FIRST, FOLLOW, Zásobníkové automaty

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

viliam.hromada@stuba.sk

- Je zrejmé, že to, aké reťazce generujú bezkontextové gramatiky - a či existuje derivácia nejakého reťazca v gramatike - úzko súvisí s pravidlami danej gramatiky.
- Preto sa pri bezkontextových gramatikách vyšetrujú tzv. množiny *FIRST* a *FOLLOW*, ktoré bližšie špecifikujú, aké vlastnosti majú vetné formy, resp. reťazce vznikajúce počas derivácií.
- Následne má potom zmysel tieto vlastnosti využívať pri zisťovaní, či derivácie reťazcov v gramatike existujú alebo nie.
- Množina *FIRST* pojednáva o tom, aké terminálne symboly môžu stáť **na prvom mieste** reťazca, ktorý viem derivovať z nejakej postupnosti symbolov gramatiky.
- Množina *FOLLOW* pojednáva o tom, aké terminálne symboly môžu **nasledovať** nejaký neterminál vo vetnej forme.

Množina *FIRST*

Definícia

Nech $G = (N, T, P, S)$ je redukovaná bezkontextová gramatika a $\alpha \in (N \cup T)^*$. Potom

$$FIRST(\alpha) = \{a \in T \mid \alpha \Rightarrow^* a\beta, \beta \in (N \cup T)^*\} \cup \{\varepsilon \mid \alpha \Rightarrow^* \varepsilon\}. \tag{1}$$

T.j. pre reťazec zložený z neterminálov a terminálov α je $FIRST(\alpha)$ množina terminálov, ktorými môžu začínať reťazce odvodené z α. A ak je možné z α odvodiť ε, tak do $FIRST(\alpha)$ patrí aj ε).

Odvodenie ε z neterminálov

Uvažujme o gramatike $G = (N, T, P, S)$. Budeme hľadať množinu neterminálov N_ε, z ktorých je možné odvodiť prázdny reťazec. Teda neterminály, ktoré sa môžu počas odvodzovania „stratiť“, t.j. prepísať na prázdne slovo ε.

$$N_\varepsilon = \{A \in N \mid A \Rightarrow^* \varepsilon\} \tag{2}$$

Množinu tvoria tie neterminály, z ktorých možno odvodiť ε.

Algoritmus nájdenia N_ε

Vstup: bezkontextová gramatika $G = (N, T, P, S)$
Výstup: množina N_ε (2)

$N_\varepsilon \leftarrow \emptyset$.
opakuj
  $\acute{N}_\varepsilon \leftarrow N_\varepsilon$
  $N_\varepsilon \leftarrow \acute{N}_\varepsilon \cup \{A \mid A \to \alpha \in P \land \alpha \in \acute{N}_\varepsilon{}^*\}$
pokiaľ $N_\varepsilon \neq \acute{N}_\varepsilon$

Množina *FIRST* - rôzne situácie

1. ak gramatika obsahuje pravidlo $A \rightarrow a\alpha$, potom $a \in FIRST(A)$
2. ak $A \in N_\varepsilon$, potom $\varepsilon \in FIRST(A)$
3. ak gramatika obsahuje pravidlo $A \rightarrow \alpha$, potom $FIRST(\alpha) \subseteq FIRST(A)$

Množina *FIRST* - príklad

Nech je daná gramatika $G = (N, T, P, S)$, kde
$N = \{S, A, B, C\}$, $T = \{a, b, +, (,), \$\}$ a pravidlá:

$$S \to C\$$$

$$A \to b \mid \varepsilon$$

$$B \to +S \mid \varepsilon$$

$$C \to A(C) \mid aB$$

Potom:

- $FIRST(B) = \{+, \varepsilon\}$
- $FIRST(C) = \{b, a, (\}$
- $FIRST(A) = \{b, \varepsilon\}$
- $FIRST(S) = \{a, b, (\}$

$FIRST(aBA) = \{a\}$

$FIRST(AB) = \{b, \varepsilon, +\}$

$FIRST(A\$B) = \{b, \$\}$

Množina $FIRST(X)$ I

Vstup: Redukovaná bezkontextová gramatika $G = (N, T, P, S)$.
Výstup: Množiny $FIRST(X)$ pre $X \in N \cup T$

pre všetky $A \in N$ **rob**
    $FIRST(A) \leftarrow \emptyset$
koniec pre
pre všetky $a \in T$ **rob**
    $FIRST(a) \leftarrow \{a\}$
koniec pre
pre všetky pravidlá $A \rightarrow a\alpha$ **rob**
    $FIRST(A) \leftarrow FIRST(A) \cup \{a\}$
koniec pre
pre všetky $A \in N_\varepsilon$ **rob**
    $FIRST(A) \leftarrow FIRST(A) \cup \{\varepsilon\}$
koniec pre

Množina $FIRST(X)$ II

opakuj
    pre všetky pravidlá $A \to B\alpha$ **rob**
        $FIRST(A) \leftarrow FIRST(A) \cup FIRST(B\alpha)$
    koniec pre
pokiaľ zmenila sa niektorá z množín $FIRST(A), A \in N$

Množina $FIRST(\alpha)$

Vstup: množiny $FIRST(X), X \in N \cup T$; reťazec $\alpha \in (N \cup T)^*$.

Výstup: $FIRST(\alpha), \alpha \in (N \cup T)^*$

```
ak α = ε potom
    vráť {ε}
inak
    nech α = X_1 X_2 ... X_n
    FIRST(α) ← FIRST(X_1) − {ε}
    i ← 1
    pokiaľ ε ∈ FIRST(X_i) ∧ i < n rob
        i ← i + 1
        FIRST(α) ← FIRST(α) ∪ (FIRST(X_i) − {ε})
    koniec pokiaľ
    ak i = n ∧ ε ∈ FIRST(X_n) potom
        FIRST(α) ← FIRST(α) ∪ {ε}
    koniec ak
    vráť FIRST(α)
koniec ak
```

Množina *FIRST* - príklad

Nech je daná gramatika $G = (N, T, P, S)$, kde
$N = \{S, A, B, C\}$, $T = \{a, b, +, (,), \$\}$ a pravidlá:

$$S \to C\$$$

$$A \to b \mid \varepsilon$$

$$B \to +S \mid \varepsilon$$

$$C \to A(C) \mid aB$$

Určte $FIRST(X)$ pre všetky $X \in N \cup T$, $FIRST(BS)$, $FIRST(AB)$, $FIRST(CB)$

Množina *FIRST* - príklad

$FIRST(X)$:

$FIRST$	S	A	B	C	a	b	$+$	(	)	\$
Terminály:					a	b	$+$	(	)	\$
Pravidlá $A \to a\alpha$		b	$+$	a						
Neterminály $A \in N_\varepsilon$		ε	ε							
1. iterácia:	a			b, (						
2. iterácia:	b, (									
3. iterácia:										
Celkom:	a, b, (	b, ε	$+$, ε	a, b, (	a	b	$+$	(	)	\$

Množina *FIRST* - príklad

$$FIRST(BS) = \{+, a, b, (\}$$
$$FIRST(AB) = \{b, +, \varepsilon\}$$
$$FIRST(CB) = \{a, b, (\}$$

Množina *FOLLOW*

Definícia

Nech $G = (N, T, P, S)$ je redukovaná bezkontextová gramatika a $A \in N$. Potom

$$FOLLOW(A) = \{a \in T \mid S \Rightarrow^{+} \alpha A a \beta\} \cup \{\varepsilon \mid S \Rightarrow^{*} \gamma A\}, \tag{3}$$

kde $\alpha, \beta, \gamma \in (N \cup T)^{*}$.

T.j. pre neterminál A je množina $FOLLOW(A)$ množina takých terminálov, ktoré môžu v nejakej vetnej forme stáť **hneď za neterminálom** A. Do množiny patrí aj ε, ak môže neterminál A stáť **na konci** nejakej vetnej formy.

Príklad: Nech je daná gramatika $G = (N, T, P, S)$, kde $N = \{S, A, B, C\}$, $T = \{a, b, +, (,), \$\}$ a pravidlá:

$$S \to C\$$$

$$A \to b \mid \varepsilon$$

$$B \to +S \mid \varepsilon$$

$$C \to A(C) \mid aB$$

Potom:

$$FOLLOW(S) = \{\varepsilon, \$,)\}$$

$$FOLLOW(A) = \{(\}$$

$$FOLLOW(B) = \{\$,)\}$$

$$FOLLOW(C) = \{\$,)\}$$

Množina $FOLLOW(A)$

- Predpokladajme, že gramatika obsahuje pravidlo $A \to \alpha B \beta; \alpha, \beta \in (N \cup T)^*; A, B \in N$
- Predpokladajme, že existuje odvodenie: $S \Rightarrow^* ...Aa... \Rightarrow^* ...\alpha B \beta a... \Rightarrow^* ...$
  1. Vždy platí, že $\varepsilon \in FOLLOW(S)$
  2. $a \in FOLLOW(A)$
  3. $(FIRST(\beta) - \{\varepsilon\}) \subseteq FOLLOW(B)$
  4. (ak $\varepsilon \in FIRST(\beta)$, potom $FOLLOW(A) \subseteq FOLLOW(B)$.

Množina $FOLLOW(A)$ pre $A \in N$

Vstup: Redukovaná bezkontextová gramatika $G = (N, T, P, S)$, množiny $FIRST(X)$ pre $X \in N \cup T$

Výstup: množiny $FOLLOW(A)$ pre $A \in N$

pre všetky $A \in N$ **rob**
    $FOLLOW(A) \leftarrow \emptyset$
koniec pre
$FOLLOW(S) \leftarrow \{\varepsilon\}$
opakuj
    pre všetky $B \in N$ v pravých stranách pravidiel $A \to \alpha B \beta$ **rob**
        $FOLLOW(B) \leftarrow FOLLOW(B) \cup (FIRST(\beta) - \{\varepsilon\})$
        ak $\varepsilon \in FIRST(\beta)$ **potom**
            $FOLLOW(B) \leftarrow FOLLOW(B) \cup FOLLOW(A)$
        koniec ak
    koniec pre
pokiaľ sa zmenila niektorá z množín $FOLLOW(B)$, $B \in N$

Množina $FOLLOW(A)$ - príklad

Príklad: Nech je daná gramatika $G = (N, T, P, S)$, kde
$N = \{S, A, B, C\}$, $T = \{a, b, +, (,), \$\}$ a pravidlá:

$$S \to C\$$$

$$A \to b \mid \varepsilon$$

$$B \to +S \mid \varepsilon$$

$$C \to A(C) \mid aB$$

Množina $FOLLOW(A)$ - príklad (pokr.)

$FOLLOW$	S	A	B	C
Začiatočný symbol	ε			
1. iterácia: $S \to \underline{C}\$$				$\$$
$B \to +\underline{S}$				
$C \to \underline{A}(C)$		(		
$C \to A(\underline{C})$				)
$C \to a\underline{B}$			$\$,)$	
2. iterácia: $S \to \underline{C}\$$				
$B \to +\underline{S}$	$\$,)$			
...				

Zvyšok tabuľky už budú prázdne riadky, pričom po 3. iterácii sa už množiny nezmenia a algoritmus končí. **V tabuľke uvádzame len novo-zistené symboly.**

Zásobníkové automaty

- Pri **regulárnych jazykoch** sme spomínali výpočtové zariadenie, ktoré ich dokáže akceptovať - **konečné automaty.**
- Spomínali sme si, že **konečné automaty** dokážu akceptovať **práve tie isté jazyky**, ktoré dokážu generovať **regulárne gramatiky.**
- Pre **bezkontextové jazyky** uvažujeme tiež istý typ výpočtového zariadenia, ktoré ich dokáže akceptovať - tzv. **zásobníkové automaty**
- Zjednodušene povedané: zásobníkový automat je variantou konečného automatu, ktorý **navyše disponuje pamäťovým médiom organizovaným ako zásobník** (stack)

Zásobníkové automaty - definícia I

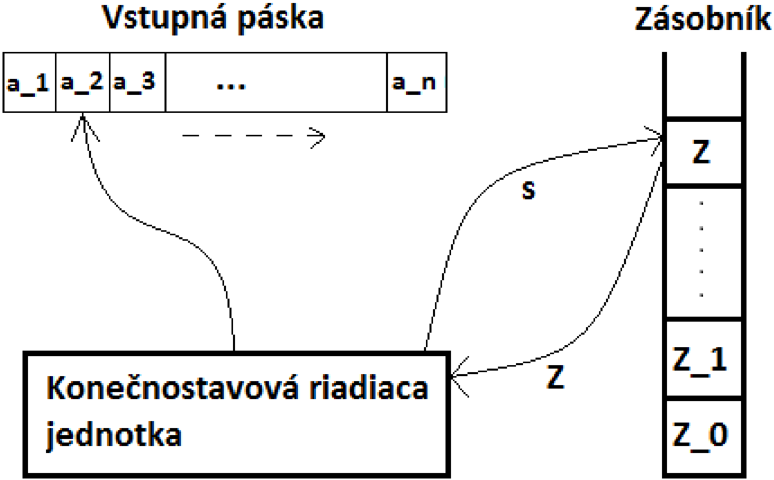
Nedeterministický zásobníkový automat (ZA) M je usporiadaná sedmica $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde:

- Q je konečná množina **stavov** automatu,
- Σ je konečná množina **vstupných symbolov** automatu,
- Γ je konečná množina **zásobníkových symbolov**,
- δ je **prechodová funkcia** ZA,

$$\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Q\times\Gamma^*}_{\text{kon}},$$

  pričom symbol $2^{Q\times\Gamma^*}_{\text{kon}}$ je množina všetkých **konečných** podmnožín množiny $Q \times \Gamma^*$,
- q_0 je **začiatočný stav** automatu,
- Z_0 je zásobníkový symbol, nachádzajúci sa na začiatku výpočtu na dne zásobníka automatu, $Z_0 \in \Gamma$,
- F je množina **akceptujúcich** stavov automatu, $F \subseteq Q$.

Zásobníkový automat



Zásobníkový automat

Zásobníkový automat sa skladá z:

1. Vstupnej pásky - je na nej napísané vstupné slovo, ktoré sa spracúva znak po znaku, zľava doprava.
2. Zásobníka - potenciálne nekonečné pamäťové úložisko; operácia s ním je ako so štandardným zásobníkom, t.j. prvky sa vkladajú na vrch zásobníka (push) a taktiež sa vyberajú z vrchu zásobníka (pop).
3. Konečnostavovej riadiacej jednotky - logika automatu, na základe aktuálneho stavu, vstupného symbolu a obsahu zásobníka rozhoduje, aký bude ďalší stav automatu a nový obsah zásobníka.

Zásobníkový automat

1. Konečnostavová jednotka sa vždy nachádza v nejakom stave, vidí aktuálny vstupný symbol na vstupnej páske a vrchný symbol zásobníka.
2. V každom kroku prečíta aktuálny vstupný symbol (alebo miesto neho „prečíta“ε) a posunie čítaciu hlavu na ďalší vstupný symbol (ak prečítal ε, tak hlavu neposúva). Zároveň z vrchu zásobníka odstráni vrchný symbol (prípadne neodstráni nič, t.j. odstráni ε).
3. Na základe prechodovej funkcie potom prejde do ďalšieho stavu a do zásobníka uloží reťazec zo symbolov zásobníkovej abecedy (prípadne ε, t.j. do zásobníka sa neuloží nič).

Zásobníkový automat - začiatok & koniec

1. Na začiatku výpočtu je zásobníkový automat v stave q_0, na vstupnej páske je napísané vstupné slovo, čítacia hlava je na prvom symbole vstupného slova a v zásobníku je len počiatočný zásobníkový symbol Z_0.
2. Automat vstupné slovo akceptuje, ak sa po jeho prečítaní dostane ZA do niektorého z akceptujúcich stavov (t.j. stavov z množiny F). V tomto prípade **nezáleží** na konečnom obsahu zásobníka.

Zásobníkový automat

Definícia

Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ *je zásobníkový automat. Potom:*

- *usporiadanú trojicu* $(q, w, s) \in Q \times \Sigma^* \times \Gamma^*$ *nazývame* ***konfigurácia zásobníkového automatu*** *M.*
- *konfiguráciu* (q_0, w, Z_0), *kde w je vstupný reťazec, nazývame* ***začiatočná konfigurácia*** *automatu M,*
- *konfiguráciu* (q, ε, s), *kde* $q \in F, s \in \Gamma^*$, *nazývame* ***akceptujúca (koncová) konfigurácia*** *automatu M.*

Zásobníkový automat

Definícia

Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ *je zásobníkový automat. Potom nad množinou konfigurácií* $Q \times \Sigma^* \times \Gamma^*$ *definujeme* ***reláciu prechodu (krok výpočtu)*** *zásobníkového automatu* $\vdash$ *takto: Nech* $p, q \in Q, w \in \Sigma^*, a \in \Sigma_\varepsilon, Z \in \Gamma_\varepsilon, s, t \in \Gamma^*$*. Potom* $(q, aw, Zs) \vdash (p, w, ts)$ *práve vtedy, keď* $(p, t) \in \delta(q, a, Z)$*.*

- Ak $a = \varepsilon$, automat neposúva čítaciu hlavu.
- Ak $Z = \varepsilon$, automat neodstráni žiaden symbol zo zásobníka.
- Ak $t = \varepsilon$, automat nepridá žiadne symboly do zásobníka.

Zásobníkový automat

Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat. Potom postupnosť konfigurácií $(q_0, w_0, s_0), (q_1, w_1, s_1), ..., (q_k, w_k, s_k)$ takých, že

$$(q_0, w_0, s_0) \vdash (q_1, w_1, s_1) \vdash ... \vdash (q_k, w_k, s_k),$$

$q_i \in Q, w_i \in \Sigma^*, s_i \in \Gamma^*, i = 0, 1, ..., k$ sa nazýva **výpočet zásobníkového automatu** M z (q_0, w_0, s_0) do (q_k, w_k, s_k). Výpočet budeme zapisovať pomocou mocniny relácie $\vdash$ v tvare $(q_0, w_0, s_0) \vdash^k (q_k, w_k, s_k)$.

Zásobníkový automat

Analogicky, ako v prípade konečných automatov:

- $(q, w, s) \vdash^{+} (p, x, t)$, ak existuje $k \geq 1$, že $(q, w, s) \vdash^{k} (p, x, t)$ (tranzitívny uzáver)
- $(q, w, s) \vdash^{*} (p, x, t)$, ak existuje $k \geq 0$, že $(q, w, s) \vdash^{k} (p, x, t)$ (reflexívny uzáver)

Zásobníkový automat

Definícia

Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat. Potom ***jazyk*** *$L(M)$* ***rozpoznávaný (akceptovaný)*** *zásobníkovým automatom M je množina:*

$$L(M) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, s), w \in \Sigma^*, q \in F, s \in \Gamma^*\}$$

Horeuvedený spôsob sa nazýva **akceptácia koncovým stavom.** Niekedy sa uvažuje aj o druhom spôsobe akceptácie slov - ZA akceptuje slovo, ak existuje výpočet $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon), q \in Q$, tzv. **akceptovanie prázdnym zásobníkom.** Dá sa ukázať, že oba spôsoby sú ekvivalentné.

Vizuálna reprezentácia ZA

- Tak, ako v prípade KA, aj v prípade ZA je ich možné vizualizovať pomocou prechodového diagramu.
- Od prechodových diagramov KA sa líšia iným ohodnotením hrán:
- Ak $(p, t) \in \delta(q, a, Z)$, potom prechodový diagram obsahuje hranu z vrcholu q do vrcholu p ohodnotenú $a, Z \to t$.

Zásobníkový automat - príklad

Príklad: ZA akceptujúci jazyk $L = \{a^n b^n \mid n \in \{1, 2, ...\}\}$.

Idea:

- Pre každý jeden prečítaný symbol a automat vloží do zásobníka jeden zásobníkový symbol a.
- Následne pre každý jeden prečítaný symbol b automat zo zásobníka jeden symbol a vyberie.
- V prípade, že slovo na vstupe spĺňalo tvar $a^n b^n$, tak v zásobníku po dočítaní posledného b ostane len počiatočný symbol Z_0 a automat prejde do akceptačného stavu q_f.
- V prípade, že slovo na vstupe nepatrilo do jazyka L, tak sa automat alebo zasekne v neakceptačnom stave, alebo sa dostane do akceptačného stavu, avšak neprečíta celý vstup.
- Upozorňujem, že $n > 0$, takže $\varepsilon \notin L$ a teda ZA nesmie akceptovať ε.

Zásobníkový automat - príklad (pokr.)

Formálne:

- $Q = \{q_0, q_1, q_f\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, Z_0\}$
- δ :
  - $\delta(q_0, a, Z_0) = \{(q_0, aZ_0)\}$
  - $\delta(q_0, a, a) = \{(q_0, aa)\}$
  - $\delta(q_0, b, a) = \{(q_1, \varepsilon)\}$
  - $\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$
  - $\delta(q_1, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$
- $F = \{q_f\}$.

Zásobníkový automat - príklad (pokr.)

Vstupné slovo: *aabb* (existuje akceptačná konfigurácia)

$$(q_0, aabb, Z_0) \vdash (q_0, abb, aZ_0) \vdash (q_0, bb, aaZ_0) \vdash (q_1, b, aZ_0) \vdash (q_1, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$$

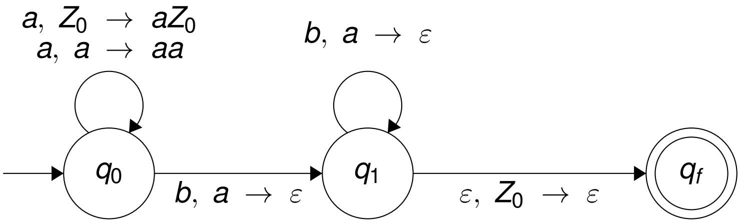
Vstupné slovo: *aabbb* (neexistuje akceptačná konfigurácia) - jeden z výpočtov...:

$$(q_0, aabbb, Z_0) \vdash (q_0, abbb, aZ_0) \vdash (q_0, bbb, aaZ_0) \vdash (q_1, bb, aZ_0) \vdash (q_1, b, Z_0) \vdash (q_f, b, \varepsilon)$$

Horeuvedený výpočet nie je akceptačný, pretože na vstupe zostala neprečítaná časť vstupného slova.

Zásobníkový automat - príklad (pokr.)

Grafická reprezentácia ZA by bola nasledovná:



Vzťah medzi zásobníkovými automatmi a bezkontextovými gramatikami

Veta

Nech $G = (N, T, P, S)$ je bezkontextová gramatika. Potom k nej existuje zásobníkový automat $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ taký, že $L(M) = L(G)$.

Veta

Nech $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat. Potom k nemu existuje bezkontextová gramatika $G = (N, T, P, S)$ taká, že $L(G) = L(M)$.

Zásobníkové automaty teda zohrávajú pri akceptácii bezkontextových jazykov tú istú úlohu, ako konečné automaty pri regulárnych jazykoch. Bezkontextový jazyk teda vieme popísať alebo bezkontextovou gramatikou, ktorá ho generuje, alebo zásobníkovým automatom, ktorý ho akceptuje.

Použitá literatúra

Aho, A., Lam, M., Sethi, R., Ullman, J.: *Compilers: Principles, techniques and tools.*

Dedera, Ľ: *Počítačové jazyky a ich spracovanie*.

Linz, P.: *An Introduction to Formal Languages and Automata*.