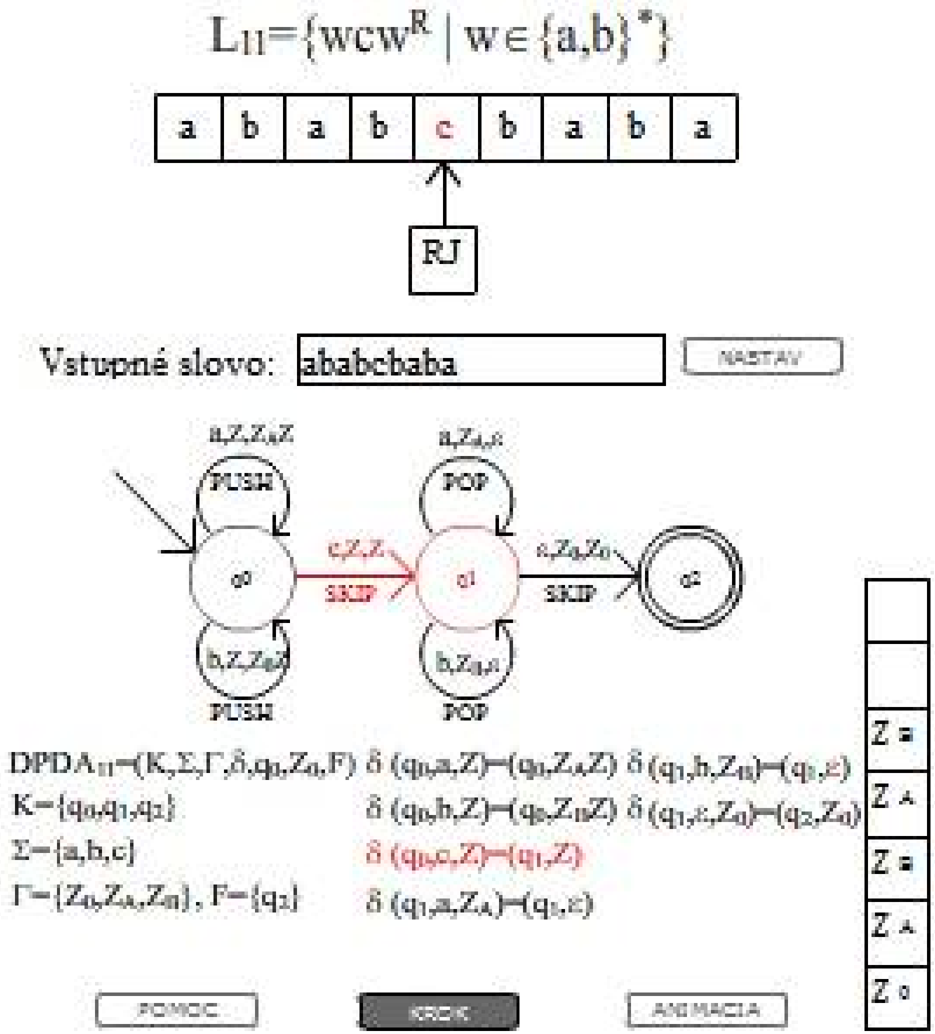
Frame 6


L11={wcw^R | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo:
ababcbaba
NASTAV
a,Z,ZAZ
PUSH
b,Z,ZBZ
PUSH
c,Z,Z
SKIP
a,ZA,ε
POP
b,ZB,ε
POP
ε,Z0,Z0
SKIP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
δ (q0,a,Z)=(q0,ZAZ)
δ (q0,b,Z)=(q0,ZBZ)
δ (q0,c,Z)=(q1,Z)
δ (q1,a,ZA)=(q1,ε)
δ (q1,b,ZB)=(q1,ε)
δ (q1,ε,Z0)=(q2,Z0)
Z A
Z B
Z A
Z 0
POMOC
KROK
ANIMACIA

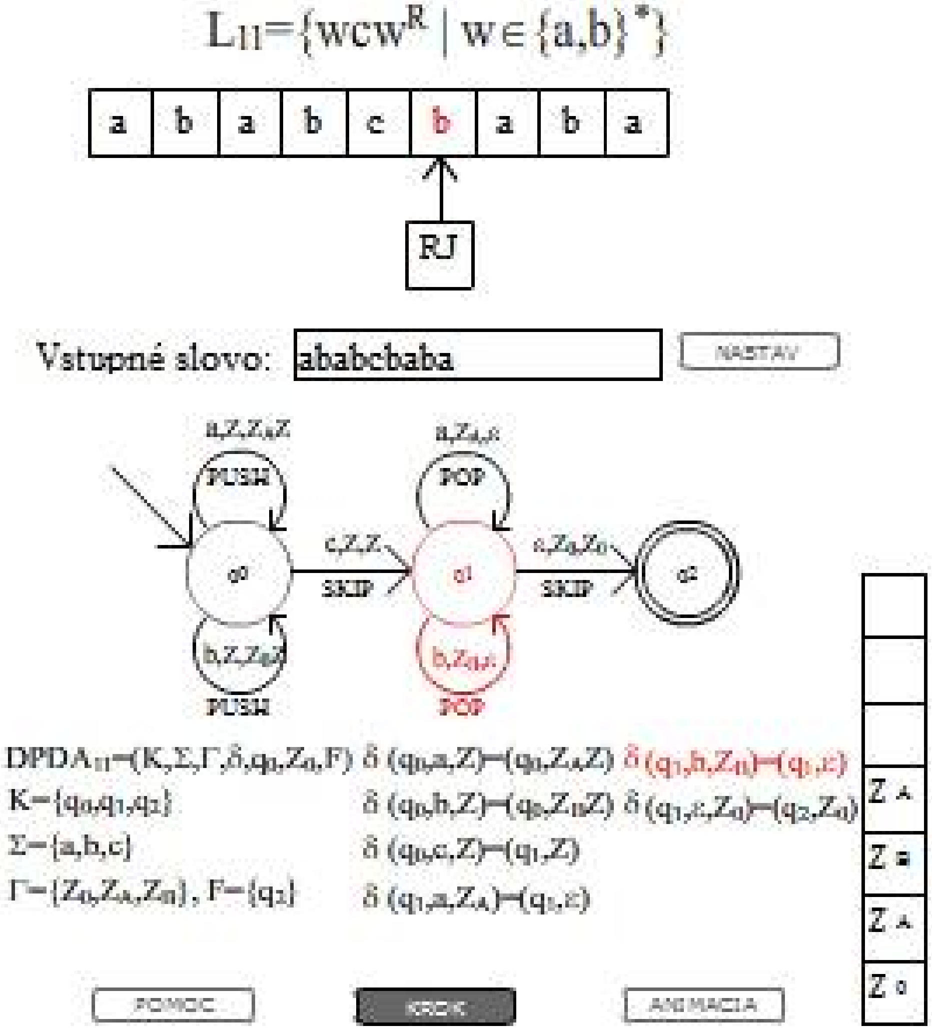
Frame 9


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,ZAZ
PUSH
a,ZA,ε
POP
c,Z,Z
SKIP
ε,Z0,Z0
SKIP
q0
q1
q2
b,Z,ZBZ
PUSH
b,ZB,ε
POP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
δ (q0,a,Z)=(q0,ZAZ)
δ (q0,b,Z)=(q0,ZBZ)
δ (q0,c,Z)=(q1,Z)
δ (q1,a,ZA)=(q1,ε)
δ (q1,b,ZB)=(q1,ε)
δ (q1,ε,Z0)=(q2,Z0)
Z B
Z A
Z B
Z A
Z 0
POMOC
KROK
ANIMACIA

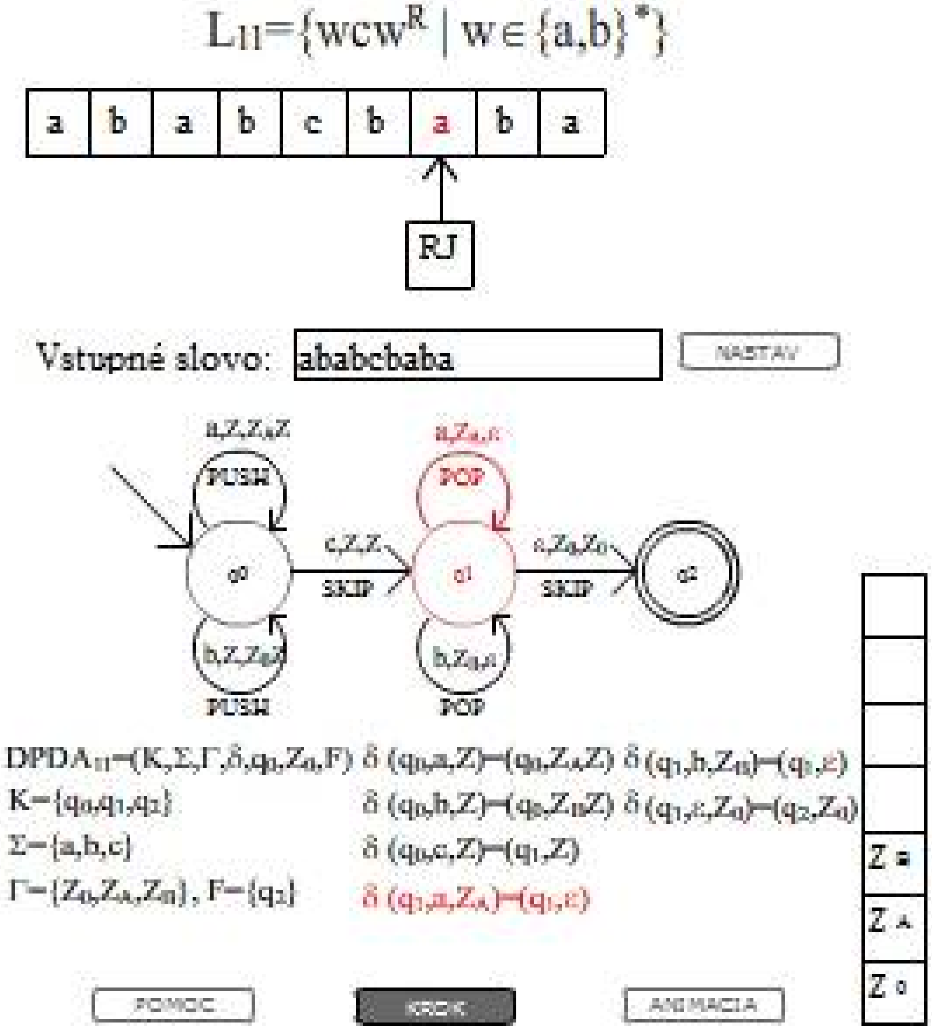
Frame 12


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo:
ababcbaba
NASTAV
a,Z,ZAZ
PUSH
b,Z,ZBZ
PUSH
c,Z,Z
SKIP
a,ZA,ε
POP
b,ZB,ε
POP
ε,Z0,Z0
SKIP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
δ (q0,a,Z)=(q0,ZAZ)
δ (q0,b,Z)=(q0,ZBZ)
δ (q0,c,Z)=(q1,Z)
δ (q1,a,ZA)=(q1,ε)
δ (q1,b,ZB)=(q1,ε)
δ (q1,ε,Z0)=(q2,Z0)
POMOC
KROK
ANIMACIA

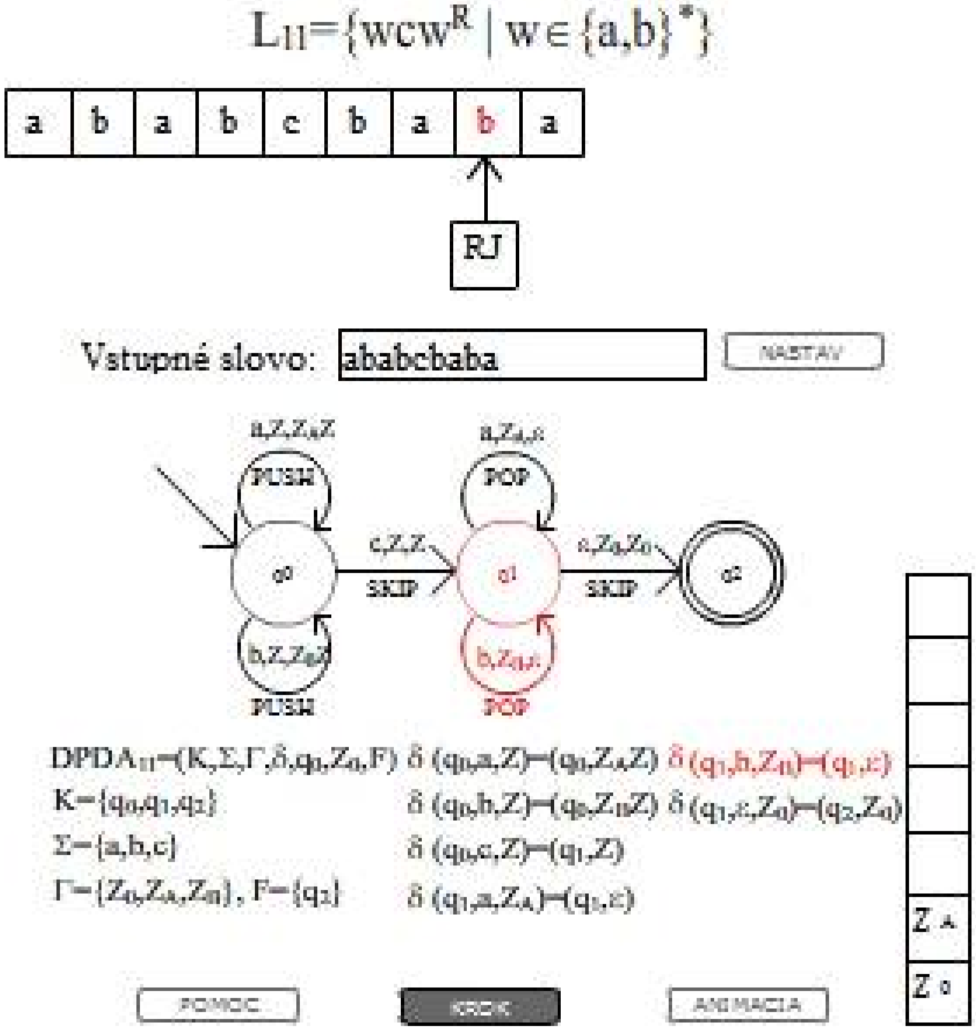
Frame 15


$L_{11}=\{wcw^R \mid w\in\{a,b\}^*\}$
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,Z$_A$Z
PUSH
b,Z,Z$_B$Z
PUSH
c,Z,Z
SKIP
a,Z$_A$,ε
POP
b,Z$_B$,ε
POP
ε,Z$_0$,Z$_0$
SKIP
$DPDA_{11}=(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$
$K=\{q_0,q_1,q_2\}$
$\Sigma=\{a,b,c\}$
$\Gamma=\{Z_0,Z_A,Z_B\}$, $F=\{q_2\}$
$\delta(q_0,a,Z)=(q_0,Z_AZ)$
$\delta(q_0,b,Z)=(q_0,Z_BZ)$
$\delta(q_0,c,Z)=(q_1,Z)$
$\delta(q_1,a,Z_A)=(q_1,\varepsilon)$
$\delta(q_1,b,Z_B)=(q_1,\varepsilon)$
$\delta(q_1,\varepsilon,Z_0)=(q_2,Z_0)$
Z A
Z B
Z A
Z 0
POMOC
KROK
ANIMACIA

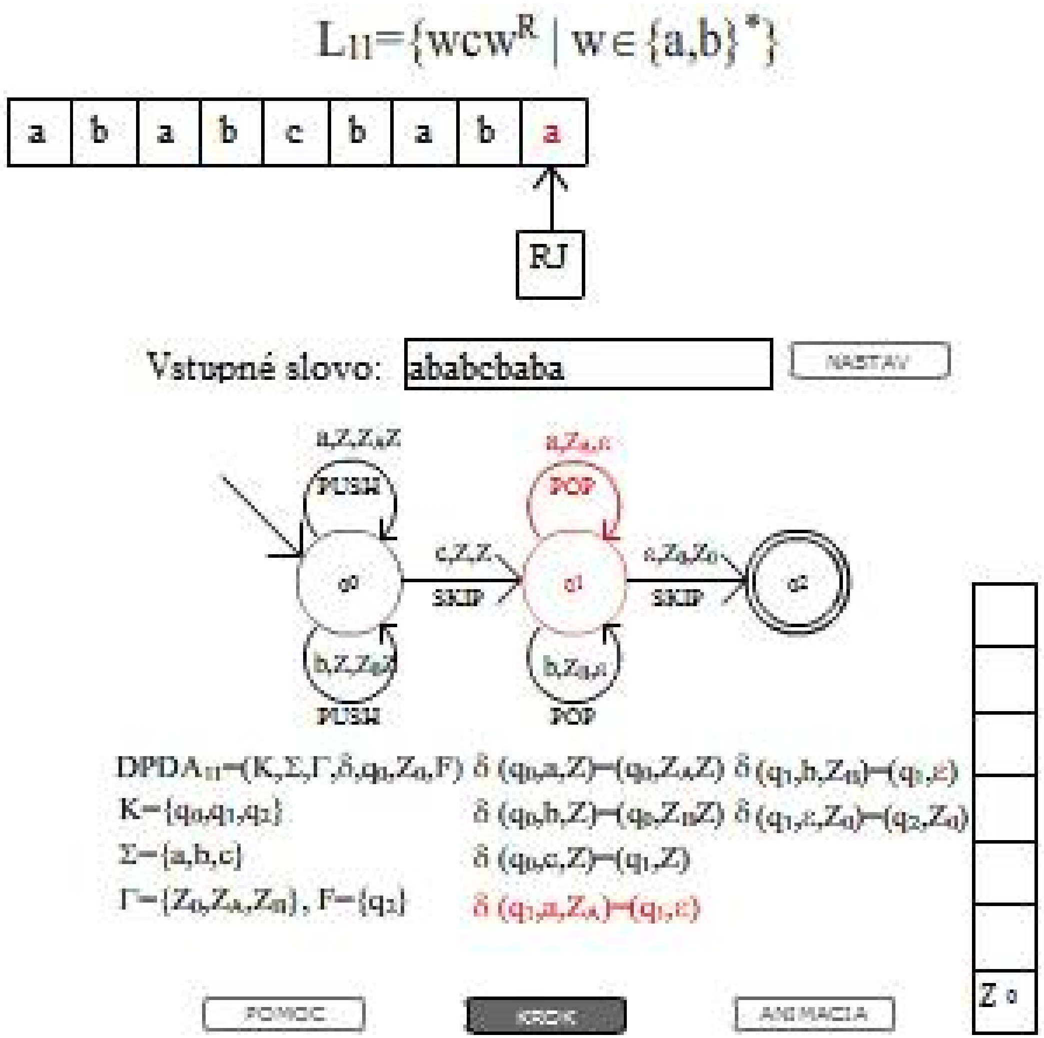
Frame 18


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,ZaZ
PUSH
b,Z,ZBZ
PUSH
c,Z,Z
SKIP
a,Za,ε
POP
b,ZB,ε
POP
ε,Z0,Z0
SKIP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
δ (q0,a,Z)=(q0,ZAZ)
δ (q0,b,Z)=(q0,ZBZ)
δ (q0,c,Z)=(q1,Z)
δ (q1,a,ZA)=(q1,ε)
δ (q1,b,ZB)=(q1,ε)
δ (q1,ε,Z0)=(q2,Z0)
Z B
Z A
Z 0
POMOC
KROK
ANIMACIA

Frame 21

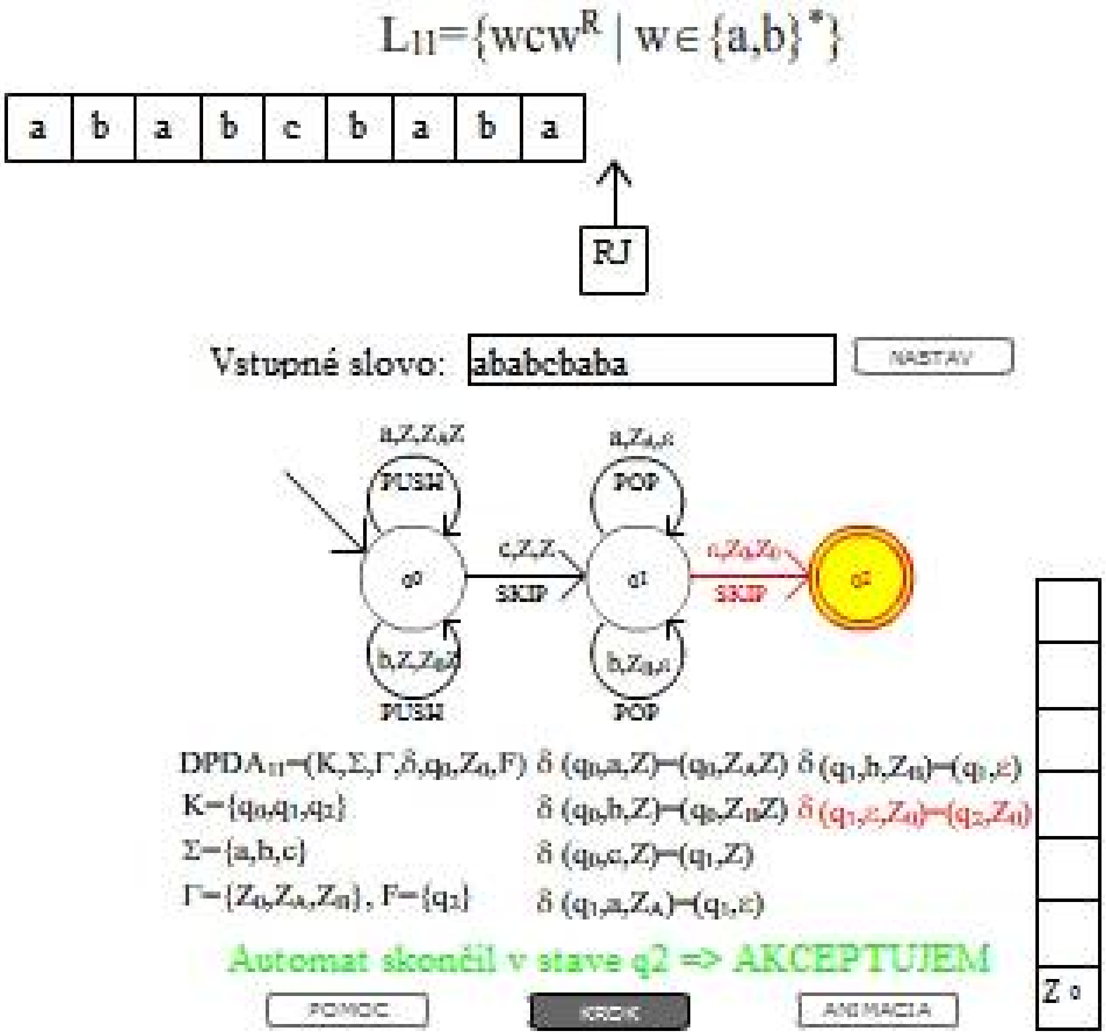

L11={wcw^R | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
DPDA11=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2}
Σ={a,b,c}
Γ={Z0,ZA,ZB}, F={q2}
POMOC
KROK
ANIMACIA

Frame 24



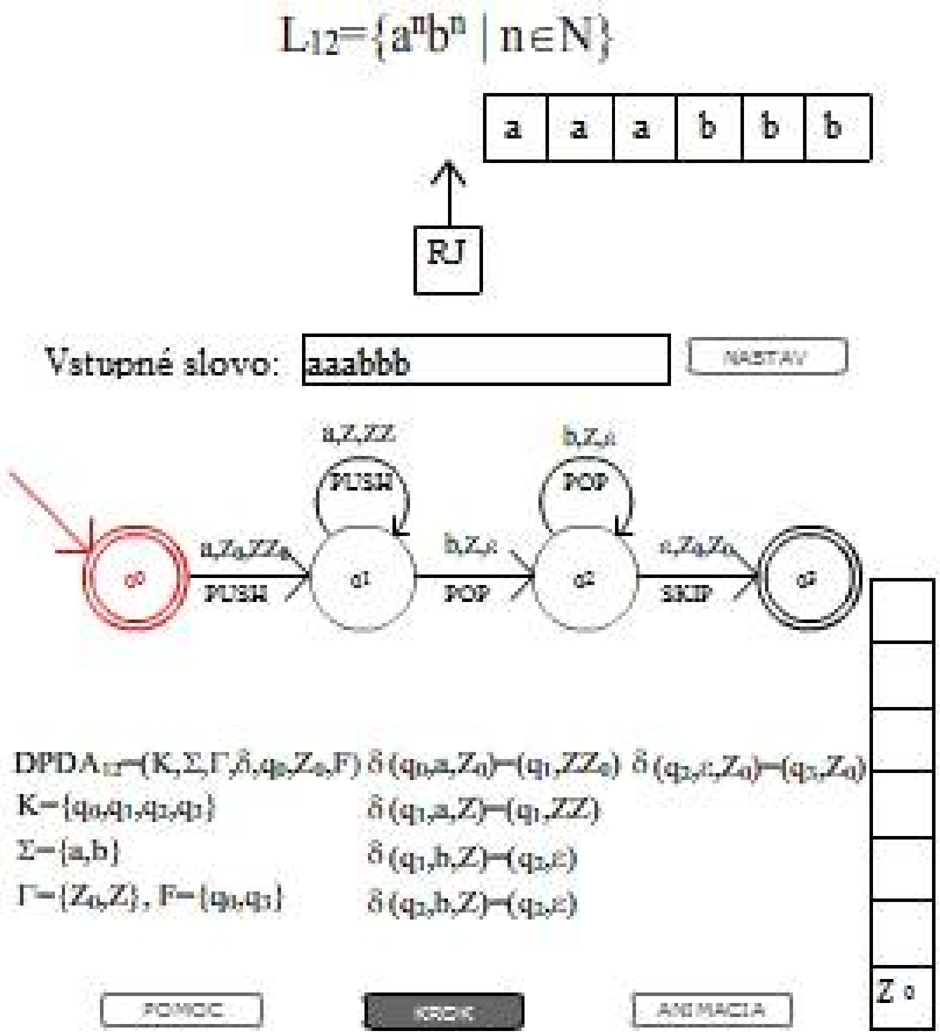
L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,ZaZ
PUSH
c,Z,Z
SKIP
a,Za,ε
POP
ε,Z0,Z0
SKIP
b,Z,ZbZ
PUSH
b,Zb,ε
POP
DPDA11=(K,Σ,Γ,δ,q0,Z0,F) δ (q0,a,Z)=(q0,ZaZ) δ (q1,b,Zb)=(q1,ε)
K={q0,q1,q2} δ (q0,b,Z)=(q0,ZbZ) δ (q1,ε,Z0)=(q2,Z0)
Σ={a,b,c} δ (q0,c,Z)=(q1,Z)
Γ={Z0,Za,Zb}, F={q2} δ (q1,a,Za)=(q1,ε)
Z0
POMOC
KROK
ANIMACIA

Frame 27

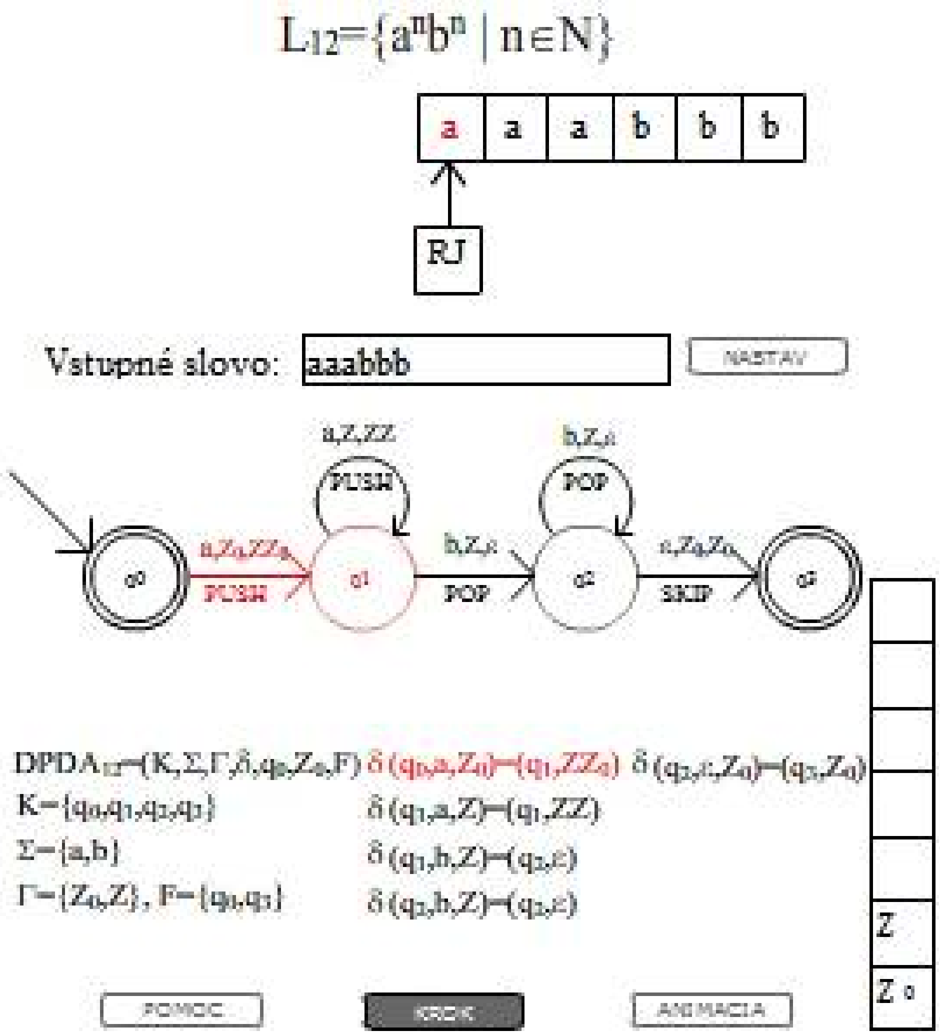


L11={wcwR | w∈{a,b}*}
a b a b c b a b a
RJ
Vstupné slovo: ababcbaba
NASTAV
a,Z,ZaZ
PUSH
b,Z,ZBZ
PUSH
c,Z,Z
SKIP
a,ZA,ε
POP
b,ZB,ε
POP
ε,Z0,Z0
SKIP
q0
q1
q2
DPDA11=(K,Σ,Γ,δ,q0,Z0,F) δ (q0,a,Z)=(q0,ZAZ) δ (q1,b,ZB)=(q1,ε)
K={q0,q1,q2} δ (q0,b,Z)=(q0,ZBZ) δ (q1,ε,Z0)=(q2,Z0)
Σ={a,b,c} δ (q0,c,Z)=(q1,Z)
Γ={Z0,ZA,ZB}, F={q2} δ (q1,a,ZA)=(q1,ε)
Automat skončil v stave q2 => AKCEPTUJEM
POMOC
KROK
ANIMACIA
Z0

Frame 30

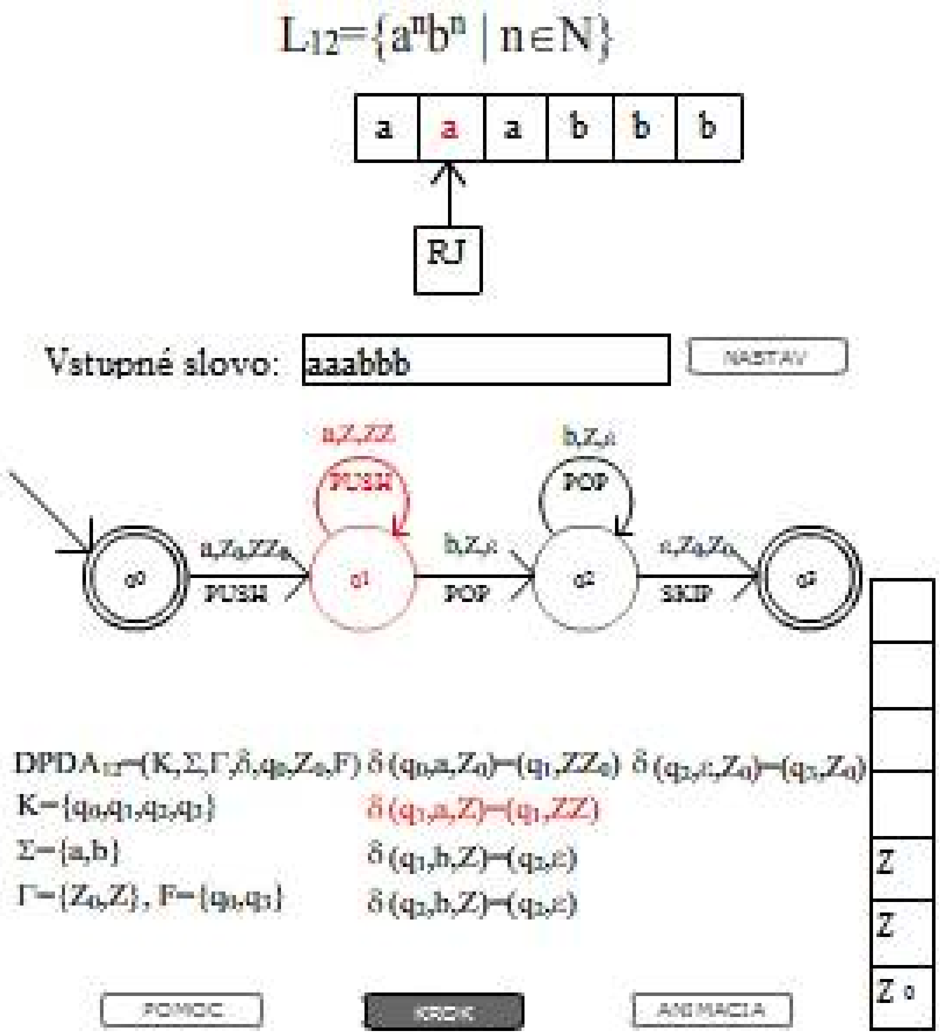

L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo:
aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
q0
q1
b,Z,ε
POP
q2
ε,Z0,Z0
SKIP
q3
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Z0
POMOC
KROK
ANIMACIA

Frame 2

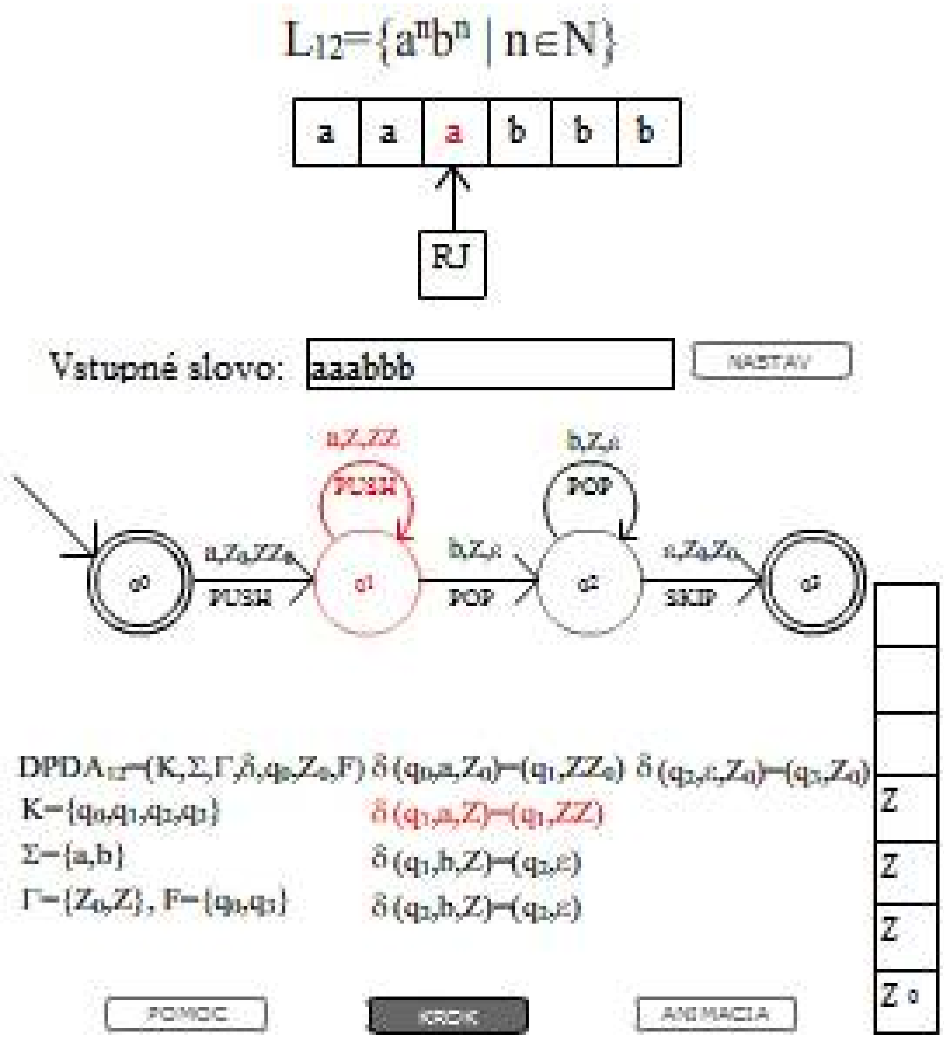


L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo:
aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
b,Z,ε
POP
ε,Z0,Z0
SKIP
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Z
Z0
POMOC
KROK
ANIMACIA

Frame 3

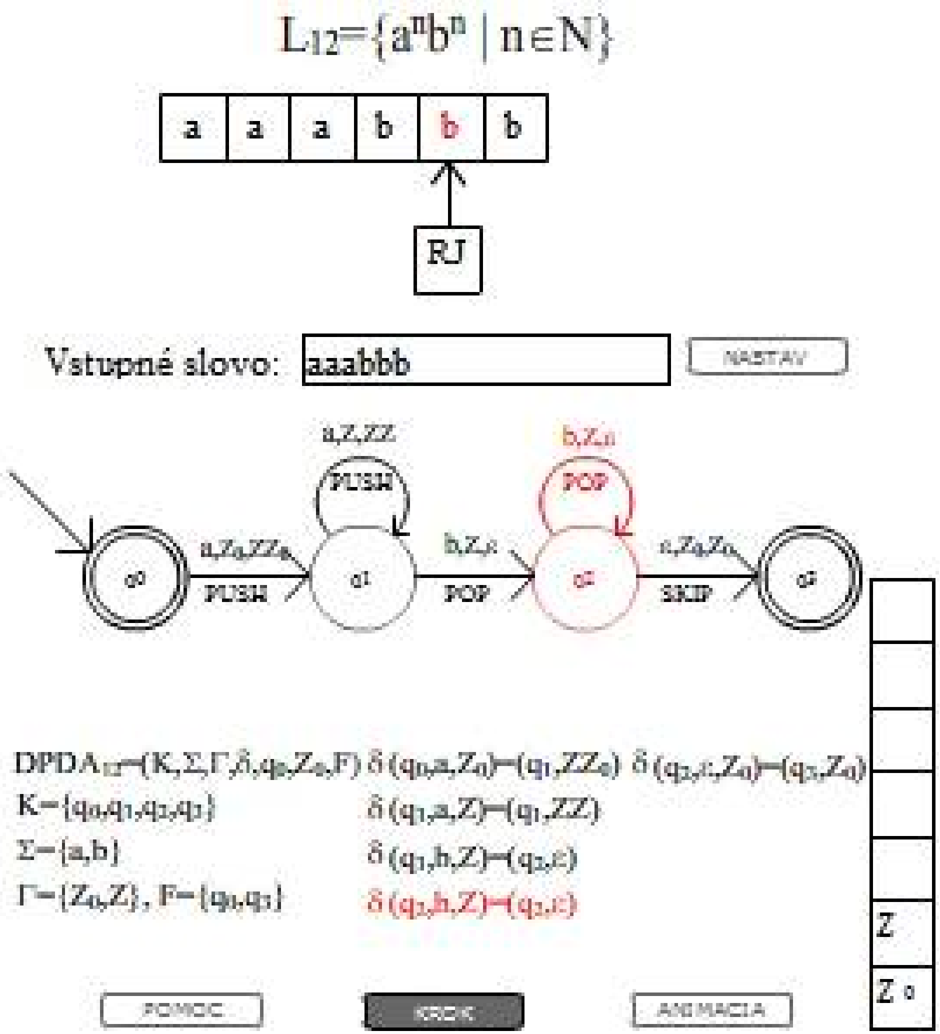

L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo: aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
b,Z,ε
POP
ε,Z0,Z0
SKIP
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Z
Z
Z0
POMOC
KROK
ANIMACIA

Frame 6

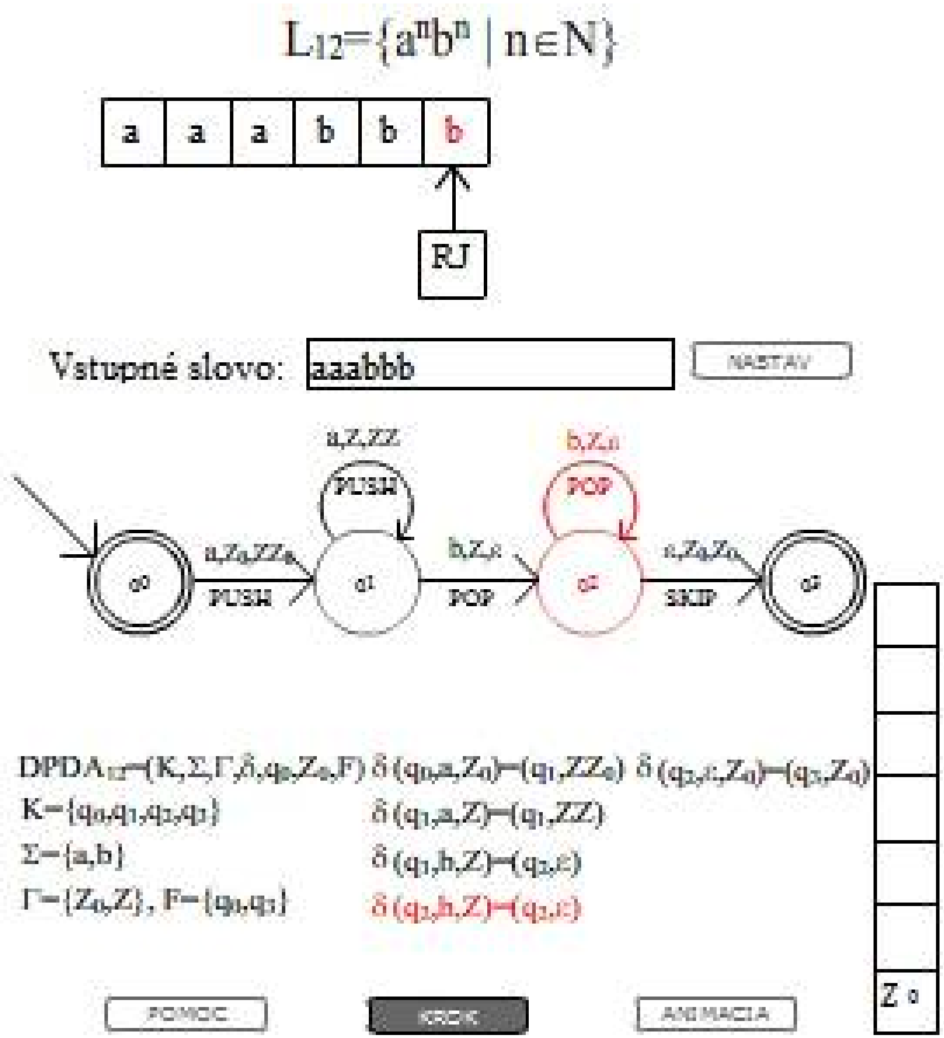


L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo: aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
b,Z,ε
POP
ε,Z0,Z0
SKIP
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Z
Z
Z
Z0
POMOC
KROK
ANIMACIA

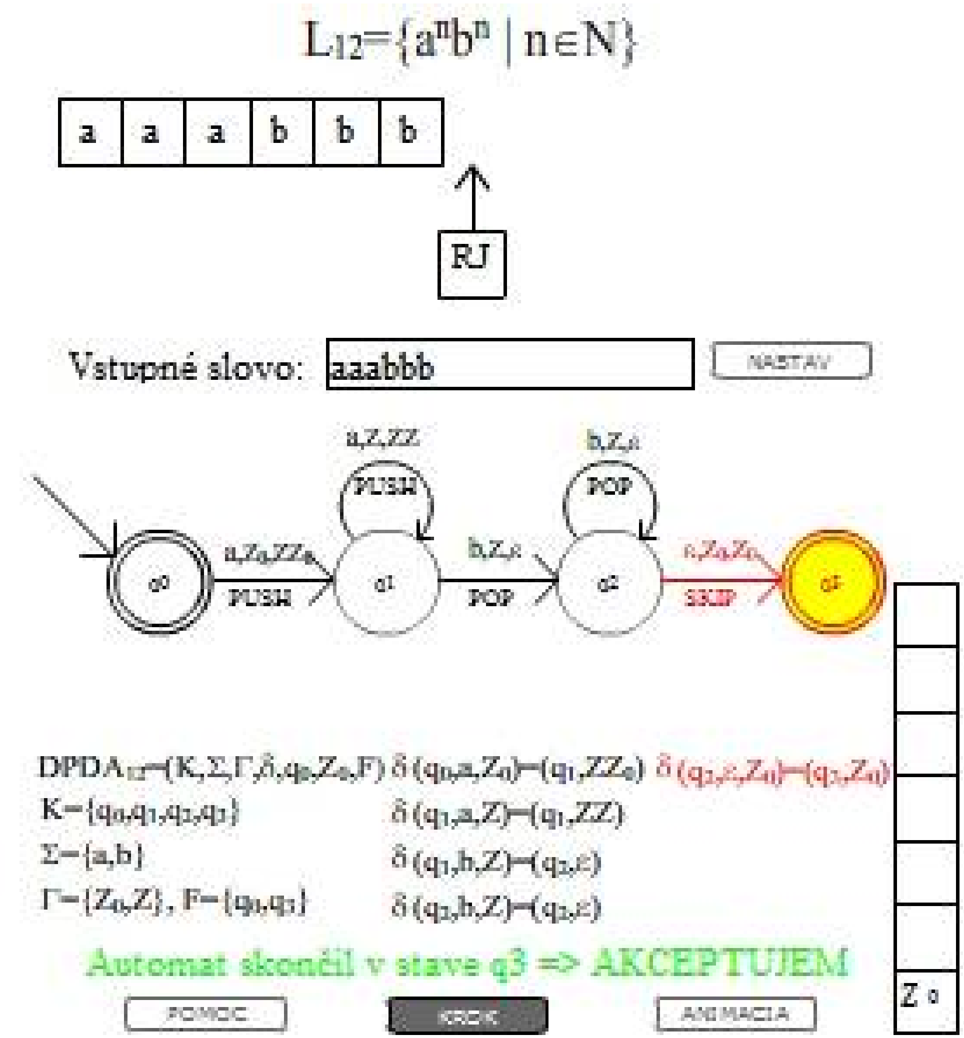
Frame 9


L12={a^n b^n | n∈N}
a a a b b b
RJ
Vstupné slovo: aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
b,Z,ε
POP
ε,Z0,Z0
SKIP
q0
q1
q2
q3
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Z
Z0
POMOC
KROK
ANIMACIA

Frame 15

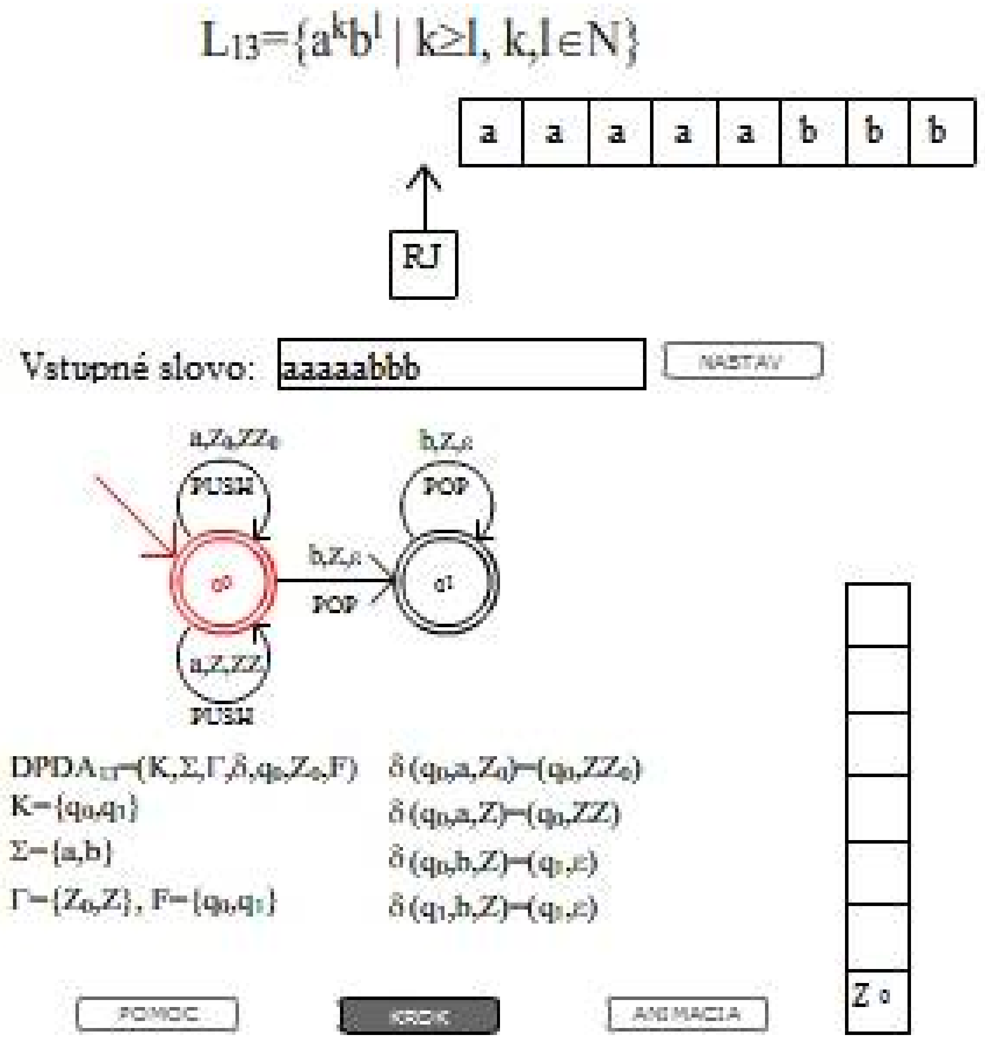

L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo: aaabbb
NASTAV
PUSH
POP
PUSH
POP
SKIP
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
POMOC
KROK
ANIMACIA

Frame 18

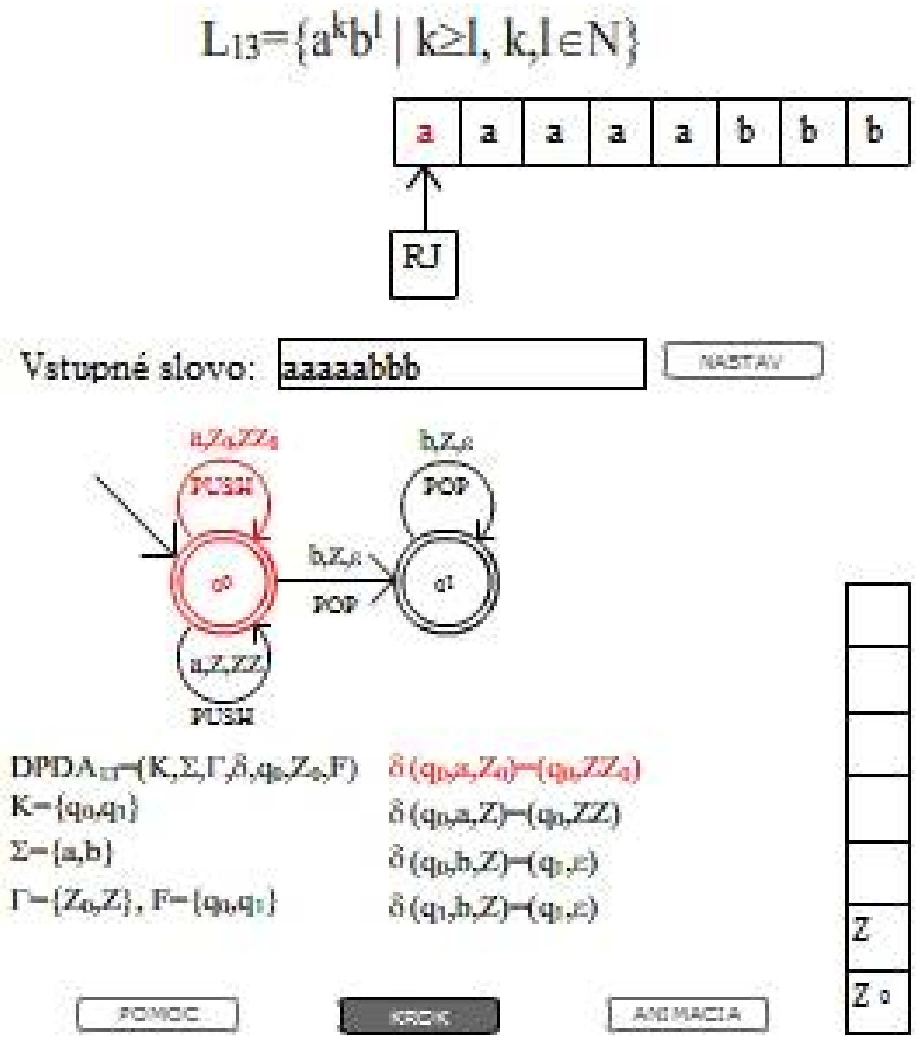


L12={aⁿbⁿ | n∈N}
a a a b b b
RJ
Vstupné slovo: aaabbb
NASTAV
a,Z,ZZ
PUSH
b,Z,ε
POP
a,Z0,ZZ0
PUSH
b,Z,ε
POP
ε,Z0,Z0
SKIP
q0
q1
q2
q3
DPDA12=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b}
Γ={Z0,Z}, F={q0,q3}
δ(q0,a,Z0)=(q1,ZZ0)
δ(q1,a,Z)=(q1,ZZ)
δ(q1,b,Z)=(q2,ε)
δ(q2,b,Z)=(q2,ε)
δ(q2,ε,Z0)=(q3,Z0)
Automat skončil v stave q3 => AKCEPTUJEM
POMOC
KROK
ANIMACIA
Z0

Frame 21

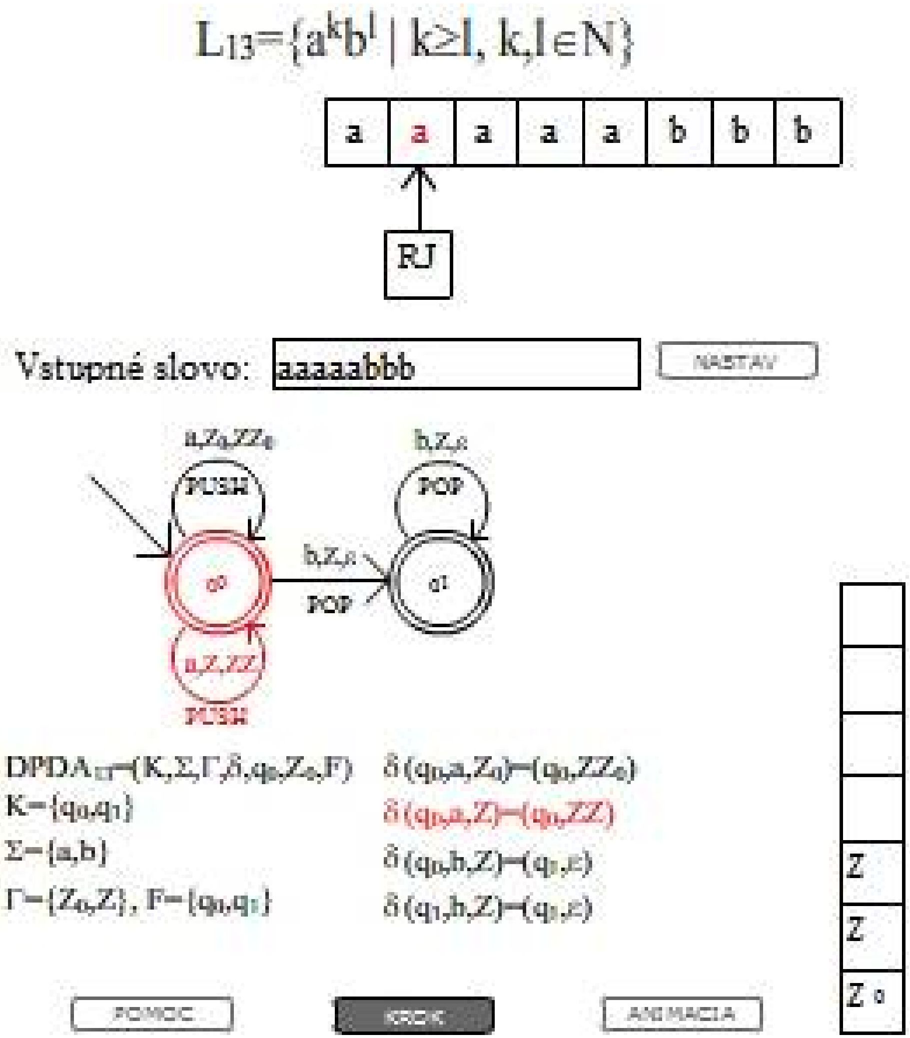

$L_{13}=\{a^k b^l \mid k \geq l,\ k,l \in N\}$
a a a a a b b b
RJ
Vstupné slovo:
aaaaabbb
NASTAV
PUSH
POP
PUSH
$DPDA_{13}=(K,\Sigma,\Gamma,\delta,q_0,Z_0,F)$
$K=\{q_0,q_1\}$
$\Sigma=\{a,b\}$
$\Gamma=\{Z_0,Z\}$, $F=\{q_0,q_1\}$
$\delta(q_0,a,Z_0)=(q_0,ZZ_0)$
$\delta(q_0,a,Z)=(q_0,ZZ)$
$\delta(q_0,b,Z)=(q_1,\varepsilon)$
$\delta(q_1,b,Z)=(q_1,\varepsilon)$
POMOC
KROK
ANIMACIA
Z_0

Frame 2

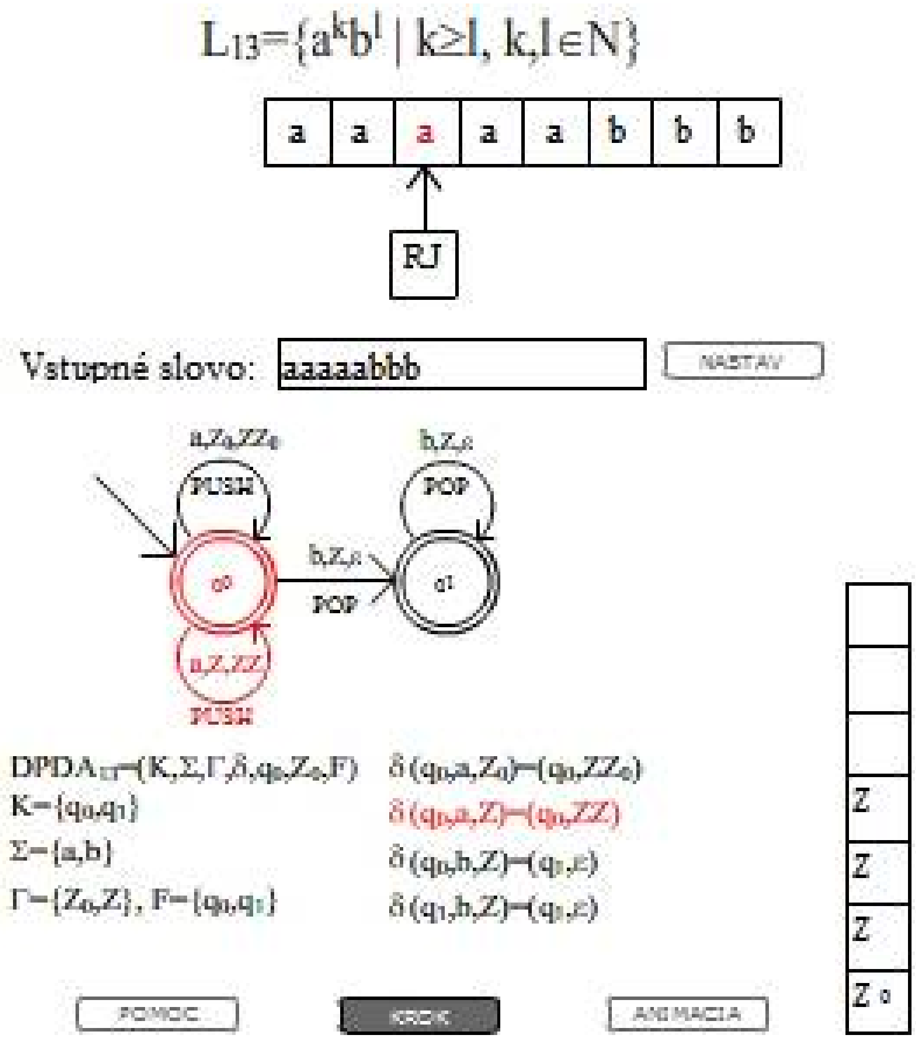


L13={a^k b^l | k≥l, k,l∈N}
a
a
a
a
a
b
b
b
RJ
Vstupné slovo:
aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
q0
q1
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
Z
Z0
POMOC
KROK
ANIMACIA

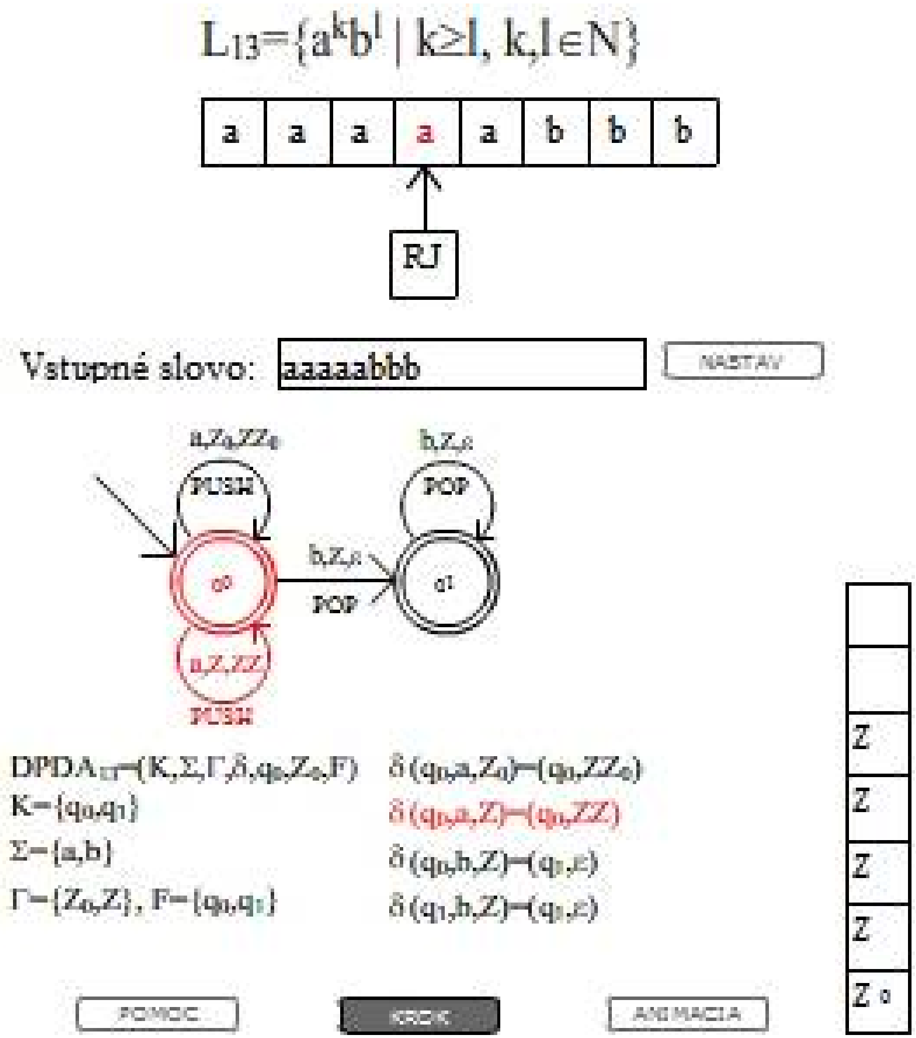
Frame 3


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo: aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
b,Z,ε
POP
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA
Z
Z
Z0

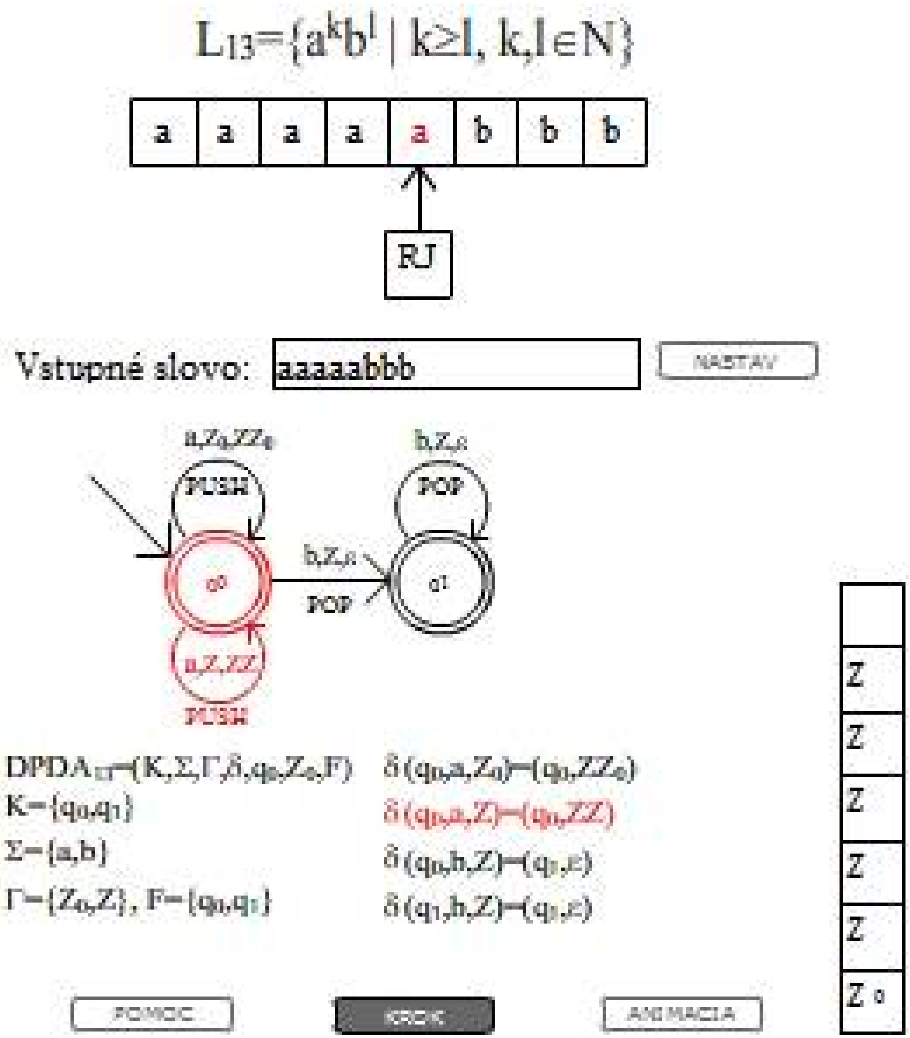
Frame 6


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo: aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
b,Z,ε
POP
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA
Z
Z
Z
Z0

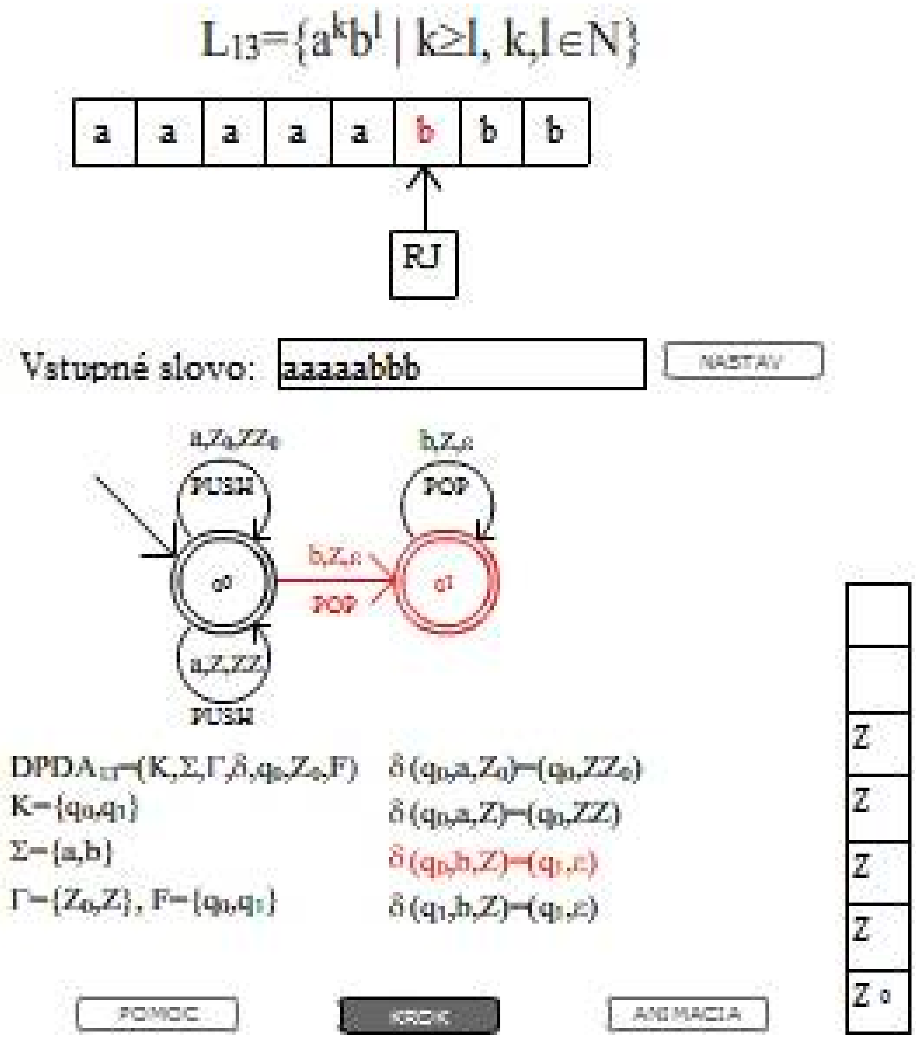
Frame 9


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo: aaaaabbb
NASTAV
PUSH
POP
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA

Frame 12

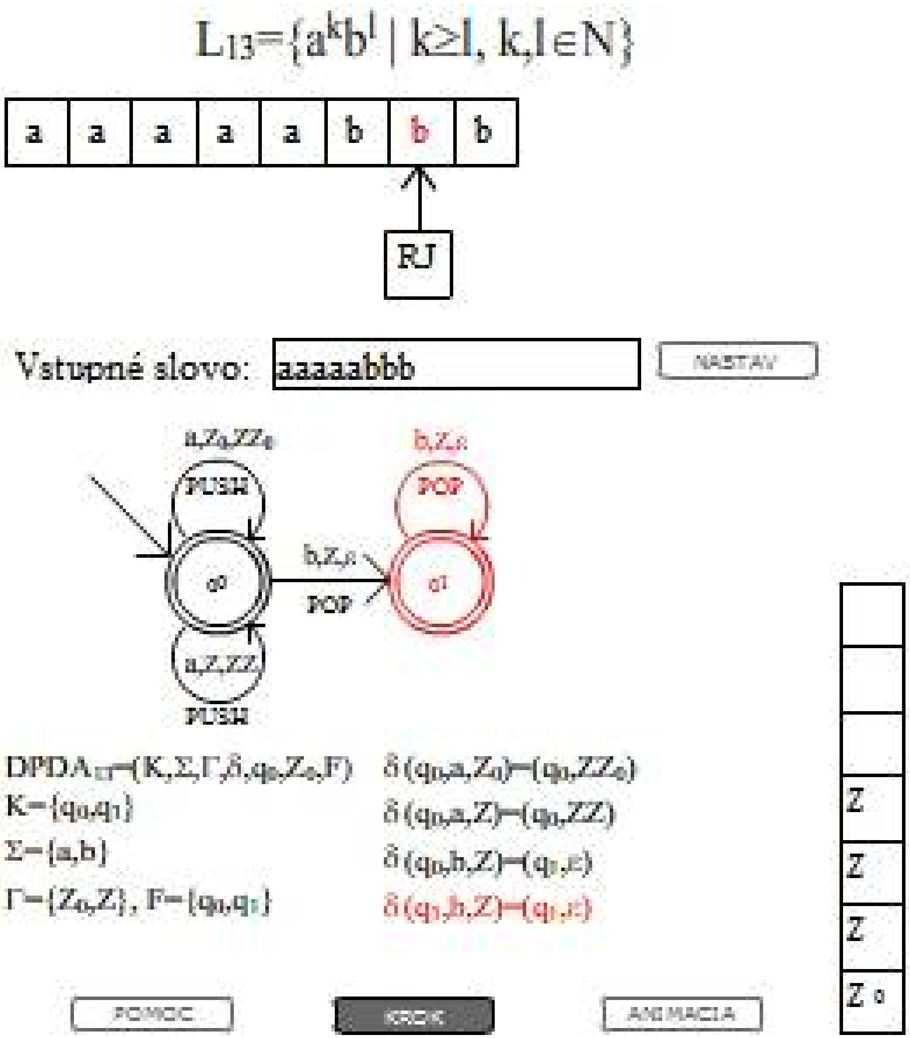

L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo:
aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
b,Z,ε
POP
q0
q1
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA
Z
Z
Z
Z
Z
Z0

Frame 15

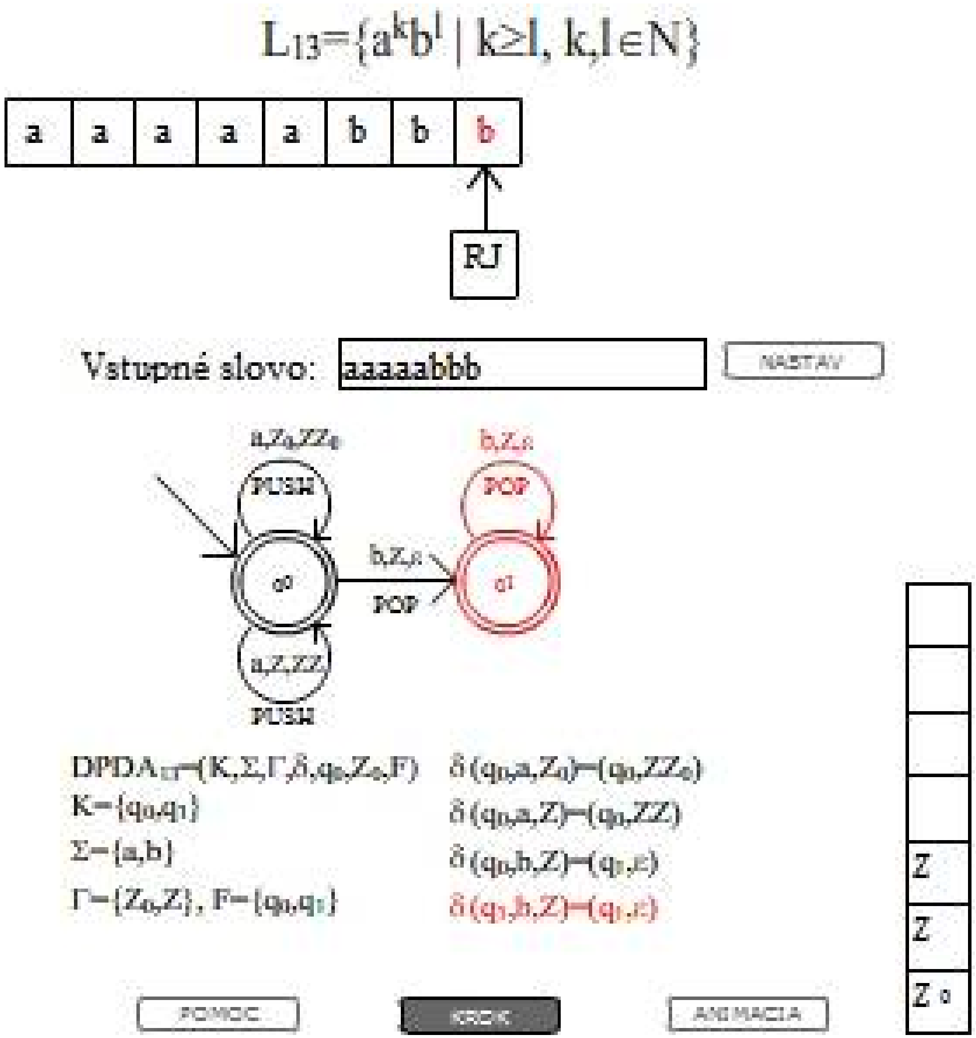


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo:
aaaaabbb
NASTAV
PUSH
POP
b,Z,ε
POP
PUSH
DPDA=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA
Z
Z
Z
Z
Z0

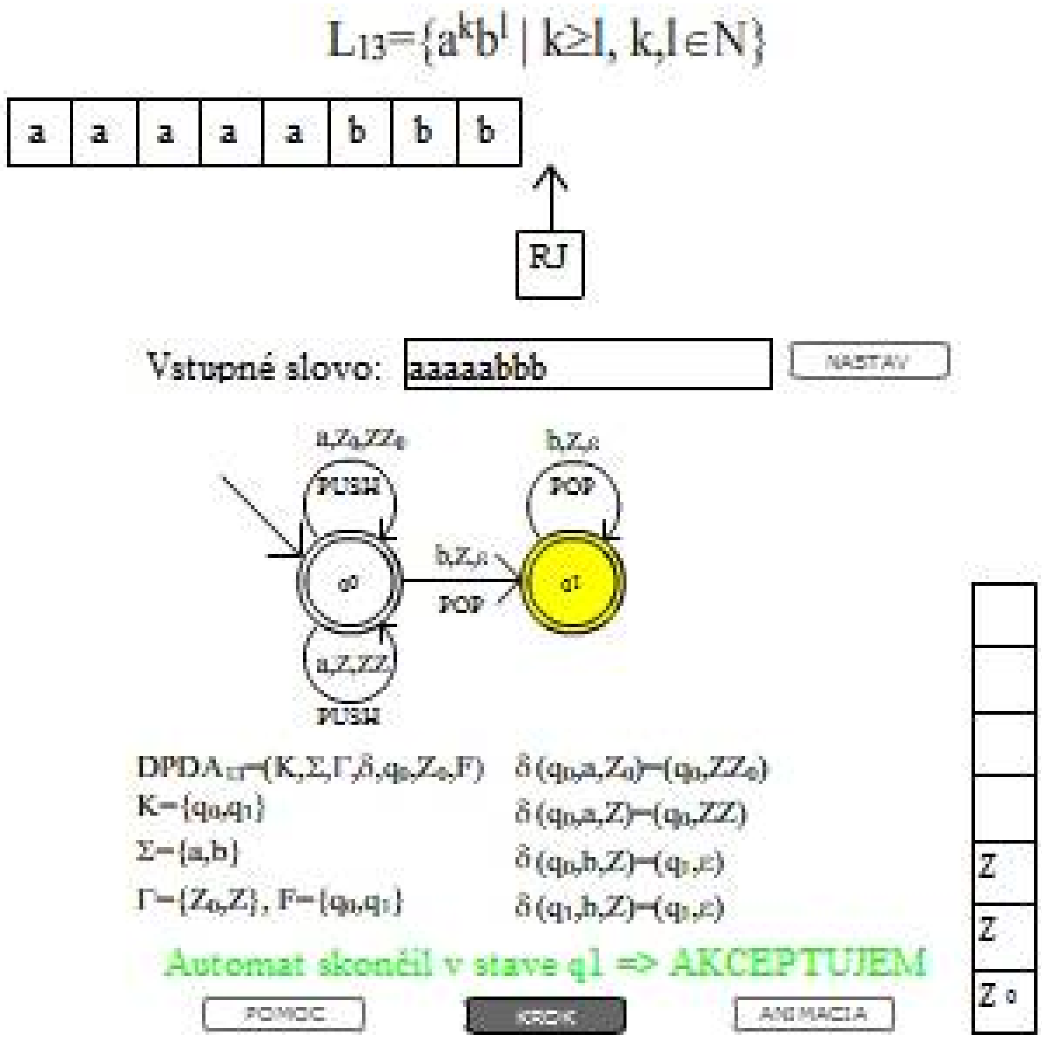
Frame 18


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo:
aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
b,Z,ε
POP
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
Z
Z
Z
Z0
POMOC
KROK
ANIMACIA

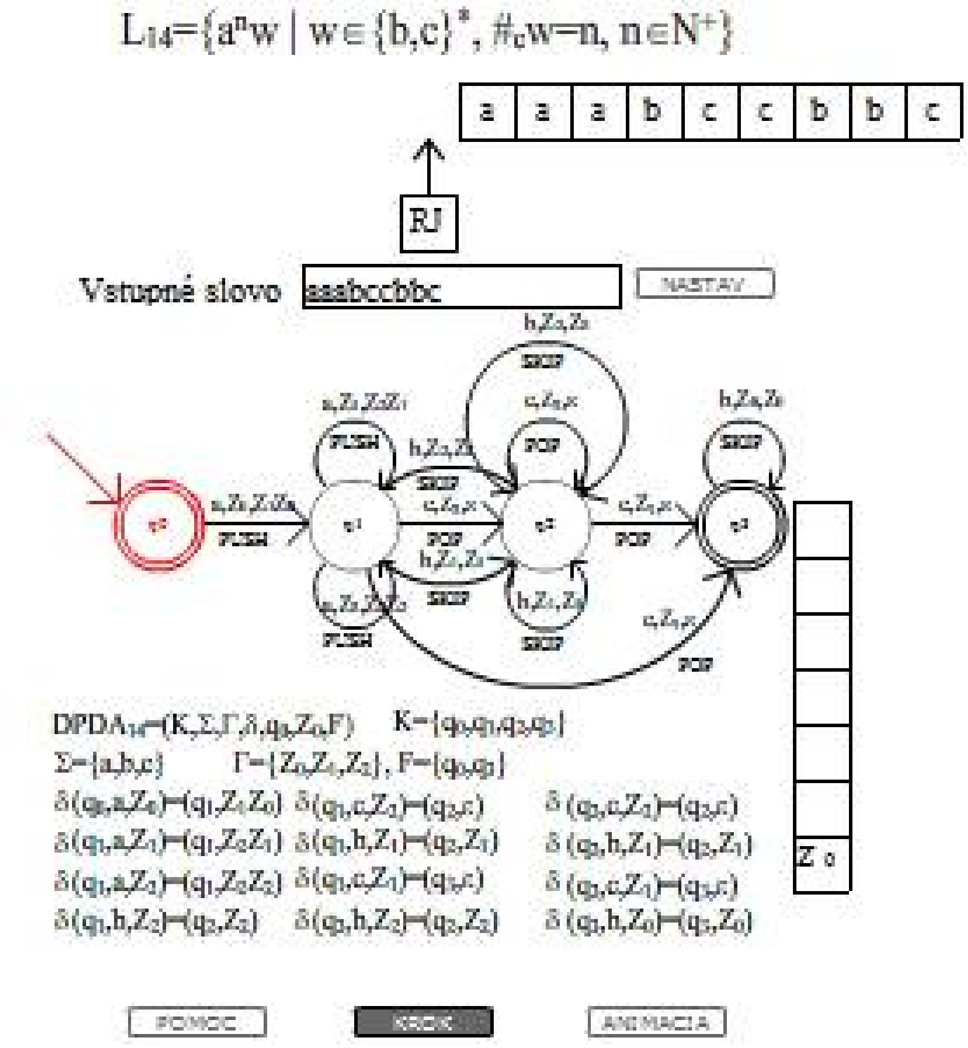
Frame 21


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo: aaaaabbb
NASTAV
a,Z0,ZZ0
PUSH
b,Z,ε
POP
b,Z,ε
POP
a,Z,ZZ
PUSH
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
POMOC
KROK
ANIMACIA
Z
Z
Z0

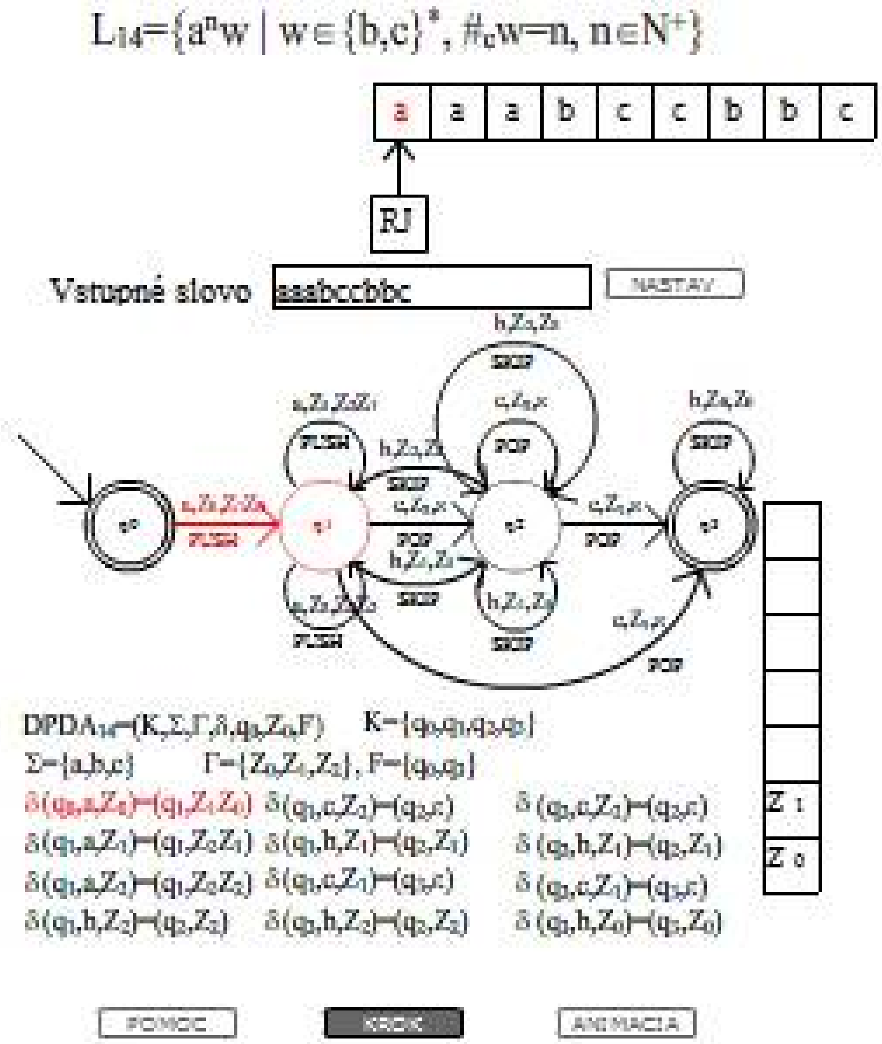
Frame 24


L13={a^k b^l | k≥l, k,l∈N}
a a a a a b b b
RJ
Vstupné slovo: aaaaabbb
NASTAV
PUSH
POP
POP
PUSH
q0
q1
DPDA13=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1}
Σ={a,b}
Γ={Z0,Z}, F={q0,q1}
δ(q0,a,Z0)=(q0,ZZ0)
δ(q0,a,Z)=(q0,ZZ)
δ(q0,b,Z)=(q1,ε)
δ(q1,b,Z)=(q1,ε)
Automat skončil v stave q1 => AKCEPTUJEM
POMOC
KROK
ANIMACIA
Z
Z
Z0

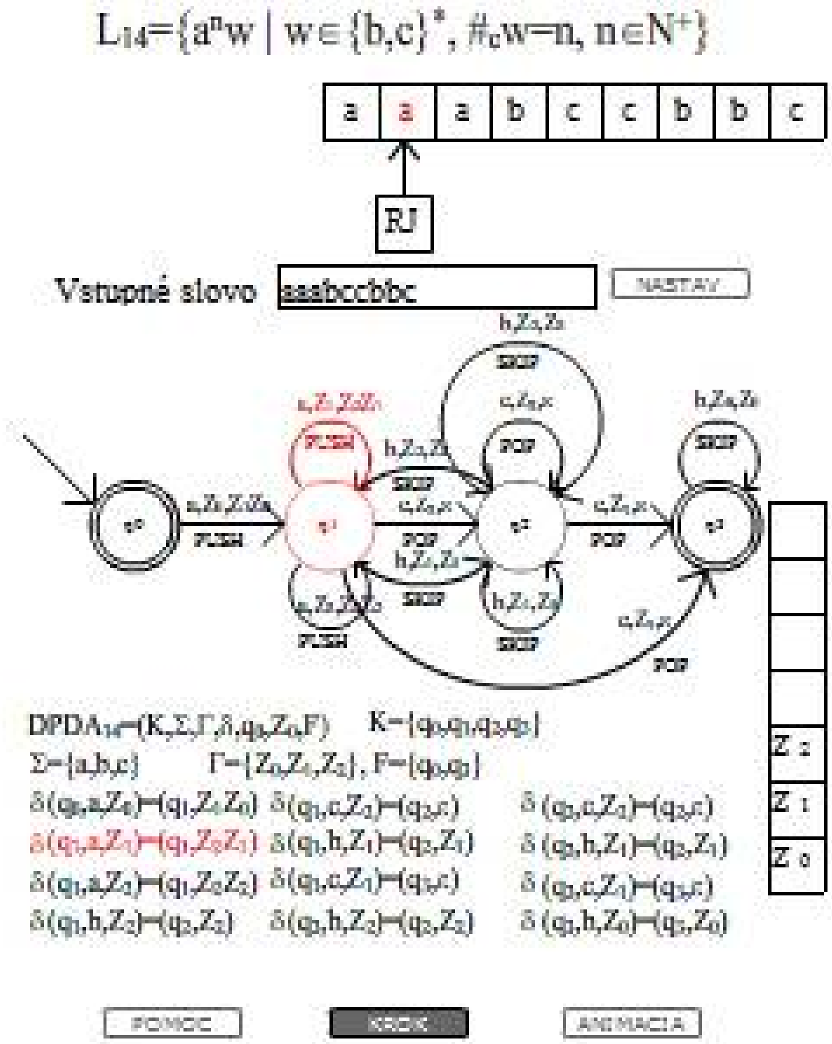
Frame 27


L14={aⁿw | w∈{b,c}*, #cw=n, n∈N⁺}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0) δ(q1,c,Z2)=(q2,ε) δ(q2,c,Z2)=(q2,ε)
δ(q1,a,Z1)=(q1,Z2Z1) δ(q1,b,Z1)=(q2,Z1) δ(q2,b,Z1)=(q2,Z1)
δ(q1,a,Z2)=(q1,Z2Z2) δ(q1,c,Z1)=(q3,ε) δ(q2,c,Z1)=(q3,ε)
δ(q1,b,Z2)=(q2,Z2) δ(q2,b,Z2)=(q2,Z2) δ(q3,b,Z0)=(q3,Z0)
POMOC
KROK
ANIMACIA

Frame 2

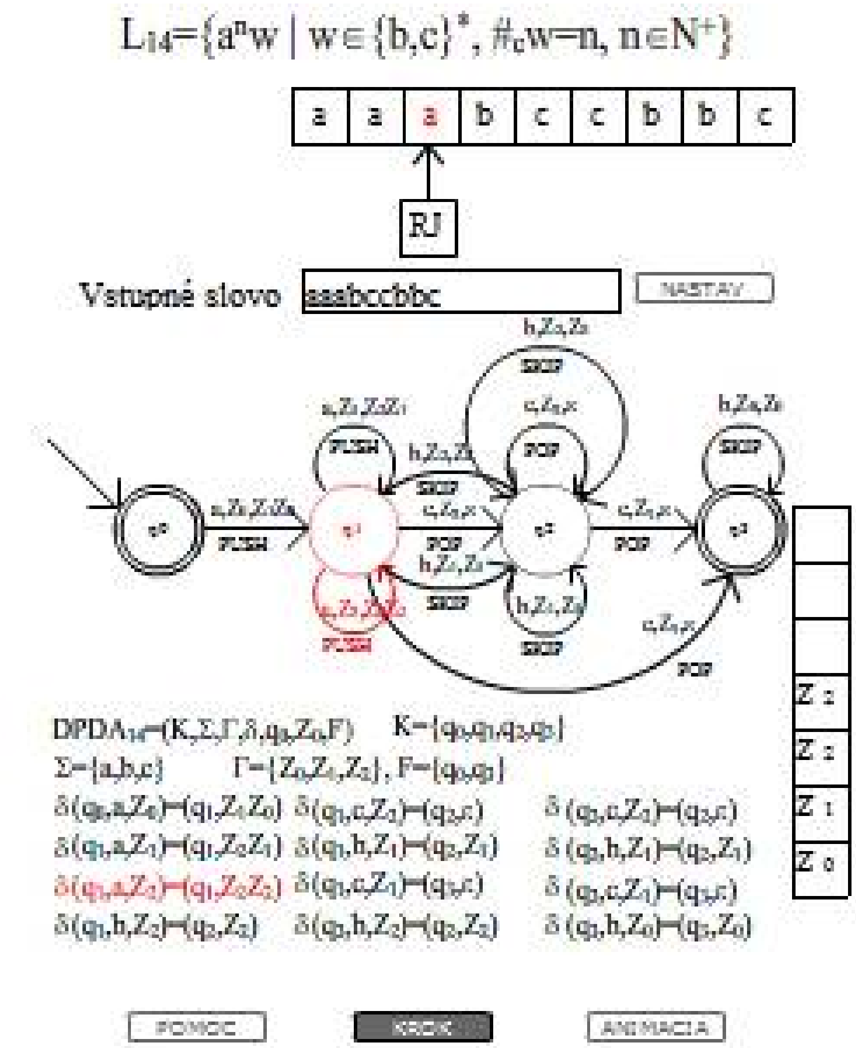

L14={aⁿw | w∈{b,c}*, #cw=n, n∈N⁺}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
SKIP
POP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0) δ(q1,c,Z2)=(q2,ε) δ(q2,c,Z2)=(q2,ε)
δ(q1,a,Z1)=(q1,Z2Z1) δ(q1,b,Z1)=(q2,Z1) δ(q2,b,Z1)=(q2,Z1)
δ(q1,a,Z2)=(q1,Z2Z2) δ(q1,c,Z1)=(q3,ε) δ(q2,c,Z1)=(q3,ε)
δ(q1,b,Z2)=(q2,Z2) δ(q2,b,Z2)=(q2,Z2) δ(q3,b,Z0)=(q3,Z0)
Z1
Z0
POMOC
KROK
ANIMACIA

Frame 3



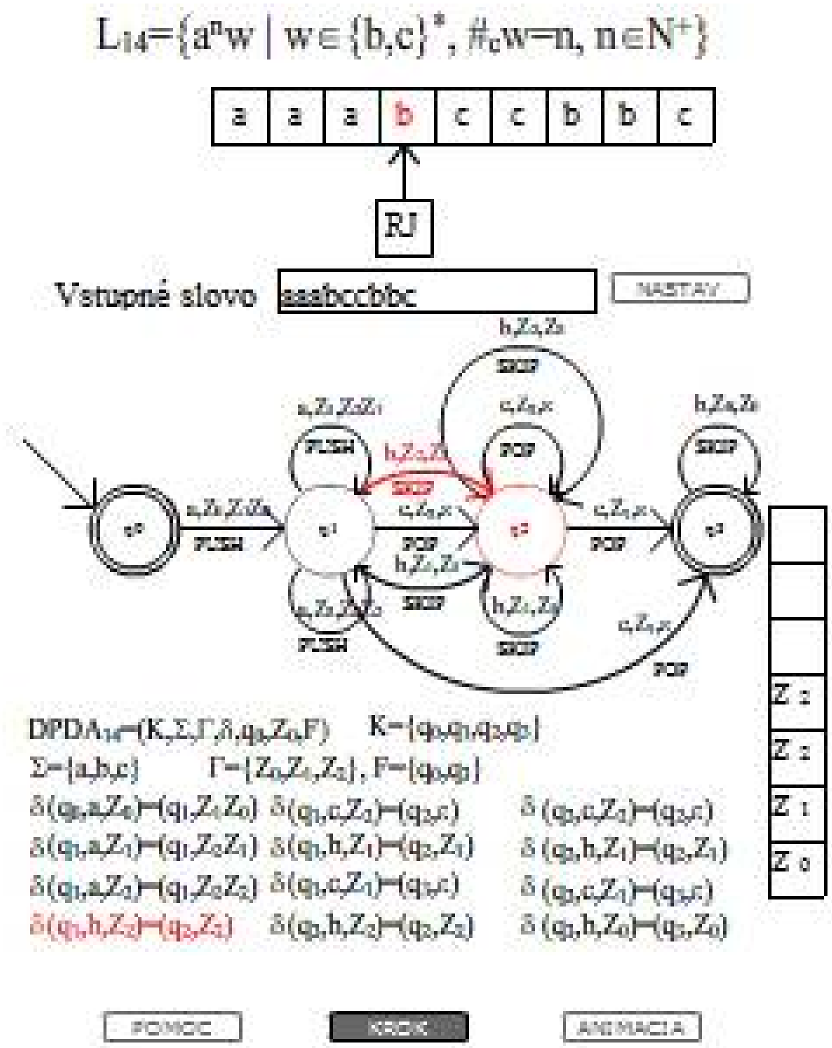
L14={a^n w | w∈{b,c}*, #c w=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
POP
SKIP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0) δ(q1,c,Z2)=(q2,ε) δ(q2,c,Z1)=(q3,ε)
δ(q1,a,Z1)=(q1,Z2Z1) δ(q1,b,Z1)=(q2,Z1) δ(q2,b,Z1)=(q2,Z1)
δ(q1,a,Z2)=(q1,Z2Z2) δ(q1,c,Z1)=(q3,ε) δ(q3,c,Z1)=(q3,ε)
δ(q1,b,Z2)=(q2,Z2) δ(q2,b,Z2)=(q2,Z2) δ(q3,b,Z0)=(q3,Z0)
Z2
Z1
Z0
POMOC
KROK
ANIMACIA

Frame 5

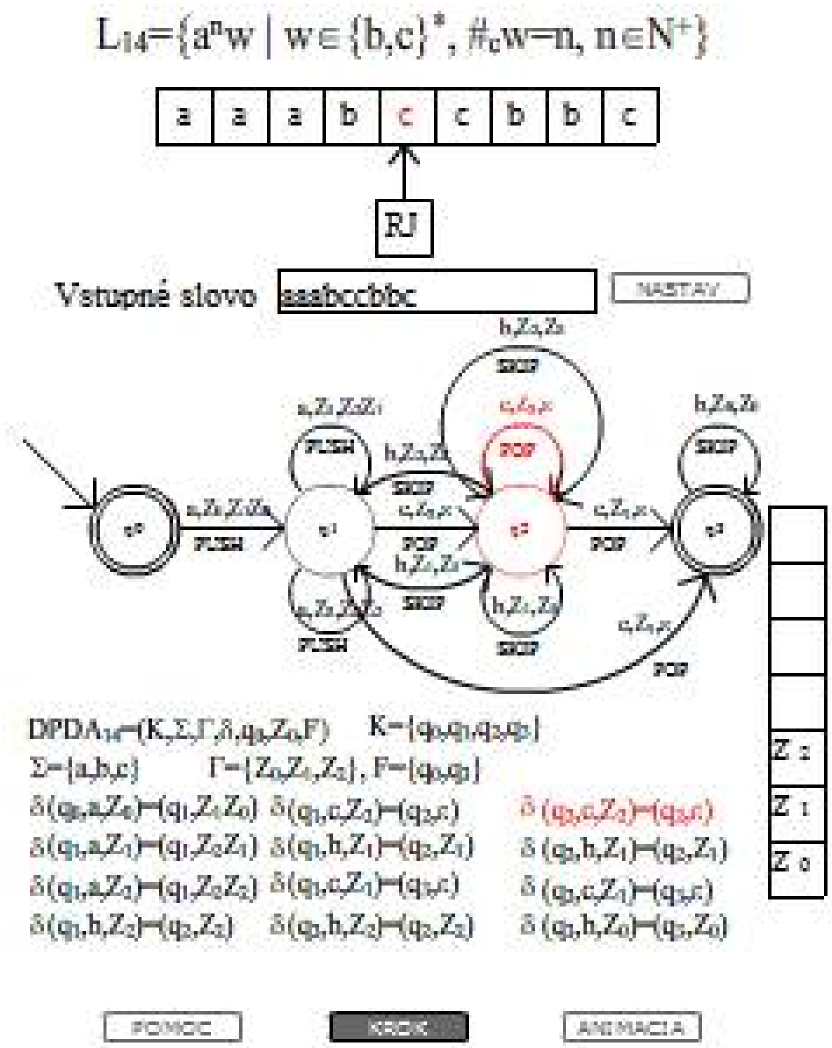


L14={a^n w | w∈{b,c}*, #cw=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0) δ(q1,c,Z2)=(q2,ε) δ(q2,c,Z2)=(q2,ε)
δ(q1,a,Z1)=(q1,Z2Z1) δ(q1,b,Z1)=(q2,Z1) δ(q2,b,Z1)=(q2,Z1)
δ(q1,a,Z2)=(q1,Z2Z2) δ(q1,c,Z1)=(q3,ε) δ(q2,c,Z1)=(q3,ε)
δ(q1,b,Z2)=(q2,Z2) δ(q2,b,Z2)=(q2,Z2) δ(q3,b,Z0)=(q3,Z0)
POMOC
KROK
ANIMACIA

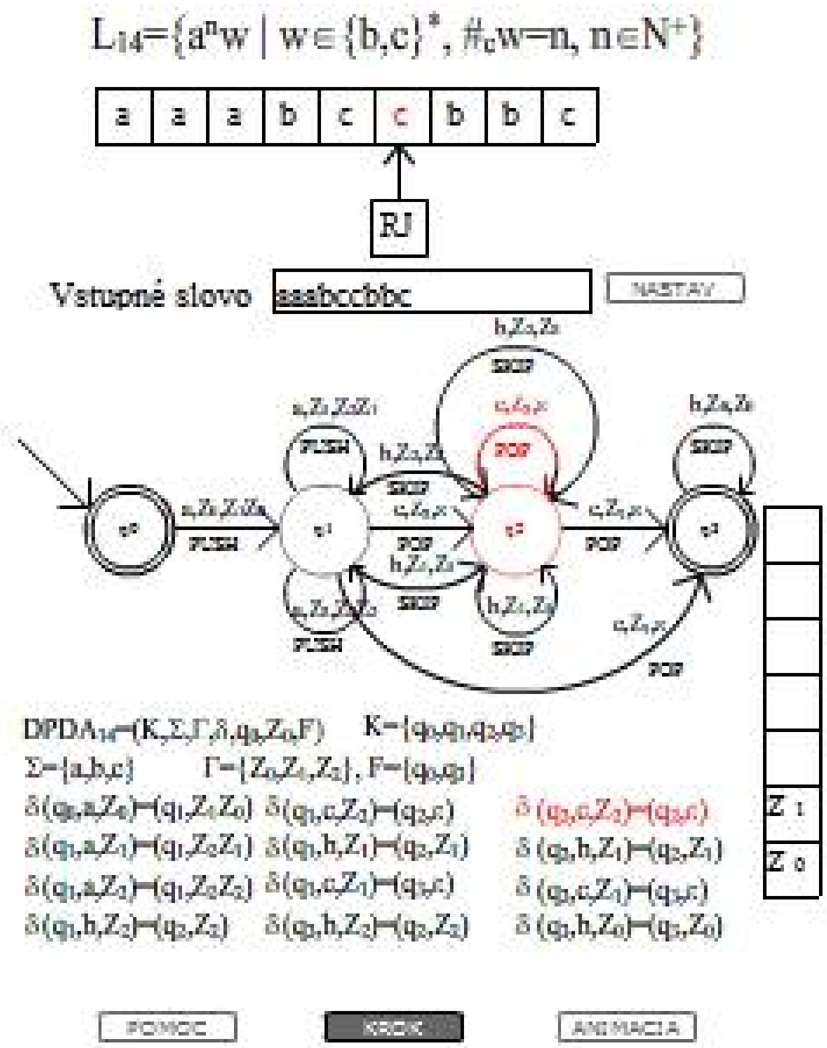
Frame 7


L14={a^n w | w∈{b,c}*, #cw=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
POP
SKIP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F)
Σ={a,b,c}
Z 2
Z 2
Z 1
Z 0

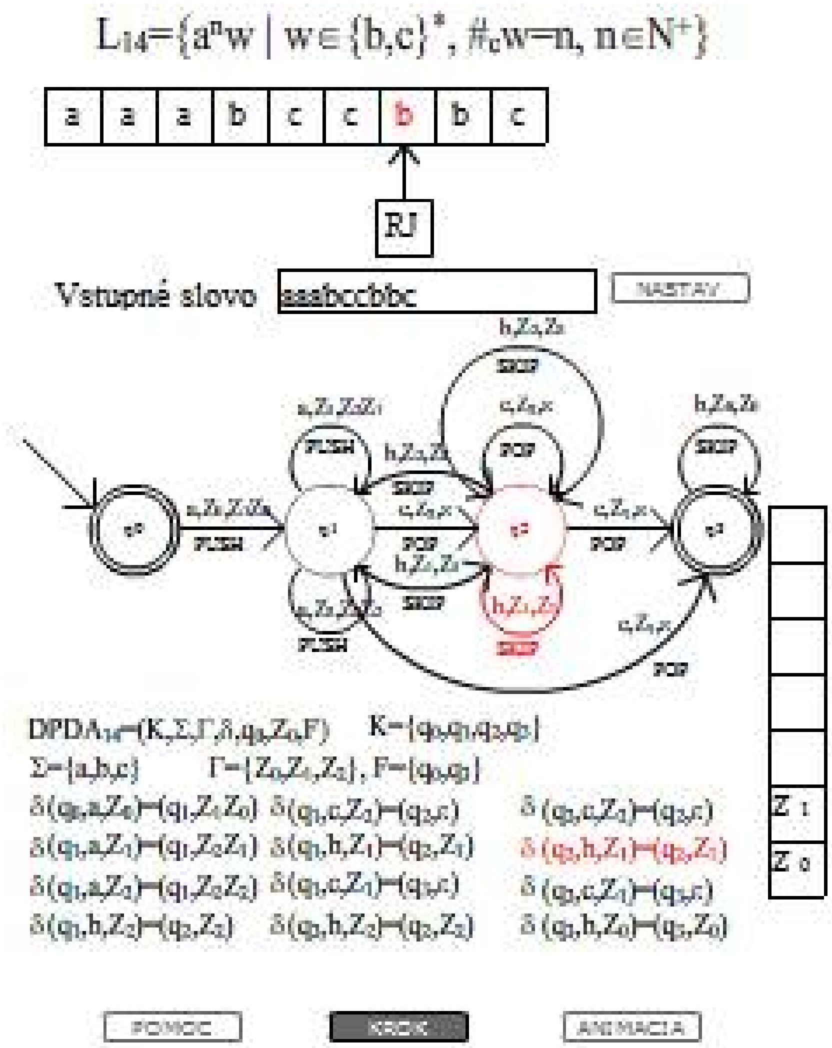
Frame 9


L14={aⁿw | w∈{b,c}*, #cw=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
SKIP
POP
PUSH
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0) δ(q1,c,Z2)=(q2,ε) δ(q2,c,Z2)=(q2,ε)
δ(q1,a,Z1)=(q1,Z2Z1) δ(q1,b,Z1)=(q2,Z1) δ(q2,b,Z1)=(q2,Z1)
δ(q1,a,Z2)=(q1,Z2Z2) δ(q1,c,Z1)=(q3,ε) δ(q2,c,Z1)=(q3,ε)
δ(q1,b,Z2)=(q2,Z2) δ(q2,b,Z2)=(q2,Z2) δ(q3,b,Z0)=(q3,Z0)
Z2
Z1
Z0
POMOC
KROK
ANIMACIA

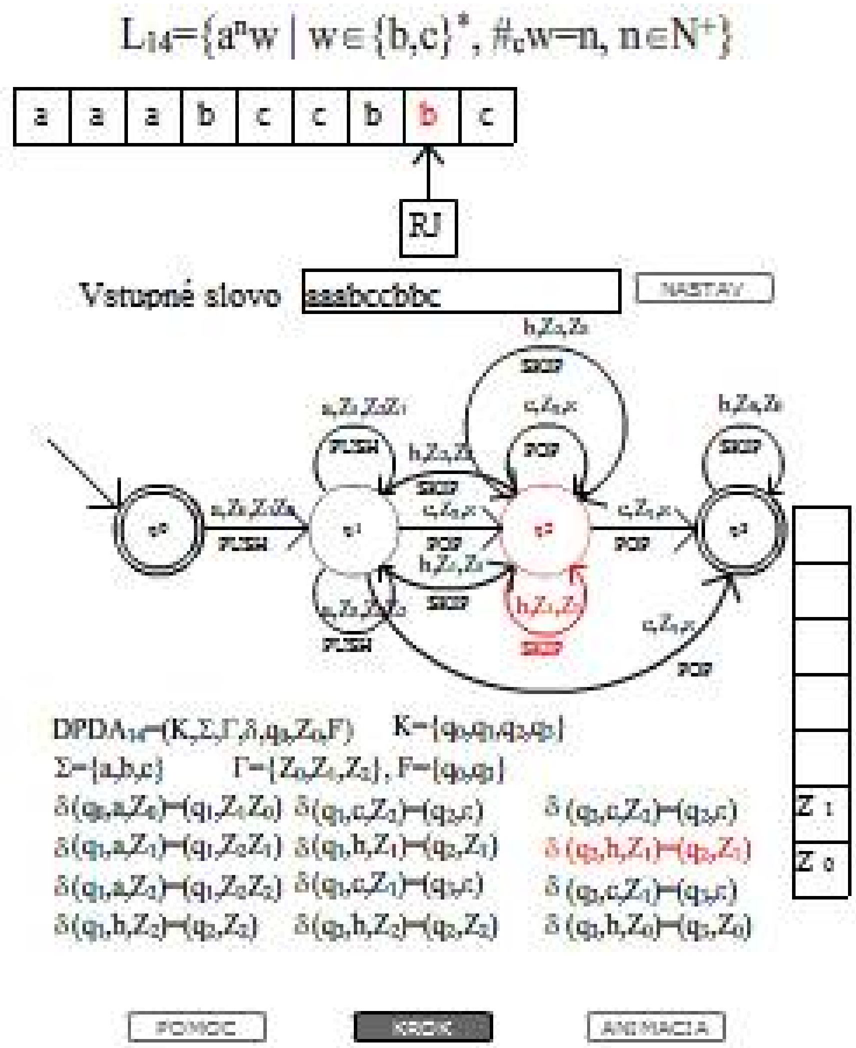
Frame 11


L14={a^n w | w∈{b,c}*, #c w=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
SKIP
POP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b,c}
Γ={Z0,Z1,Z2}, F={q0,q3}
Z 1
Z 0
POMOC
KROK
ANIMACIA

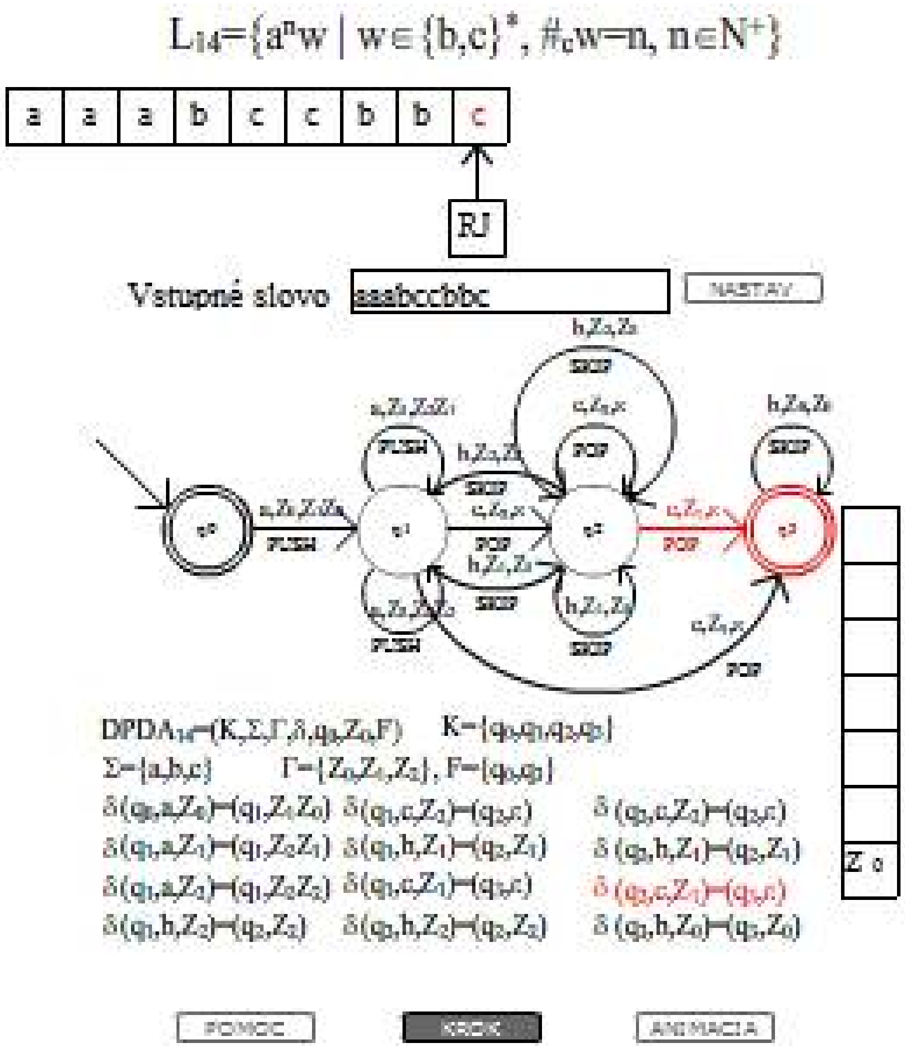
Frame 13


L14={aⁿw | w∈{b,c}*, #cw=n, n∈N⁺}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
POP
SKIP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b,c}
Γ={Z0,Z1,Z2}, F={q0,q3}
δ(q0,a,Z0)=(q1,Z1Z0)
δ(q1,a,Z1)=(q1,Z2Z1)
δ(q1,a,Z2)=(q1,Z2Z2)
δ(q1,b,Z2)=(q2,Z2)
δ(q1,c,Z2)=(q2,ε)
δ(q1,b,Z1)=(q2,Z1)
δ(q1,c,Z1)=(q3,ε)
δ(q2,b,Z2)=(q2,Z2)
δ(q2,c,Z2)=(q2,ε)
δ(q2,b,Z1)=(q2,Z1)
δ(q2,c,Z1)=(q3,ε)
δ(q3,b,Z0)=(q3,Z0)
Z1
Z0
POMOC
KROK
ANIMACIA

Frame 15

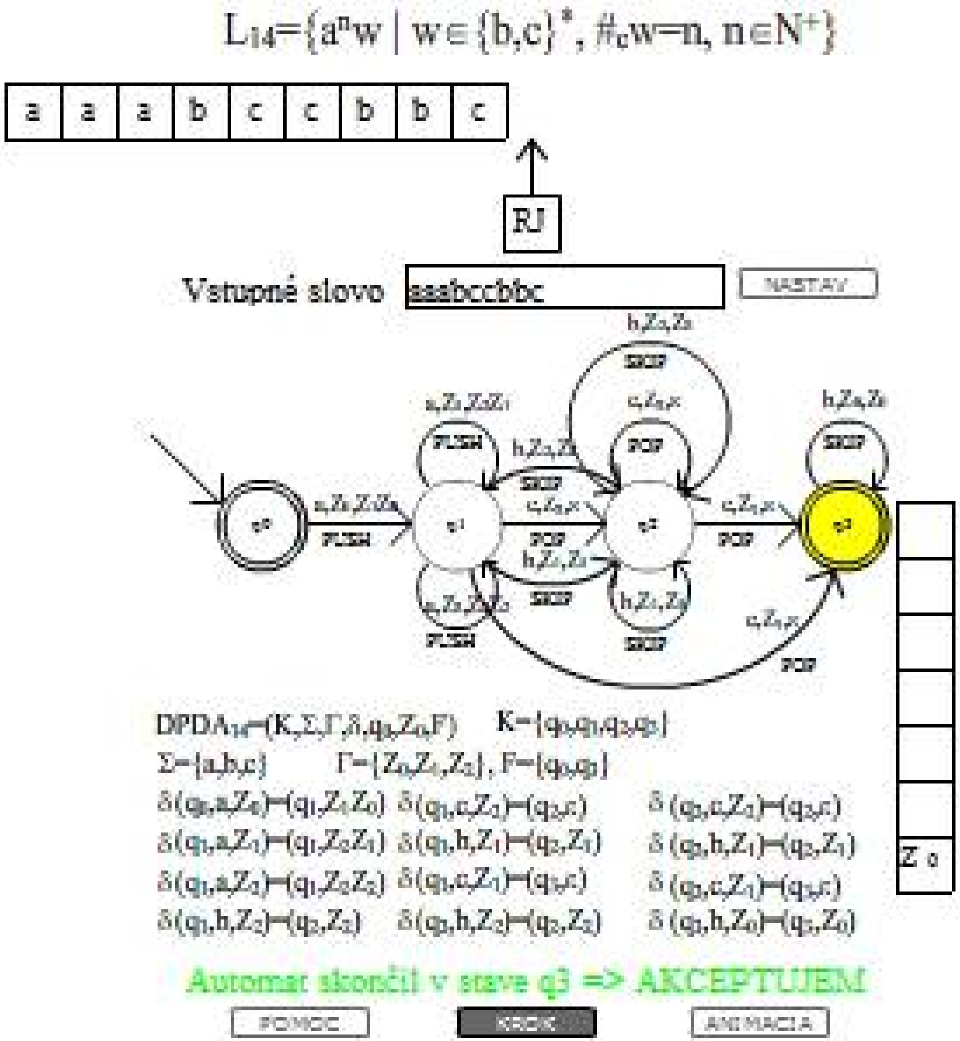

L14={aⁿw | w∈{b,c}*, #cw=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
SKIP
POP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b,c}
Γ={Z0,Z1,Z2}, F={q0,q3}
POMOC
KROK
ANIMACIA

Frame 17



$L_{14}=\{a^n w \mid w\in\{b,c\}^*, \#_c w=n, n\in N^+\}$
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
PUSH
SKIP
POP
DPDA14=(K,Σ,Γ,δ,q0,Z0,F)
K={q0,q1,q2,q3}
Σ={a,b,c}
Γ={Z0,Z1,Z2}, F={q0,q3}
Z 0
POMOC
KROK
ANIMACIA

Frame 19



L14={a^n w | w∈{b,c}*, #_c w=n, n∈N+}
a a a b c c b b c
RJ
Vstupné slovo
aaabccbbc
NASTAV
SKIP
POP
PUSH
DPDA14=(K,Σ,Γ,δ,q0,Z0,F) K={q0,q1,q2,q3}
Σ={a,b,c} Γ={Z0,Z1,Z2}, F={q0,q3}
Automat skončil v stave q3 => AKCEPTUJEM
POMOC
KROK
ANIMACIA

Frame 21

STU
FIIT

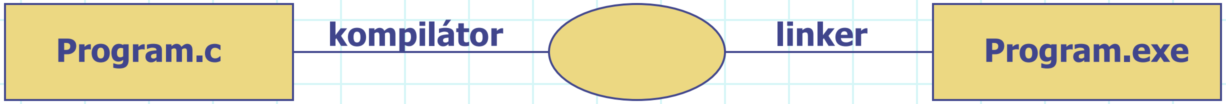
Teoretické základy informatiky

Syntaktická analýza

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Syntaktická analýza

PDA sú formálnym modelom syntaktickej analýzy programovacích jazykov.

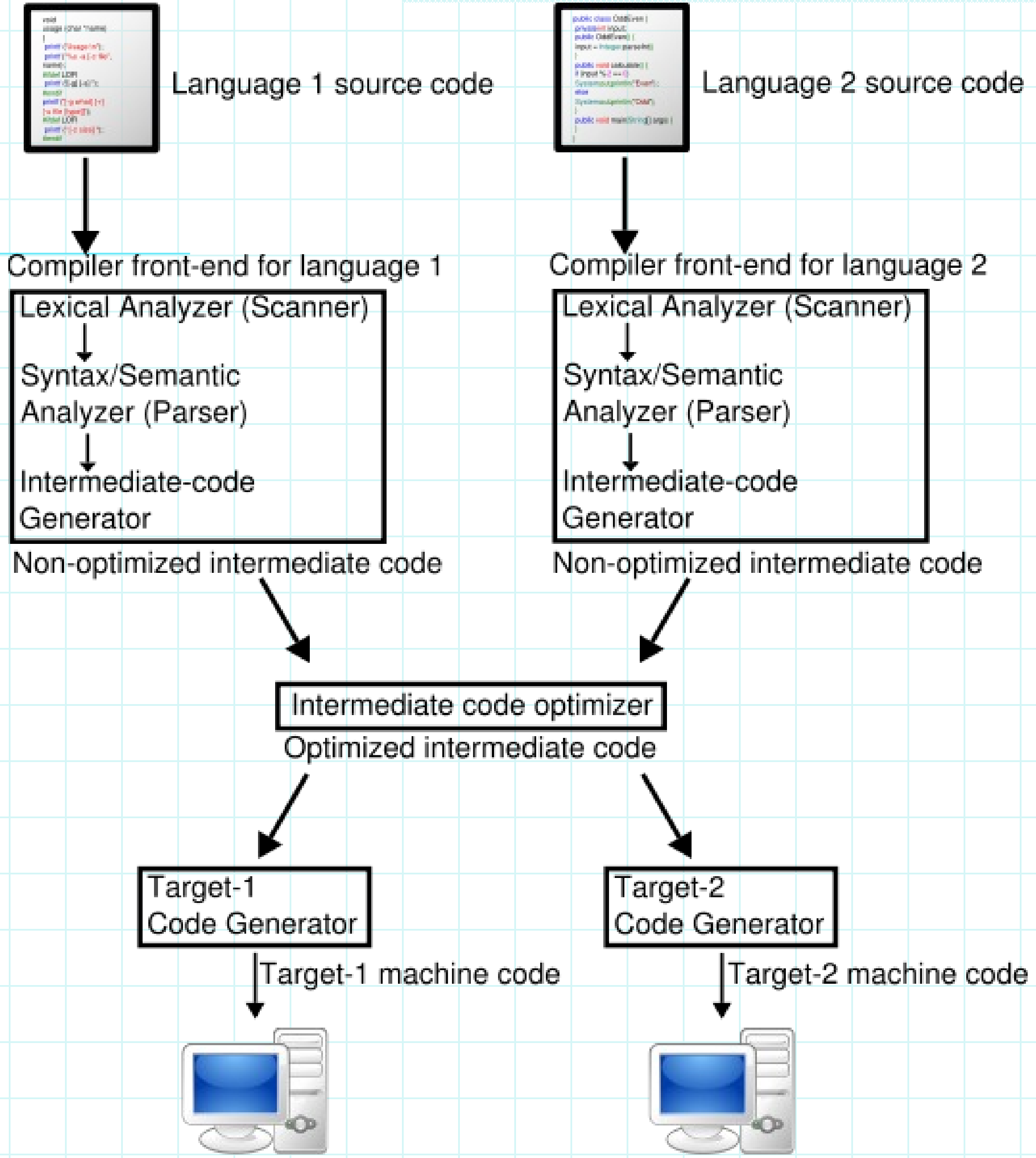


Moduly kompilátora:

Lexikálny analyzátor - rozpoznáva regulárny podjazyk
- simulácia konečného automatu

Syntaktický analyzátor - rozpoznáva celý jazyk
- simulácia zásobníkového automatu
- z gramatiky ⇨ NPDA, úprava gramatiky na vhodnejší tvar – odstránenie rekurzie

...


Language 1 source code
Language 2 source code
Compiler front-end for language 1
Lexical Analyzer (Scanner)
Syntax/Semantic Analyzer (Parser)
Intermediate-code Generator
Compiler front-end for language 2
Lexical Analyzer (Scanner)
Syntax/Semantic Analyzer (Parser)
Intermediate-code Generator
Non-optimized intermediate code
Non-optimized intermediate code
Intermediate code optimizer
Optimized intermediate code
Target-1 Code Generator
Target-2 Code Generator
Target-1 machine code
Target-2 machine code

Odstránenie ľavej rekurzie

A→Ax

A

↙ ↘

A x

↙ ↘

A x

↙

...

Pre každé pravidlo

A→Ax | y

Na pravidlá

A→yB

B →xB|ε

Pr.: Odstránenie ľavej rekurzie

$A \rightarrow Ax \mid y$

$A \rightarrow yB$

$B \rightarrow xB \mid \varepsilon$

Príklad s ľavou rekurziou:

$E \rightarrow E+T$

$E \rightarrow E-T$

$E \rightarrow T$

$T \rightarrow (E)$

$T \rightarrow 0|..|9$

Po odstránení ľavej rekurzie:

$E \rightarrow TR$

$R \rightarrow +TR$

$R \rightarrow -TR$

$R \rightarrow \varepsilon$

$T \rightarrow (E)$

$T \rightarrow 0|..|9$

Syntaktická analýza

- zhora nadol – LL(k), k∈N, napr. LL(1)
- zdola nahor - LR(k) : LR(0), LR(1),... LR(k)

sLL(k)

Definícia sLL(k) :

Nech kϵN^{+}. Gramatika sa nazýva jednoduchá LL(k) ak:

1. Pravá strana každého pravidla začína k terminálnymi symbolmi.
2. Ak pravidlá majú rovnakú ľavú stranu, ich pravé strany sa líšia nanajvýš v k-tom terminálnom symbole.

Pr.: sLL(k)

sLL(1)

S→aSA	S→aSA
S→bAA	S→bAA
S→bS	S→cS

pre slovo **aab** neviem podľa ktorého pravidla redukovať

sLL(3)

A→aaaA | bbbB	A→aaaA | bbbB
B→abcBA	B→abcBA
B→abcCA	B→abdCA

Riadiaca tabuľka

Gramatika:

E →TR

R →+TR

R →-TR

R →ε

T →(E)

T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Syntaktická analýza LL metódou

```
VSTUP:   w$ , wєΣ*, G=(N,T,P,E)
VÝSTUP:  najľavejšie odvodenie pre  wєL(G)
INIT:    STACK:=E$
         ^IN:=1.symbol vstupu
REPEAT
    X:=TOP(STACK); a=READ(IN);
    IF XєT U {$} THEN
       IF X==a THEN  [POP(STACK); NEXT(IN);]
       ELSE ERROR;
    ELSE
       IF M[X,a]==X→Y1…Yk THEN
          [POP(STACK);
          FOR i:=k TO 1 DO PUSH(Yi);
          OUTPUT("X→Y1…Yk"); ]
    ELSE ERROR;
UNTIL X==$
```

Strom odvodenia pre slovo (3)

```
          E
        /    \
       T      R
     / | \    |
    (  E  )   ε
      / \
      T  R
      |  |
      3  ε
```

Syntaktická analýza – stromový prehľadávací algoritmus.
Spôsob prehľadávania – do hĺbky, zľava doprava.

Ďakujem za pozornosť.

chuda@fiit.stuba.sk

Syntaktická analýza

Využitie konečných a zásobníkových automatov v kompilátoroch

Mgr. Daniela Chudá, PhD., **chuda@fiit.stuba.sk**

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

```
INIT:        STACK:=E$
             ^IN:=1.symbol vstupu
REPEAT
             X:=TOP(STACK); a=READ(IN);
             IF XєT U {$} THEN
                IF X==a THEN
                  [POP(STACK); NEXT(IN);]
                ELSE ERROR;
             ELSE
                IF M[X,a]==X→Y1...Yk THEN
                         [POP(STACK);
                         FOR i:=k TO 1 DO
                            PUSH(Yi);
                         OUTPUT("X→Y1...Yk");
                         ]
             ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

Zásobník:

výstup:

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1.symbol vstupu

REPEAT
X:=TOP(STACK); a=READ(IN);
IF XєT U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR;
ELSE
IF M[X,a]==X→Y_1...Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X→Y_1...Y_k");
]
ELSE ERROR;
UNTIL X==$

Strom odvodenia:

Zásobník:

E
$

INIT

výstup:

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1.symbol vstupu
REPEAT
X:=TOP(STACK); a=READ(IN);
IF XєT U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR;
ELSE
IF M[X,a]==X→Y_1…Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X→Y_1…Y_k");
]
ELSE ERROR;
UNTIL X==$

Strom odvodenia:

```
    E
  /   \
 T     R
```

Zásobník:

T
R
$

1. REPEAT X=E a=(

výstup: E →TR

vstupné slovo: (3) $

▲

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1.symbol vstupu
REPEAT
X:=TOP(STACK); a=READ(IN);
IF XєT U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR;
ELSE
IF M[X,a]==X→Y_1…Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X→Y_1…Y_k");
]
ELSE ERROR;
UNTIL X==$

Strom odvodenia:

```
      E
    /   \
   T     R
 / | \
 (  E  )
```

Zásobník:

(
E
)
R
$

2. REPEAT X=T a=(

výstup: E →TR , T →(E)

vstupné slovo: (3) $

Gramatika:

E → TR
R → +TR
R → -TR
R → ε
T → (E)
T → 0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E → TR		E → TR		
R		R → ±TR		R → ε	R → ε
T	T → 0|..|9		T → (E)		

Algoritmus:

```
INIT:     STACK:=E$
          ^IN:=1.symbol vstupu
REPEAT
          X:=TOP(STACK); a=READ(IN);
          IF XєT U {$} THEN
            IF X==a THEN
              [POP(STACK); NEXT(IN);]
            ELSE ERROR;
          ELSE
            IF M[X,a]==X→Y1...Yk THEN
                      [POP(STACK);
                      FOR i:=k TO 1 DO
                        PUSH(Yi);
                      OUTPUT("X→Y1...Yk");
                      ]
          ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
   ( E )
```

Zásobník:

E
)
R
$

3. REPEAT X=(a=(

výstup: E → TR , T → (E)

vstupné slovo: (3) $

▲

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

INIT: STACK:=E$

^IN:=1.symbol vstupu

REPEAT

X:=TOP(STACK); a=READ(IN);

IF XϵT U {$} THEN

IF X==a THEN

[POP(STACK); NEXT(IN);]

ELSE ERROR;

ELSE

IF M[X,a]==X→Y_1…Y_k THEN

[POP(STACK);

FOR i:=k TO 1 DO

PUSH(Y_i);

OUTPUT(“X→Y_1…Y_k”);

]

ELSE ERROR;

UNTIL X==$

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
  (  E  )
    / \
   T   R
```

Zásobník:

T
R
)
R
$

4. REPEAT X=E a=3

výstup: E →TR , T →(E) , E →TR

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

```
INIT:     STACK:=E$
          ^IN:=1.symbol vstupu
REPEAT
          X:=TOP(STACK); a=READ(IN);
          IF XєT U {$} THEN
             IF X==a THEN
                [POP(STACK); NEXT(IN);]
             ELSE ERROR;
          ELSE
             IF M[X,a]==X→Y1...Yk THEN
                   [POP(STACK);
                   FOR i:=k TO 1 DO
                      PUSH(Yi);
                   OUTPUT("X→Y1...Yk");
                   ]
          ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
   ( E )
    / \
   T   R
   |
   3
```

Zásobník:

3
R
)
R
$

5. REPEAT X=T a=3

výstup: E →TR , T →(E) , E →TR , T →3

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1.symbol vstupu
REPEAT
X:=TOP(STACK); a=READ(IN);
IF XϵT U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR;
ELSE
IF M[X,a]==X→Y_1...Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT(“X→Y_1...Y_k”);
]
ELSE ERROR;
UNTIL X==$

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
  (  E  )
    / \
    T  R
    |
    3
```

Zásobník:

R
)
R
$

6. REPEAT X=3 a=3

výstup: E →TR , T →(E) , E →TR , T →3

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

```
INIT:      STACK:=E$
           ^IN:=1.symbol vstupu
REPEAT
           X:=TOP(STACK); a=READ(IN);
           IF XϵT U {$} THEN
              IF X==a THEN
                [POP(STACK); NEXT(IN);]
              ELSE ERROR;
           ELSE
              IF M[X,a]==X→Y1...Yk THEN
                      [POP(STACK);
                      FOR i:=k TO 1 DO
                         PUSH(Yi);
                      OUTPUT("X→Y1...Yk");
                      ]
           ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
   ( E )
    / \
    T  R
    |  |
    3  ε
```

Zásobník:

)
R
$

7. REPEAT X=R a=)

výstup: E →TR , T →(E) , E →TR , T →3 , R →ε

vstupné slovo: (3) $

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

```
INIT:        STACK:=E$
             ^IN:=1.symbol vstupu
REPEAT
             X:=TOP(STACK); a=READ(IN);
             IF XєT U {$} THEN
                IF X==a THEN
                  [POP(STACK); NEXT(IN);]
                ELSE ERROR;
             ELSE
                IF M[X,a]==X→Y1...Yk THEN
                             [POP(STACK);
                             FOR i:=k TO 1 DO
                                PUSH(Yi);
                             OUTPUT("X→Y1...Yk");
                             ]
             ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \
  (  E  )
    / \
   T   R
   |   |
   3   ε
```

Zásobník:

R
$

8. REPEAT X=) a=)

výstup: E →TR , T →(E) , E →TR , T →3 , R →ε

vstupné slovo: (3) $

Gramatika:

E → TR
R → +TR
R → -TR
R → ε
T → (E)
T → 0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E → TR		E → TR		
R		R → ±TR		R → ε	R → ε
T	T → 0|..|9		T → (E)		

Algoritmus:

```
INIT:       STACK:=E$
            ^IN:=1.symbol vstupu
REPEAT
            X:=TOP(STACK); a=READ(IN);
            IF XϵT U {$} THEN
               IF X==a THEN
                  [POP(STACK); NEXT(IN);]
               ELSE ERROR;
            ELSE
               IF M[X,a]==X→Y1...Yk THEN
                     [POP(STACK);
                     FOR i:=k TO 1 DO
                        PUSH(Yi);
                     OUTPUT("X→Y1...Yk");
                     ]
            ELSE ERROR;
UNTIL X==$
```

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \   |
  (  E  )  ε
    / \
    T  R
    |  |
    3  ε
```

Zásobník:

$

9. REPEAT X=R a=$

výstup: E → TR , T → (E) , E → TR , T → 3 , R → ε , R → ε

vstupné slovo: (3) $

▲

Gramatika:

E →TR
R →+TR
R →-TR
R →ε
T →(E)
T →0|..|9

Riadiaca tabuľka:

	id	+/-	(	)	$
E	E →TR		E →TR		
R		R →±TR		R →ε	R →ε
T	T →0|..|9		T →(E)		

Algoritmus:

```
INIT:       STACK:=E$
            ^IN:=1.symbol vstupu
REPEAT
            X:=TOP(STACK); a=READ(IN);
            IF XєT U {$} THEN
              IF X==a THEN
                [POP(STACK); NEXT(IN);]
              ELSE ERROR;
            ELSE
              IF M[X,a]==X→Y1...Yk THEN
                        [POP(STACK);
                        FOR i:=k TO 1 DO
                          PUSH(Yi);
                        OUTPUT("X→Y1...Yk");
                        ]
            ELSE ERROR;
UNTIL X==$
```

10. REPEAT X=$ a=$

Strom odvodenia:

```
        E
      /   \
     T     R
   / | \   |
   ( E )   ε
    / \
    T R
    | |
    3 ε
```

Zásobník:

výstup: E →TR , T →(E) , E →TR , T →3 , R →ε , R →ε

Ďakujem za pozornosť.

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
           IF X==a THEN
              [POP(STACK); NEXT(IN);]
           ELSE ERROR
        ELSE
           IF M[X,a]==X->Y1...Yk THEN
              [POP(STACK);
              FOR i:=k TO 1 DO
                 PUSH(Yi);
              OUTPUT("X->Y1...Yk");
              ]
           ELSE ERROR
UNTIL X==$
```

Výstup:

Zásobník:

E

KROK POMOC

(c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
           IF X==a THEN
              [POP(STACK); NEXT(IN);]
           ELSE ERROR
        ELSE
           IF M[X,a]==X->Y1...Yk THEN
              [POP(STACK);
              FOR i:=k TO 1 DO
                 PUSH(Yi);
              OUTPUT("X->Y1...Yk");
              ]
           ELSE ERROR
UNTIL X==$
```

INIT

Výstup:

Zásobník:

E
$

E

KROK POMOC

(c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1. symbol vstupu
REPEAT
X:=TOP(STACK); a:=READ(IN);
IF X∈T U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR
ELSE
IF M[X,a]==X->Y_1...Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X->Y_1...Y_k");
]
ELSE ERROR
UNTIL X==$

1. REPEAT X=E a=(

Výstup:

E→TR

Zásobník:

T
R
$

E ⟹ TR

KROK POMOC

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
           IF X==a THEN
              [POP(STACK); NEXT(IN);]
           ELSE ERROR
        ELSE
           IF M[X,a]==X->Y1...Yk THEN
              [POP(STACK);
              FOR i:=k TO 1 DO
                 PUSH(Yi);
              OUTPUT("X->Y1...Yk");
              ]
           ELSE ERROR
UNTIL X==$
```

2. REPEAT X=T a=(

Výstup:

E→TR , T→(E)

Zásobník:

(
E
)
R
$

E ⟹ TR ⟹ (E)R

KROK POMOC

STU FIIT (c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
           IF X==a THEN
              [POP(STACK); NEXT(IN);]
           ELSE ERROR
        ELSE
           IF M[X,a]==X->Y1...Yk THEN
              [POP(STACK);
              FOR i:=k TO 1 DO
                 PUSH(Yi);
              OUTPUT("X->Y1...Yk");
              ]
           ELSE ERROR
UNTIL X==$
```

3. REPEAT X=(a=(

Výstup:

E→TR , T→(E)

Zásobník:

E
)
R
$

E ⟹ TR ⟹ (E)R

KROK POMOC

STU FIIT (c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
           IF X==a THEN
              [POP(STACK); NEXT(IN);]
           ELSE ERROR
        ELSE
           IF M[X,a]==X->Y1...Yk THEN
              [POP(STACK);
              FOR i:=k TO 1 DO
                 PUSH(Yi);
              OUTPUT("X->Y1...Yk");
              ]
           ELSE ERROR
UNTIL X==$
```

4. REPEAT X=E a=3

Výstup:

E→TR , T→(E) , E→TR

Zásobník:

T
R
)
R
$

$E \Longrightarrow TR \Longrightarrow (E)R \Longrightarrow (TR)R$

KROK POMOC

(c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:  STACK:=E$
       ^IN:=1. symbol vstupu
REPEAT
       X:=TOP(STACK); a:=READ(IN);
       IF X∈T U {$} THEN
          IF X==a THEN
             [POP(STACK); NEXT(IN);]
          ELSE ERROR
       ELSE
          IF M[X,a]==X->Y1...Yk THEN
             [POP(STACK);
             FOR i:=k TO 1 DO
                PUSH(Yi);
             OUTPUT("X->Y1...Yk");
             ]
          ELSE ERROR
UNTIL X==$
```

5. REPEAT X=T a=3

Výstup:

E→TR , T→(E) , E→TR , T→3

Zásobník:

3
R
)
R
$

E ⟹ TR ⟹ (E)R ⟹ (TR)R ⟹ (3R)R

KROK POMOC

(c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

$E \to TR$
$R \to +TR$
$R \to -TR$
$R \to \varepsilon$
$T \to (E)$
$T \to 0|..|9$

Riadiaca tabuľka:

	id	+	-	(	)	$
E	$E \to TR$			$E \to TR$		
R		$R \to +TR$	$R \to -TR$		$R \to \varepsilon$	$R \to \varepsilon$
T	$T \to 0\|..\|9$			$T \to (E)$		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
          IF X==a THEN
             [POP(STACK); NEXT(IN);]
          ELSE ERROR
        ELSE
          IF M[X,a]==X->Y1...Yk THEN
             [POP(STACK);
             FOR i:=k TO 1 DO
                PUSH(Yi);
             OUTPUT("X->Y1...Yk");
             ]
          ELSE ERROR
UNTIL X==$
```

6. REPEAT X=3 a=3

Výstup:

$E \to TR$, $T \to (E)$, $E \to TR$, $T \to 3$

Zásobník:

R
)
R
$

$E \Longrightarrow TR \Longrightarrow (E)R \Longrightarrow (TR)R \Longrightarrow (3R)R$

KROK POMOC

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:   STACK:=E$
        ^IN:=1. symbol vstupu
REPEAT
        X:=TOP(STACK); a:=READ(IN);
        IF X∈T U {$} THEN
          IF X==a THEN
            [POP(STACK); NEXT(IN);]
          ELSE ERROR
        ELSE
          IF M[X,a]==X->Y1...Yk THEN
            [POP(STACK);
            FOR i:=k TO 1 DO
              PUSH(Yi);
            OUTPUT("X->Y1...Yk");
            ]
          ELSE ERROR
UNTIL X==$
```

7. REPEAT X=R a=)

Výstup:

E→TR , T→(E) , E→TR , T→3 , R→ε

Zásobník:

)
R
$

E ⟹ TR ⟹ (E)R ⟹ (TR)R ⟹ (3R)R ⟹ (3)R

KROK POMOC

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1. symbol vstupu
REPEAT
X:=TOP(STACK); a:=READ(IN);
IF X∈T U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR
ELSE
IF M[X,a]==X->Y_1...Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X->Y_1...Y_k");
]
ELSE ERROR
UNTIL X==$

8. REPEAT X=) a=)

Výstup:

E→TR , T→(E) , E→TR , T→3 , R→ε

Zásobník:

R
$

E ⟹ TR ⟹ (E)R ⟹ (TR)R ⟹ (3R)R ⟹ (3)R

KROK POMOC

(c) 2007

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

INIT: STACK:=E$
^IN:=1. symbol vstupu
REPEAT
X:=TOP(STACK); a:=READ(IN);
IF X∈T U {$} THEN
IF X==a THEN
[POP(STACK); NEXT(IN);]
ELSE ERROR
ELSE
IF M[X,a]==X->Y_1...Y_k THEN
[POP(STACK);
FOR i:=k TO 1 DO
PUSH(Y_i);
OUTPUT("X->Y_1...Y_k");
]
ELSE ERROR
UNTIL X==$

9. REPEAT X=R a=$

Výstup:

E→TR , T→(E) , E→TR , T→3 , R→ε , R→ε

Zásobník:

$

E ⟹ TR ⟹ (E)R ⟹ (TR)R ⟹ (3R)R ⟹ (3)R ⟹ (3)

KROK POMOC

Syntaktická analýza LL metódou

Vstupné slovo: (3) NASTAV

(	3	)	$

Gramatika:

E→TR
R→+TR
R→-TR
R→ε
T→(E)
T→0|..|9

Riadiaca tabuľka:

	id	+	-	(	)	$
E	E→TR			E→TR		
R		R→+TR	R→-TR		R→ε	R→ε
T	T→0|..|9			T→(E)		

Algoritmus:

```
INIT:  STACK:=E$
       ^IN:=1. symbol vstupu
REPEAT
    X:=TOP(STACK); a:=READ(IN);
    IF X∈T U {$} THEN
      IF X==a THEN
        [POP(STACK); NEXT(IN);]
      ELSE ERROR
    ELSE
      IF M[X,a]==X->Y1...Yk THEN
        [POP(STACK);
        FOR i:=k TO 1 DO
          PUSH(Yi);
        OUTPUT("X->Y1...Yk");
        ]
      ELSE ERROR
UNTIL X==$
```

10. REPEAT X=$ a=$

Výstup:

E→TR , T→(E) , E→TR , T→3 , R→ε , R→ε

Zásobník:

E ⟹ TR ⟹ (E)R ⟹ (TR)R ⟹ (3R)R ⟹ (3)R ⟹ (3)

KROK POMOC

STU
FIIT

Teoretické základy informatiky

Univerzálny Turingov stroj

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Turingov stroj (opakovanie)

Definícia:

Nedeterministický Turingov stroj je šestica

$A = (K, \Sigma, \Gamma, \delta, q_0, F)$, kde

K je konečná množina stavov,

Σ je vstupná abeceda,

Γ ja pracovná abeceda ($\Sigma \subseteq \Gamma; B \in \Gamma; B - blank$)

δ je prechodová funkcia $\delta : K \times \Gamma \to 2^{K \times \Gamma \times \{-1,0,1\}}$,

$q_0 \in K$ je počiatočný stav,

$F \subseteq K$ je množina koncových stavov.

Poznámka:

Ak platí: $\#(\delta(q, a)) \leq 1$, hovoríme, že Turingov stroj je deterministický.

Univerzalita TS

Dôležitá vlastnosť Turingových strojov je tzv. univerzalita.

Znamená to, že sa dajú zostrojiť Turingove stroje, na ktorých je možné simulovať výpočet iných Turingových strojov.

Univerzálny TS a problém zastavenia

Existuje Turingov stroj (univerzálny TS), ktorý keď sa predloží kód ľubovolného Turingovho Stroja T a kód slova x, bude simulovať činnosť stroja T pri spracúvaní slova x.

Univerzálny TS možno pokladať za mnohoúčelový počítač, ktorý má dostatočne veľké možnosti pre simuláciu ľubovolného počítača, vrátane seba samého.

Neexistuje algoritmus (TS, ktorý sa zastaví pre všetky vstupné slová), ktorý by určil pre ľubovolný Turingov stroj T a ľubovolný vstup x, či sa T so vstupom x niekedy zastaví.

Hopcroft, Ullman: Formálne jazyky a automaty

Univerzálny Turingov Stroj -UTM

Definícia: Univerzálny Turingov stroj

Nech Σ *je abeceda a nech* T_x *je Turingov stroj, ktorý je repreznetovaný svojim kódom* $x \in \Sigma^*$. Univerzálny Turingov stroj U *pre všetky vstupy* y, *pre ktoré je výstup* $T_x(y)$ *definovaný, počíta hodnotu*

$$U(x, y) = T_x(y)$$

pre všetky prípustné kódy $x \in \Sigma^*$.

Veta 6.1.1 *Existuje univerzálny Turingov stroj.*

Turingov stroj sa používa aj na formalizáciu pojmu T-vypočítateľná funkcia.

Dôkaz: UTM, myšlienka = zakódovať TS

1. Zvolenie kódovania C
   - kódovanie vstupu
   - kódovanie TS (prechodovej funkcie)
2. Konštrukcia U pre C
   - konštrukcia dvojpáskového TS
   - simulovanie TS
   - Ukončenie TS

STU FIIT

Kódovanie vstupu

- potrebujeme zakódovať vstup, máme symboly 0,1
- vstup zakódujeme cez bezprefixový kód

Bezprefixový kód –jednoznačne určím znaky

Bezprefixový kód $Code(x) = 1^{|x|}0x; x \in \{0,1\}^*$
$|Code(x)| = 2.|x| + 1$

<u>Príklad</u>:

a	b	c	d	e	f	g
ϵ	0	1	00	01	10	11

$Code(a) = 0$
$Code(b) = 100$
$Code(c) = 101$
$Code(d) = 11000$
$Code(e) = 11001$
$Code(f) = 11010$
$Code(g) = 11011$

Každý symbol dokážem pretransformovať do postupnosti 0 a 1.

Kódovanie Turingovho stroja

Kódovanie Turingovho stroja

$L = \{w \in \{0,1\}^* \,|\, \#_0 w = 2k; k \in \mathbb{N}\}$

$A = (K, \Sigma, \Gamma, \delta, q_1, F); F = \{q_3\}$

$\delta(q_1, 0) = (q_2, 0, R)$

$\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_2, 0) = (q_1, 0, R)$

$\delta(q_2, 1) = (q_2, 1, R)$

$\delta(q_1, B) = (q_3, B, L)$

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Páska:

ccc11R0c1R1c111LBcc1R0c11R1c0cc0c0c0ccc

Kódovanie Turingovho stroja - vysvetlenie

Zakódujem všetky TS nad abecedou $\{0,1\}^*$

- počet stĺpcov tabuľky – vždy 3 → (0,1,B)
- počet riadkov tabuľky - závisí od počtu stavov
- koncový stav – jediný, riadok, ktorý je nedefinovaný pre každý symbol
- páska = na pásku kódujem kontinuálne tabuľku:
  - ccc - koncové a počiatočné oddelovače ,
  - cc – oddeľuje riadky,
  - c – oddeľuje políčka v riadkoch.
- kódujem stavy:
  - q_1 1
  - q_2 11
  - q_3 111
  - \- 0
- pri kódovaní vymieňam zapisovaný znak a pohyb hlavy, aby boli oddelené 0 a 1 → $q_2$0R - 11R0

Návrh TS

Univerzálny Turingov stroj

$$U = (K, \Sigma, \Gamma, \delta, q_U, F); F = \{q_F\}$$

$$\Sigma = \{0, 1, c, L, R\}$$

$$\Gamma = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} | x \in \{B, m\}, y \in \Sigma \bigcup \{B\} \right\}$$

- UTS využíva dvojstopú pásku $\begin{pmatrix} x \\ y \end{pmatrix}$
- vstupná abeceda Σ,
- vstupné symboly – z kódovania
- koncový stav – q_F
- dvojice (6x2=12):
  - horné - B,m symboly
  - dolné – Σ u {B} symboly

Simulácia TS

horná stopa:

dolná stopa:

		m																			
c	**c**	**c**	**1**	**1**	**R**	**0**	**c**	**1**	**R**	**1**	**c**	**1**	**1**	**1**	**L**	**B**	**c**	**c**	**1**	**R**	**0**

...

...

																m				
c	**1**	**1**	**R**	**1**	**c**	**0**	**c**	**c**	**0**	**c**	**0**	**c**	**0**	**c**	**c**	**c**	**0**	**1**	**0**	**B**

dolná stopa:

- prvá časť - kód TS, ktorý chcem simulovať
- druhá časť – dáta **010**

prvý symbol m ← aktuálny stav

druhý symbol m ← poloha čítacej hlavy

$U(x,y)=T_x(y)$

Simulácia TS

horná stopa:
dolná stopa:

		m																			
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

																	m			
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:			m																			
dolná stopa:	c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

																m			
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Simulácia TS

horná stopa:
dolná stopa:

		m																				
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0	...

																		m			
...	c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:																			**m**				
dolná stopa:	**c**	**c**	**c**	**1**	**1**	**R**	**0**	**c**	**1**	**R**	**1**	**c**	**1**	**1**	**1**	**L**	**B**	**c**	**c**	**1**	**R**	**0**	**...**

																			m		
...	**c**	**1**	**1**	**R**	**1**	**c**	**0**	**c**	**c**	**0**	**c**	**0**	**c**	**0**	**c**	**c**	**c**	**0**	**1**	**0**	**B**

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:

dolná stopa:

c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

m																		m		
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Simulácia TS

horná stopa:

dolná stopa:

c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0	...

	m																	m		
...	c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Simulácia TS

horná stopa:

dolná stopa:

																		m				
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0	...

																				m	
...	c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Simulácia TS

horná stopa:
dolná stopa:

																		m			
c	**c**	**c**	**1**	**1**	**R**	**0**	**c**	**1**	**R**	**1**	**c**	**1**	**1**	**1**	**L**	**B**	**c**	**c**	**1**	**R**	**0**

...

...

																			m	
c	**1**	**1**	**R**	**1**	**c**	**0**	**c**	**c**	**0**	**c**	**0**	**c**	**0**	**c**	**c**	**c**	**0**	**1**	**0**	**B**

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:																		**m**				
dolná stopa:	**c**	**c**	**c**	**1**	**1**	**R**	**0**	**c**	**1**	**R**	**1**	**c**	**1**	**1**	**1**	**L**	**B**	**c**	**c**	**1**	**R**	**0**

...

...

																			m	
c	**1**	**1**	**R**	**1**	**c**	**0**	**c**	**c**	**0**	**c**	**0**	**c**	**0**	**c**	**c**	**c**	**0**	**1**	**0**	**B**

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Simulácia TS

horná stopa:
dolná stopa:

																		m			
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

																			m	
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:
dolná stopa:

		m																			
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

																					m
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B	

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:
dolná stopa:

											m										
c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0

...

...

																				m
c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	—
q_3	—	—	—

Simulácia TS

horná stopa:																							
dolná stopa:	c	c	c	1	1	R	0	c	1	R	1	c	1	1	1	L	B	c	c	1	R	0	...

													m							m	
...	c	1	1	R	1	c	0	c	c	0	c	0	c	0	c	c	c	0	1	0	B

akceptuje q_3

	0	1	B
q_1	$q_2 0 R$	$q_1 1 R$	$q_3 B L$
q_2	$q_1 0 R$	$q_2 1 R$	$-$
q_3	$-$	$-$	$-$

Ukončenie TS

U akceptuje <=> T_x akceptuje

U sa zasekne <=> T_x sa zasekne

U cyklí <=> T_x cyklí

Univerzálny TS a problém zastavenia

Problém zastavenia pre TS je definovaný takto:

Nech je daný TS v ľubovolnej konfigurácii a konečne dlhým reťazcom neprázdnych páskových symbolov, zastaví sa niekedy tento TS?

O tomto probléme hovoríme, že je rekurzívne neriešiteľný (alebo nerozhodnuteľný) v tom zmysle, že neexistuje algoritmus, ktorý pre každý TS a každú konfiguráciu určí, či tento TS v uvedenej konfigurácii nakoniec zastaví. Toto neznamená, že nemôžeme určiť, či sa konkrétny TS v nejake konfigurácii zastaví.

Hopcroft, Ullman: Formálne jazyky a automaty

Univerzálne Turingove stroje (1)

	0	1	2	3	4	5
q_0	$q_0 17L$	$q_0 4R$	$q_0 17L$	$q_0 0R$	$q_0 3L$	$q_0 7R$
q_1	$q_1 2R$	$q_1 2R$	$q_1 1L$	$q_1 4R$	$q_1 0L$	$q_0 10R$
	6	7	8	9	10	11
q_0	$q_0 9R$	$q_0 5L$	$q_0 5R$	$q_0 8L$	$q_1 11L$	$q_1 8L$
q_1	$q_1 7R$	$q_1 6L$	$q_1 9R$	$q_1 6L$	$q_0 5R$	$q_1 9R$
	12	13	14	15	16	17
q_0	$q_1 1L$	$q_0 14L$	$q_0 15L$	$q_1 16R$	$-$	$q_0 2R$
q_1	$q_1 14R$	$q_1 14R$	$q_1 13L$	$q_1 17R$	$q_0 17L$	$q_1 12L$

UTM(2, 18), Rogozhin (1995)

	q_0	q_1	q_2	q_3
0	$q_0 3L$	$q_1 4R$	$q_2 0R$	$q_3 4R$
1	$q_0 2R$	$q_2 2L$	$q_3 3R$	$q_1 5L$
2	$q_0 1L$	$q_1 3R$	$q_2 1R$	$q_3 3R$
3	$q_0 4R$	$q_1 2L$	$-$	$-$
4	$q_0 3L$	$q_1 0L$	$q_0 5R$	$q_1 5L$
5	$q_3 4R$	$q_1 1R$	$q_0 0R$	$q_3 1R$

UTM(4, 6), Rogozhin (1982)

Univerzálne Turingove stroje (2)

	q_0	q_1	q_2	q_3	q_4	q_5
0	$q_4 0 R$	$q_0 1 R$	$q_3 0 L$	$q_{11} 1 L$	$q_0 1 R$	$q_6 0 L$
1	$q_1 1 R$	$q_2 1 L$	$q_1 0 L$	$q_8 0 L$	$q_5 0 L$	$q_6 1 L$
	q_6	q_7	q_8	q_9	q_{10}	q_{11}
0	$q_7 0 L$	$q_6 0 L$	$q_{18} 0 R$	$q_3 1 L$	$q_3 0 L$	$q_{18} 0 R$
1	$q_5 0 L$	$q_1 1 R$	$q_3 1 L$	$q_{12} 0 R$	—	$q_{13} 1 L$
	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}	q_{17}
0	$q_9 0 R$	$q_{14} 0 L$	$q_{15} 0 R$	$q_{14} 0 R$	$q_{15} 0 R$	$q_{18} 0 R$
1	$q_{23} 1 R$	$q_{10} 1 R$	$q_{16} 1 R$	$q_9 1 R$	$q_{20} 1 R$	$q_{19} 1 R$
	q_{18}	q_{19}	q_{20}	q_{21}	q_{22}	q_{23}
0	$q_2 1 L$	$q_{17} 1 R$	$q_{21} O R$	$q_9 1 L$	$q_{20} 1 R$	$q_{12} 0 R$
1	$q_{17} 1 R$	$q_{17} 0 R$	$q_{22} 1 R$	$q_{20} 1 R$	$q_{20} 0 R$	$q_2 0 L$

UTM(24, 2), Rogozhin (1982)

Church-Turingova téza

Každý proces, ktorý je možné intuitívne nazvať algoritmus, sa dá realizovať na Turingovom stroji.

Ďakujem za pozornosť.

chuda@fiit.stuba.sk

Teoretické základy informatiky

Vypočítateľnosť

Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Turingov stroj (opakovanie)

Definícia:
Nedeterministický Turingov stroj je šestica
$A = (K, \Sigma, \Gamma, \delta, q_0, F)$, kde
K je konečná množina stavov,
Σ je vstupná abeceda,
Γ ja pracovná abeceda ($\Sigma \subseteq \Gamma; B \in \Gamma; B - blank$)
δ je prechodová funkcia $\delta : K \times \Gamma \to 2^{K \times \Gamma \times \{-1,0,1\}}$,
$q_0 \in K$ je počiatočný stav,
$F \subseteq K$ je množina koncových stavov.

Poznámka:
Ak platí: $\#(\delta(q, a)) \leq 1$, hovoríme, že Turingov stroj je Deterministický.

Algoritmizovateľnosť

Teória vypočítateľnosti skúma otázky **algoritmizovateľnosti** na rôznych výpočtových modeloch.

Pojem **vypočítateľného problému** sa používa ako synonymum pre **algoritmizovateľný problém**.

Exaktné vymedzenie samotného pojmu **algoritmus** je však vysoko netriviálne. Kvôli tomu sa v teórii, ktorá sa zaoberá vypočítateľnosťou skúmajú rôznorodé problémy, z ktorých najdôležitejšie sú nasledujúce:

- algoritmus a jeho vlastnosti,
- problémy, ktoré sa dajú algoritmizovať a tie, ktoré sa algoritmizovať nedajú.

Výpočtové modely

Kľúčom k riešeniu otázok algoritmizovateľnosti sú rôzne výpočtové modely, ktoré vznikli preto, aby čo možno najvernejšie reprezentovali pojmy algoritmus a počítač. Ich vlastnosti a vzájomné vzťahy (ekvivalencia) dokážu v značnej miere odpovedať na uvedené otázky.

Budeme zaoberať tromi vzájomne ekvivaletnými výpočtovými modelmi. Sú to nasledujúce modely:

- Turingov stroj,
- Počítadlový stroj,
- stroj RAM.

STU FIIT

T-vypočítateľná funkcia

Definícia: **T-vypočítateľná funkcia**

Nech $k \in \mathbb{N}^+$. *Funkcia* $f : \mathbb{N}^k \to \mathbb{N}$ *sa nazýva* T-vypočítateľná funkcia, *ak existuje Turingov stroj* A, *ktorý rozpoznáva jazyk*

$$L = \{\ 1^{x_1}¥1^{x_2}¥\ldots¥1^{x_k}\$1^{f(\overline{x})}\ \}$$

pre všetky $\overline{x} = (x_1, \ldots, x_k) \in \mathbb{N}^k$.

Príklady T-vypočítateľných funkcií:

- lineárne funkcie,
- polynomiálne funkcie,
- exponenciálne funkcie,
- logaritmické funkcie,
- lineárne kombinácie T-vypočítateľných funkcií.

Príklad: T-vypočítateľná funkcia f(a,b)=(a+b)

Zadanie:
Dokážte, že funkcia **f(*a*,*b*)=(*a*+*b*)** je T-vypočítateľná.

Vstup:

$$\overset{a}{\downarrow}\,11¢\overset{b}{\downarrow}\,111\$$$

Výstup:

$$\underbrace{\underline{11}}_{a}¢\underbrace{\underline{111}}_{b}\$\underbrace{11111}_{f(a,b)=(a+b)}$$

Neformálne riešenie:
- z ***a*** okopírovať každú 1 za $
- z ***b*** okopírovať každú 1 za $

Príklad: T-vypočítateľná funkcia f(a,b)=(a+b)

Formálne

$$copy\ a \begin{cases} \delta(q_0, 1) = (q_1, \underline{1}, R) & \\ \delta(q_1, x) = (q_1, x, R) & x \in \{1, \not{C}, \$\} \\ \delta(q_1, B) = (q_2, 1, L) & \\ \delta(q_2, x) = (q_2, x, L) & \\ \delta(q_2, \underline{1}) = (q_0, \underline{1}, R) & \end{cases}$$

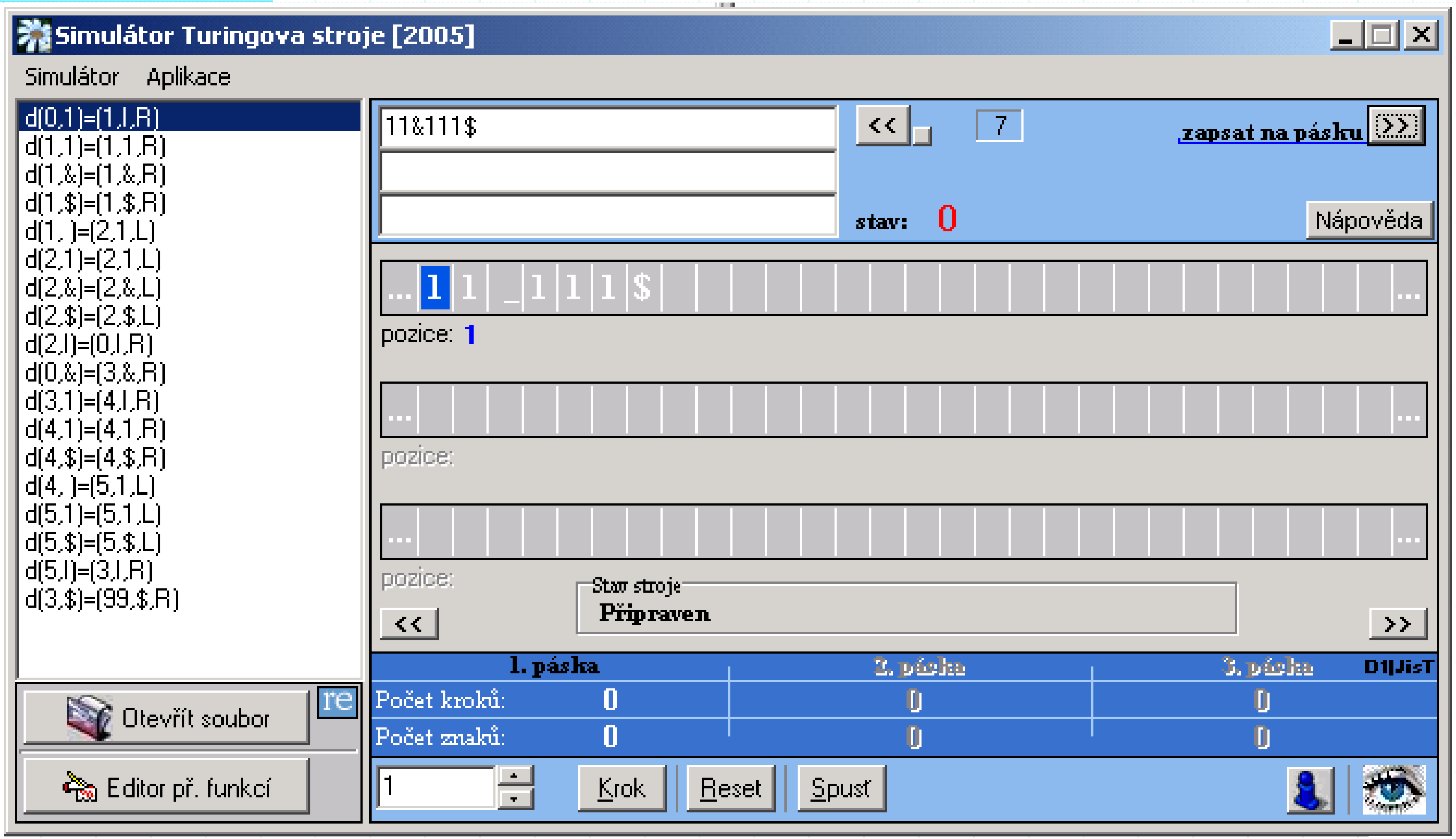
musím prejsť do q_3 a nie q_0, lebo by spracoval nekonečný cyklus - nekonečné množstvo úsekov $11...1\not{C}11...1\not{C}11...1\not{C}$

$$\delta(q_0, \not{C}) = (q_3, \not{C}, R)$$

$$copy\ b \begin{cases} \delta(q_3, 1) = (q_4, \underline{1}, R) & \\ \delta(q_4, y) = (q_4, y, R) & y \in \{1, \$\} \\ \delta(q_4, B) = (q_5, 1, L) & \\ \delta(q_5, y) = (q_5, y, L) & \\ \delta(q_5, \underline{1}) = (q_3, \underline{1}, R) & \end{cases}$$

$$\delta(q_3, \$) = (q_6, \$, R) \qquad F \in \{q_6\}$$

Simulácia T-vypočítateľnej funkcie **f(a,b)=(a+b)** na simulátore TS



Ďakujem za pozornosť.

chuda@fiit.stuba.sk

STU
FIIT

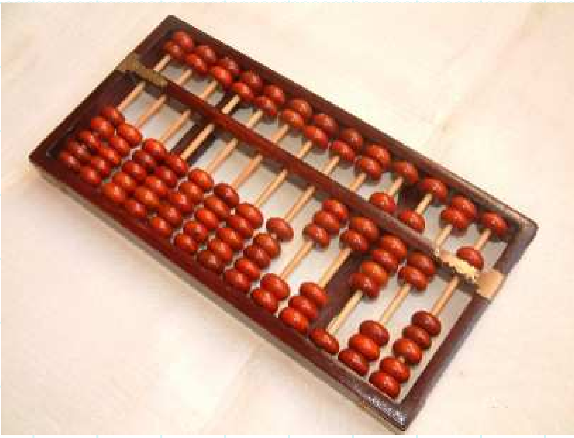
Teoretické základy informatiky

Abacus Machine

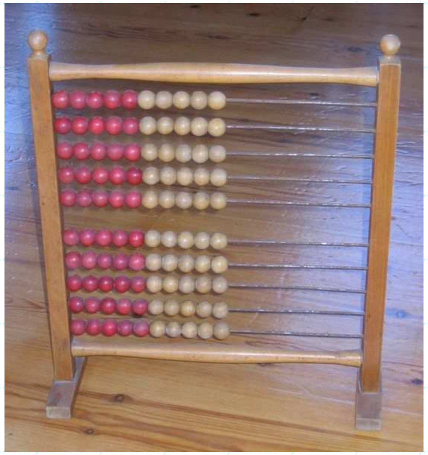
Mgr. Daniela Chudá, PhD., chuda@fiit.stuba.sk

Abacus počiatky: Babylon pred 5000 rokmi, Čína pre 3000 rokmi

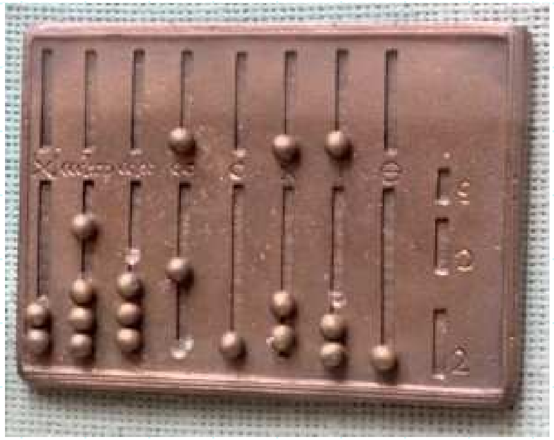
Abacus = počítadlo



A Chinese abacus



School abacus used in Danish elementary school. Early 20th century.



Reconstructed Roman Abacus

Zdroj: http://en.wikipedia.org/wiki/Abacus

Abacus Machine

- Abacus Machine = Počítadlový stroj
- Výpočtový model s operáciami inkrementovania a dekrementovania
- Počítadlový stroj nad prirodzenými číslami s operáciami:
  - +1
  - -1
  - cyklus

Definícia: Abacus Machine

- **Primitívy:**
  Symboly a_k, s_k ($k \in N$) sú jednoduché počítadlové stroje hĺbky 0
- **Rekurzívna definícia:**
  Nech $M_1, M_2, \dots M_Z$ sú jednoduché počítadlové stroje s hĺbkami $h_1, h_1, \dots h_Z$, postupnosť $M_1, M_2, \dots M_Z$ sa nazýva počítadlový stroj s hĺbkou

  $$\overline{h} = \max \{h_1, h_1, \dots h_Z\}$$
- **Kompozícia:**
  Nech M je počítadlový stroj s hĺbkou h. Jednoduchý počítadlový stroj s hĺbkou h+1 je postupnosť $(M)_k$, $k \in N$.

Neformálny opis počítadlového stroja registre

- pracuje s nekonečným počtom registrov R_i,i$\in$N, v danom čase s použitým konečným počtom registrov,
- v registroch môžu byť uložené ľubovoľne veľké prirodzené (nezáporné celé) čísla,
- predpokladá sa, že na začiatku výpočtu sú registre nastavené na 0,
- do zvolených registrov sa zapíšu vstupné hodnoty, počítadlový stroj vykoná operácie a po skončení výpočtu sú v zvolených registroch výstupné údaje .

Neformálny opis počítadlového stroja sémantika

- a_i pripočítavanie (addition) +1 $R_i \leftarrow R_i+1$
- s_i odpočítavanie (subtraction) ⊖1 $R_i \leftarrow R_i \ominus 1$

  len pre nezáporné čísla

$$x \ominus y = \begin{cases} x - y & \text{ak } x > y \\ 0 & \text{inak} \end{cases}$$

- $M_1M_2 \ldots M_n$ zreťazenie počítadlových strojov, vykonávanie sekvenčne za sebou
- $(M)_k$ cyklus počítadlových strojov, otestuje sa register R_k, v prípade, že je nenulový vykoná sa operácia zodpovedajúca stroju M. Stroj M sa cyklicky vykonáva až kým R_k nenadobudne nulovú hodnotu.

  while (R_k>0) M

Prepis AM pomocou G

$G=(N,T,P,M)$

$N=\{M\}$

$T=\{a_k , s_k , (,)_k\}$

P: $M \to a_k \mid s_k$

$M \to MM$

$M \to (M)_k$

Príklad AM $f(i)=2+i$

Vstup R_1: i

Výstup R_1: $f(i)=2+i$

Výpočet na AM

$a_1 a_1$

Príklad AM $f(i,j)=i+j$

Vstup R_1: i

R_2: j

Výstup R_1: $f(i,j)=i+j$

Výpočet na AM (deštruktívne kopírovanie)

R_1:3	R_2:2
4	1
5	0

$(s_2 a_1)_2$

zovšeobecnenie: $(s_j a_i)_j$ $R_j \to R_i$

R_2 odpočítavam –1 až po 0 a zároveň pripočítavam +1 ku R_1

Príklad AM $f(i,j)=i+j$

Vstup R_1: i

R_2: j

Výstup R_3: $f(i,j)=i+j$

Výpočet na AM (deštruktívne kopírovanie)

$$(s_1a_3)_1(s_2a_3)_2$$

Príklad AM $f(i,j)=i+j$

Vstup R_i: i

R_j: j

Výstup R_j: $f(i,j)=i+j$

Výpočet na AM (nedeštruktívne kopírovanie, zachováva len R_i)

$(s_i a_k a_j)_i\ (s_k a_i)_k$

Príklad AM $f(i,j)=i+j$

Vstup R_1: i

R_2: j

Výstup R_3: $f(i,j)=i+j$

Výpočet na AM (nedeštruktívne kopírovanie)

$$(s_1a_0a_3)_1\,(s_0a_1)_0\,(s_2a_0a_3)_2\,(s_0a_2)_0$$

Príklady

$(s_i)_i$	vynulovanie registra R_i
$a_j(a_j)_j$	nekonečný cyklus
$(a_j)_j$	ak $R_j > 0$, nekonečný cyklus

Poznámky

Keď konštruujem počítadlové stroje, môžu byť syntakticky správne, ale nemusia byť sémanticky zmysluplné.

Môže byť veľmi ťažké zistiť, čo robí počítadlový stroj.

Nie je triviálne zistiť, či niekedy skončí počítadlový stroj, ktorý syntakticky správne zapíšem – algoritmicky neriešiteľné (problém zastavenia TM).

Veta o ekvivalencii medzi TM a AM

Funkcia f: $N^k \rightarrow N$ je T-vypočítateľná práve vtedy ak f je vypočítateľná na počítadlovom stroji (AM).

Ekvivalencia výpočtových modelov

Veta 6.4.1 *(O ekvivalencií výpočtových modeloch) Nasledujúce výpočtové modely sú ekvivalentné:*

1. *Turingov stroj*
2. *Počítadlový stroj*
3. *Stroj RAM*

$$\begin{tikzcd} \mathrm{RAM} \arrow[dr, Rightarrow] & & \mathrm{TM} \arrow[ll, Rightarrow] \\ & \mathrm{AM} \arrow[ur, Rightarrow] & \end{tikzcd}$$

Ekvivalencia výpočtových modelov

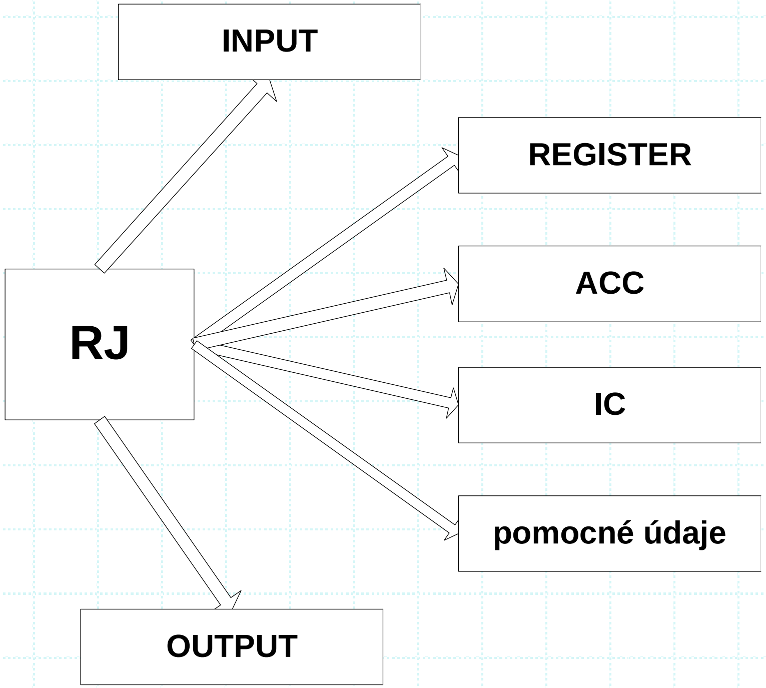
Veta 6.4.1 *(O ekvivalencií výpočtových modeloch) Nasledujúce výpočtové modely sú ekvivalentné:*

1. *Turingov stroj*
2. *Počítadlový stroj*
3. *Stroj RAM*

$$\begin{tikzcd}
\mathrm{RAM} \arrow[dr, Rightarrow] & & \mathrm{TM} \arrow[ll, Rightarrow] \\
& \mathrm{AM} \arrow[ur, Rightarrow] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathrm{RAM} \arrow[dr, Leftrightarrow] \arrow[rr, Leftrightarrow] & & \mathrm{TM} \\
& \mathrm{AM} \arrow[ur, Leftrightarrow] &
\end{tikzcd}$$

Ekvivalencia RAM <=> TS

Simulácia RAM na TS – 6 páskový TS, simuluje sa každá inštrukcia zvlášť podľa polohy IC



AM ⇨ TM reprezentácia registrov

AM má nekonečne veľa registrov $R_0, R_1, \ldots$. tie potrebujem reprezentovať na TM → na páske v unárnej sústave.

dátová časť:

0	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

$R_0=2$ $R_1=1$ $R_2=0$ $R_3=2$ $R_4=1$ $R_5=3$

AM ⇨ TM definícia 4 jednoduchých TM

SEARCH (R_k)

na začiatku je čítacia hlava na pevnej nule, SEARCH (R_k) posunie hlavu ne register R_k – vyhľadanie hodnoty, ktorá sa nachádza v registri R_k

SHIFT (0R)

posun nuly o jedno políčko doprava, modifikuje pásku

SHIFT (0L)

posun nuly o jedno políčko doľava, modifikuje pásku

TEST (ZERO)

kontrolujem, či daný register (kde sa nachádza hlava) obsahuje 0 hodnotu (posunie sa vľavo)

AM ⇨ TM

definícia 4 jednoduchých TM

SHIFT (0R)

0	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**B**
0	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**

$\delta(q_x,1)=(q_x,1,R)$
$\delta(q_x,0)=(q_{x+1},1,R)$
$\delta(q_{x+1},1)=(q_x,0,R)$
$\delta(q_{x+1},0)=(q_{x+1},0,R)$
$\delta(q_{x+1},B)=(q_{x+2},0,L)$
$\delta(q_{x+2},n)=(q_{x+2},n,L)$ $\quad n\epsilon\{0,1\}$
$\delta(q_{x+2},B)=(q_{xF},B,R)$

AM ⇨ TM   definícia 4 jednoduchých TM

TEST (ZERO)

posuniem sa na najľavejší symbol na páske a buď skončím na q_{F0} alebo q_{F1} . Akceptujem: q_{F0} test bol nulový,
q_{F1} test bol nenulový.

$\delta(q_y,0)=(q_{y0},0,L)$
$\delta(q_y,1)=(q_{y1},1,L)$
$\delta(q_{y0},u)=(q_{y0},u,L)$ $u \in \{0,1\}$
$\delta(q_{y1},u)=(q_{y1},u,L)$
$\delta(q_{y0},B)=(q_{F0},B,R)$
$\delta(q_{y1},B)=(q_{F1},B,R)$

AM ⇨ TM

dôkaz indukciou a_k, s_k

a_k: SEARCH (R_k) → SHIFT (0R) +1

s_k: SEARCH (R_k) → SHIFT (0L) -1

AM ⇨ TM dôkaz indukciou $M_1M_2\,M_3\ldots\,M_Z$

Mám $M_1, M_2, M_{3\ldots}, M_Z$ a z nich skonštruujem $M_1M_2M_3\ldots\,M_Z = M$ môžem ich vedľa seba naukladať.

indukčný predpoklad: nech M_i ⇨ TM_i

potom mám $TM_1, TM_2, TM_3\ldots, TM_Z$

? ako zostrojím TM, ktorý bude reprezentovať M? M ⇨ TM

zreťazím ich: $TM_1 \rightarrow TM_2 \rightarrow TM_3\ldots \rightarrow TM_Z$

AM ⇨ TM   dôkaz indukciou $(M)_k$

mám M a zostrojím $(M)_k$

indukčný predpoklad M ⇨ TM

? pýtam sa ako zostrojím TM, ktorý zodpovedá $(M)_k$?

SEARCH (R_k) → TEST (ZERO) $\xrightarrow{q_{F0}}$ END

TEST (ZERO) $\xrightarrow{q_{F1}}$ TM → SEARCH (R_k)

Veta o ekvivalencii medzi RAM a AM

Funkcia f je vypočítateľná na počítadlovom stroji (AM) práve vtedy ak existuje RAM, ktorý počíta f.

Ekvivalencia výpočtových modelov

AM ⇦ RAM ⇦ TM ⇦ AM

Pozn.

Platí aj opačná implikácia, dokazuje sa ťažšie.

AM ⇨ RAM

štrukturálna indukcia

jednojednoznačné zobrazenie namapovania registrov:

problém R_0 v RAM - akumulátor
predpoklad pre AM - R_1, R_2, R_3, R_4,... R_n
Zostrojím nový AM, kde prečíslujem registre

$R_i \rightarrow R_{i+1}$

Namapujem registre

$\mathbf{AM_{NEW}}$		**RAM**
R_i	$\rightarrow$	R_i

Využívam skutočnosť, že RAM aj AM majú spočítateľne nekonečne veľa registrov.

AM ⇨ RAM dôkaz indukciou a_k, s_k

a_k:

LOAD k

ADD =1

STORE k

s_k:

LOAD k

SUB =1

STORE k

AM ⇨ RAM dôkaz indukciou $M_1, M_2, M_3 \dots, M_Z$

predpokladám, že mám $M_1, M_2, M_3 \dots, M_Z$

indukčný predpoklad $M_j \sim I_j$

AM $M_1 M_2 M_3 \dots M_Z$ skonštruujem tak, že inštrukcie budú za sebou

$\}\ I_1$

$\}\ I_2$

…

$\}\ I_Z$

AM ⇨ RAM

dôkaz indukciou $(M)_k$

chcem skonštruovať $(M)_k$

indukčný predpoklad $M_j \sim I_j$

musím zostrojiť cyklus

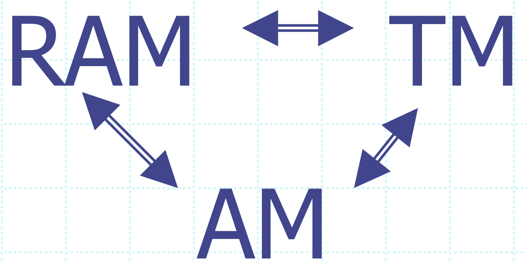
$(M)_k$

```
next:   LOAD k
        JZERO end
        ______
        ______  } I
        ______
        JUMP next
end:
```

Ekvivalencia výpočtových modelov

Veta 6.4.1 *(O ekvivalencií výpočtových modeloch) Nasledujúce výpočtové modely sú ekvivalentné:*

1. *Turingov stroj*
2. *Počítadlový stroj*
3. *Stroj RAM*



Ďakujem za pozornosť.

chuda@fiit.stuba.sk

Zadania príkladov na riešenia na cvičeniach - množiny

1. definícia množina, mohutnosť, konečnosť
2. |Odd|=|N|
3. |N+|=|N|
4. |Q|=|N|
5. {7k+1 | k patrí N} U {7k+5 | k patrí N} ~ N
6. {ak+b | a,b,k patrí N ^ a<>0} ~ N
7. {7k | k patrí N} U {5k | k patrí N} ~ N
8. N X {2,4,6} ~ N
9. Z X {2,4,6} ~ N
10. (0,1) ~ (0,2)
11. (0,1) ~ (a,b)
12. (-pi/2,pi/2) ~ R
13. |P(A)| = 2^|A|
14. N X Z ~ N

****

1. Q ~ N
2. <0,1) ~ (0,1)
3. <0,1> ~ (a,b) alebo <a,b> ~ (a,b)
4. (0,1) ~ (0,2> U (4,7)
5. (0,1) ~ (0,oo)
6. <0,1) ~ <0,1) X {0,1,2,3}
7. <0,1) ~ <0,1) X N
8. dokážte tvrdenie: Ir je nespočítateľná
9. dokážte tvrdenie: R \ {2 na -k | k patrí Z} je nespočítateľná

Zadania príkladov na riešenia na cvičeniach - jazyky a gramatiky

1. definícia gramatiky
2. c^(4n)
3. a^i b^j
4. prvé dve písmená rovnaké
5. práve jedna 0 na posledných dvoch miestach
6. obsahuje podslovo ab
7. záporné čísla deliteľné 5
8. a^n b^n
9. a^(n+1) b^n, a^(3n+1) b^n, a^(3n+1) b^(n+3), a^(3n+2) c b^(n+3)
10. a^x b^y a^y b^x, a^(x+3) b^(2y) a^y b^x
11. 0^x 1^(3x) 0^y 1^(y+5)

----

1. #_0 w = #_1 w; w je z {0,1}^*
2. wcw; w je z {a, b}^*
3. a^n b^n c^n; n je z N
4. a^n b^n c^n d^n; n je z N
5. a^i b^j c^i d^j; i,j je z N
6. Dokážte, že gramatika je viacznačná: S -> aaSaa | aS | a
7. Pre danú gramatiku nájdite homomorfizmus h tak, aby nová gramatika bola viacznačná. Príklad 2.5.15 zo skrípt.

Zadania príkladov na riešenia na cvičeniach - konečné automaty

1. DFA: |w|=3k+1 ; w je z {a,b}^*
2. DFA: a^i b^j c^k ; i,j,k je z N alebo i,j,k je z N+ ; zdôrazniť záchytné stavy
3. DFA: #_c w + #_d w = 3k+1 ; w je z {a, b, c, d}^*
4. DFA: w obsahuje podslovo ab & #_b w je nepárny ; w je z {a, b}^* - DÚ
5. NFA: w obsahuje podslovo aaa alebo bbb ; w je z {a, b}^*
6. NFA: w má práve jednu 0 na posledných dvoch miestach ; w je z {0, 1}^*
7. prevod NFA -> DFA pre príklad 6
****
1. w obsahuje ab alebo #b(w) je nepárny
2. práve jedna 0 na posledných dvoch miestach A -> G
3. w začína aj končí tým istým písmenom, w patrí {a,b}* , G->NKA->DKA
4. zovšeobecnenie úloh #a w = 3k+2 , #a w + #b w = 3k+1 , #a w + 3.#b w - 2.#c w = 4k+1
5. NKA, DKA: xaaby
6. DKA: obsahuje aaa alebo bbb
7. DKA: rozpoznáva kladné čísla, celé čísla
8. porozmýšlať: rozpoznáva reálne čísla, desiatkové párne čísla, záporné deliteľné 5
9. |w|=2k+1 v #a w= 3k+1 -čo je zjednotenie jazykov L1 u L2, L1={|w|=2k+1,w patrí {a,b}* }, L2 ={#a w= 3k+1,w patrí {a,b}* }
10. zreťazenie L1.L2
11. cleeneho iterácia L1*
12. kladný uzáver L1+

Zadania príkladov na riešenia na cvičeniach - zásobníkové automaty

DPDA wcw^R , w patrí {a,b}*
DPDA 0 a^n 12 b^n 34 , n je z N
DPDA a^n w , #_c w = n , w patrí {b,c}*
DÚ: DPDA w=uv , #_(w = #_) w , u patrí {a,(}* , v patrí {a,)}*
DPDA a^3n b^2n , n je z N
DPDA w=uv , 2|u| = 3|v| , u patrí {a,b}* , v patrí {c,d}*
NPDA ww^R , w patrí {a,b}*
NPDA uav , |u| = |v| , u,v patrí {a,b}*
DPDA wcw^R , w patrí {a,b}*
DPDA a^Nb^N
DPDA 0 a^n 12 b^n 34 , n je z N
DPDA a^3n b^2n , n je z N
DPDA w=uv , 2|u| = 3|v| , u patrí {a,b}* , v patrí {c,d}*
DÚ: DPDA w=uv , #_(w = #_) w , u patrí {a,(}* , v patrí {a,)}*
w patrí {a,b}*, #_a w = #_b w ,
a^k b^l , k>=l, k,l je z N
DPDA, NPDA a^n w , #_c w = n , w patrí {b,c}*
NPDA ww^R , w patrí {a,b}*
NPDA uav , |u| = |v| , u,v patrí {a,b}*
urobte prevod G -> PDA: G=(N,T,P,S) P={S->aAA|a , A->aSA|bS|b}
urobte prevod G -> PDA: G=(N,T,P,S) P={S->aACA|a|b , A->aABAS|aBB|bB|b, B->+SC|bB|aAA|a, C->+}

Zadania príkladov na riešenia na cvičeniach - turingove stroje a LOA

0. TM definícia, konfigurácia
1. DTM formálne, delta funkcia aj konfigurácia: 0^n 1^n , n je z N+
2. DTM formálne bol na prednáške, iba pripomenúť myšlienku: a^n b^n c^n; n je z N
3. DTM neformálne: a^n b^3n c^n d^2n; n je z N
4. DTM formálne, zdôrazniť zapamätanie symbolu v stave: ww^R , w patrí {a,b}*
5. DTM neformálne, zdôrazniť zapamätanie symbolu v stave: ww^R , w patrí {a,b,c}*
6. DTM formálne: #_a w = #_b w, w patrí {a,b}*
7. DTM neformálne: ww, w patrí {a,b}*
8. DÚ, premyslieť nápad DTM neformálne: www , w patrí {a,b}*

----

1. LBA neformálne: ww, w patrí {a,b}*
2. LBA neformálne: www , w patrí {a,b}*
3. formálne T vypočítateľná funkcia: dolná_časť(a/2)+2*b
4. neformálne T vypočítateľná funkcia: a*b
5. formálne T vypočítateľná funkcia: (n nad dvomi)
6. formálne T vypočítateľná funkcia: dolná_časť(log2n)+1

Zadania príkladov na riešenia na cvičeniach – RAM

1. 2^2^n

napíš RAM; vypočítaj jednotkovú (pamäťovú, časovú) zložitosť; logaritmickú (pamäťovú, časovú) zložitosť

2. rozpoznať postupnosť čísel 0,1,2 ; ktorá končí 0 a má rovnaký počet 1 ako 2

napíš RAM; vypočítaj jednotkovú (pamäťovú, časovú) zložitosť; logaritmickú (pamäťovú, časovú) zložitosť

ak sa stihne, tak len RAM pre príklady, ak nie na DÚ:

3. n!
4. 2^n
5. horná časť (log2 n)
6. n^n
7. načítať 100 čísel a vypísať tie ktoré sú väčšie ako priemer
8. načítať číslo a zistiť či je prvočíslo

Rekurzívna definícia počítadlového stroja:

- **a_k, s_k ($k \in N$) sú počítadlové stroje**
- **ak $M_1, M_2, \ldots M_n$ sú počítadlové stroje, tak $M_1, M_2, \ldots M_n$ je počítadlový stroj**
- **ak M je počítadlový stroj, tak $(M)_k$ je počítadlový stroj**

---

1. príklad: deštruktívne kopírovanie vstup R_1:a, R_2:b výstup R_3:a+b
   $(s_1a_3)_1(s_2a_3)_2$
   nedeštruktívne kopírovanie vstup R_1:a, R_2:b výstup R_3:a+b
   $(s_1a_0a_3)_1(s_0a_1)_0(s_2\, a_0a_3)_2(s_0a_2)_0$
2. príklad: mínus v krúžku $\ominus$
3. príklad: násobenie
4. príklad: horná celá časť a/2
5. príklad: dolná celá časť a/2
6. príklad: dolná celá časť $\log_2 a$
7. príklad: $f(a,b)=a^b$ (dú)
8. príklad: $f(n)=\binom{n+1}{2}$
9. príklad: f(n)=n!
10. príklad: f(a)=a mod 2
11. príklad: f(a,b)=a div b (dú)
12. príklad: f(a,b)=a mod b (dú)
13. príklad: if (R_i>0) M
14. príklad: if (R_i=0) M

5 Turingove stroje a LBA

Zadania na projekt č. 2

Dokážte, že nasledujúce funkcie sú T-vypočítateľné.

1 $f(i,j,k) = \lceil \log_2(1 + i + j \ominus k) \rceil$
2 $f(n) = 2n + \lceil \log_2 n \rceil$
3 $f(i,j) = \binom{i+j}{2} + 1$
4 $f(i,j,k) = 2i + 3j + 3k \ominus 2$
5 $f_4(i,j,k) = \lfloor i/2 \rfloor + \lfloor j/3 \rfloor + \lfloor k/2 \rfloor$
6 $f(i,j,k) = \lceil \log_2(2i + j + k) \rceil$
7 $f(i,j,k) = \lceil j/3 \rceil + \lceil \log_2(i + k) \rceil$
8 $f(i,j,k) = 3j + \lceil \log_2(i \ominus k) \rceil$
9 $f(n) = n + (2n)^2$
10 $f(n) = (n \ominus 2)^2$
11 $f(n) = n + n^2 \ominus 3$
12 $f(i,j,k) = (i + k)^2 + 3j + 1$
13 $f(i,j,k) = \lceil i/4 \rceil + \lceil j/3 \rceil \ominus k$
14 $f(i,j) = i + \binom{2j}{2}$
15 $f(i,j) = i^2 + 2j$

Definujte deterministické Turingove stroje, ktoré rozpoznávajú nasledujúce jazyky.

16 $L = \{\ u\%v\%w \mid |u| = |v| = |w|, u \in \Sigma_1^+, v \in \Sigma_2^+, w \in \Sigma_3^+\ \}$
17 $L = \{\ w0h(w)0h(w) \mid w \in \{a,b,c\}^+\ \}$, kde $h(a) = 4,\ h(b) = 6,\ h(c) = 8$
18 $L = \{\ u\%v\%w \mid \#_a u = \#_b v = \#_c w, u \in \{a,d\}^*, w \in \{b,d\}^*, w \in \{c,d\}^*\ \}$
19 $L = \{\ a^{3i}uv \mid |u| = \#_b v = i, i \in \mathbb{N}, u \in \{c,d\}^*, v \in \{a,b\}^*\ \}$
20 $L = \{\ a^{3i}b^j c^{2i} d^{2j} \mid i,j \in \mathbb{N}^+\ \}$
21 $L = \{\ uvw \mid u \in \{a,b\}^+,\ v \in \{0,1\}^+,\ w \in \{2,3\}^+,\ 2 \cdot |u| = |v| = 3 \cdot |w|\ \}$
22 $L = \{\ a^{2i}b^j c^{2k} d^l \mid i,j,k,l \in \mathbb{N},\ i \neq k\ \vee\ j \neq l\ \vee\ i \neq l\ \}$
23 $L = \{\ ww^R w \mid w \in \{b,c,d\}^*\}$
24 $L = \{\ a^{2n}b^n 0 c^{4n} d^{3n} \mid n \in \mathbb{N}\ \}$
25 $L = \{\ w\%\%w \mid w \in \Sigma^+\ \}$, kde $|\Sigma| \geq 4$ a $\% \notin \Sigma$
26 $L = \{\ w0h(w)0w0h(w) \mid w \in \{b,d\}^*\ \}$, kde $h(b) = a,\ h(d) = c$

Dokážte, že nasledujúce funkcie sú T-vypočítateľné.

27 $f(i,j,k) = \lceil \log_2 \lceil (i + j + k)/3 \rceil \rceil$
28 $f(i,j) = \binom{i \ominus j}{2}$
29 $f(i,j) = 2(i + j)^2$
30 $f(n) = \lceil \log_2(3n) \rceil$
31 $f(i,j) = (i + 2j)^2$
32 $f(i,j) = (i \ominus j)^2 + 1$
33 $f(n) = \lceil \log_2(n + 2) \rceil$
34 $f(i,j,k) = (i \ominus j + k)^2$
35 $f(n) = \binom{n+3}{2}$
36 $f(i,j,k) = 1 + i + k + \lceil \log_2 j \rceil$
37 $f(i,j,k) = \lceil i/3 \rceil \ominus 3j + \lceil k/2 \rceil + 2$
38 $f(i,j,k) = \lceil \log_2(3i + \lceil j/3 \rceil + k) \rceil$
39 $f(n) = \binom{2n}{2}$

40 $f(n) = 2n + \binom{n}{2}$
41 $f(n) = (n+3)^2 + 1$
42 $f(i,j,k) = \lceil \log_2(j+k+2) \rceil \ominus i$

Definujte deterministické Turingove stroje, ktoré rozpoznávajú nasledujúce jazyky.
43 $L = \{\ wcwc(w)^R cwc \mid w \in \{a,b\}^*\ \}$
44 $L = \{\ a^{3n+1} b^{2n+1} c^{5n+3} d^i \mid n \in \mathbb{N}, i \in \mathbb{N}^+\ \}$
45 $L = \{\ w\%w\%w \mid w \in \Sigma^*\ \}$, kde $|\Sigma| \geq 3$ a $\% \notin \Sigma$
46 $L = \{\ wa^j b^j w \mid w \in \{4,5,6\}^*, j \in \mathbb{N}^+\ \}$
47 $L = \{\ 9a^{2n} b^{n+2} c^{3n} d^{2n} \mid n \in \mathbb{N}\ \}$
48 $L = \{\ u\%v\%w \mid |u| = |v| = \#_a w, u \in \Sigma_1^*, v \in \Sigma_2^*, w \in \{a,b,c\}^*\ \}$
49 $L = \{\ w0h(w) \mid w \in \{a,b,c\}^+\ \}$, kde $h(a) = 1,\ h(b) = \{2,3\},\ h(c) = 4$
50 $L = \{\ ww^R \mid w \in \{3,4,5,6,7\}^+ \}$
51 $L = \{\ a^i b^{3j} c^{2k} \mid i,j,k \in \mathbb{N},\ i \neq j\ \vee\ j \neq k\ \vee\ i \neq k\ \}$
52 $L = \{\ uvw \mid u \in \{a,b\}^+,\ v \in \{c,d\}^+,\ w \in \{0,1\}^+,\ |u| \neq |v|\ \vee\ |v| \neq |w|\ \}$

Cvičenia

5.1 Nasledujúce jazyky patria do triedy $\mathcal{L}(NPDA) \setminus \mathcal{L}(DPDA)$. Neformálne opíšte deterministické lineárne ohraničené automaty, ktoré rozpoznávajú nasledujúce jazyky.
a) $L_1 = \{ww^R \mid w \in \{a,b\}^*\}$
b) $L_2 = \{ww^R \mid w \in \{1,2,3,4\}^+\}$
c) $L_3 = \{a^i b^j c^k \mid i,j,k \in \mathbb{N},\ \text{buď } i \neq j \text{ alebo } j \neq k\}$
d) $L_4 = \{uavbw \mid u,v,w \in \{a,b\}^*,\ |v| = |u| + |w|\}$

5.2 Neformálne opíšte nedeterministické lineárne ohraničené automaty, ktoré rozpoznávajú nasledujúce jazyky.
a) $L_1 = \{ww \mid w \in \{a,b\}^*\}$
b) $L_2 = \{www \mid w \in \{a,b,c\}^*\}$
c) $L_3 = \{1w2w3w4 \mid w \in \{1,2,3,4\}^*\}$
d) $L_4 = \{uavbw \mid u,v,w \in \{a,b\}^*,\ |v| = |u| + |w|\}$

5.3 Platí, že nasledujúce jazyky nie su bezkontextové. Dokážte, že sú kontextové. (Pre parameter t platí: $t \in \mathbb{N}$, $t \geq 3$.)
a) $L_1 = \{a^{\alpha.n} b^{\beta.n} c^{\gamma.n} \mid n \in \mathbb{N}\}$ $\alpha,\ \beta,\ \gamma \in \mathbb{N}^+$
b) $L_2 = \{wcw^R cw \mid w \in \{a,b\}^*\}$
c) $L_3 = \{c(a^n c)^t \mid n \in \mathbb{N}\}$
d) $L_4 = \{w_1 c w_2 c \dots c w_t \mid w_i \in \{a,b\}^*,\ i \in \{1,\dots,t\},\ |w_1| = |w_2| = \dots = |w_t|\ \}$

5.4 Dokážte, že nasledujúce funkcie sú T-vypočítateľné.
a) $f_1(i,j,k) = 2i + 2^{(j+1\ominus k)}$
b) $f_2(i,j,k) = 2^{(1+i+j\ominus k)}$
c) $f_3(i,j,k) = 2^{(\lceil i/2 \rceil + j + 3k)}$
d) $f_4(i,j,k) = 1 + 2^{\lceil (i+j+k)/3 \rceil}$
e) $f_5(i,j,k) = i \ominus k + 2^{2j}$
f) $f_6(i,j,k) = 3k + 2^{\lceil (1+i+j)/3 \rceil}$
g) $f_7(i,j,k) = \lceil j/2 \rceil + 2^{(i+2k)}$
h) $f_8(n) = n!$

Zadania projektu č.2 – Počítače RAM

1. Vypíšte prvých N prirodzených čísel, ktoré nie sú mocninou čísla 2, v poradí od najväčšieho po najmenšie.
2. Zistite, či kladné celé číslo N je mocninou nejakého iného prirodzeného čísla menšieho ako N.
3. Zistite, či možno vyjadriť zadané kladné celé číslo N ako súčet faktoriálov nejakých dvoch prirodzených čísel (napr. 744 = 6! + 4!)
4. Dokonalé číslo je také číslo N, pre ktoré platí, že súčet jeho deliteľov menších ako N je rovný N (napr. 6 = 3 + 2 + 1). Preverte, že číslo 8128 je dokonalé. Zistite, koľkým dokonalým číslom v poradí je.
5. Dvojica kladných celých čísel A, B sa nazýva *spriatelená*, ak súčet deliteľov čísla A sa rovná číslu B a súčet deliteľov čísla B sa rovná A. Napr. čísla 220 a 284 sú spriatelené: 220 = 1 + 2 + 4 + 71 + 142 a 284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110. Vypíšte všetky dvojice spriatelených čísel menších ako N.
6. Medzi prvočíslami od 1 po N nájdite dvojicu susedných prvočísel s najväčším rozdielom.
7. Polynóm $f(x)$ je rovný $x^2 - 79x + 1601$. Vstupom je prirodzené číslo N a výstupom je celočíselná hodnota v percentách určujúca podiel medzi počtom prvočíselných funkčných hodnôt pre x=1,2,3,....N a počtom všetkým funkčných hodnôt pre x=1,2,3,....N.
Upresnenie: Pod hľadaným podielom sa myslí pomerná časť, ktorú predstavuje množina $\{f(x) \mid x=1,2,\ldots,N, f(x)$ je prvočíslo$\}$ vzhľadom na celú množinu $\{f(x) \mid x=1,2,\ldots,N\}$
8. Číslo N rozložte (ak sa rozložiť dá) na súčet dvoch prvočísel.
9. Vypíšte prvočíslo, ktoré v prvočíselnom rozklade čísla N vystupuje s maximálnou mocninou.
10. Načítajte kladné celé číslo N; jeho ciferný zápis cyklicky posuňte o 3 miesta doprava a získané číslo vypíšte.
11. Načítajte kladné celé číslo N; jeho ciferný zápis obráťte a získané číslo vypíšte.
12. N-ciferné číslo sa nazýva Armstrongovo, ak súčet N-tých mocnín jeho cifier je rovný jemu samému (napr. $153 = 1^3 + 5^3 + 3^3$; $407 = 4^3 + 0^3 + 7^3$). Nájdite a vypíšte všetky dvoj-, troj- a štvorciferné Armstrongove čísla.
13. Vypíšte prvých N čísel takých, ktorých posledná cifra sa rovná súčtu všetkých predchádzajúcich cifier.
14. Načítajte kladné celé číslo N a prirodzené číslo z $(2 <= z <= 9)$ označujúce číselnú sústavu. Preveďte číslo N do sústavy o základe z.
15. Načítajte tri prirodzené čísla M, N, O a vypíšte ich v takom poradí, aby počet núl v ich dvojkových zápisoch neklesal.
16. Načítajte kladné celé číslo N a po ňom nasledujúcich N celých čísel. Usporiadajte načítané čísla algoritmom INSERT SORT.
17. Načítajte kladné celé číslo N a po ňom nasledujúcich N celých čísel. Usporiadajte načítané čísla algoritmom SELECT SORT.
18. Načítajte kladné celé číslo N a po ňom nasledujúcich N celých čísel. Usporiadajte načítané čísla algoritmom BUBBLE SORT.
19. Napíšte program, ktorý rozpozná jazyk $L=\{ww^R w \mid w$ je slovo nad abecedou $A=\{0,1,2,\ldots,9\}\}$

20. Vytvorte storočný kalendár pre roky 2000 až 2100. Vstupom bude rok, mesiac a deň, výstupom poradové číslo dňa v týždni. Rok 2000 bol prestupný, rok 2100 prestupný nie je, 1.1.2000 bola sobota