Formális nyelvek és szintaktikus elemzésük
– reguláris nyelvek –

Fülöp Zoltán

Szegedi Tudományegyetem
Számítástudomány Alapjai Tanszék

– Hogyan, milyen módszerrel adható meg programozási nyelvek szintaxisa?

A legelterjedtebb módszer a *generatív nyelvtannal* történő szintaxis megadás.

Adjuk meg az A, B és C változókból, a 0 és 1 konstansokból, a + és a $*$ műveleti jelekből, valamint a (és) zárójelekből felépíthető aritmetikai kifejezések szintaxisát!

Ilyenek például az A, 1, $A+1$, $A+B$, $A*(B+1)$ aritmetikai kifejezések.

Aritmetikai kifejezéseknek az A, B és C változó jelekből, a 0 és 1 konstans jelekből, a + és $*$ műveleti jelekből a (és a) csoportosító jelekből, a

$$
\begin{array}{lll}
< kif > & \to & < tag > \mid < kif > + < tag > \\
< tag > & \to & < fakt > \mid < tag > * < fakt > \\
< fakt > & \to & (< kif >) \mid < valt > \mid < konst > \\
< valt > & \to & A \mid B \mid C \\
< konst > & \to & 0 \mid 1
\end{array}
$$

szabályok alkalmazásával felépíthető jelsorozatokat (szavakat) nevezzük.

Ez egy *generatív nyelvtan*.

Levezetés:

$$
\begin{array}{ll}
 & \underline{< kif >} \\
\Rightarrow & \underline{< tag >} \\
\Rightarrow & < tag > * \underline{< fakt >} \\
\Rightarrow & < tag > *(\underline{< kif >}) \\
\Rightarrow & < tag > *(\underline{< kif >} + < tag >) \\
\Rightarrow & \underline{< tag >} * (< tag > + < tag >) \\
\Rightarrow & < fakt > *(\underline{< tag >} + < tag >) \\
\Rightarrow & < fakt > *(< fakt > + \underline{< tag >}) \\
\Rightarrow & < fakt > *(< fakt > + < fakt >)
\end{array}
$$

$$
\begin{aligned}
&\Rightarrow \underline{< fakt >} * (< fakt > + < fakt >)\\
&\Rightarrow < valt > *(\underline{< fakt >} + < fakt >)\\
&\Rightarrow < valt > *(< valt > + \underline{< fakt >})\\
&\Rightarrow \underline{< valt >} * (< valt > + < konst >)\\
&\Rightarrow A * (\underline{< valt >} + < konst >)\\
&\Rightarrow A * (B + \underline{< konst >})\\
&\Rightarrow A * (B + 1)
\end{aligned}
$$

Jelölés: $< kif > \Rightarrow^* A * (B + 1)$

Az $A, B, C, 0, 1, +, *, ($ és $)$ jelekből álló aritmetikai kifejezés szintaxisa:

egy jelsorozat (vagy szó) akkor és csak akkor aritmetikai kifejezés, ha a $< kif >$-ből a fenti szintaktikai szabályok alkalmazásával történő levezetéssel megkapható.

Röviden:

w szó aritmetikai kifejezés $\Leftrightarrow$ $< kif > \Rightarrow^* w$.

Egy egyszerű programozási nyelv, a PÉLDA szintaxisa:

$$
\begin{array}{lll}
< program > & \rightarrow & < ut.lista > .\\
< ut.lista > & \rightarrow & < ut > \mid < ut >; < ut.lista >\\
< ut > & \rightarrow & < ert.ado > \mid < ifut > \mid\\
 & & < whileut > \mid < blokk >\\
< ert.ado > & \rightarrow & < valt > := < kif >\\
< ifut > & \rightarrow & \textbf{if} < relacio > \textbf{then} < ut > \textbf{else} < ut >\\
< whileut > & \rightarrow & \textbf{while} < relacio > \textbf{do} < ut >\\
< blokk > & \rightarrow & \textbf{begin} < ut.lista > \textbf{end}\\
< relacio > & \rightarrow & < kif > < relaciojel > < kif >\\
< relaciojel > & \rightarrow & < \mid > \mid = \mid \neq
\end{array}
$$

$$
\begin{array}{lll}
< kif > & \rightarrow & < tag > \mid < kif > + < tag >\\
< tag > & \rightarrow & < fakt > \mid < tag > * < fakt >\\
< fakt > & \rightarrow & (< kif >) \mid < valt > \mid < konst >\\
< valt > & \rightarrow & A \mid B \mid C\\
< konst > & \rightarrow & 0 \mid 1
\end{array}
$$

Egy w jelsorozat akkor és csakis akkor szintaktikusan helyes PÉLDA nyelvű program, ha $< program >\Rightarrow^* w$.

Ilyen például a következő jelsorozat:

$A := 0;$
while $A < C$ **do**
  begin $A := A + 1;$
  $B := B * C;$
  end;
$C := C * B.$

Az elemzés alapkérdése:

– Amennyiben adott egy programozási nyelv szintaxisa és adott egy – ezen a nyelven írott – program, akkor hogyan tudjuk eldönteni azt, hogy az adott program engedelmeskedik-e a szintaxisnak, vagyis szintaktikusan helyes-e?

Röviden: igaz-e, hogy $< program >\Rightarrow^* w$?

Fogalmak, jelölések

Ábécé: szimbólumoknak egy tetszőleges véges, nemüres halmaza.
Példa: $\Sigma = \{a, b\}$

Σ *ábécé feletti szó*: egy $a_1 \dots a_k$ alakú sorozat, ahol $k \geq 0$ és $a_1, \dots, a_k \in \Sigma$.
Példa: $a, b, aba, aababba, \dots$

$k = 0$ eset: *üres szónak nevezzük*, jele λ.

Összes szavak halmaza:
$\Sigma^* = \{a_1 \dots a_k \,|\, k \geq 0,\ a_1, \dots, a_k \in \Sigma\}$
Példa: $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \dots\}$

$\Sigma^+ = \Sigma^* - \{\lambda\} = \{a, b, aa, ab, ba, bb, aaa, \dots\}$

Konkatenáció: az $u, v \in \Sigma^*$ szavak egymás után írásával kapott $uv \in \Sigma^*$ szó.
$w^n = \overbrace{ww \dots w}^{n}$, $w^0 = \lambda$.
Példa: ha $u = abb$, $v = ba$, akkor $uv = abbba$, $u\lambda = \lambda u = u$, $u^3 = abbabbabb$.

Ha $w = xy$, akkor x a w *prefixe*, y a w *suffixe*.

Példa: abb pefixei λ, a, ab, abb, suffixei abb, bb, b, λ

w *hosszát* $|w|$-vel jelöljük:

- $|w| = 0$ ha $w = \lambda$,
- $|w| = 1 + |v|$ ha $w = av$, valamely $a \in \Sigma$ és $v \in \Sigma^*$-ra.

Példa: $|\lambda| = 0$, $|a| = 1$, $|ab| = 2$, $|abb| = 3$.

$k \geq 0$ esetén $\Sigma^{*,k} = \{w \in \Sigma^* \mid |w| \leq k\}$.

Nyelv: Σ^* tetszőleges részhalmazát Σ feletti nyelvnek nevezzük.

Példa: $\Sigma = \{a, b\}$ esetén

$L_1 = \{aa, bab, abba\}$,
$L_2 = \{w \in \Sigma^* \mid |w| \text{ páros } \}$,
$L_3 = \{\lambda, ab, aabb, aaabbb, \ldots\} = \{a^n b^n \mid n \geq 0\}$

Σ feletti nyelvek.

Műveletek nyelvekkel: $L_1, L_2 \subseteq \Sigma^*$ nyelvek esetén $L_1 \cup L_2$, $L_1 \cap L_2$ a halmazelméleti műveletek, továbbá

- $\overline{L_1} = \Sigma^* - L_1$ az L_1 *komplementere*,
- $L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}$ az L_1 és L_1 *konkatenációja*,
- $L_1^* = \{\lambda\} \cup L_1 \cup L_1 L_1 \cup L_1 L_1 L_1 \cup \ldots$ az L_1 *iteráltja*.

Megjegyzés: $\lambda \in L^*$, minden L nyelv esetén, továbbá
$\emptyset^* = \{\lambda\}$.

Generatív nyelvtan: egy $G = (N, \Sigma, P, S)$ négyes, ahol:

- N egy ábécé, a *nemterminális ábécé*,
- Σ egy ábécé, a *terminális (befejező, végső)* ábécé, amire $N \cap \Sigma = \emptyset$,
- $S \in N$ a *kezdő szimbólum* (vagy *start szimbólum*),
- P pedig $\alpha \to \beta$ alakú ún. *átírási szabályok* véges halmaza, ahol $\alpha, \beta \in (N \cup \Sigma)^*$ és α-ban van legalább egy nemterminális betű.

(α a szabály bal oldala, β a jobb oldala.)

Példa: $G = (\{S, A\}, \{a, b\}$,
$\{S \to aAb, aA \to aaAb, A \to \lambda\}, S)$ egy nyelvtan.

Közvetlen levezetés (*deriváció*):

tetszőleges $\gamma, \delta \in (N \cup \Sigma)^*$ esetén $\gamma \Rightarrow_G \delta$, ha van olyan $\alpha \to \beta \in P$ szabály és vannak olyan $\alpha', \beta' \in (N \cup \Sigma)^*$ szavak, amelyekre fennállnak, hogy $\gamma = \alpha'\alpha\beta'$, $\delta = \alpha'\beta\beta'$.

$$\underbrace{\overset{\alpha'}{\vdash\!\!-\!\!-}\overset{\alpha}{\vdash\!\!-\!\!-}\overset{\beta'}{\vdash\!\!-\!\!-\!\!\dashv}}_{\gamma} \Rightarrow_G \underbrace{\overset{\alpha'}{\vdash\!\!-\!\!-}\overset{\beta}{\vdash\!\!-\!\!-}\overset{\beta'}{\vdash\!\!-\!\!-\!\!\dashv}}_{\delta}$$

Példa:

$b\underline{aA}Aa \Rightarrow_G b\underline{aaAb}Aa$ az $aA \to aaAb$ szabállyal

$baaAb\underline{A}a \Rightarrow_G baaAba$ az $A \to \lambda$ szabállyal

Levezetések:

$\alpha \Rightarrow_G \beta$: egy lépés

$\alpha \Rightarrow_G^n \beta$: $n \geq 0$ lépés $(\alpha \Rightarrow_G^0 \beta \Leftrightarrow \alpha = \beta)$

$\alpha \Rightarrow_G^+ \beta$: legalább egy lépés

$\alpha \Rightarrow_G^* \beta$: valamennyi (esetleg 0) lépés

Ha nem okoz félreértést, $\Rightarrow_G$ helyett $\Rightarrow$-t írunk.

Példa:

$$\begin{aligned} S &\Rightarrow aAb && \Rightarrow aaAbb \\ &\Rightarrow aaaAbbb && \Rightarrow aaabbb \end{aligned}$$

Tehát: $S \Rightarrow^* aaabbb$, $S \Rightarrow^+ aaabbb$, $S \Rightarrow^4 aaabbb$ mind teljesülnek.

Továbbá: $aAb \Rightarrow aaAbb$, $aAb \Rightarrow^2 aaaAbbb$, tehát $aAb \Rightarrow^* aaAbb$, $aAb \Rightarrow^* aaaAbbb$.

$G = (N, \Sigma, P, S)$ *nyelvtan által generált nyelv*:

$$L(G) = \{u \in \Sigma^* \mid S \Rightarrow_G^* u\}.$$

A $G = (\{S, A\}, \{a, b\},$
$\{S \to aAb, aA \to aaAb, A \to \lambda\}, S)$ példában

$$L(G) = \{a^n b^n \mid n \geq 1\}.$$

A $G = (N, \Sigma, P, S)$ és $G' = (N', \Sigma', P', S')$ nyelvtanok *ekvivalensek*, ha $L(G) = L(G')$.

Példa: $G_{ar} = (N, \Sigma, P, S)$, ahol

- $N = \{K, T, F\}$,
- $\Sigma = \{+, *, (,), a\}$,
- $S = K$,
- $P = \{$
  - $K \to K + T,\ K \to T$,
  - $T \to T * F,\ T \to F$,
  - $F \to (K),\ F \to a\}$.

Ekkor $L(G_{ar})$ az a-ból valamint a $(,), +$ és $*$ jelekből képezhető aritmetikai kifejezések halmaza.

Jelölések:

- Az a, b, c, d szimbólumok Σ elemeit,
- az A, B, C, D, S szimbólumok N elemeit,
- az U, V, W, X, Y, Z szimbólumok $N \cup \Sigma$ elemeit,
- az $\alpha, \beta, \gamma, \delta$ szimbólumok $(N \cup \Sigma)^*$ elemeit,
- és az u, v, w, x, y, z szimbólumok Σ^* elemeit

fogják jelölni.

Chomsky nyelvosztályok

A $G = (N, \Sigma, P, S)$ nyelvtan

- 0 típusú (vagy *kifejezés struktúrájú*), ha rá semmilyen korlátozás nincs.
- 1 típusú (vagy *környezetfüggő*), ha P-ben minden szabály $\alpha A \beta \to \alpha \delta \beta$ alakú, ahol $\delta \neq \lambda$. Kivétel, az $S \to \lambda$ szabály, ekkor azonban az S nem szerepelhet semelyik szabály jobb oldalán.
- 2 típusú (vagy *környezetfüggetlen*), ha P-ben minden szabály $A \to \alpha$ alakú.
- 3 típusú (vagy *reguláris*, vagy *jobblineáris*), ha P-ben minden szabály $A \to xB$ vagy $A \to x$ alakú.

Egy $L \subseteq \Sigma^*$ nyelvet i típusúnak mondunk, ahol $0 \leq i \leq 3$, ha van olyan i típusú $G = (N, \Sigma, P, S)$, nyelvtan, ami éppen L-t generálja, tehát amire $L = L(G)$.

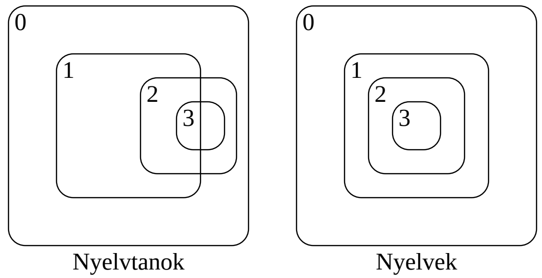
Az i típusú nyelvek osztályát $\mathcal{L}_i$-vel jelöljük.

A *Chomsky nyelvhierarchia*:

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

és erősebb alakja:

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

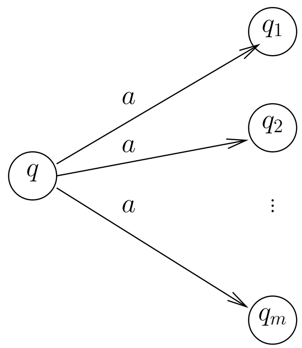


Véges automata

Az $M = (Q, \Sigma, \delta, q_0, F)$ rendszert *nemdeterminisztikus automatának* nevezzük, ahol:

- Q egy nem üres, véges halmaz, az *állapotok halmaza*,
- Σ egy ábécé, az *input ábécé*,
- $q_0 \in Q$ a *kezdő állapot*,
- $F \subseteq Q$ a *végállapotok halmaza*,
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ egy leképezés, az *átmenetfüggvény*.

$\delta(q, a) = \{q_1, \ldots, q_n\}$



M konfigurációinak halmaza: $C = Q \times \Sigma^*$. A $(q, a_1 \dots a_n)$ konfiguráció azt a jelenti, hogy M a q állapotban van és az $a_1 \dots a_n$ szót kapja inputként.

Átmeneti reláció:

$(q, w), (q', w') \in C$ esetén $(q, w) \vdash_M (q', w')$ ha $w = aw'$, valamely $a \in \Sigma$-ra és fennáll a $q' \in \delta(q, a)$ tartalmazás.

$(q, w) \vdash_M (q', w')$, egy lépés

$(q, w) \vdash_M^n (q', w')$, $n \geq 0$ lépés

$(q, w) \vdash_M^+ (q', w')$, legalább egy lépés

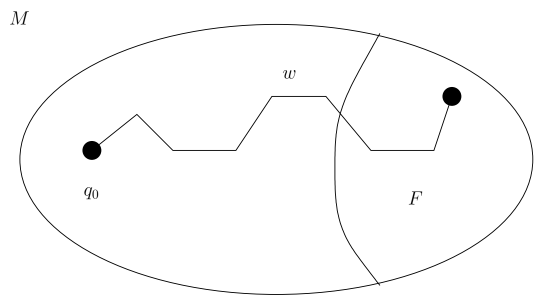
$(q, w) \vdash_M^* (q', w')$, valamennyi (esetleg 0) lépés

Az $M = (Q, \Sigma, \delta, q_0, F)$ automata által felismert nyelven az

$$L(M) = \{w \in \Sigma^* \mid (q_0, w) \vdash_M^* (q, \lambda) \text{ valamely } q \in F\text{-re}\}$$

nyelvet értjük.

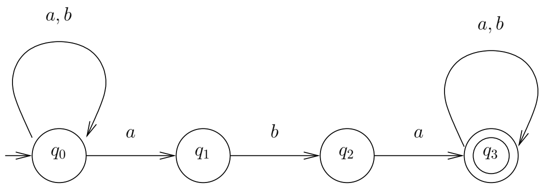
Szavakkal: q_0-ból a w hatására elérhető valamely $q \in F$ végállapot (ugyanakkor esetleg nem végállapotok is elérhetők).



$w \in L(M)$

Példa: $M = (Q, \Sigma, \delta, q_0, F)$ egy automata, ahol

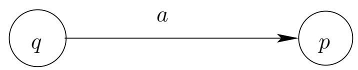
- $Q = \{q_0, q_1, q_2, q_3\}$,
- $\Sigma = \{a, b\}$,
- $F = \{q_3\}$,
- továbbá
  - $\delta(q_0, a) = \{q_0, q_1\}$, $\delta(q_0, b) = \{q_0\}$,
  - $\delta(q_1, a) = \emptyset$, $\delta(q_1, b) = \{q_2\}$,
  - $\delta(q_2, a) = \{q_3\}$, $\delta(q_2, b) = \emptyset$,
  - $\delta(q_3, a) = \delta(q_3, b) = \{q_3\}$.



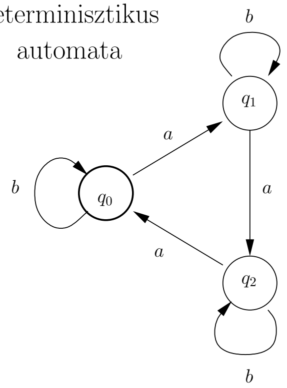
$L(M) = \{uabav \mid u, v \in \Sigma^*\}$, tehát M pontosan azon Σ^*-beli szavakat ismeri fel amelyekben előfordul az *aba* rész-szó.

M nemdeterminisztikus!

• Egy $M = (Q, \Sigma, \delta, q_0, F)$ automata *determinisztikus*, ha teljesül, hogy minden $q \in Q$ és $a \in \Sigma$ esetén a $\delta(q, a)$ halmaz legfeljebb egyelemű.



determinisztikus automata



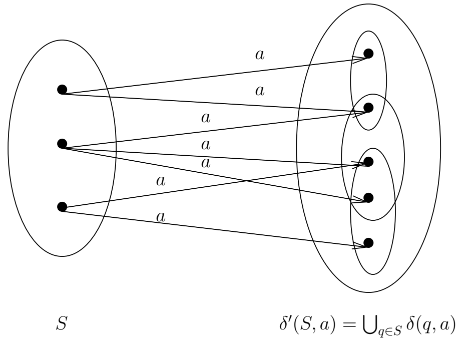
Tétel A determinisztikus automatákkal felismerhető nyelvek osztálya megegyezik a nemdeterminisztikus automatákkal felismerhető nyelvek osztályával.

Bizonyítás. a) Ha egy nyelv felismerhető determinisztikus automatával akkor felismerhető nemdeterminisztikus automatával is.

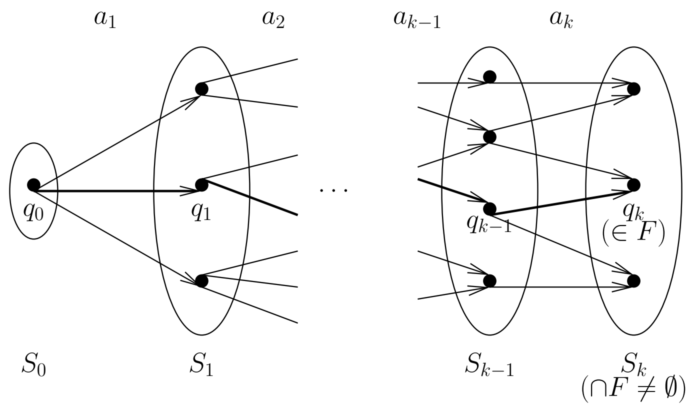
b) Fordítva: legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy nemdeterminisztikus automata. Megadunk egy $M' = (Q', \Sigma, \delta', q'_0, F')$ determinisztikus automatát, amelyre $L(M') = L(M)$.

$M' = (Q', \Sigma, \delta', q'_0, F')$, ahol

- $Q' = \mathcal{P}(Q) (= \{S \mid S \subseteq Q\})$,
- $q'_0 = \{q_0\}$,
- $F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$,
- $\delta' : Q' \times \Sigma \to Q'$ az a leképezés amelyre tetszőleges $S \in Q'$ és $a \in \Sigma$ esetén $\delta'(S,a) = \bigcup_{q \in S} \delta(q,a)$.



Az $L(M') = L(M)$ bizonyítása:
vegyünk egy $w = a_1 \dots a_k$ szót.



Az $M = (Q, \Sigma, \delta, q_0, F)$ automata *teljesen definiált*, ha teljesül, hogy minden $q \in Q$ és $a \in \Sigma$ esetén $\delta(q, a)$ legalább egy elemű.

A teljesen definiált automaták minden szót végig tudnak olvasni, mivel minden (q, ax) alakú konfigurációjukra van legalább egy rákövetkező konfiguráció.

Tétel. Tetszőleges $M = (Q, \Sigma, \delta, q_0, F)$ automatához megadható olyan $M' = (Q', \Sigma, \delta', q_0, F)$ teljesen definiált automata, melyre $L(M) = L(M')$.

Bizonyítás. Legyen $Q' = Q \cup \{q_c\}$, ahol $q_c \notin Q$, vagyis egy új állapot (a "csapda" állapot).
Továbbá, minden $q \in Q$ és $a \in \Sigma$ esetén, legyen

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{ha} \quad \delta(q, a) \neq \emptyset \\ \{q_c\} & \text{ha} \quad \delta(q, a) = \emptyset. \end{cases}$$

Végül, minden $a \in \Sigma$-ra, legyen $\delta'(q_c, a) = \{q_c\}$.

Reguláris kifejezések és nyelvek

Egy Σ ábécé feletti *reguláris kifejezések* halmaza a $(\Sigma \cup \{\emptyset, (,), +, *\})^*$ halmaz legszűkebb olyan U részhalmaza, amelyre az alábbi feltételek teljesülnek:

(i) az $\emptyset$ (áthúzott nulla) szimbólum eleme U-nak;

(ii) Minden $a \in \Sigma$-ra az a szimbólum eleme U-nak;

(iii) Ha $R_1, R_2 \in U$, akkor $(R_1) + (R_2)$, $(R_1)(R_2)$, és $(R_1)^*$ is elemei U-nak.

Példa: Legyen $\Sigma = \{a, b\}$. Akkor például a $(\emptyset)^*, ((a) + (b))^*$ és $((a)(b))((a)^*)$, Σ feletti reguláris kifejezések.

Az R reguláris kifejezés *által meghatározott* (*reprezentált*) $|R|$ *nyelvet* a következőképpen definiáljuk:

(i) Ha $R = \emptyset$ (áthúzott nulla), akkor $|R| = \emptyset$ (üres nyelv);

(ii) Ha $R = a$ (mint szimbólum), akkor $|R| = \{a\}$ (mint nyelv);

(iii) a) Ha $R = (R_1) + (R_2)$, akkor $|R| = |R_1| \cup |R_2|$;

(iii) b) Ha $R = (R_1)(R_2)$, akkor $|R| = |R_1||R_2|$;

(iii) c) Ha $R = (R_1)^*$, akkor $|R| = |R_1|^*$.

Egy $L \subseteq \Sigma^*$ nyelv reguláris, ha van olyan Σ feletti R reguláris kifejezés, melyre $|R| = L$.

Példa

- $|(\emptyset)^*| = |\emptyset|^* = \emptyset^* = \{\lambda\}$;
- $|((a) + (b))^*| = |(a) + (b)|^* = |\{a\} \cup \{b\}|^* = \{a, b\}^*$;
- $|((a)(b))((a)^*)| = |((a)(b))||(a)^*| = |(a)||(b)||(a)|^* = \{a\}\{b\}(\{a\}^*) =$
  $\{ab\}\{\lambda, a, aa, \ldots\} = \{ab, aba, abaa, \ldots\}$.

Tehát a $\{\lambda\}$, $\{a, b\}^*$ és $\{ab, aba, abaa, \ldots\}$ nyelvek reguláris nyelvek.

Megállapodás: a prioritási sorrend legyen *, konkatenáció, $+$.

Továbbá: $\cup$ és konkatenáció asszociatív

Akkor a reguláris kifejezések zárójelezése egyszerűsíthető:
$(\emptyset)^*$ helyett $\emptyset^*$
$((a) + (b))^*$ helyett $(a + b)^*$
$((a)(b))((a)^*)$ helyett aba^*
vagy
$(a + b)^* aba(a + b)^*$, $a(a + b)^* b(a + b)^* a$, stb.

Példák reguláris nyelvekre

Tétel. Minden véges nyelv reguláris.

Bizonyítás. Legyen $L = \{x_1, \ldots, x_n\}$, $n \geq 1$. Akkor $L = |R|$, ahol

$$R = R_1 + \ldots + R_n$$

és

$$R_i = \begin{cases} a_{i_1} \ldots a_{i_{n_i}} & \text{ha } x_i = a_{i_1} \ldots a_{i_{n_i}} \\ \emptyset^* & \text{ha } x_i = \lambda. \end{cases}$$

További példák: $\Sigma = \{a, b\}$

a) $L = \{uabav \mid u, v \in \Sigma^*\}$ ($w \in L \Leftrightarrow w$-ben előfordul az aba rész-szó)
reguláris, mert $L = |(a + b)^* aba(a + b)^*|$

b) $L = \{aubva \mid u, v \in \Sigma^*\}$ ($w \in L \Leftrightarrow w$ a-val kezdődik és végződik és van benne legalább egy b)
reguláris, mert $L = |a(a + b)^* b(a + b)^* a|$.

Tétel. A reguláris nyelvek zártak az $\cup$, a konkatenáció és a * műveletekre.
(Ha L_1, L_2 reguláris, akkor $L_1 \cup L_2$, L_1L_2 és L_1^* is regulárisak.

Bizonyítás. Triviális, ezért csak az $\cup$-t bizonyítjuk.
Ha L_1, L_2 regulárisak, akkor vannak olyan R_1, R_2 reguláris kifejezések, melyekre $L_1 = |R_1|$ és $L_2 = |R_2|$.
Legyen $R = R_1 + R_2$.
Ekkor $|R| = |R_1 + R_2| = |R_1| \cup |R_2| = L_1 \cup L_2$, tehát $L_1 \cup L_2$ is reguláris.

Tétel. Tetszőleges Σ ábécé esetén

(1) a Σ feletti reguláris nyelvek osztálya,

(2) a Σ feletti 3-típusú nyelvek osztálya,

(3) a Σ feletti automatával felismerhető nyelvek osztálya,

egymással megegyeznek.

Bizonyítás.

1. Lemma: (1) $\subseteq$ (2)
2. Lemma: (2) $\subseteq$ (3)
3. Lemma: (3) $\subseteq$ (1)

Akkor (1) = (2) = (3). □

1. Lemma. (1) $\subseteq$ (2): Tetszőleges, Σ feletti L reguláris nyelv generálható 3-típusú nyelvtannal.

Bizonyítás. Az L-et reprezentáló R reguláris kifejezés struktúrája szerinti indukcióval.

Az indukció alapja.

(i) $R = \emptyset$ Ekkor $L = |R| = \emptyset$, mely generálható a $G = (\{S\}, \Sigma, \emptyset, S)$ 3-típusú nyelvtannal.

(ii) $R = a,\ a \in \Sigma$ Ekkor $L = |R| = \{a\}$, mely generálható a $G = (\{S\}, \Sigma, \{S \to a\}, S)$ 3-típusú nyelvtannal.

Indukciós lépés.

(iii) a) $R = (R_1) + (R_2)$
Ekkor $L = |R| = L_1 \cup L_2$, ahol $L_1 = |R_1|$ és $L_2 = |R_2|$.

Indukciós feltevés: L_i generálható a $G_i = (N_i, \Sigma, P_i, S_i)$ 3-típusú nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

Akkor L generálható a

$G = (N_1 \cup N_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S)$

3-típusú nyelvtannal, ahol S egy új szimbólum.

$$S \Rightarrow_G^* w$$

akkor és csak akkor, ha

$$S_1 \Rightarrow_{G_1}^* w \text{ vagy } S_2 \Rightarrow_{G_2}^* w.$$

Indukciós lépés.

(iii) b) $R = (R_1)(R_2)$
Ekkor $L = |R| = L_1L_2$, ahol $L_1 = |R_1|$ és $L_2 = |R_2|$.

Indukciós feltevés: L_i generálható a $G_i = (N_i, \Sigma, P_i, S_i)$ 3-típusú nyelvtannal, $i = 1, 2$. ($N_1 \cap N_2 = \emptyset$.)

Akkor L generálható a $G = (N_1 \cup N_2, \Sigma, P, S_1)$ 3-típusú nyelvtannal, ahol P a legszűkebb olyan szabályhalmaz amire teljesülnek a következő feltételek:

- Ha $A \to xB \in P_1$, akkor $A \to xB \in P$,
- Ha $A \to x \in P_1$, akkor $A \to xS_2 \in P$,
- P_2 minden eleme P-nek is eleme.

$$S_1 \Rightarrow_{G_1}^* w_1 \text{ és } S_2 \Rightarrow_{G_2}^* w_2$$

akkor és csak akkor, ha

$$S_1 \Rightarrow_G^* w_1S_2 \Rightarrow_G^* w_1w_2.$$

Indukciós lépés.

(iii) c) $R = (R_1)^*$
Ekkor $L = |R| = L_1^*$, ahol $L_1 = |R_1|$.

Indukciós feltevés: L_1 generálható a $G_1 = (N_1, \Sigma, P_1, S_1)$ 3-típusú nyelvtannal.

Akkor L generálható a $G = (N_1 \cup \{S\}, \Sigma, P, S)$ 3-típusú nyelvtannal, ahol S egy új szimbólum, P pedig a legszűkebb olyan szabályhalmaz amire teljesülnek a következő feltételek:

- $S \to S_1, S \to \lambda \in P$,
- Ha $A \to xB \in P_1$, akkor $A \to xB \in P$,
- Ha $A \to x \in P_1$, akkor $A \to xS \in P$.

$S \Rightarrow_G \lambda$

$S \Rightarrow_G S_1 \Rightarrow_G^* w_1 S \Rightarrow_G w_1 \ (\in L_1)$

$w_1 S \Rightarrow_G w_1 S_1 \Rightarrow_G^* w_1 w_2 S \Rightarrow_G w_1 w_2 (\in L_1 L_1)$

$w_1 w_2 S \Rightarrow_G w_1 w_2 S_1$ ...

Tétel. Tetszőleges Σ ábécé esetén

(1) a Σ feletti reguláris nyelvek osztálya,
(2) a Σ feletti 3-típusú nyelvek osztálya,
(3) a Σ feletti automatával felismerhető nyelvek osztálya,

egymással megegyeznek.

Bizonyítás.

1. Lemma: (1) $\subseteq$ (2) $\surd$
2. Lemma: (2) $\subseteq$ (3)
3. Lemma: (3) $\subseteq$ (1)

Definíció. Egy nyelvtanban az $A \to B$ alakú szabályokat láncszabályoknak nevezzük.

Tétel. Minden $G = (N, \Sigma, P, S)$ 3-típusú nyelvtanhoz van vele ekvivalens $G' = (N, \Sigma, P', S)$ láncszabálymentes 3-típusú nyelvtan.

Bizonyítás. Legyen $G = (N, \Sigma, P, S)$ egy 3-típusú nyelvtan.
a) Minden $A \in N$-re számoljuk ki az

$$N_A = \{B \in N \mid A \Rightarrow^* B\}$$

halmazt.

b) Konstruáljuk meg $G' = (N, \Sigma, P', S)$-t:

$P' = \emptyset$;
Minden $A \in N$-re,
minden $B \in N_A$-ra, $/{*}N_A = \{B \in N \mid A \Rightarrow^* B\}{*}/$
minden $B \to \alpha \in P$ szabály esetén:
Ha $B \to \alpha$ nem láncszabály akkor
vegyük fel az $A \to \alpha$ szabályt P'-be;

Ekkor $L(G) = L(G')$ lesz.

Példa láncszabálymentesítésre.

G: $S \to A \,|\, ab$, $A \to B \,|\, bA$, $B \to bB \,|\, C \,|\, a$, $C \to bb$

$N_S = \{S, A, B, C\}$, $N_A = \{A, B, C\}$, $N_B = \{B, C\}$ és $N_C = \{C\}$.

G':

- $S \to ab \,|\, bA \,|\, bB \,|\, a \,|\, bb$
- $A \to bA \,|\, bB \,|\, a \,|\, bb$
- $B \to bB \,|\, bb \,|\, a$
- $C \to bb$ $\qquad L(G) = L(G')$.

2. Lemma. (2) $\subseteq$ (3): Minden 3-típusú nyelv felismerhető automatával.

Bizonyítás. Legyen L egy 3-típusú nyelv és tegyük fel, hogy $L = L(G)$, ahol G láncszabálymentes 3-típusú nyelvtan.

2.1. Lemma. Minden $G = (N, \Sigma, P, S)$ 3-típusú láncszabálymentes nyelvtanhoz megadható vele ekvivalens $G' = (N', \Sigma, P', S)$ 3-típusú nyelvtan, úgy hogy P'-ben minden szabály $A \to aB$ vagy $A \to \lambda$ alakú, ahol $A, B \in N$ és $a \in \Sigma$.

2.2. Lemma. Minden olyan $G = (N, \Sigma, P, S)$ 3-típusú nyelvtanhoz melynek csak $A \to aB$ vagy $A \to \lambda$ alakú szabályai vannak megadható olyan
$M = (Q, \Sigma, \delta, q_0, F)$ automata, amelyre $L(M) = L(G)$.

2.1. Lemma. Minden $G = (N, \Sigma, P, S)$ 3-típusú láncszabálymentes nyelvtanhoz megadható vele ekvivalens $G' = (N', \Sigma, P', S)$ 3-típusú nyelvtan, úgy hogy P'-ben minden szabály $A \to aB$ vagy $A \to \lambda$ alakú, ahol $A, B \in N$ és $a \in \Sigma$.

Bizonyítás. Konstruáljuk meg P'-t a következőképpen:

(i) Minden $A \to aB$ vagy $A \to \lambda$ alakú P-beli szabályt vegyünk fel P'-be.

(ii) Minden $A \to a_1 \dots a_n B$, P-beli szabály esetén (ahol $n > 1, a_1, \dots, a_n \in \Sigma$) vegyük fel P'-be az

$$A \to a_1 A_1,\ A_1 \to a_2 A_2, \dots, A_{n-1} \to a_n B$$

szabályokat, ahol $A_1, \dots, A_{n-1}$ új nemterminális szimbólumok.

(iii) Minden $A \to a_1 \dots a_n$, P-beli szabály esetén (ahol $n \geq 1, a_1, \dots, a_n \in \Sigma$) vegyük fel P'-be az

$$A \to a_1 A_1,\ A_1 \to a_2 A_2, \dots, A_{n-1} \to a_n A_n,\ A_n \to \lambda$$

szabályokat, ahol $A_1, \dots, A_n$ új nemterminálisok.

Legyen $N' = N \cup \{$ új nemterminálisok $\}$.

Minden $A \in N$-re és $w \in \Sigma^*$-ra

$$A \Rightarrow_G^* w \text{ akkor és csak akkor, ha } A \Rightarrow_{G'}^* w.$$

Ugyanis

$A \to a_1 \dots a_n B \in P$ akkor és csak akkor, ha

$A \Rightarrow_{G'} a_1 A_1 \Rightarrow_{G'} \dots \Rightarrow_{G'} a_1 \dots a_{n-1} A_{n-1} \Rightarrow_{G'} a_1 \dots a_n B$

és

$A \to a_1 \dots a_n \in P$ akkor és csak akkor, ha

$A \Rightarrow_{G'} a_1 A_1 \Rightarrow_{G'} \dots \Rightarrow_{G'} a_1 \dots a_n A_n \Rightarrow_{G'} a_1 \dots a_n$.

Az $A = S$ választással kapjuk, hogy $L(G) = L(G')$.

Példa a szabályok szétdarabolására

$$\begin{array}{llll} G: & S & \to & abA \mid bB \\ & A & \to & bbB \mid \lambda \\ & B & \to & ab \quad \mid bA \end{array}$$

$$\begin{array}{lllllllllll} G': & S & \to & aA_1, & A_1 & \to & bA, & S & \to & bB \\ & A & \to & bA_2, & A_2 & \to & bB, & A & \to & \lambda \\ & B & \to & aA_3, & A_3 & \to & bA_4, & A_4 & \to & \lambda \\ & B & \to & bA & & & & & & \end{array}$$

2.2. Lemma. Minden olyan $G = (N, \Sigma, P, S)$ 3-típusú nyelvtanhoz melynek csak $A \to aB$ vagy $A \to \lambda$ alakú szabályai vannak megadható olyan
$M = (Q, \Sigma, \delta, q_0, F)$ automata, amelyre $L(M) = L(G)$.

Bizonyítás. Konstruáljuk meg M-et a következőképpen:

- $Q = N$,
- $q_0 = S$,
- $F = \{B \in N \mid B \to \lambda \in P\}$,
- minden $A \in N$ és $a \in \Sigma$ esetén legyen

$$\delta(A, a) = \{B \in N \mid A \to aB \in P\}.$$

<u>G-ben</u>:		<u>M-ben</u>:
$A \to aB \in P$	$\Leftrightarrow$	$A \xrightarrow{a} B$
$B \to \lambda \in P$		$B \in F$

Ekkor minden $n \geq 1$, $A, B \in N$ és $w \in \Sigma^*$ esetén

$$A \Rightarrow_G^n wB \text{ akkor és csak akkor ha } (A, w) \vdash_M^n (B, \lambda).$$

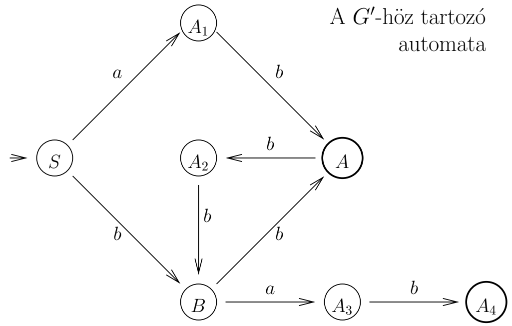
$$A \Rightarrow_G a_1 A_1 \Rightarrow_G \ldots \Rightarrow_G a_1 \ldots a_{n-1} A_{n-1} \Rightarrow_G a_1 \ldots a_n B$$

akkor és csak akkor, ha

$$(A, a_1 \ldots a_n) \vdash_M (A_1, a_2 \ldots a_n) \vdash_M \ldots \vdash_M (B, \lambda).$$

Az $A = S$, $B \in F$ választással adódik, hogy $L(M) = L(G)$.

A G'-höz tartozó automata



Tétel. Tetszőleges Σ ábécé esetén

(1) a Σ feletti reguláris nyelvek osztálya,

(2) a Σ feletti 3-típusú nyelvek osztálya,

(3) a Σ feletti automatával felismerhető nyelvek osztálya,

egymással megegyeznek.

Bizonyítás.

1. Lemma: (1) $\subseteq$ (2) $\checkmark$
2. Lemma: (2) $\subseteq$ (3) $\checkmark$
3. Lemma: (3) $\subseteq$ (1)

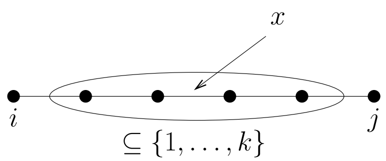
Lemma 3. (3) $\subseteq$ (1): Minden, automatával felismerhető nyelv reguláris. (S. C. Kleene tétele, 1956.)

Bizonyítás. Legyen $L = L(M)$, ahol $M = (Q, \Sigma, \delta, q_0, F)$ determinisztikus automata. Megmutatjuk, hogy L reguláris.

Tételezzük fel, hogy $Q = \{1, \ldots, n\}$ és $q_0 = 1$.

Minden $0 \leq k \leq n$ és $1 \leq i, j \leq n$ esetén definiáljuk a $L_{i,j}^{(k)} \subseteq \Sigma^*$ nyelvet a következőképpen:

$$x \in L_{i,j}^{(k)} \;\Leftrightarrow\; \begin{array}{l} (i,x) \vdash_M^* (j,\lambda) \text{ és} \\ \text{minden } (i,x) \vdash_M^+ (i',x') \vdash_M^+ (j,\lambda) \text{ esetén} \\ i' \in \{1,\ldots,k\}. \end{array}$$



Mivel

$$L(M) = \bigcup_{j \in F} L_{1,j}^{(n)}$$

és a reguláris nyelvek zártak az egyesítésre, elegendő megmutatni, hogy minden $j \in F$-re,

az $L_{1,j}^{(n)}$ nyelv reguláris.

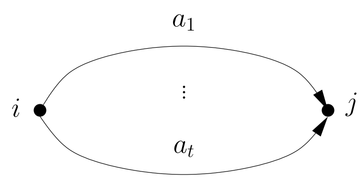
Többet bizonyítunk: minden $0 \le k \le n$-ra és $1 \le i, j \le n$-re,

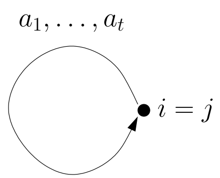
$L_{i,j}^{(k)}$ reguláris.

$L_{i,j}^{(k)}$ reguláris: k szerinti indukció.

$k = 0$:

$$L_{i,j}^{(0)} = \begin{cases} \{a \in \Sigma \mid \delta(i,a) = j\}, & \text{ha } i \ne j \\ \{a \in \Sigma \mid \delta(i,a) = j\} \cup \{\lambda\}, & \text{ha } i = j. \end{cases}$$





Mindkét esetben $L_{i,j}^{(0)}$ véges, tehát reguláris.

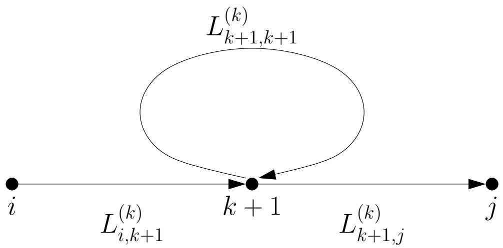
$k \Rightarrow k+1$:

$(\forall i,j) L_{i,j}^{(k)}$ reguláris $\quad\Rightarrow\quad (\forall i,j) L_{i,j}^{(k+1)}$ reguláris

Először azt bizonyítjuk be, hogy

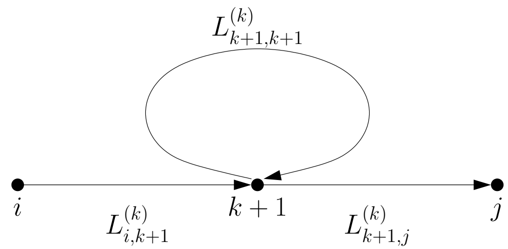
$$L_{i,j}^{(k+1)} = L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)} (L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)}.$$

a) $L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)}(L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)} \subseteq L_{i,j}^{(k+1)}$: nyilvánvaló



b) $L_{i,j}^{(k+1)} \subseteq L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)}(L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)}$:

Ha $x \in L_{i,j}^{(k+1)}$, akkor vagy $x \in L_{i,j}^{(k)}$ vagy $x = x_0\, x_1 \ldots x_t\, x_{t+1}$, ahol $t \geq 1$ és $x_0 \in L_{i,k+1}^{(k)}$, $x_1, \ldots, x_t \in L_{k+1,k+1}^{(k)}$ és $x_{t+1} \in L_{k+1,j}^{(k)}$.



$\underline{k \Rightarrow k+1}$: $(\forall i,j) L_{i,j}^{(k)}$ reguláris $\Rightarrow$ $(\forall i,j) L_{i,j}^{(k+1)}$ reguláris

Megmutattuk, hogy

$$L_{i,j}^{(k+1)} = L_{i,j}^{(k)} \cup L_{i,k+1}^{(k)}(L_{k+1,k+1}^{(k)})^* L_{k+1,j}^{(k)}.$$

1) Indukciós feltevés: $L_{i,j}^{(k)}, L_{i,k+1}^{(k)}, L_{k+1,k+1}^{(k)}$ és $L_{k+1,j}^{(k)}$ regulárisak.

2) A reguláris nyelvek zártak az $\cup$, konkatenáció és * műveletekre.

1) + 2) $\Rightarrow$ $L_{i,j}^{(k+1)}$ is reguláris.

Tehát $L(M) = \bigcup_{j \in F} L_{1,j}^{(n)}$ is reguláris.

Tétel. Tetszőleges Σ ábécé esetén

(1) a Σ feletti reguláris nyelvek osztálya,
(2) a Σ feletti 3-típusú nyelvek osztálya,
(3) a Σ feletti automatával felismerhető nyelvek osztálya,

egymással megegyeznek.

Bizonyítás.

1. Lemma: $(1) \subseteq (2)$ $\surd$
2. Lemma: $(2) \subseteq (3)$ $\surd$
3. Lemma: $(3) \subseteq (1)$ $\surd$

Tehát $(1) = (2) = (3)$. $\square$

Lemma. (Pumpáló lemma reguláris nyelvekre.) Minden $L \subseteq \Sigma^*$ reguláris nyelv esetén megadható olyan (L-től függő) $k > 0$ egész szám, hogy minden $w \in L$-re, ha $|w| \geq k$, akkor vannak olyan $w_1, w_2, w_3 \in \Sigma^*$ szavak, melyekre teljesülnek az alábbi feltételek:

1) $w = w_1 w_2 w_3$,

2) $0 < |w_2| \leq k$,

3) minden $n \geq 0$-ra, $w_1 w_2^n w_3 \in L$.

(Szükséges feltétele annak, hogy egy nyelv reguláris legyen.)

Bizonyítás. Mivel L reguláris, van olyan $M = (Q, \Sigma, \delta, q_0, F)$ determinisztikus automata, melyre $L = L(M)$.

Legyen $k = ||Q||$ (Q elemeinek a száma), ez a k jó lesz.

Vegyünk egy $w \in L$ szót úgy, hogy $|w| \geq k$. Akkor $w = a_1 a_2 \dots a_{k'}$, ahol $k' \geq k$ és vannak olyan $q_1, q_2, \dots, q_{k'} \in Q$ állapotok, melyekre

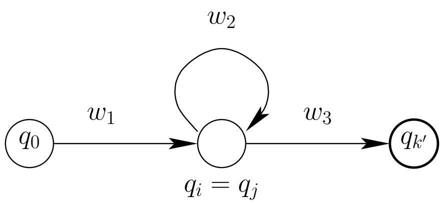
$$\begin{aligned} (q_0, a_1 \dots a_{k'}) &\vdash (q_1, a_2 \dots a_{k'}) \\ &\vdash (q_2, a_3 \dots a_{k'}) \\ &\dots \\ &\vdash (q_{k'-1}, a_{k'}) \\ &\vdash (q_{k'}, \lambda), \end{aligned}$$

továbbá $q_{k'} \in F$.

Mivel $k' \geq k = ||Q||$, van olyan $0 \leq i, j \leq k'$ számpár, melyre $i < j$ és $q_i = q_j$. Továbbá i és j választhatók úgy, hogy a $q_{i+1}, \dots, q_j$ sorozatban már nincsenek megegyező elemek.

Legyenek $w_1 = a_1 \dots a_i$, $w_2 = a_{i+1} \dots a_j$ és $w_3 = a_{j+1} \dots a_{k'}$. (Ha $i = 0$ akkor $w_1 = \lambda$, ha $j = k'$ akkor $w_3 = \lambda$)

Minden $n \geq 0$-ra $w_1 w_2^n w_3 \in L$:



Lemma. Az $L = \{a^n b^n \mid n \geq 0\}$ nyelv nem reguláris.

Bizonyítás. Tegyük fel, hogy igen (indirekt bizonyítás).

Legyen k pumpáló lemma szerint az L-hez tarozó szám, és vegyük a $w = a^k b^k$ szót ($|w| = 2k \geq k$).

Akkor $a^k b^k = w_1 w_2 w_3$ és minden $n \geq 0$-ra $w_1 w_2^n w_3 \in L$.

a) w_2 csak a betűkből áll. Ekkor a $w_1 w_2 w_2 w_3$ szóban az a-k száma nagyobb mint a b-k száma, tehát $w_1 w_2 w_2 w_3 \notin L$, $\lightning$.

b) w_2 tartalmaz a betűket és b betűket is. Ekkor a $w_1 w_2 w_2 w_3$ szóban a b betű megelőz a betűt, tehát $w_1 w_2 w_2 w_3 \notin L$, $\lightning$.

c) w_2 csak b betűkből áll. $\lightning$, ugyanúgy mint az a) esetben.

Mindhárom esetben $\lightning$, tehát az $L = \{a^n b^n \mid n \geq 0\}$ nyelv nem reguláris.

Tétel. $\mathcal{L}_3 \subset \mathcal{L}_2$.

Bizonyítás.

a) $\mathcal{L}_3 \subseteq \mathcal{L}_2$, mivel minden 3 típusú nyelvtan egyúttal 2 típusú is.

b) $L = \{a^n b^n \,|\, n \geq 0\} \in \mathcal{L}_2$, mivel az $S \to aSb \,|\, \lambda$ szabályokból álló környezetfüggetlen nyelvtan éppen L-et generálja.

Ugyanakkor $L \notin \mathcal{L}_3$.

Reguláris nyelvek zártsági tulajdonságai

Reguláris műveletek: $\cup$, konkatenáció, *.

Tétel. A reguláris nyelvek osztálya zárt a reguláris műveletekre.

Boole műveletek: $\cup$, $\cap$, komplementer.

Tétel. A reguláris nyelvek osztálya zárt a Boole műveletekre.

Bizonyítás. Az automaták direkt szorzata konstrukcióval.

Automaták direkt szorzata

Az $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ és $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ teljesen definiált és determinisztikus automaták direkt szorzata az $M = (Q_1 \times Q_2, \Sigma, \delta, [q_1, q_2], F)$ automata, ahol minden $p_1 \in Q_1$ és $p_2 \in Q_2$ állapot és $a \in \Sigma$ input szimbólum esetén

$$\delta([p_1, p_2], a) = [\delta_1(p_1, a), \delta_2(p_2, a)],$$

továbbá $F \subseteq Q_1 \times Q_2$.

Ekkor minden $p_1, r_1 \in Q_1$, $p_2, r_2 \in Q_2$ és $x \in \Sigma^*$ esetében

$$([p_1, p_2], x) \vdash_M^* ([r_1, r_2], \lambda)$$
$$\Leftrightarrow$$
$$(p_1, x) \vdash_{M_1}^* (r_1, \lambda) \text{ és } (p_2, x) \vdash_{M_2}^* (r_2, \lambda).$$

a) Ha $F = F_1 \times F_2$, akkor $L(M) = L(M_1) \cap L(M_2)$. Tehát ha L_1 és L_2 felismerhetők, akkor $L_1 \cap L_2$ is felismerhető.

b) Ha $F = F_1 \times (Q_2 - F_2)$, akkor $L(M) = L(M_1) - L(M_2)$. Tehát ha L_1 és L_2 felismerhetők, akkor $L_1 - L_2$ is felismerhető.

Következmény: ha L felismerhető, akkor $\overline{L} = \Sigma^* - L$ is felismerhető.

c) Ha $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$, akkor $L(M) = L(M_1) \cup L(M_2)$. Tehát ha L_1 és L_2 felismerhetők, akkor $L_1 \cup L_2$ is felismerhető. (Újabb bizonyítás.)

Környezetfüggetlen (cf) nyelvek

$G = (N, \Sigma, P, S)$, minden szabály $A \to \alpha$ alakú

Korlátozás nélküli deriváció (levezetés):

$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$

Bal olodali deriváció

(α_i legbaloldalibb nemterminálisát helyettesítjük:

$\alpha_0 \Rightarrow_l \alpha_1 \Rightarrow_l \ldots \Rightarrow_l \alpha_n$

Jobb olodali deriváció

(α_i legjobboldalibb nemterminálisát helyettesítjük:

$\alpha_0 \Rightarrow_r \alpha_1 \Rightarrow_r \ldots \Rightarrow_r \alpha_n$

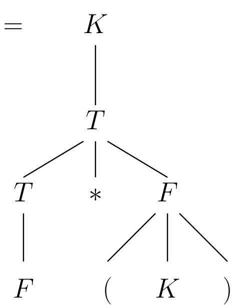
Példák

A aritmetikai kifejezéseket generáló G_{ar} nyelvtanban

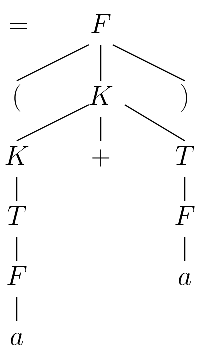
$$\begin{array}{lllllllll}
\underline{K} & \Rightarrow & \underline{T} & \Rightarrow & \underline{T} * F & \Rightarrow & F * \underline{F} & \Rightarrow & F * (K) \\
\underline{K} & \Rightarrow_l & \underline{T} & \Rightarrow_l & \underline{T} * F & \Rightarrow_l & \underline{F} * F & \Rightarrow_l & a * F \\
\underline{K} & \Rightarrow_r & \underline{T} & \Rightarrow_r & T * \underline{F} & \Rightarrow_r & T * (\underline{K}) & \Rightarrow_r & T * (K + T)
\end{array}$$

Ha α olyan, hogy $S \Rightarrow^* \alpha$ ($S \Rightarrow_l^* \alpha$, $S \Rightarrow_r^* \alpha$), akkor α *mondatforma (bal mondatforma, jobb mondatforma).*

$t_1 =$



$t_2 =$



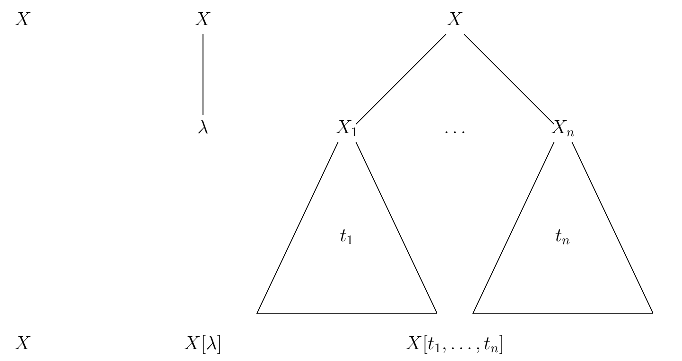
derivációs fák: $K \Rightarrow^* F * (K)$, $K \Rightarrow^* (a + a)$

Az $X \in (N \cup \Sigma)$ gyökerű derivációs fák halmaza a legszűkebb olyan D_X halmaz, amelyre

(i) Az a fa, amelynek egyetlen szögpontja (vagyis csak gyökere) az X, eleme D_X-nek. (Ezt a fát X-szel jelöljük.)

(ii) Ha $X \to \lambda \in P$, akkor az a fa, amelynek gyökere X, a gyökerének egyetlen leszármazottja a λ, eleme D_X-nek. (Ezt a fát $X[\lambda]$-val jelöljük.)

(iii) Ha $X \to X_1 \ldots X_n \in P$ és $t_1 \in D_{X_1}, \ldots, t_n \in D_{X_n}$, akkor az a fa, amelynek gyökere X, a gyökérből n él indul rendre a $t_1, \ldots, t_n$ fák gyökeréhez, eleme D_X-nek. (Ezt a fát $X[t_1, \ldots, t_n]$-nel jelöljük.



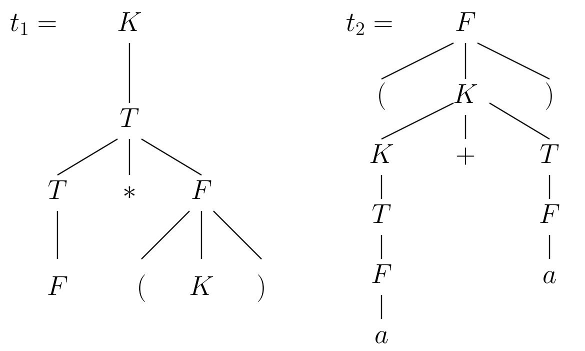
Legyen t egy X gyökerű derivációs fa. Akkor t magasságát $h(t)$-vel, t határát $fr(t)$-vel jelöljük és az alábbi módon definiáljuk:

(i) Ha $t = X$, akkor $h(t) = 0$ és $fr(t) = X$.

(ii) Ha $t = X[\lambda]$, akkor $h(t) = 1$ és $fr(t) = \lambda$.

(iii) Ha $t = X[t_1, \ldots, t_n]$, akkor $h(t) = 1 + max\{h(t_i) \mid 1 \leq i \leq n\}$ és $fr(t) = fr(t_1) \ldots fr(t_n)$.

Informálisan: $h(t)$ a t-ben lévő olyan utak hosszának maximuma, amelyek t gyökerétől annak valamely leveléhez vezetnek. Továbbá, $fr(t)$ az az $(N \cup \Sigma)^*$-beli szó, amelyet t leveleinek balról jobbra történő leolvasásával kapunk.



$fr(t_1) = F * (K)$ $\qquad$ $fr(t_2) = (a + a)$

$h(t_1) = 3$ $\qquad$ $h(t_2) = 5$

Tétel Tetszőleges $X \in (N \cup \Sigma)$ és $\alpha \in (N \cup \Sigma)^*$ esetén $X \Rightarrow^* \alpha$ akkor és csak akkor, ha van olyan $t \in D_X$ derivációs fa amelyre $fr(t) = \alpha$.

Bizonyítás. (a) Tfh $X \Rightarrow^* \alpha$, vagyis $X \Rightarrow^n \alpha$ valamilyen $n \geq 0$-ra.

(i) $\underline{n = 0}$: $X = \alpha$. A $t = X$ fa megfelelő lesz, mert erre $t \in D_X$ és $fr(t) = X (= \alpha)$.

(ii) $\underline{n \Rightarrow n + 1}$: $X \Rightarrow^{n+1} \alpha$ vagyis

$$X \Rightarrow X_1 \ldots X_k \Rightarrow^n \alpha_1 \ldots \alpha_k = \alpha.$$

1) $X \to X_1 \ldots X_k \in P$

2) Minden $1 \leq i \leq k$ esetén $X_i \Rightarrow^{n_i} \alpha_i$, ahol $n_i \leq n$. (Mellesleg: $n = n_1 + \ldots + n_k$.)

I. F: Minden $1 \le i \le k$-ra van olyan $t_i \in D_{X_i}$, hogy $fr(t_i) = \alpha_i$.

Legyen $t = X[t_1, \ldots, t_k]$. Ekkor $t \in D_X$, továbbá

$$fr(t) = fr(t_1) \ldots fr(t_k) = \alpha_1 \ldots \alpha_k = \alpha.$$

(b) Tfh az X gyökerű t derivációs fára teljesül, hogy $fr(t) = \alpha$.

t magassága szerinti indukció.

(i) $\underline{h(t) = 0}$: Akkor $t = X$, tehát $fr(t) = \alpha = X$. Következésképpen $X \Rightarrow^* \alpha (= X)$.

(ii) $\underline{h(t) = n + 1}$: $t = X[t_1, \ldots, t_k]$, valamilyen $k \ge 1$ szám és $t_1 \in D_{X_1}, \ldots, t_k \in D_{X_k}$ derivációs fák esetén.

1) $X \to X_1 \ldots X_k \in P$

2) Legyen minden $1 \le i \le k$-re $\alpha_i = fr(t_i)$.

I. F.: Minden $1 \le i \le k$-ra $X_i \Rightarrow^* \alpha_i$.

Akkor

$$X \Rightarrow X_1 \ldots X_k \Rightarrow^* \alpha_1 \ldots \alpha_k = \alpha.$$

Megjegyzések

(a) Ha $X \Rightarrow^* \alpha$, akkor általában nem csak egy olyan X gyökerű derivációs fa létezik melynek határa α.

Például legyenek az

$$\begin{array}{lcl} A & \to & aB \mid Ab \\ A & \to & a \\ B & \to & b \end{array}$$

szabályok egy környezetfüggetlen nyelvtan szabályai.

Akkor $A \Rightarrow^* ab$. Ugyanakkor a $t_1 = A[a, B[b]]$ és a $t_2 = A[A[a], b]$ fákra $fr(t_1) = ab$ és $fr(t_2) = ab$.

(b) Ha $t \in D_X$, akkor csak azt tudjuk, hogy $X \Rightarrow^* \alpha$, de nem igaz, hogy ezen deriváció lépései egyértelműen meghatározottak.

Például a G_{ar} nyelvtan esetén $t = K[T[T[F], *, F[(, K,)]]]$ egy K gyökerű derivációs fa, melynek határa $F * (K)$. Tudjuk, hogy $K \Rightarrow^* F * (K)$.

Ugyanakkor ez a deriváció két különböző lépéssorozattal is elvégezhető:

$$\begin{array}{lllll} K \Rightarrow & T \Rightarrow & T * F \Rightarrow & F * F & \Rightarrow F * (K) \\ K \Rightarrow & T \Rightarrow & T * F \Rightarrow & T * (K) & \Rightarrow F * (K) \end{array}$$

A különböző levezetési módok kapcsolata

Ha $X \Rightarrow_l^* \alpha$ vagy $X \Rightarrow_r^* \alpha$, akkor $X \Rightarrow^* \alpha$ is fennáll.

Fordítva nem igaz, például G_{ar}-ban

$$K \Rightarrow \underline{K} + T \Rightarrow K + \underline{T} + T \Rightarrow K + F + T,$$

de sem $K \Rightarrow_l^* K+F+T$ sem $K \Rightarrow_r^* K+F+T$ nem teljesül.

Ellenben, ha α terminális szó ($\alpha \in \Sigma^*$), akkor akkor az állítás már megfordítható, vagyis igaz lesz, hogy bal oldali (jobb oldali) levezetésekkel ugyanazok a terminális szavak kaphatók meg mint korlátozás nélküli levezetésekkel.

Lemma. Minden $X \in (N \cup \Sigma)$ és $w \in \Sigma^*$ esetén:

$$X \Rightarrow^* w \quad \Leftrightarrow \quad X \Rightarrow_l^* w \quad \Leftrightarrow \quad X \Rightarrow_r^* w.$$

Bizonyítás. Csak a baloldali levezetést vizsgáljuk (szimmetria).

(a) Ha $X \Rightarrow_l^* w$, akkor $X \Rightarrow^* w$.

(b) Tfh $X \Rightarrow^* w$, vagyis $X \Rightarrow^n w$, $n \geq 0$.

n szerinti indukció.

(i) $\underline{n = 0}$: $X = w$, tehát $X \Rightarrow_l^0 w$.

(ii) $\underline{X \Rightarrow^{n+1} w}$: $\quad X \Rightarrow X_1 \ldots X_k \Rightarrow^* w_1 w_2 \ldots w_k = w$

1) $X \to X_1 \ldots X_k \in P$

2) Minden $1 \leq i \leq k$-ra teljesül $X_i \Rightarrow^{n_i} w_i$, $n_i \leq n$.

I. F.: Minden $1 \leq i \leq k$-ra $X_i \Rightarrow_l^{n_i} w_i$ is fennáll.

Tehát

$$\begin{aligned} X &\Rightarrow_l X_1 X_2 \ldots X_k \\ &\Rightarrow_l w_1 X_2 \ldots X_k \\ &\Rightarrow_l w_1 w_2 \ldots X_k \\ &\ldots \\ &\Rightarrow_l w_1 w_2 \ldots w_k = w, \text{ vagyis } X \Rightarrow_l^* w. \end{aligned}$$

Összefoglalás

Egy $G = (N, \Sigma, G, S)$ cf nyelvtan által generált nyelv a most bevezetett fogalmak segítségével a következőképpen írható fel.

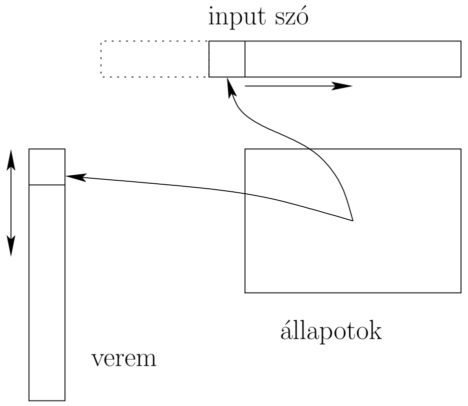
Levezetésekkel: $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\} = \{w \in \Sigma^* \mid S \Rightarrow_l^* w\} = \{w \in \Sigma^* \mid S \Rightarrow_r^* w\}$.

Derivációs fákkal: $L(G) = \{fr(t) \mid t \in D_S,\ fr(t) \in \Sigma^*\}$.

Környezetfüggetlen nyelvek és veremautomaták

Veremautomatának (vagy *pushdown automatának*, röviden *pda*-nak) nevezzük a $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ rendszert, ahol

- Q egy véges halmaz, az *állapotok halmaza*;
- Σ az *input ábécé*;
- Γ a *verem ábécé*;
- $q_0 \in Q$ a *kezdő állapot*;
- $Z_0 \in \Gamma$ a *verem kezdőszimbólum*;
- $F \subseteq Q$ a *végállapotok halmaza*;
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \mathcal{P}_f(Q \times \Gamma^*)$ az *átmenet függvény.*



Tetszőleges $q \in Q$, $a \in (\Sigma \cup \{\lambda\})$ input- és $Z \in \Gamma$ veremszimbólum esetén $\delta(q, a, Z) = \{(q_1, \alpha_1), \ldots, (q_n, \alpha_n)\}$, ahol $n \geq 0$, $q_1, \ldots, q_n \in Q$ és $\alpha_1, \ldots, \alpha_n \in \Gamma^*$.
(Az $n = 0$ esetben a képhalmaz az $\emptyset$ lesz.)

A $C = Q \times \Sigma^* \times \Gamma^*$ halmazt a P *konfigurációi halmazának* nevezzük.

Egy $(q, w, \gamma) \in C$ konfiguráció jelentése az, hogy P a $q \in Q$ állapotban van, a $w \in \Sigma^*$ input szót kapja és vermének tartalma γ.
A konfigurációt $(q, aw, Z\gamma)$ alakban is megadhatjuk.

A $\vdash_P \subseteq C \times C$ *átmeneti reláció*: tetszőleges $p, q \in Q$, $a \in (\Sigma \cup \{\lambda\})$, $w \in \Sigma^*$, $Z \in \Gamma$ és $\alpha, \gamma \in \Gamma^*$-ra

$$(q, aw, Z\gamma) \vdash_P (p, w, \alpha\gamma)$$

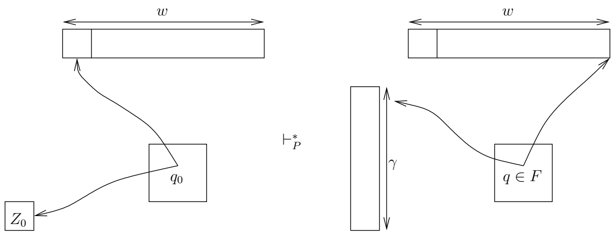
akkor és csakis akkor áll fenn, ha $(p, \alpha) \in \delta(q, a, Z)$.

A $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ által *végállapotokkal felismert nyelv*:

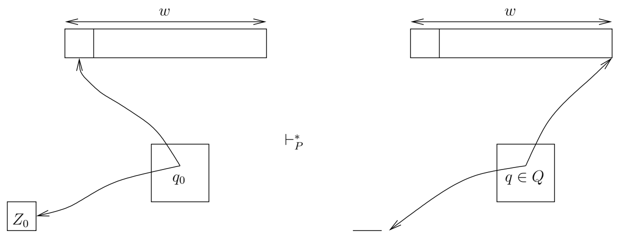
$$L_f(P) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \lambda, \gamma),\ \text{ahol } q \in F, \gamma \in \Gamma^*\}.$$

A P által *üres veremmel felismert nyelv*:

$$L_\emptyset(P) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \lambda, \lambda),\ \text{ahol } q \in Q\}.$$



felismerés végállapotokkal



felismerés üres veremmel

Példa $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, ahol

- $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, Z_0\}$, $F = \{q_0\}$,
- δ pedig a következő átmenetfüggvény:
  - $\delta(q_0, a, Z_0) = \{(q_1, aZ_0)\}$,
  - $\delta(q_1, a, a) = \{(q_1, aa)\}$,
  - $\delta(q_1, b, a) = \{(q_2, \lambda)\}$,
  - $\delta(q_2, b, a) = \{(q_2, \lambda)\}$,
  - $\delta(q_2, \lambda, Z_0) = \{(q_0, \lambda)\}$.

Az *aabb* input szóra

$$(q_0, aabb, Z_0) \vdash (q_1, abb, aZ_0) \vdash (q_1, bb, aaZ_0) \vdash$$
$$(q_2, b, aZ_0) \vdash (q_2, \lambda, Z_0) \vdash (q_0, \lambda, \lambda)$$

tehát $aabb \in L_f(P)$.

Az *abb* input szóra

$$(q_0, abb, Z_0) \vdash (q_1, bb, aZ_0) \vdash (q_2, b, Z_0) \vdash (q_0, b, \lambda),$$

mely utóbbi konfigurációból nem lehet tovább menni, tehát $abb \notin L_f(P)$.

Be lehet látni, hogy

$$L_f(P) = \{a^n b^n \mid n \geq 0\}.$$

Egy konkrét P veremautomatára általában $L_f(P) \neq L_\emptyset(P)$.

Ugyanakkor érvényes a következő tétel.

Tétel. A veremautomatákkal végállapotokkal felismerhető nyelvek osztálya megegyezik a veremautomatákkal üres veremmel felismerhető nyelvek osztályával.

$$\{L \mid L = L_f(P) \text{ valamely } P \text{ veremautomatára }\}$$
$$=$$
$$\{L \mid L = L_\emptyset(P) \text{ valamely } P \text{ veremautomatára }\}$$

Bizonyítás. a) Minden $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ pdahoz megkonstruálható olyan $P' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F')$ pda amelyre $L_\emptyset(P') = L_f(P)$.

- $Q' = Q \cup \{q_0', q_e\}$, ahol q_0' és q_e új állapotok;
- $\Gamma' = \Gamma \cup \{Z_0'\}$, ahol Z_0' egy új szimbólum;
- F' tetszőleges részhalmaza Q'-nek;
- (1) $\delta'(q_0', \lambda, Z_0') = \{(q_0, Z_0Z_0')\}$,

  (2) minden $q \in Q, a \in (\Sigma \cup \{\lambda\}), Z \in \Gamma$ esetén

$$\delta'(q, a, Z) = \begin{cases} \delta(q, a, Z) \cup \{(q_e, \lambda)\} & \text{ha } a = \lambda \text{ és } q \in F \\ \delta(q, a, Z) & \text{ha } a \neq \lambda \text{ vagy } q \notin F \end{cases}$$

  (3) minden $Z \in \Gamma'$-re $\delta'(q_e, \lambda, Z) = \{(q_e, \lambda)\}$.

$L_\emptyset(P') = L_f(P)$, mert:

$$\begin{array}{llll}
 & w \in L_f(P) & & \\
\Leftrightarrow & (q_0, w, Z_0) & \vdash_P^* \ (q, \lambda, \gamma) \text{ ahol } q \in F & \text{(def. szerint)} \\
\Leftrightarrow & (q_0', w, Z_0') & \vdash_{P'} \ (q_0, w, Z_0Z_0') & \text{((1) miatt)} \\
 & & \vdash_{P'}^* \ (q, \lambda, \gamma Z_0') \text{ ahol } q \in F & \text{((2) miatt)} \\
 & & = \ (q, \lambda, Z_1 \ldots Z_k Z_0') & (\gamma = Z_1 \ldots Z_k) \\
 & & \vdash_{P'} \ (q_e, \lambda, Z_2 \ldots Z_k Z_0') & \text{((2) miatt)} \\
 & & \vdash_{P'}^* \ (q_e, \lambda, \lambda) & \text{((3) miatt)} \\
\Leftrightarrow & & & \\
 & w \in L_\emptyset(P') & & \text{(def. szerint).}
\end{array}$$

b) Minden $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ pdahoz megkonstruálható olyan $P' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F')$ pda amelyre $L_f(P') = L_\emptyset(P)$

- $Q' = Q \cup \{q_0', q_f\}$, ahol q_0' és q_f új állapotok;
- $\Gamma' = \Gamma \cup \{Z_0'\}$, ahol Z_0' egy új szimbólum;
- $F' = \{q_f\}$;
- δ' az alábbi módon definiált átmenetfüggvény:

  (1) $\delta'(q_0', \lambda, Z_0') = \{(q_0, Z_0Z_0')\}$,

  (2) minden $q \in Q, a \in (\Sigma \cup \{\lambda\}), Z \in \Gamma$ esetén $\delta'(q, a, Z) = \delta(q, a, Z)$,

  (3) minden $q \in Q$ esetén, $\delta'(q, \lambda, Z_0') = \{(q_f, \lambda)\}$.

$L_f(P') = L_\emptyset(P)$, mert:

$$
\begin{array}{llll}
 & w \in L_\emptyset(P) & & \\
\Leftrightarrow & (q_0, w, Z_0) & \vdash_P^* \ (q, \lambda, \lambda) \text{ ahol } q \in Q & \text{(def. szerint)} \\
\Leftrightarrow & (q_0', w, Z_0') & \vdash_{P'} \ (q_0, w, Z_0 Z_0') & \text{((1) miatt)} \\
 & & \vdash_{P'}^* \ (q, \lambda, Z_0') \text{ ahol } q \in Q & \text{((2) miatt)} \\
 & & \vdash_{P'} \ (q_f, \lambda, \lambda) & \text{((3) miatt)} \\
\Leftrightarrow & & & \\
 & w \in L_f(P') & & \text{(def. szerint).}
\end{array}
$$

A cf nyelvtanok és a pdak ekvivalenciája

Tétel. Minden cf nyelv felismerhető pdaval.

Bizonyítás. Vegyünk egy $G = (N, \Sigma, R, S)$ környezetfüggetlen nyelvtant. Legyen $P = (\{q\}, \Sigma, \Gamma, \delta, q, Z_0, \emptyset)$ az a veremautomata, amelyben

- $\Gamma = N \cup \Sigma$,
- $Z_0 = S$,
- δ átmenetfüggvény pedig a következő módon van definiálva:

(1) minden $A \in N$-re $\delta(q, \lambda, A) = \{(q, \alpha) \mid A \to \alpha \in R\}$,

(2) minden $a \in \Sigma$-ra $\delta(q, a, a) = \{(q, \lambda)\}$.

Megmutatjuk, hogy $L_\emptyset(P) = L(G)$. Elegendő igazolni, hogy minden $X \in (N \cup \Sigma)$ és $w \in \Sigma^*$ esetén

$X \Rightarrow^* w$ akkor és csakis akkor, ha $(q, w, X) \vdash^* (q, \lambda, \lambda)$

(a) Tfh $X \Rightarrow^n w$

(i) <u>$n = 0$</u>: $X = w \in \Sigma$, ezért a (2) pont szerint

$$(q, w, X) = (q, w, w) \vdash (q, \lambda, \lambda).$$

(ii) <u>$n \Rightarrow n+1$</u>: $X \Rightarrow X_1 \ldots X_k \Rightarrow^n w_1 \ldots w_k = w$

1) $X \to X_1 \ldots X_k \in R$

2) $X_i \Rightarrow^{n_i} w_i$ minden $1 \le i \le k$-ra, ahol $n_i \le n$

I. F.: minden $1 \le i \le k$ esetén $(q, w_i, X_i) \vdash^* (q, \lambda, \lambda)$.

Innen kapjuk, hogy

$$
\begin{array}{lll}
(q, w, X) & = & (q, w_1 \ldots w_k, X) \\
 & \vdash & (q, w_1 \ldots w_k, X_1 \ldots X_k) \\
 & \vdash^* & (q, w_2 \ldots w_k, X_2 \ldots X_k) \\
 & & \ldots \\
 & \vdash^* & (q, w_k, X_k) \\
 & \vdash^* & (q, \lambda, \lambda).
\end{array}
$$

A felismerés során P a w-nek egy bal oldali levezetését szimulálja.

(b) Tfh $(q, w, X) \vdash^* (q, \lambda, \lambda)$. Az átmenetek során alkalmazott (1) típusú átmenetek száma szerinti indukcióval igazolhatjuk, hogy $X \Rightarrow^* w$.

Példa

G_{ar} nyelvtanhoz a $P_{ar} = (\{q\}, \Sigma, \Gamma, \delta, q, K, \emptyset)$ veremautomata, ahol

- $\Sigma = \{a, +, *, (,)\}$,
- $\Gamma = \{K, T, F, a, +, *, (,)\}$,
- δ pedig a következő átmenet függvény:
  - $\delta(q, \lambda, K) = \{(q, K+T), (q, T)\}$,
  - $\delta(q, \lambda, T) = \{(q, T*F), (q, F)\}$,
  - $\delta(q, \lambda, F) = \{(q, (K)), (q, a)\}$,
  - minden $x \in \{a, +, *, (,)\}$ esetén $\delta(q, x, x) = \{(q, \lambda)\}$.

Például P_{ar} a következőképpen ismeri fel az $a + a \in L(G_{ar})$ szót:

$$\begin{array}{llllll}
(q, a+a, K) & \vdash & (q, a+a, K+T) & \vdash^2 & (q, a+a, F+T) & \vdash \\
(q, a+a, a+T) & \vdash & (q, +a, +T) & \vdash & (q, a, T) & \vdash \\
(q, a, F) & \vdash & (q, a, a) & \vdash & (q, \lambda, \lambda) &
\end{array}$$

Tétel. Minden pdaval felismerhető nyelv környezetfüggetlen.

Bizonyítás. Legyen $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ egy pda. Konstruáljuk meg $G = (N, \Sigma, R, S)$ cf nyelvtant a következőképpen.

- Legyen S egy új szimbólum,
- legyen $N = \{S\} \cup \{[qZr] \mid q, r \in Q, Z \in \Gamma\}$,

- legyen R szabályok legszűkebb olyan halmaza, amire teljesülnek a következő feltételek:

(1) minden $q \in Q$-ra legyen $S \to [q_0 Z_0 q]$ szabály R-ben,

(2) minden $q \in Q, a \in (\Sigma \cup \{\lambda\}), Z \in \Gamma$-ra, ha $(s_0, Z_1 \dots Z_k) \in \delta(q, a, Z)$, (ahol $k \geq 1, Z_1, \dots, Z_k \in \Gamma$) akkor minden $s_1, \dots, s_k \in Q$ sorozatra legyen $[qZs_k] \to a[s_0 Z_1 s_1] \dots [s_{k-1} Z_k s_k]$ szabály R-ben,

(3) minden $q \in Q, a \in (\Sigma \cup \{\lambda\}), Z \in \Gamma$-ra, ha $(s_0, \lambda) \in \delta(q, a, Z)$, akkor legyen $[qZs_0] \to a$ szabály R-ben (az előző eset $k = 0$-ra).

Elegendő megmutatni, hogy minden $q, r \in Q, Z \in \Gamma$ és $w \in \Sigma^*$ esetén

$(q, w, Z) \vdash^* (r, \lambda, \lambda)$ akkor és csak akkor, ha $[qZr] \Rightarrow^* w$.

Ugyanis ezen ekvivalencia a $q = q_0$ és $Z = Z_0$ esetben éppen az $L(G) = L_\emptyset(P)$ egyenlőséget jelenti:

$$\begin{aligned} & w \in L_\emptyset(P) \\ \Leftrightarrow\ & (q_0, w, Z_0) \vdash^* (r, \lambda, \lambda) \\ \Leftrightarrow\ & [q_0 Z_0 r] \Rightarrow^* w \\ \Leftrightarrow\ & S \Rightarrow [q_0 Z_0 r] \Rightarrow^* w \\ \Leftrightarrow\ & w \in L(G) \end{aligned}$$

(a) Tfh $[qZr] \Rightarrow^n w$ valamilyen $n \geq 1$-re.

(i) $\underline{n = 1}$: $[qZr] \Rightarrow w$, ez csak úgy lehet, ha $w = a \in (\Sigma \cup \{\lambda\})$ és $(r, \lambda) \in \delta(q, a, Z)$. Tehát $(q, w, Z) = (q, a, Z) \vdash (r, \lambda, \lambda)$.

(ii) $\underline{n \Longrightarrow n + 1}$:

$$[qZr] \Rightarrow a[s_0 Z_1 s_1] \ldots [s_{k-1} Z_k s_k] \Rightarrow^n a w_1 \ldots w_k = w$$

1) $[qZr] \rightarrow a[s_0 Z_1 s_1] \ldots [s_{k-1} Z_k s_k] \in R$, $r = s_k$, azaz $(s_0, Z_1 \ldots Z_k) \in \delta(q, a, Z)$

2) Minden $1 \leq i \leq k$-ra $[s_{i-1} Z_i s_i] \Rightarrow^{n_i} w_i$, és $n_i \leq n$.

I. F.: minden $1 \leq i \leq k$-ra $(s_{i-1}, w_i, Z_i) \vdash^* (s_i, \lambda, \lambda)$.
Kapjuk, hogy

$$\begin{array}{rcl} (q, w, Z) & = & (q, a w_1 \ldots w_k, Z) \\ & \vdash & (s_0, w_1 \ldots w_k, Z_1 \ldots Z_k) \\ & \vdash^* & (s_1, w_2 \ldots w_k, Z_2 \ldots Z_k) \\ & & \ldots \\ & \vdash^* & (s_{k-1}, w_k, Z_k) \\ & \vdash^* & (s_k, \lambda, \lambda) \\ & = & (r, \lambda, \lambda) \end{array}$$

Az ekvivalencia megfordítását hasonló módon, a $(q, w, Z) \vdash^n (r, \lambda, \lambda)$ feltételben szereplő n szerinti indukcióval bizonyíthatjuk be.

Cf nyelvtanok λ-mentesítése

Egy $G = (N, \Sigma, P, S)$ cf nyelvtan λ-*mentes*, ha P nem tartalmaz $A \rightarrow \lambda$ alakú szabályokat, kivéve esetleg az $S \rightarrow \lambda$ szabályt. Ha azonban $S \rightarrow \lambda \in P$, akkor S nem szerepel semelyik P-beli szabály jobb oldalán.

Tétel. Tetszőleges $G = (N, \Sigma, P, S)$ cf nyelvtanhoz megkonstruálható vele ekvivalens ($L(G) = L(G')$) λ-mentes $G' = (N', \Sigma, P', S')$ környezetfüggetlen nyelvtan.

Bizonyítás.

a) Először megadunk egy olyan λ-mentes $G_1 = (N, \Sigma, P_1, S)$ nyelvtant, melyre $L(G_1) = (L(G) - \{\lambda\})$.

Számoljuk ki a $H = \{A \in N \mid A \Rightarrow^* \lambda\}$ halmazt.

Legyen P_1 a legszűkebb olyan halmaz, melyre teljesül, hogy

- minden olyan $A \to \alpha \in P$ szabály esetén melyre $\alpha \neq \lambda$,
- minden olyan $A \to \alpha_1$ szabály P_1-ben van, melyre $\alpha_1 \neq \lambda$ és α_1 úgy keletkezik α-ból, hogy töröljük belőle H-beli nemterminálisok 0 vagy több előfordulását.

(Ha $A, B \in H$ és az $A \to aCBbAB \in P$, akkor az $A \to aCBbAB$, $A \to aCbAB$, $A \to aCBbB$, $A \to aCBbA$, $A \to aCbB$, $A \to aCbA$, $A \to aCBb$, $A \to aCb \in P_1$.

Ha $C \to AB \in P$, akkor a $C \to A$, $C \to B \in P_1$, de $C \to \lambda \notin P_1$.)

Ekkor $L(G_1) = (L(G) - \{\lambda\})$.

b) A G_1 ismeretében a következőképpen adjuk meg G'-t

Ha $\lambda \notin L(G)$ (vagyis, ha $S \notin H$), akkor legyen $G' = G_1$, különben pedig legyen

$$G' = (N \cup \{S'\}, \Sigma, P_1 \cup \{S' \to S, S' \to \lambda\}, S').$$

Nyilvánvaló, hogy mindkét esetben G' is λ-mentes, és $L(G') = L(G)$.

Nem minden környezetfüggetlen nyelvtan környezetfüggő (definíció). Ugyanakkor minden λ-mentes környezetfüggetlen nyelvtan egyben környezetfüggő is. Ezért az előző tételből azonnal következik az alábbi.

Tétel. $\mathcal{L}_2 \subseteq \mathcal{L}_1$.

Bizonyítás. Legyen $L \in \mathcal{L}_2$. Akkor van olyan $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtan, melyre $L = L(G)$. Ha G λ-mentes, akkor egyben környezetfüggő is, tehát $L \in \mathcal{L}_1$. Az ellenkező esetben, az előző tétel szerint van olyan G' λ-mentes nyelvtan, melyre $L = L(G')$. Mivel ekkor G' környezetfüggő, megint csak azt kapjuk, hogy $L \in \mathcal{L}_1$.

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma.

Tétel. Minden $L \subseteq \Sigma^*$ környezetfüggetlen nyelvhez megadhatók olyan (L-től függő) $p, q > 0$ egész számok, hogy minden $w \in L$ esetén, ha $|w| > p$, akkor vannak olyan $w_1, w_2, w_3, w_4, w_5 \in \Sigma^*$ szavak, melyekre teljesülnek az alábbi feltételek:

1) $w = w_1 w_2 w_3 w_4 w_5$,

2) $|w_2 w_3 w_4| \leq q$,

3) $w_2 w_4 \neq \lambda$,

4) minden $n \geq 0$-ra, $w_1 w_2^n w_3 w_4^n w_5 \in L$.

(Szükséges feltétele annak, hogy egy nyelv cf legyen.)

Bizonyítás. Legyen $L = L(G)$, ahol $G = (N, \Sigma, P, S)$ láncszabály és λ-mentes cf nyelvtan.

Legyen $n = ||N||$, $k = \max\{|\alpha| \mid A \to \alpha \in P\}$ és

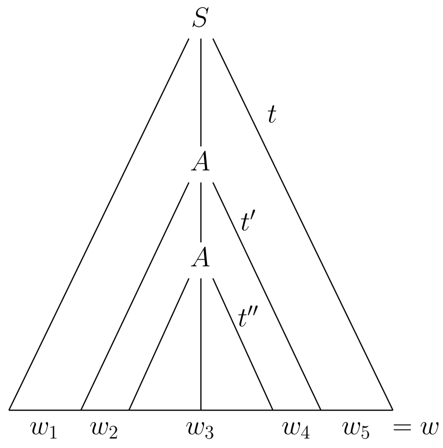
$$p := k^n \quad \text{és} \quad q := k^{n+1}.$$

Megmutatjuk, hogy ez a p és q megfelelő lesz.

Vegyünk egy $w \in L$ szót, melyre $|w| > p$. Akkor van olyan $t \in D_S$ derivációs fa, melyre $fr(t) = w$ és $h(t) > n$.

Következésképpen t-ben van olyan, S-től valamely levélhez vezető út melynek hossza legalább $n+1$ és amelyen ezért (egy terminális és) legalább $n+1$ nemterminális szimbólum szerepel.

Mivel $n = ||N||$, van olyan nemterminális ami ezen az úton legalább kétszer előfordul.

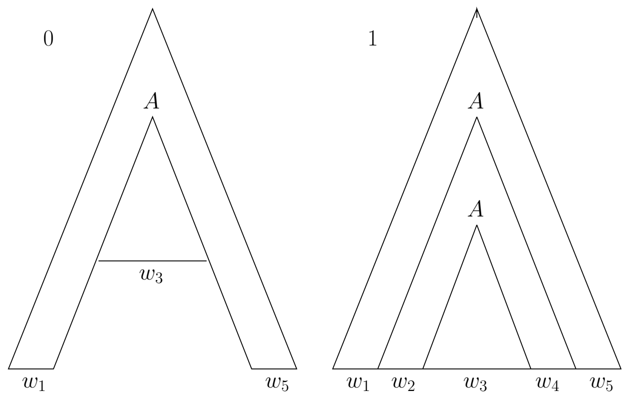


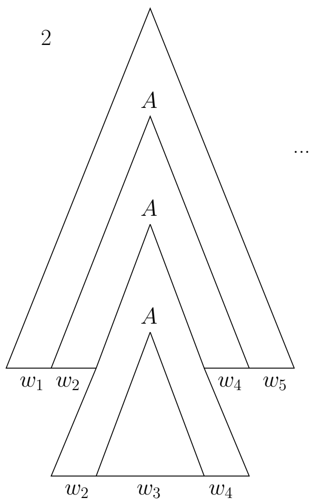
Legyen A az a nemterminálist amelyik ezen úton a terminális levéltől indulva a gyökér felé legelőször megismétlődik.

Definiáljuk w_1, w_2, w_3, w_4 és w_5 szavakat az ábrán látható módon.

Ekkor:

1) $w = w_1w_2w_3w_4w_5$, mert t határát osztottuk fel,
2) $|w_2w_3w_4| \leq q$, mert a legelső ismétlődést vettük,
3) $w_2w_4 \neq \lambda$, mert G láncszabály és λ-mentes,
4) minden $n \geq 0$-ra, $w_1w_2^nw_3w_4^nw_5 \in L$, mert a t' fa iterálható.





Tétel. Az $L = \{a^n b^n c^n \mid n \geq 1\}$ nyelv nem környezetfüggetlen.

Bizonyítás. Tfh igen. A Bar Hillel lemma szerint vannak olyan $p, q > 0$ számok, hogy minden $w \in L$ szóra, ha $|w| > p$, akkor teljesülnek ezen lemmában szereplő 1) – 4) feltételek.

Vegyük a $w = a^p b^p c^p \in L$ szót. Mivel $|w| = 3p$, az is igaz, hogy $|w| \geq p$.

4) feltétel: $w = w_1 w_2 w_3 w_4 w_5$, $w_2 w_4 \neq \lambda$, ahol minden $n \geq 0$-ra $w_1 w_2^n w_3 w_4^n w_5 \in L$.

$a^p b^p c^p = w_1 w_2 w_3 w_4 w_5$,

$w_2 w_4 \neq \lambda$,

minden $n \geq 0$-ra $w_1 w_2^n w_3 w_4^n w_5 \in L$.

Hogyan helyezkedhet w_2 és w_4 a $a^p b^p c^p$-ben?

Sem w_2 sem w_4 nem tartalmazhat két különböző betűt, mert ekkor például a $w_1 w_2 w_2 w_3 w_4 w_4 w_5$ szóban a betűk sorrendje nem $a - b - c$ lenne!

Tehát mind w_2 mind w_4 legfeljebb egy fajta betűt tartalmaz.

Akkor n-et növelve a $w_1 w_2^n w_3 w_4^n w_5$ szóban csak legfeljebb két fajta betűnek a száma növekszik, tehát $w_1 w_2^n w_3 w_4^n w_5 \notin L$. $\lightning$

Tétel. $\mathcal{L}_2 \subset \mathcal{L}_1$.

Bizonyítás. Azt már láttuk, hogy $\mathcal{L}_2 \subseteq \mathcal{L}_1$ (λ-mentesítés).

$L = \{a^n b^n c^n \mid n \geq 1\} \in \mathcal{L}_1$ (környezetfüggő), mert a következő nyelvtan generálja:

$S \to SC_1$	$S \to AC_1$	$A \to aB_1$
$A \to aAB_1$	$B_1C_1 \to B_1C_3$	$B_1C_3 \to BC_3$
$BC_3 \to BC$	$B_1B \to B_1B_2$	$B_1B_2 \to BB_2$
$BB_2 \to BB_1$	$CC_1 \to C_2C_1$	$C_2C_1 \to C_2C$
$C_2C \to C_1C$	$B \to b$	$C \to c$

Ugyanakkor $L \notin \mathcal{L}_2$.

Determinisztikus veremautomaták

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ veremautomata *determinisztikus*, ha minden $q \in Q$-ra és $Z \in \Gamma$-ra a következő két feltétel valamelyike teljesül:

(1) $\delta(q, \lambda, Z) = \emptyset$ és minden $a \in \Sigma$-ra a $\delta(q, a, Z)$ halmaz legfeljebb egy elemű, vagy

(2) minden $a \in \Sigma$-ra $\delta(q, a, Z) = \emptyset$ és a $\delta(q, \lambda, Z)$ halmaz legfeljebb egy elemű.

Tétel. A determinisztikus veremautomatákkal végállapotokkal (!) felismerhető nyelvek osztálya valódi részosztálya a veremautomatákkal felismerhető nyelvek osztályának.

Megjegyzések.

1. A közönséges automaták esetében egyenlőség van.

2. Determinisztikus veremautomaták esetében csak a végállapotokkal való felismerés hatékony

3. ... és nem igaz, hogy a determinisztikus veremautomatákkal végállapotokkal felismerhető nyelvek osztálya megegyezik a a determinisztikus veremautomatákkal üres veremmel felismerhető nyelvek osztályával.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Tétel. A cf nyelvek osztálya zárt a reguláris műveletekre nézve.

Bizonyítás. Legyenek $G_1 = (N_1, \Sigma, P_1, S_1)$ és $G_2 = (N_2, \Sigma, P_2, S_2)$ cf nyelvtanok, $L_1 = L(G_1)$ és $L_2 = L(G_2)$.

- Legyen $G = (N_1 \cup N_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S)$, ahol S egy új szimbólum. Akkor $L(G) = L_1 \cup L_2$.
- Legyen $G = (N_1 \cup N_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$. Akkor $L(G) = L_1 L_2$.
- Legyen $G = (N_1 \cup \{S\}, \Sigma, P_1 \cup \{S \to S_1 S, S \to \lambda\}, S)$. Akkor $L(G) = L_1^*$.

Boole műveletekre való zártság

Tétel. A cf nyelvek osztálya nem zárt sem a metszetre sem a komplementerre (ugyanakkor zárt az egyesítésre).

Bizonyítás.

a) metszet: $L_1 = \{a^m b^m c^n \mid m, n \geq 0\}$ cf, mert generálható az alábbi környezetfüggetlen nyelvtannal:

$$\begin{aligned} S &\to AB \\ A &\to aAb|\lambda \\ B &\to cB|\lambda \end{aligned}$$

Hasonlóan látható be, hogy az $L_2 = \{a^m b^n c^n \mid m, n \geq 0\}$ is környezetfüggetlen.

Ugyanakkor $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$, ami nem cf.

b) komplementer: Legyenek $L_1 \subseteq \Sigma^*$ és $L_2 \subseteq \Sigma^*$ környezetfüggetlen nyelvek.

A de Morgan azonosság szerint $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. Továbbá, a környezetfüggetlen nyelvek osztálya zárt az egyesítésre.

Ezért, ha a komplementer képzésre is zárt lenne, akkor a metszetre is zárt lenne.

Elemzési módszerek

Adott: A PL programozási nyelv szintaxisa: $G_{PL} = (N, \Sigma, P, S)$ cf nyelvtan

és egy (felhasználó által megírt) PL nyelvű w program (szó, sztring).

Kérdés: A w program megfelel-e a szintaktikai előírásoknak?

Válasz: w program szintaktikusan helyes $\Leftrightarrow$ $w \in L(G_{PL})$.

Az elemzés alapfeladata: Adjunk meg olyan algoritmust, amely tetszőleges $G = (N, \Sigma, P, S)$ cf nyelvtan és $w \in \Sigma^*$ szó esetén eldönti, hogy $w \in L(G)$ teljesül-e!

Felülről lefelé (top-down) haladó elemzési algoritmusok: az S kezdőszimbólumból kiindulva próbálunk meg felépíteni egy olyan derivációs fát aminek a határa w.

Alulról felfelé (bottom-up) haladó elemzési algoritmusok: a w szóból kiindulva próbálunk meg felépíteni egy olyan derivációs fát aminek a gyökere S, a határa pedig w.

Mindkét esetben: Ha sikerül, $w \in L(G)$, különben $w \notin L(G)$.

Felülről lefelé haladó elemzés

Előkészület: a balrekurzió megszüntetése.

Definíció. Egy $G = (N, \Sigma, P, S)$ egy cf nyelvtan *balrekurzív*, ha van olyan $A \in N$, melyre $A \Rightarrow^+ A\alpha$.

$A \Rightarrow^+ A\alpha \Rightarrow^+ A\alpha\alpha \Rightarrow^+ A\alpha\alpha\alpha \ldots$

Lemma. (Lásd tankönyv). Tetszőleges G cf nyelvtanhoz konstruálható olyan G' nem balrekurzív cf nyelvtan, amelyre $L(G) = L(G')$.

Általános felülről lefelé elemzés

Adott $G = (N, \Sigma, P, S)$ cf nyelvtan és $w \in \Sigma^*$ szó, igaz-e, hogy $w \in L(G)$

Csak balrekurziómentes nyelvtanokkal foglalkozunk.

Az S kezdőszimbólumból kiindulva próbálunk meg felépíteni egy olyan derivációs fát aminek a határa w.

Kiterjesztés: A derivációs fában egy nemterminálist helyettesítünk valamely alternatívájával.

Illesztés: Annak az ellenőrzése, hogy a kiterjesztésnél alkalmazott alternatívában szereplő terminálisok illeszkednek-e az elemezendő szó (w) megfelelő részéhez.

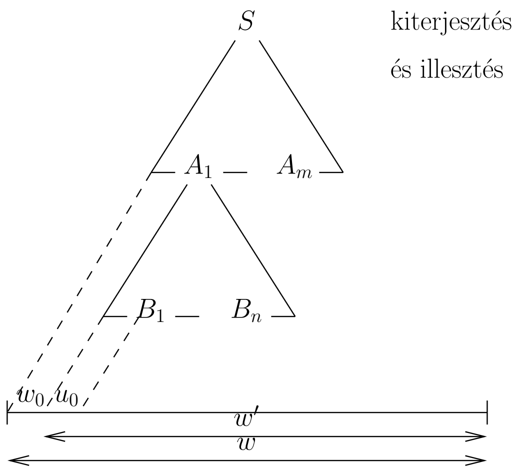
Az általános felülről lefelé haladó elemzés alapötlete.

– Minden $A \in N$-re rögzítsük le az A alternatíváinak egy $A \to \gamma_1 \mid \ldots \mid \gamma_k$ sorrendjét.

– Legyenek S alternatívái, $S \to \alpha_1 \mid \ldots \mid \alpha_k$.

– Keressük meg az első olyan $\alpha_i = w_0 A_1 w_1 \ldots A_m w_m$ alternatívát, melyre w_0 illeszkedik-e w elejéhez, azaz fennáll $w = w_0 w'$. ($w_0 = \lambda$ mindig jó!)

– Ha nincs ilyen α_i, akkor azt mondhatjuk, hogy $w \notin L(G)$ és az algoritmus leáll.



kiterjesztés

és illesztés

– Ha van ilyen i, azaz $\alpha_i = w_0 A_1 w_1 \ldots A_m w_m$ és $w = w_0 w'$, akkor A_1-et terjesztjük ki. Az $A_1 \to \beta_1 \mid \ldots \mid \beta_l$ alternatívák között megkeressük a legelső olyat, ami illeszkedik w' elejéhez: $\beta_j = u_0 B_1 u_1 \ldots B_n u_n$ és $w' = u_0 w''$.

– Ha egy nemterminális kiterjesztése esetén (pl. most A_1) nem találunk olyan alternatívát ami illeszkedik w megmaradt részéhez, akkor nem mondhatjuk azt, hogy $w \notin L(G)$, mivel valamennyi előző kiterjesztés esetében a *legelső* olyan alternatívát választottuk ami illeszkedett.

– Ezért, ha egy szinten nem találunk illeszkedő alternatívát, akkor egy szintet visszamegyünk (ún. *backtrack*-et hajtunk végre) és az ott választott alternatíva után következő első olyan alternatívát kell választanunk ami illeszkedik.

Felülről lefelé haladó általános elemzési algoritmus.

Input Egy nem balrekurzív $G = (N, \Sigma, P, S)$ cf nyelvtan és egy $w = a_1 \ldots a_n$, $n \geq 0$ input szó.

Ouput *Igen* és a w egy bal oldali levezetése, ha $w \in L(G)$. Különben *Nem*.

Módszer

- Minden $A \in N$ esetében rögzítsük le A alternatíváinak egy $A \to \gamma_1 \,|\, \gamma_2 \,|\, \ldots \,|\, \gamma_k$ sorrendjét. Az A i-edik alternatívájára A_i-vel fogunk hivatkozni.
- Az elemzés (s, i, α, β) alakú konfigurációk sorozata.

– $s \in \{ q, t, b \}$, normál, elfogadó és backtrack *állapot*.

– i egy pointer, $1 \leq i \leq n + 1$, $a_{n+1} := \$$.

– α egy verem, melynek teteje a jobb végén van. Tartalmazza az elemzés mindenkori „történetét”. Kezdő értéke: λ.

– β egy verem, melynek teteje a bal végén van. Tartalmazza a levezetett bal oldali mondatformának azt a részét amelyet még ki kell terjeszteni. Kezdőértéke: S.

- A konfigurációk halmazán megadunk egy $\vdash$ *átmeneti relációt*. Minden (s, i, α, β) konfigurációhoz legfeljebb egy $(s', i', \alpha', \beta')$ konfiguráció van, melyre $(s, i, \alpha, \beta) \vdash (s', i', \alpha', \beta')$.
- $w \in L(G) \Leftrightarrow$, ha $(q, 1, \lambda, S) \vdash^* (t, n + 1, \alpha, \lambda)$.

Az elemzési algoritmus

1. $C := (q, 1, \lambda, S)$;
2. Amíg van olyan C', melyre $C \vdash C'$, legyen $C := C'$;
3. Ha $C := (t, n + 1, \alpha, \lambda)$ valamely α-ra, akkor output: *Igen* különben output: *Nem*.

Az algoritmusban szereplő $\vdash$ átmeneti relációt a következőképpen definiáljuk: az alábbi (1) – (6) pontok közül sorrendben a legelső alkalmazhatót alkalmazzuk a C konfigurációra.

Átmeneti reláció

(1) *Kiterjesztés:* Ha $C = (q, i, \alpha, A\beta)$, vagyis az aktív szimbólum nemterminális, akkor $C \vdash C'$, ahol $C' = (q, i, \alpha A_1, \gamma_1\beta)$, és γ_1 az A első alternatívája.

(2) *Sikeres input illesztés:* Ha $C = (q, i, \alpha, a\beta)$ és $a = a_i$, vagyis az aktív szimbólum terminális és illeszkedik az input pointer által mutatott input szimbólumhoz, akkor $C \vdash C'$, ahol $C' = (q, i + 1, \alpha a, \beta)$. (A lépés $i = n + 1$ esetén soha nem alkalmazható, mert $a_{n+1} = \$ \neq a$, semmilyen $a \in \Sigma$-ra.)

(3) *Sikeres elemzés:* Ha $C = (q, n + 1, \alpha, \lambda)$, akkor $C \vdash C'$, ahol $C' = (t, n + 1, \alpha, \lambda)$, vagyis elérjük a befejező konfigurációt.

(4) *Sikertelen input illesztés:* Ha $C = (q, i, \alpha, a\beta)$ de $a \neq a_i$, akkor $C \vdash C'$, ahol $C' = (b, i, \alpha, a\beta)$. (Átmegy backtrack állapotba.)

(5) *Backtrack az inputban:* Ha $C = (b, i, \alpha a, \beta)$, akkor $C \vdash C'$, ahol $C' = (b, i-1, \alpha, a\beta)$. (A passzív veremből visszatesszük az inputot.)

(6) *Backtrack a kiterjesztésben:* Ha $C = (b, i, \alpha A_j, \gamma_j\beta)$ akkor az alábbi esetek lehetségesek:

(i) Ha A-nak van $j+1$-edik alternatívája is, akkor $C \vdash C'$, ahol $C' = (q, i, \alpha A_{j+1}, \gamma_{j+1}\beta)$. ($A$-t a következő, vagyis a $j+1$-edik alternatívájával terjesztjük ki. Utána ezt illesztjük ezért átmegyünk normál állapotba.)

(ii) Ha $i = 1, A = S$ és S-nek csak j alternatívája van, akkor nincs átmenet semelyik konfigurációba.

(iii) Ha az előző feltételek egyike sem teljesül, akkor $C \vdash C'$, ahol $C' = (b, i, \alpha, A\beta)$. ($A$-nak már minden alternatíváját kipróbáltuk, ezért vissza kell térnünk az előző szintre.)

Példa. G_e: $K \to T + K \,|\, T$
$T \to a \,|\, b$

Az alternatívák sorrendje a fenti, tehát K_1 a $K \to T + K$, K_2 a $K \to T$, T_1 a $T \to a$ és végül T_2 a $T \to b$ alternatívát jelentik. Elemezzük a $w = b + a$ szót az előbb ismertetett általános elemzési algoritmussal. (Tehát most $n = 3$, $a_1 = b$, $a_2 = +$, $a_3 = a$ és $a_4 = \$$.)

Példa. $K \to T + K \,|\, T$
$T \to a \,|\, b$

$$
\begin{aligned}
&(q,1,\lambda,K) \vdash (q,1,K_1,T+K) \vdash (q,1,K_1T_1,a+K) \vdash \\
&(b,1,K_1T_1,a+K) \vdash (q,1,K_1T_2,b+K) \vdash (q,2,K_1T_2b,+K) \vdash \\
&(q,3,K_1T_2b+,K) \vdash (q,3,K_1T_2b+K_1,T+K) \vdash \\
&(q,3,K_1T_2b+K_1T_1,a+K) \vdash (q,4,K_1T_2b+K_1T_1a,+K) \vdash \\
&(b,4,K_1T_2b+K_1T_1a,+K) \vdash (b,3,K_1T_2b+K_1T_1,a+K) \vdash \\
&(q,3,K_1T_2b+K_1T_2,b+K) \vdash (b,3,K_1T_2b+K_1,T+K) \vdash \\
&(q,3,K_1T_2b+K_2,T) \vdash (q,3,K_1T_2b+K_2T_1,a) \vdash \\
&(q,4,K_1T_2b+K_2T_1a,\lambda) \vdash (t,4,K_1T_2b+K_2T_1a,\lambda)
\end{aligned}
$$

Következésképpen $b + a \in L(G_e)$.

$LL(k)$ nyelvtanok és elemzésük

Az $LL(k)$ elemzés (mint speciális felülről lefelé elemzés) azon alapul, hogy amikor egy A nemterminális kiterjesztését keressük akkor pótlólagos információként megnézzük az input még feldolgozatlan részének k hosszúságú prefixét és ennek ismeretében egyértelműen ki tudjuk választani A-nak azt az alternatíváját amely szerint ki kell őt terjeszteni, ha van ilyen. Amennyiben nincs ilyen alternatíva, akkor azt tudjuk mondani, hogy az elemzett szó nem eleme $L(G)$-nek.

Csak azokat nyelvtanokat lehet $LL(k)$ módon elemezni, amelyek kielégítik az ún. $LL(k)$ feltételt.

A $FIRST_k$ halmaz definíciója

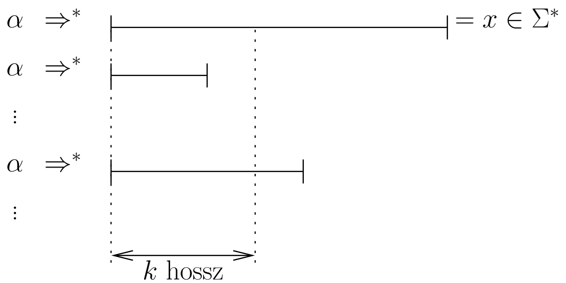
Definíció. Legyen $k \geq 0$ egy egész szám.

Tetszőleges $\alpha \in (N \cup \Sigma)^*$-ra

$FIRST_k(\alpha) = \{w \mid \alpha \Rightarrow^* wx \in \Sigma^*$ és $|w| = k$ vagy $(|w| < k$ és $x = \lambda)\}$

Tetszőleges $L \subseteq (N \cup \Sigma)^*$-ra

$$FIRST_k(L) = \bigcup_{\alpha \in L} FIRST_k(\alpha)$$



Megjegyzés.

1. $FIRST_k(\alpha), FIRST_k(L) \subseteq \Sigma^{*,k}$ (és így véges halmazok).

2. $u \in \Sigma^*$-ra

$$FIRST_k(u) = \begin{cases} \{\, u \,\} & \text{ha } |u| < k \\ \{\, w \,\} & \text{ha } u = wx, \text{ ahol } |w| = k. \end{cases}$$

Ilyenkor $FIRST_k(u) = \{\, u \,\}$ és $FIRST_k(u) = \{\, w \,\}$ helyett rendre $FIRST_k(u) = u$ és $FIRST_k(u) = w$-t írunk.

3. A továbbiakban a $FIRST_k$ jelölés helyett a rövidebb FI_k-t használjuk.

4. $FI_1(aba) = a$, $FI_2(aba) = ab$ és $FI_k(aba) = aba$ minden $k \geq 3$-ra.

Az $LL(k)$ nyelvtan definíciója

Definíció.
Legyen $k \geq 1$ egy egész szám. Azt mondjuk, hogy a G nyelvtan $LL(k)$, ha valahányszor teljesülnek az

- $S \Rightarrow_l^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx$
- $S \Rightarrow_l^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy$
- $FI_k(x) = FI_k(y)$

feltételek, mindannyiszor $\beta = \gamma$.

Észrevétel. Ha egy nyelvtan $LL(k)$, akkor az $LL(k+1)$ is.

Az $LL(k)$ feltétel átfogalmazása

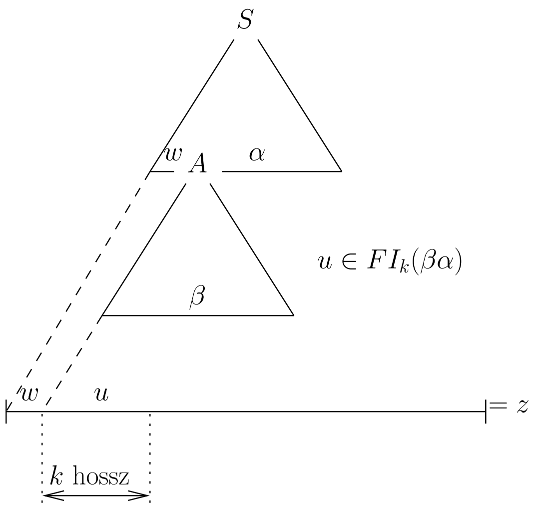
Tétel. G akkor és csakis akkor $LL(k)$, ha valahányszor $S \Rightarrow_l^* wA\alpha$ levezetés, $A \to \beta$ és $A \to \gamma$ pedig különböző P-beli szabályok, mindannyiszor $FI_k(\beta\alpha) \cap FI_k(\gamma\alpha) = \emptyset$.

<u>$LL(k)$ elemzés alapötlete.</u> Kérdés: $z \in L(G)$?

Tudjuk: $S \Rightarrow_l^* wA\alpha$ és $z = wz'$. Előrenézés: $u = FI_k(z')$.

Akkor A-t azzal az egyértelmű $A \to \beta$ alternatívával terjesztjük ki, melyre $u \in FI_k(\beta\alpha)$.

Ha nincs ilyen β, akkor $z \notin L(G)$.



Példa. G_{ar}: $K \to K+T$, $K \to T$,
$T \to T*F$, $T \to F$,
$F \to (K)$, $F \to a$

Nem $LL(1)$, mert

- $K \Rightarrow_l^* K \Rightarrow K+T \Rightarrow^* a+a$
- $K \Rightarrow_l^* K \Rightarrow T \Rightarrow^* a$
- $FI_1(a+a) = FI_1(a) = a$

mégis $K+T \neq T$.

A következőkben két kérdést szeretnénk tisztázni:

(1) Ha adott egy G nyelvtan és egy k szám akkor eldönthető-e, hogy G teljesíti-e az $LL(k)$ feltételt?

(2) Hogyan működik az $LL(k)$ elemzés?

Csak egy egyszerűbb esetre, az ún. erősen $LL(k)$ nyelvtanokra adjuk meg a válaszokat.

Mindkét kérdés megválaszolásához szükség van a $FI_k(\alpha)$-t kiszámító algoritmusra. Ehhez szükség lesz a következőkre.

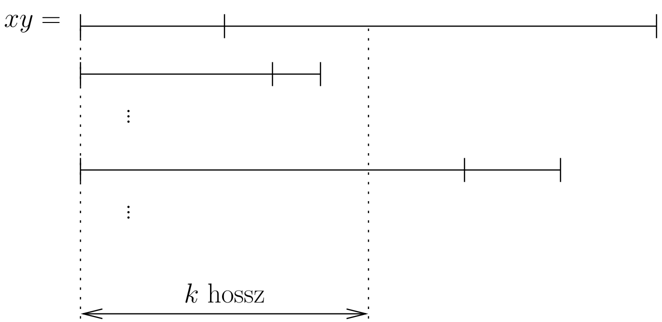
Definíció. Legyenek $L_1, L_2 \subseteq \Sigma^*$ nyelvek és $k \geq 0$. Akkor

$$L_1 \oplus_k L_2 = \{FI_k(xy) \mid x \in L_1, y \in L_2\}.$$

Példa. Ha $L_1 = \{\lambda, abb\}$ és $L_2 = \{b, bab\}$, akkor $L_1 \oplus_2 L_2 = \{b, ba, ab\}$.

Megjegyzés. $L_1 \oplus_k L_2$ mindig véges, mivel $L_1 \oplus_k L_2 \subseteq \Sigma^{*,k}$. Továbbá a $\oplus_k$ művelet asszociatív és könnyen igazolható, hogy minden α, β-ra

$$FI_k(\alpha\beta) = FI_k(\alpha) \oplus_k FI_k(\beta).$$



$FI_k(\alpha)$-t kiszámítása

Legyen $\alpha = X_1 \dots X_n$. Ekkor az előzőek miatt

$$FI_k(\alpha) = FI_k(X_1) \oplus_k \dots \oplus_k FI_k(X_n).$$

Mivel a $\oplus_k$ műveletet könnyű végrehajtani, elegendő csak $FI_k(X_i)$-t kiszámítani, ahol $X_i \in (N \cup \Sigma)$.

Sőt, a $FI_k(\alpha)$ definíciójából következik, hogy ha $X_i \in \Sigma$, akkor $FI_k(X_i) = X_i$.

Mindezt egybevetve, azt kapjuk, hogy amennyiben minden $A \in N$-re $FI_k(A)$-t ki tudjuk számolni, akkor minden α szóra $FI_k(\alpha)$-t is ki tudjuk számolni. Tehát elegendő $FI_k(A)$-val foglalkozni.

$FI_k(A)$ **halmazok kiszámítása.**

Input Egy G környezetfüggetlen nyelvtan.

Output Minden $A \in N$-re a $FI_k(A)$ halmaz.

Módszer $FI_k(A)$-t iteráljuk a $H_0(A), H_1(A), \ldots$ sorozattal.

Algoritmus

(1) Legyen minden $a \in \Sigma$ és $i \geq 0$ esetén $H_i(a) = \{a\}$.

(2) Legyen minden $A \in N$-re $H_0(A) =$
$\{x \in \Sigma^* \mid A \to x\alpha \in P$
ahol $(|x| = k)$ vagy $(|x| < k$ és $\alpha = \lambda)\}$ és legyen $i = 0$.

(3) Minden $A \in N$-re $H_0(A), \ldots, H_i(A)$ már ismertek.
Legyen $H_{i+1}(A) =$
$H_i(A) \cup \{x \in \Sigma^* \mid x \in H_i(X_1) \oplus_k \ldots \oplus_k H_i(X_n)$
valamely $A \to X_1 \ldots X_n \in P$ esetén $\}$.

(4) Ha minden $A \in N$-re $H_i(A) = H_{i+1}(A)$, akkor álljunk meg, különben legyen $i = i + 1$ és menjünk vissza (3)-ra.

Példa. G_{ar}: $K \to K + T$, $K \to T$,
$T \to T * F$, $T \to F$,
$F \to (K)$, $F \to a$

Számítsuk ki a $FI_1(K), FI_1(T)$ és $FI_1(F)$ halmazokat. A $H_0, H_1, \ldots$ közelítések egy táblázat soraiban ábrázolhatók a következő módon:

	K	T	F
H_0	$\emptyset$	$\emptyset$	$\{(, a\}$
H_1	$\emptyset$	$\{(, a\}$	$\{(, a\}$
H_2	$\{(, a\}$	$\{(, a\}$	$\{(, a\}$
H_3	$\{(, a\}$	$\{(, a\}$	$\{(, a\}$

Példa. Jelöljük G'_{ar}-ral azt a nyelvtant melynek szabályai:

1: $K \to TT'$
2: $T' \to +TT'$
3: $T' \to \lambda$
4: $T \to FF'$
5: $F' \to *FF'$
6: $F' \to \lambda$
7: $F \to (K)$
8: $F \to a$

G'_{ar} ugyancsak az aritmetikai kifejezéseket generálja, tehát $L(G_{ar}) = L(G'_{ar})$.

Számoljuk ki $FI_1(X)$-et minden $X \in \{K, T, T', F, F'\}$-re.

	K	T	T'	F	F'
H_0	$\emptyset$	$\emptyset$	$\{+, \lambda\}$	$\{(, a\}$	$\{*, \lambda\}$
H_1	$\emptyset$	$\{(, a\}$	$\{+, \lambda\}$	$\{(, a\}$	$\{*, \lambda\}$
H_2	$\{(, a\}$	$\{(, a\}$	$\{+, \lambda\}$	$\{(, a\}$	$\{*, \lambda\}$
H_3	$\{(, a\}$	$\{(, a\}$	$\{+, \lambda\}$	$\{(, a\}$	$\{*, \lambda\}$

Tehát $FI_1(K) = FI_1(T) = FI_1(F) = \{(, a\}$, $FI_1(T') = \{+, \lambda\}$ és $FI_1(F') = \{*, \lambda\}$.

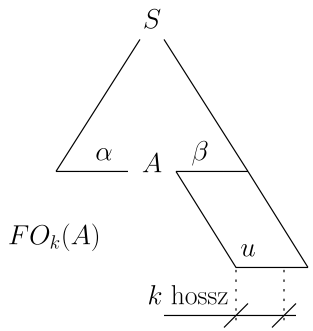
A $FOLLOW_k$ halmaz definíciója

Definíció. Legyen $A \in N$ és $k \geq 1$. Akkor

$$FOLLOW_k(A) = \bigcup \{FI_k(\beta) \mid S \Rightarrow^* \alpha A \beta, \text{ valamilyen } \alpha, \beta \in (N \cup \Sigma)^*\text{-ra}\}.$$

Az olyan terminális szavak FI_k-i, amelyek az A-t tartalmazó mondatformák (mint $\alpha A \beta$) A-t követő részéből (vagyis β-ból) vezethetők le.

A továbbiakban a $FOLLOW_k$ jelölés helyett a rövidebb FO_k-t használjuk.



A $FO_k(A)$ halmazok kiszámítása.

Input Egy G cf nyelvtan és egy $k \geq 1$ egész szám.

Output Minden $A \in N$-re $FO_k(A)$.

Módszer $FO_k(A)$-t iteráljuk a $H_0(A), H_1(A), \ldots$ sorozattal.

Algoritmus

(1) Legyen $i = 0$, $H_0(S) = \{\lambda\}$ és minden $A \neq S$-re legyen $H_0(A) = \emptyset$.

(2) Minden $A \in N$-re $H_0(A), \ldots, H_i(A)$ már ismertek. Legyen $H_{i+1}(A) =$
$H_i(A) \cup \{x \in \Sigma^* \mid x \in FI_k(\beta H_i(B))$ valamely $B \to \alpha A\beta$ szabály esetén $\}$.

(3) Ha minden $A \in N$-re $H_i(A) = H_{i+1}(A)$, akkor álljunk meg, különben legyen $i = i + 1$ és menjünk (2)-re.

Minden A-ra $FO_k(A) = H_i(A)$.

Példa. Jelöljük G'_{ar}-ral azt a nyelvtant melynek szabályai:

1: $K \to TT'$
2: $T' \to +TT'$
3: $T' \to \lambda$
4: $T \to FF'$
5: $F' \to *FF'$
6: $F' \to \lambda$
7: $F \to (K)$
8: $F \to a$

Számoljuk ki $FO_1(X)$-et minden $X \in \{K, T, T', F, F'\}$-re.

Számoljuk ki $FO_1(X)$-et minden $X \in \{K, T, T', F, F'\}$-re.

	K	T	T'	F	F'
H_0	$\{\lambda\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
H_1	$\{\lambda\}$	$\{+, \lambda\}$	$\{\lambda\}$	$\emptyset$	$\emptyset$
H_2	$\{\lambda\}$	$\{+, \lambda\}$	$\{\lambda\}$	$\{+, *, \lambda\}$	$\{+, \lambda\}$
H_3	$\{), \lambda\}$	$\{+, \lambda\}$	$\{\lambda\}$	$\{+, *, \lambda\}$	$\{+, \lambda\}$
H_4	$\{), \lambda\}$	$\{+,), \lambda\}$	$\{), \lambda\}$	$\{+, *, \lambda\}$	$\{+, \lambda\}$
H_5	$\{), \lambda\}$	$\{+,), \lambda\}$	$\{), \lambda\}$	$\{+, *,), \lambda\}$	$\{+,), \lambda\}$
H_6	$\{), \lambda\}$	$\{+,), \lambda\}$	$\{), \lambda\}$	$\{+, *,), \lambda\}$	$\{+,), \lambda\}$

Tehát $FO_1(K) = FO_1(T') = \{), \lambda\}$, $FO_1(T) = FO_1(F') = \{+,), \lambda\}$ és $FO_1(F) = \{+, *,), \lambda\}$.

Az erősen $LL(k)$ nyelvtan definíciója

Definíció. Legyen $k \geq 1$. Azt mondjuk, hogy G erősen $LL(k)$, ha tetszőleges $A \in N$ és $A \to \beta$, $A \to \gamma$ különböző szabályok esetén teljesül, hogy

$$FI_k(\beta FO_k(A)) \cap FI_k(\gamma FO_k(A)) = \emptyset,$$

(ahol $\beta FO_k(A) = \{\beta x \mid x \in FO_k(A)\}$.)

Tétel. Ha G erősen $LL(k)$, akkor $LL(k)$.

Bizonyítás. Indirekt: tfh G erősen $LL(k)$, de nem $LL(k)$.

Ha G nem $LL(k)$ akkor teljesülnek az

- $S \Rightarrow_l^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx$
- $S \Rightarrow_l^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy$
- $FI_k(x) = FI_k(y)$

feltételek, és mégis $\beta \neq \gamma$. Legyen $u = FI_k(x)$.

Mellesleg: $FI_k(\alpha) \subseteq FO_k(A)$.

A $FI_k(\alpha) \subseteq FO_k(A)$ feltétel miatt

$u = FI_k(x) \in FI_k(\beta\alpha) = FI_k(\beta) \oplus_k FI_k(\alpha) \subseteq$
$FI_k(\beta) \oplus_k FO_k(A) = FI_k(\beta FO_k(A))$.

Hasonlóan megmutatható, hogy $u \in FI_k(\gamma FO_k(A))$, tehát

$$u \in FI_k(\beta FO_k(A)) \cap FI_k(\gamma FO_k(A)) \neq \emptyset,$$

ami ellentmondás, mert G erősen $LL(k)$.

A következőkben két kérdést fogjuk tisztázni:

(1) Ha adott egy G nyelvtan és egy k szám akkor eldönthető-e, hogy G teljesíti-e az erősen $LL(k)$ feltételt?

(2) Hogyan működik az erősen $LL(k)$ elemzés?

Az erősen $LL(k)$ feltétel eldöntése.

Input Egy G környezetfüggetlen nyelvtan és egy $k \geq 1$ egész szám.

Output *Igen* ha G erősen $LL(k)$, különben *Nem*.

Algoritmus

(1) Válasszunk egy olyan A nemterminálist melynek legalább két alternatívája van.

(2) Válasszunk két különböző $A \to \beta$ és $A \to \gamma$ szabályt és számoljuk ki a $(FI_k(\beta FO_k(A))) \cap (FI_k(\gamma FO_k(A)))$ halmazt. Ha ez nemüres, akkor G nem erősen $LL(k)$, tehát álljunk meg és output: *Nem*.

(3) Ha választható A-nak újabb két alternatíva-párja, akkor ismételjük (2)-t.

(4) Ha választható újabb nemterminális, akkor ismételjük (1)-et.

(5) Adjunk outputra *Igen* jelzést.

Példa.

G'_{ar}-nak két olyan szabálypárja van, melyeknek bal oldala ugyanaz.

1) $T' \to +TT' \mid \lambda$: $FI_1(+TT'\,FO_1(T')) = \{+\}$ és $FI_1(\lambda\,FO_1(T')) = FI_1(\{\,),\lambda\}) = \{\,),\lambda\}$. Tehát

$FI_1(+TT'\,FO_1(T')) \cap FI_1(\lambda\,FO_1(T')) = \{+\} \cap \{\,),\lambda\} = \emptyset$.

2) $F' \to *FF' \mid \lambda$: $FI_1(*FF'\,FO_1(F')) = \{*\}$ és $FI_1(\lambda\,FO_1(F')) = FI_1(\{+,),\lambda\}) = \{+,),\lambda\}$. Tehát

$FI_1(*FF'FO_1(F')) \cap FI_1(\lambda FO_1(F')) = \{*\} \cap \{+,),\lambda\} = \emptyset$.

Következésképpen G'_{ar} erősen $LL(1)$.

Az $LL(1)$ eset.

Tétel. G akkor és csakis akkor erősen $LL(1)$, ha $LL(1)$.

Bizonyítás. $\Leftarrow$: Már beláttuk, hogy ha G erősen $LL(k)$, akkor $LL(k)$ is, minden k-ra.

$\Rightarrow$: Indirekt: tfh G nyelvtan $LL(1)$, de nem erősen $LL(1)$.

Van két olyan különböző $A \to \beta$ és $A \to \gamma$ szabály, (tehát $\beta \neq \gamma$) melyekre

$$FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A)) \neq \emptyset,$$

vagyis van olyan $a \in (\Sigma \cup \{\lambda\})$, hogy $a \in FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A))$.

(a) eset: $a \in \Sigma$. Ekkor az $a \in FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A))$ tartalmazás négyféleképpen valósulhat meg.

(a1) aleset: $a \in FI_1(\beta)$ és $a \in FI_1(\gamma)$. Ekkor léteznek az

- $S \Rightarrow_l^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wax'$
- $S \Rightarrow_l^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* way'$

levezetések, továbbá teljesül $FI_1(ax') = FI_1(ay') = a$. Ezért, mivel G nyelvtan $LL(1)$, $\beta = \gamma$-nak is teljesülnie kell, ami ellentmondás.

(a) eset: $a \in \Sigma$. Ekkor az $a \in FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A))$ tartalmazás négyféleképpen valósulhat meg.

(a2) aleset: $\beta \Rightarrow^* \lambda, a \in FO_1(A)$ és $a \in FI_1(\gamma)$. Ekkor léteznek az

- $S \Rightarrow_l^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* w\alpha \Rightarrow^* wax'$
- $S \Rightarrow_l^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* way'$

levezetések, továbbá teljesül $FI_1(ax') = FI_1(ay')$. Ezért, G nyelvtan $LL(1)$ volta miatt $\beta = \gamma$, ami ellentmondás.

(a) eset: $a \in \Sigma$. Ekkor az $a \in FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A))$ tartalmazás négyféleképpen valósulhat meg.

(a3) aleset: $a \in FI_1(\beta)$, $\gamma \Rightarrow^* \lambda$ és $a \in FO_1(A)$. Ennek az esetnek a bizonyítása az (a2)-höz hasonlóan történik.

(a4) aleset: $\beta \Rightarrow^* \lambda, a \in FO_1(A)$, továbbá $\gamma \Rightarrow^* \lambda$.

Hasonlóan.

(b) eset: $a = \lambda$. Ekkor a $\lambda \in FI_1(\beta FO_1(A)) \cap FI_1(\gamma FO_1(A))$ tartalmazás úgy valósulhat meg, hogy $\beta \Rightarrow^* \lambda, \gamma \Rightarrow^* \lambda$, továbbá $\lambda \in FO_1(A)$. Ezért léteznek az

- $S \Rightarrow_l^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* w\alpha \Rightarrow^* w$
- $S \Rightarrow_l^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* w\alpha \Rightarrow^* w$

levezetések, továbbá w után λ áll és teljesül, hogy $FI_1(\lambda) = FI_1(\lambda)$. Ezért, mivel G nyelvtan $LL(1)$ $\beta = \gamma$-nak is teljesülne kell, ami megint csak ellentmondás.

A tétel $k = 2$-re már nem igaz!

Az erősen $LL(k)$ elemzés alapötlete

Kérdés: $z \in L(G)$?

Tudjuk: $S \Rightarrow_l^* wA\alpha$ és $z = wz'$. Előrenézés: $u = FI_k(z')$.

Akkor A-t azzal az egyértelmű $A \rightarrow \beta$ alternatívával terjesztjük ki, melyre $u \in FI_k(\beta\alpha)$.

Mivel $FI_k(\alpha) \subseteq FO_k(A)$, annak is teljesülnie kell, hogy $u \in FI_k(\beta FO_k(A))$.

Az erősen $LL(K)$ definíciója szerint legfeljebb egy olyan szabály van, amire ez a tartalmazás teljesül, ha pedig nincs ilyen szabály, akkor $z \notin L(G)$.

A $FI_k(\beta FO_k(A))$ halmazok kiszámítása alapvető fontosságú!

Erősen $LL(k)$ nyelvtanok esetében egy M elemző tábla konstruálható a következőképpen.

M sorai az $N \cup \Sigma \cup \{\$\}$ halmaz elemeivel, oszlopai pedig a $\Sigma^{*,k}$ halmaz elemeivel vannak címkézve.

Ha $A \in N$, akkor M-nek az A sorhoz és $u \in \Sigma^{*,k}$ oszlophoz tartozó $M(A,u)$ eleme azt a tevékenységet írja le amit akkor kell végezni, ha az elemzés során A-t kell kiterjeszteni és u az előre nézett szó.

Ha $a \in \Sigma$, akkor M-nek az a sorhoz és az $u \in \Sigma^{*,k}$ oszlophoz tartozó $M(a,u)$ eleme azt adja meg, hogy mit kell csinálni, ha az előre nézett szó u és a-t kell illeszteni az elemzendő szóhoz.

M elemzőtábla definíciója:

(1) Minden $A \in N$ és $u \in \Sigma^{*,k}$ esetén, ha $u \in FI_k(\beta(FO_k(A))$ és $A \to \beta$ a j-edik szabály, akkor

$$M(A,u) = (\beta, j),$$

(2) Minden $a \in \Sigma$ és $u \in \Sigma^{*,k}$ esetén, ha $u = av$, akkor legyen $M(a,u)$=**pop**.

(3) Legyen $M(\$,\lambda)$=**elfogadás**,

(4) Minden más $X \in (N \cup \Sigma \cup \{\$\})$ és $u \in \Sigma^{*,k}$ esetén legyen $M(X,u)$=**hiba**.

Példa. G'_{ar} elemzőtáblája:

$M:$	a	$($	$)$	$+$	$*$	λ
K	$TT',1$	$TT',1$	**h**	**h**	**h**	**h**
T	$FF',4$	$FF',4$	**h**	**h**	**h**	**h**
T'	**h**	**h**	$\lambda,3$	$TT',2$	**h**	$\lambda,3$
F	$a,8$	$(K),7$	**h**	**h**	**h**	**h**
F'	**h**	**h**	$\lambda,6$	$\lambda,6$	$*FF',5$	$\lambda,6$
a	**p**	**h**	**h**	**h**	**h**	**h**
$($	**h**	**p**	**h**	**h**	**h**	**h**
$)$	**h**	**h**	**p**	**h**	**h**	**h**
$+$	**h**	**h**	**h**	**p**	**h**	**h**
$*$	**h**	**h**	**h**	**h**	**p**	**h**
$\$$	**h**	**h**	**h**	**h**	**h**	**e**

Erősen $LL(k)$ elemzési algoritmus.

Input G nyelvtan (ami erősen $LL(k)$) M elemző táblája és $z \in \Sigma^*$ szó.

Output *Igen* ha $z \in L(G)$, *Nem* különben.

Módszer

- Sorszámozzuk meg G szabályait.
- Az elemzés $(x, \alpha\$, \pi)$ alakú konfigurációk sorozata, ahol
  - $x \in \Sigma^*$ egy szó, az elemzendő szó hátralévő része,
  - $\alpha \in (N \cup \Sigma)^*$ egy verem, a levezetett bal mondatforma azon része, amely még további kiterjesztéseket és illesztéseket tartalmaz.
  - $\$$ az elemzendő szó végét jelző szimbólum, $\$ \notin (N \cup \Sigma)$.
  - π egy sor, szabályok sorszámaiból alkotott sorozat.
- Kezdő konfiguráció: $(z, S\$, \lambda)$. $z \in L(G)$ akkor és csakis akkor teljesül, ha $(z, S\$, \lambda) \vdash^* (\lambda, \$, \pi)$.

Az elemzési algoritmus

1. $C := (z, S\$, \lambda)$;
2. Amíg van olyan C', melyre $C \vdash C'$, legyen $C := C'$;
3. Ha $C = (\lambda, \$, \pi)$ alakú, valamely π-re, akkor output: *Igen* különben output: *Nem*.

Amennyiben az output *Igen*, π adja z egy bal oldali levezetését.

Átmeneti reláció Tegyük fel, hogy az $(x, X\alpha, \pi)$ konfigurációban vagyunk (ahol $X\alpha \in (N \cup \Sigma)^*\$$). Képezzük $u = FI_k(x)$-et.

(1) Ha $X = A \in N$ és $M(A, u) = (\beta, j)$, akkor $(x, X\alpha, \pi) \vdash (x, \beta\alpha, \pi j)$. (A verem tetején egy nemterminális állt amelyet kiterjesztettünk.)

(2) Ha $X = a \in \Sigma$ és $M(a, u) =$ **pop**, akkor $(x, X\alpha, \pi) \vdash (x', \alpha, \pi)$, ahol $x = ax'$. (Az illesztendő terminális illeszkedik az elemzendő szóhoz.)

(3) Ha $M(X, u) =$**elfogadás** (ami csak úgy lehet, ha $X = \$, \alpha = \lambda$ és $u = \lambda$), akkor nincs átmenet.

(4) Ha $M(X, u)$=**hiba**, akkor nincs átmenet.

Példa. Input G'_{ar} és a $z = a * (a + a)$ szó.

$(\underline{a} * (a + a), K\$, \lambda) \vdash (\underline{a} * (a + a), TT'\$, 1) \vdash$
$(\underline{a} * (a + a), FF'T'\$, 14) \vdash (\underline{a} * (a + a), aF'T'\$, 148) \vdash$
$(\underline{*}(a + a), F'T'\$, 148) \vdash (\underline{*}(a + a), *FF'T'\$, 1485) \vdash$
$(\underline{(}a + a), FF'T'\$, 1485) \vdash (\underline{(}a + a), (K)F'T'\$, 14857) \vdash$
...
$(\underline{\lambda}, F'T'\$, 14857148624863) \vdash (\underline{\lambda}, T'\$, 148571486248636) \vdash$
$(\underline{\lambda}, \$, 1485714862486363)$

Tehát $a * (a + a) \in L(G'_{ar})$ és 1485714862486363 a szó bal oldali levezetése.

Egy egyszerű példa $LL(1)$ nyelvtanra.

Definíció. *G egyszerű $LL(1)$*, ha λ-mentes és valahányszor $A \to \beta$ és $A \to \gamma$ két különböző szabály, mindannyiszor β és γ különböző terminális szimbólumokkal kezdődnek.

Egyszerű $LL(1)$ $\Rightarrow$ erősen $LL(1)$ $\Leftrightarrow$ $LL(1)$

Ha $A \to a\beta'$ és $A \to b\gamma'$ különböző szabályok, akkor $FI_1(a\beta' FO_1(A)) = FI_1(a\beta') = \{a\}$ és $FI_1(b\gamma'(FO_1(A)) = \{b\}$, ezért

$$FI_1(a\beta' FO_1(A)) \cap FI_1(a\gamma' FO_1(A)) = \emptyset.$$

Alulról felfelé haladó elemzés

Előkészület: ciklusmentesítés.

Definíció. Egy $G = (N, \Sigma, P, S)$ egy cf nyelvtan *ciklusmentes*, ha nincs olyan $A \in N$, melyre $A \Rightarrow^+ A$.

$A \Rightarrow^+ A \Rightarrow^+ A \Rightarrow^+ A \ldots$

Lemma. (Lásd tankönyv). Tetszőleges G cf nyelvtanhoz konstruálható olyan G' ciklusmentes cf nyelvtan, amelyre $L(G) = L(G')$.

Általános alulról felfelé elemzés

Adott $G = (N, \Sigma, P, S)$ cf nyelvtan és $w \in \Sigma^*$ szó, igaz-e, hogy $w \in L(G)$

Csak λ-mentes és ciklusmentes nyelvtanokkal foglalkozunk.

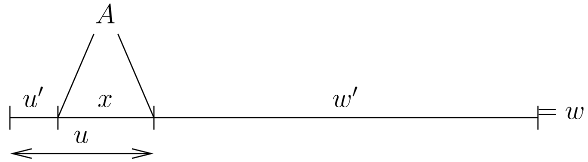
A w szóból kiindulva próbálunk meg felépíteni egy olyan derivációs fát aminek gyökere az S kezdőszimbólum.

Két művelet:

Shiftelés: egy szimbólummal tovább olvassuk az input szót (ami kezdetben w).

Redukálás: egy szabály jobb oldalát a bal oldalával helyettesítjük.

redukálás



Redukálás az $A \to x$ szabály szerint.

Az általános alulról felfelé haladó elemzés alapötlete.

Olvassuk w-t az elejéről (vagyis shifteljünk) addig, amíg az elolvasott rész jobb oldali végén ki nem alakul egy szabály jobb oldala. Tehát legyen u olyan, amire teljesül, hogy $w = uw'$ és van olyan u' és x, hogy $u = u'x$ és $A \to x \in P$. Helyettesítsük x-et a szabály A bal oldalával, vagyis redukáljunk. Eredményül kapjuk az $u'Aw'$ szót, amelyre $u'Aw' \Rightarrow w$

az $u'Aw'$ szóra végezzük ezt tovább úgy, hogy w' elejétől kezdve annak betűit hozzá shifteljük $u'A$-hoz addig, amíg az így kapott szó jobb oldali végén megint ki nem alakul egy szabály jobb oldala. Ha kialakult akkor redukálunk. Így próbáljuk meg elérni S-et.

Két probléma.

Shiftelés-redukálás konfliktus: Ha addig shifteltünk, hogy elértünk egy olyan pontot ahol redukálni is lehet, akkor nem tudjuk, hogy elvégezzük-e a redukálást vagy tovább shifteljünk egy későbbi olyan redukálás reményében, amitől azt várjuk, hogy sikeresen elvezet S-hez.

Redukálás-redukálás konfliktus: Ha elértünk egy olyan pontot, ahol redukálni lehet, akkor több olyan szabály is lehet, ami alkalmas a redukálásra. Ekkor nem tudjuk eldönteni, hogy melyik szabály szerint redukáljunk.

Megoldás.

Rögzítsük le a P-beli szabályok egy sorrendjét valamilyen módon. Ha shiftelés-redukálás konfliktussal találkozunk akkor végezzünk mindig redukálást. Továbbá, ha redukálás-redukálás konfliktusba ütközünk, akkor mindig a kisebb sorszámú szabály szerint redukáljunk.

Mivel a szabályok általunk felállított sorrendje önkényes, nem biztos, hogy a fenti elv mindig a helyes irányba visz. Mégsem mondhatjuk, hogy ekkor $w \notin L(G)$, mivel nem vettünk figyelembe minden lehetőséget. Ezért (csakúgy mint a top-down elemzésnél) fenn kell tartanunk a visszalépés, vagyis a backtrack lehetőségét.

Alulról felfelé haladó általános elemzési algoritmus.

Input Egy $G = (N, \Sigma, P, S)$ ciklusmentes és λ-mentes környezetfüggetlen nyelvtan és egy $w = a_1 \dots a_n \in \Sigma^*, n \geq 1$ szó.

Output *Igen* jelzés és a w szónak egy jobb oldali levezetése ha $w \in L(G)$. Különben *Nem* jelzés.

Módszer

- Rögzítsük le a szabályok egy sorrendjét.
- Az elemzés (p, i, α, β) alakú konfigurációk sorozata, ahol

- $p \in \{q, b, t\}$, normál, elfogadó és b=backtrack állapot. Kezdő értéke: q.

- i az input szóba mutató pointer, $1 \leq i \leq n+1$.

- α egy verem, teteje a jobb végén van. Tartalma az input szó elolvasott részéből redukálásokkal keletkezett $(N \cup \Sigma)^*$-beli szó. Kezdőértéke: λ.

- β egy verem, teteje a bal végén van. Tartalma az elemzés "története" vagyis a végrehajtott shiftelések és redukálások sorozata. Kezdőértéke: λ.

• A konfigurációk halmazán megadunk egy $\vdash$ átmeneti relációt: $(p, i, \alpha, \beta) \vdash (p', i', \alpha', \beta')$ determinisztikus reláció. Az átmeneti relációt definiáló rész (1) lépésétől elindulva az első olyan lépést alkalmazzuk, ami alkalmazható a konfigurációra.

• Kezdő konfiguráció: $(q, 1, \lambda, \lambda)$. A befejező konfigurációk alakja $(t, n+1, S, \gamma)$ ahol γ egy s (shift) szimbólumból és szabályok sorszámaiból képzett sorozat. Az teljesül, hogy $w \in L(G)$ akkor és csakis akkor, ha $(q, 1, \lambda, \lambda) \vdash^* (t, n+1, S, \gamma)$.

Az elemzési algoritmus

(0) *Kezdő konfiguráció beállítása:* Legyen $C := (q, 1, \lambda, \lambda)$.

(1) *Redukálás:* Amíg $C = (q, i, \alpha\gamma, \beta)$ alakú és van $A \to \gamma$ alakú szabály, addig $C \vdash C'$, ahol $C' = (q, i, \alpha A, j\beta)$ és j az $A \to \gamma$ szabály sorszáma, és a j-edik az első olyan szabály, amely szerint redukálás hajtható végre.

(Ha már nem lehet redukálni, akkor továbblépünk.)

(2) *Shiftelés:* Ha $C = (q, i, \alpha, \beta)$ és $i \neq n+1$, akkor $C \vdash C'$, ahol $C' = (q, i+1, \alpha a_i, s\beta)$ (a_i pedig az elemzendő szó i-edik betűje). Menjünk (1)-re (mert ha lehet, akkor redukálunk).

(3) *Elfogadás:* Ha $C = (q, n+1, S, \beta)$, akkor $C \vdash C' = (t, n+1, S, \beta)$. Az algoritmus álljon le és adjon ki *Igen* jelzést.

(4) *Átmenet backtrack állapotba:* Ha $C = (q, n+1, \alpha, \beta)$ de $\alpha \neq S$, akkor (nem értük el S-et tehát) $C \vdash C'$, ahol $C' = (b, n+1, \alpha, \beta)$. Menjünk (5)-re.

(5) Backtrack végrehajtása: (Egymást kizáró esetek.)

(i) Ha $C = (b, i, \alpha A, j\beta)$, a j-edik szabály $A \to \gamma$ és az $\alpha\gamma$ (vagyis a visszaállított szó) redukálható egy további $B \to \gamma'$ szabály szerint is (vagyis $\alpha\gamma = \alpha'\gamma'$), akkor $C \vdash C'$, ahol $C' = (q, i, \alpha' B, k\beta)$, és $k > j$ a $B \to \gamma'$ szabály sorszáma (az általunk rögzített sorrendben) és a j-edik után ez a legelső olyan szabály, amely szerint $\alpha\gamma$ szón redukálás hajtható végre.

Menjünk (1)-re (mert ha lehet, akkor redukálunk).

(5) Backtrack végrehajtása: (Egymást kizáró esetek.)

(ii) Ha $C = (b, i, \alpha A, j\beta)$, a j-edik szabály $A \to \gamma$ és $i \neq n+1$ de $\alpha\gamma$ nem redukálható más szabállyal, akkor $C \vdash C' = (q, i+1, \alpha\gamma a_i, s\beta)$.

Menjünk (1)-re (mert ha lehet, akkor redukálunk).

(iii) Ha $C = (b, n+1, \alpha A, j\beta)$ a j-edik szabály $A \to \gamma$ de $\alpha\gamma$ nem redukálható más szabállyal és már shiftelni sem tudunk akkor $C \vdash C' = (b, n+1, \alpha\gamma, \beta)$. Menjünk (5)-re.

(iv) Ha $C = (b, i, \alpha a, s\beta)$ és $i > 1$, akkor $C \vdash C' = (b, i-1, \alpha, \beta)$. Menjünk (5)-re.

(v) Ha egyik feltétel sem teljesül akkor $w \notin L(G)$, az algoritmus álljon le és adjon *Nem* jelzést outputra.

Példa. G_e: $1 : K \to K + T \quad 2 : K \to T$
$3 : T \to a \quad 4 : T \to b.$

Elemezzük a $w = b + a$ szót.

$(q, 1, \lambda, \lambda) \vdash (q, 2, b, s) \vdash (q, 2, T, 4s) \vdash (q, 2, K, 24s) \vdash$
$(q, 3, K+, s24s) \vdash (q, 4, K + a, ss24s) \vdash (q, 4, K + T, 3ss24s) \vdash$
$(q, 4, K, 13ss24s) \vdash (t, 4, K, 13ss24s)$,

Tehát $b + a \in L(G_e)$. Továbbá $13ss24s$-ből törölve az s-eket, az 1324 szabály sorszám sorozatot kapjuk, vagyis azon szabályok sorszámainak sorozatát, melyek $b + a$ jobb oldali levezetését adják.

$LR(k)$ nyelvtanok és elemzésük

$G = (N, \Sigma, P, S)$ egy tetszőleges cf nyelvtan.

Az $LR(k)$ nyelvtanok, ahol $k \geq 0$ most is egy egész szám, olyan speciális környezetfüggetlen nyelvtanok, melyek esetében az alulról felfelé elemzéskor megnézzük a jobb oldali mondatforma még feldolgozatlan részének k hosszúságú prefixét és ennek segítségével fel tudjuk oldani mind a shiftelés-redukálás mind a redukálás-redukálás konfliktust.

Technikai okok miatt feltesszük, hogy S-nek csak egyetlen alternatívája van és S nem szerepel egyetlen szabály jobb oldalán sem.

Definíció. Legyen γ jobb mondatforma, $\gamma \neq S$. Azt mondjuk, hogy β a γ nyele, ha van olyan $A \to \beta$ szabály, melyre $S \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta w = \gamma$.

Az elemzés szempontjából fontos a nyél meghatározása: ha minden γ jobb mondatformának meg tudnánk határozni egy nyelét, akkor minden jobb mondatformát tudnánk redukálni is. Mivel ekkor újra jobb mondatformát kapunk, el tudnánk dönteni, hogy egy $z \in \Sigma^*$ szó redukálható-e (több lépésben) S-re, vagyis teljesül-e $z \in L(G)$.

Az $LR(k)$ nyelvtan definíciója

Definíció. Legyen $k \geq 0$ egy egész szám. Azt mondjuk, hogy a G nyelvtan $LR(k)$, ha valahányszor teljesülnek az

- $S \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta w$
- $S \Rightarrow_r^* \gamma B x \Rightarrow_r \gamma\delta x = \alpha\beta y$
- $FI_k(w) = FI_k(y)$

feltételek, mindannyiszor $\alpha = \gamma, A = B$ és $\beta = \delta$.

Észrevétel. Ha egy nyelvtan $LR(k)$, akkor az $LR(k+1)$ is.

Tfh, hogy G $LR(k)$ és el akarjuk dönteni, hogy $z \in L(G)$ teljesül-e. Tfh, azt már tudjuk, hogy $\omega \Rightarrow_r^* z$. Ekkor az ω szónak nem lehet két különböző nyele. Ha ugyanis egyrészt $\omega = \alpha\beta w$, másrészt $\omega = \gamma\delta x$, továbbá vannak olyan $A \rightarrow \beta$ és $B \rightarrow \delta$ szabályok melyekre

- $S \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta w$ és
- $S \Rightarrow_r^* \gamma B x \Rightarrow_r \gamma\delta x = \alpha\beta w$,

akkor $FI_k(w) = FI_k(w)$, és az $LR(k)$ feltétel miatt azt kapjuk, hogy $\alpha = \gamma$, $A = B$ és $\beta = \delta$, tehát mind a nyél, mind a redukció egyértelműen meghatározott.

A továbbiakban azt is látni fogjuk, hogy az $LR(k)$ feltétel szintén feloldja a shiftelés-redukálás konfliktust is.

Az $LR(k)$ nyelvtanok esetében is a következő két kérdés megválaszolására koncentrálunk:

(1) Ha adott egy G nyelvtan és egy k szám, akkor eldönthető-e, hogy G nyelvtan $LR(k)$-e?

(2) Hogyan működik az $LR(k)$ elemzés?

Mindkét kérdés megválaszolásához szükségünk van az $LR(k)$ feltétel egy olyan átfogalmazására, amely gyakorlati szempontból jobban használható. Ehhez azonban szükségünk van a következő fogalmakra.

Definíció. Egy $\gamma \in (N \cup \Sigma)^*$ *járható prefix*, ha van olyan $\alpha\beta w$ jobb mondatforma melynek nyele β és teljesül rá, hogy γ prefixe $\alpha\beta$-nak.

A járható prefix egy olyan szó, amely prefixe egy jobb mondatformának, de nem nyúlik túl a nyél jobb oldali végén.

Az elemzés során lényegében járható prefixeket állítunk elő. Célunk az, hogy a járható prefixekhez addig shifteljünk újabb szimbólumokat, amíg el nem érjük a nyél jobb oldali végét, vagyis ki nem alakul a nyél. A definíció szerint az így kapott szó is járható prefix, ami azonban már redukálható.

Definíció. Az $[A \to \beta_1.\beta_2, u]$ alakú párokat $LR(k)$ *elemeknek* nevezzük, ahol $A \to \beta_1\beta_2 \in P$ és $u \in \Sigma^{*,k}$.) (A $\beta_1 = \lambda$ és/vagy a $\beta_2 = \lambda$ is lehetséges.)

Definíció. Az $[A \to \beta_1.\beta_2, u]$ $LR(k)$ elem *érvényes az $\alpha\beta_1$ járható prefixre*, ha létezik $S \Rightarrow_r^* \alpha A w \Rightarrow \alpha\beta_1\beta_2 w$ deriváció úgy, hogy $u = FI_k(w)$.

Definíció. Az $[A \to \beta_1.\beta_2, u]$ $LR(k)$ elem *érvényes az $\alpha\beta_1$ járható prefixre*, ha létezik $S \Rightarrow_r^* \alpha A w \Rightarrow \alpha\beta_1\beta_2 w$ deriváció úgy, hogy $u = FI_k(w)$.

Annak, hogy egy $LR(k)$ elem érvényes egy jártható prefixre, a szemléletes jelentése a következő:

- Ha az $LR(k)$ elem $[A \to \beta_1.\beta_2, u]$ alakú, ahol $\beta_2 \neq \lambda$, akkor az $A \to \beta_1\beta_2$ szabály szóba jöhet egy későbbi redukálásnál, mert ezen szabály jobb oldalának „." előtti része (vagyis β_1) már megegyezik a jártható prefix „végével".

- Ha az $LR(k)$ elem $[A \to \beta., u]$ alakú, akkor az $A \to \beta$ szabály szóba jöhet a játható prefix redukálásánál.

Adott γ járható prefix esetén a γ-ra érvényes $LR(k)$ elemek halmazát $V_k(\gamma)$-val jelöljük. Minden γ-ra $V_k(\gamma)$ véges halmaz. A $V_k(\gamma)$-kat $LR(k)$ ***tábláknak*** hívjuk (mely elnevezés onnan adódik, hogy egy $V_k(\gamma)$ halmazban lévő járható prefixeket egymás alá írva táblázatszerű elrendezést kapunk).

Bevezetjük a

$$\mathcal{T} = \{V_k(\gamma) \mid \gamma \text{ járható prefix}\}$$

jelölést.

Ismét megjegyezzük, hogy – bár általában végtelen sok γ járható prefix van – $\mathcal{T}$ is véges halmaz, ugyanis részhalmaza az $LR(k)$ elemek halmaza hatványhalmazának, ami megint csak véges.

$V_k(\gamma)$ **meghatározása.**

<u>*Input*</u> G nyelvtan és $\gamma = X_1 \ldots X_n$ szó, ahol $X_1, \ldots, X_n \in (N \cup \Sigma)$.

<u>*Output*</u> $V_k(\gamma)$ halmaz.

<u>*Módszer*</u> Kiszámoljuk a $V_k(\lambda), V_k(X_1), \ldots, V_k(X_1 \ldots X_n)$ halmazokat.

Algoritmus

(a) $V_k(\lambda)$ kiszámítása

(1) (Inicializálás.) Minden $S \to \alpha \in P$ esetén legyen $[S \to .\alpha, \lambda] \in V_k(\lambda)$.

(2) (Lezárás.) Mindaddig, amíg $V_k(\lambda)$ bővíthető, a $V_k(\lambda)$ minden $[A \to .B\beta, u]$ alakú elemére és $B \to \delta$ P-beli szabályra, legyen $[B \to .\delta, v] \in V_k(\lambda)$ minden $v \in FI_k(\beta u)$-ra.

Algoritmus

(b) Tegyük fel, hogy $V_k(X_1 \ldots X_{i-1})$-et már kiszámítottuk. Ekkor kiszámítjuk $V_k(X_1 \ldots X_i)$-t.

(1) (Léptetés.) Minden $[A \to \alpha.X_i\beta, u]$ alakú $V_k(X_1 \ldots X_{i-1})$-beli elem esetén legyen $[A \to \alpha X_i.\beta, u] \in V_k(X_1 \ldots X_i)$.

(2) (Lezárás.) Mindaddig, amíg $V_k(X_1 \ldots X_i)$ bővíthető, a $V_k(X_1 \ldots X_i)$ minden $[A \to \alpha.B\beta, u]$ alakú elemére és $B \to \delta$ P-beli szabályra, legyen $[B \to .\delta, v] \in V_k(X_1 \ldots X_i)$ minden $v \in FI_k(\beta u)$-ra.

Példa. ($LR(1)$ táblák) G_{list}

$$\begin{array}{lcl} S & \to & L \\ L & \to & L * E \,|\, E \\ E & \to & a \,|\, b \end{array}$$

$$\begin{array}{lcl} T_0 = V_1(\lambda) & = & [S \to .L, \lambda] \\ & & [L \to .L * E, \lambda \,/\, *] \\ & & [L \to .E, \lambda \,/\, *] \\ & & [E \to .a, \lambda \,/\, *] \\ & & [E \to .b, \lambda \,/\, *] \end{array}$$

$$\begin{array}{lcl} T_1 = V_1(L) & = & [S \to L., \lambda] \\ & & [L \to L. * E, \lambda \,/\, *] \end{array}$$

$\mathcal{T} = \{V_k(\gamma) \,|\, \gamma$ járható prefix $\}$ **kiszámítása.**

Input G nyelvtan.

Output $\mathcal{T} = \{V_k(\gamma) \,|\, \gamma$ járható prefix $\}$.

Algoritmus

(1) Legyen $\mathcal{T} = \{V_k(\lambda)\}$ ($V_k(\gamma)$ Algoritmus (a) pont).

(2) Mindaddig amíg $\mathcal{T}$-ben van megjelöletlen táblázat tegyük a következőt:

- Vegyünk egy megjelöletlen $T \in \mathcal{T}$ elemet. Legyen ez $T = V_k(\gamma)$.
- Jelöljük meg T-t.
- Minden $X \in (N \cup \Sigma)$-ra számoljuk ki $T' = V_k(\gamma X)$-et ($V_k(\gamma)$ Algoritmus (b) pont). Ha $T' \notin \mathcal{T}$ akkor T'-t vegyük fel $\mathcal{T}$-be, különben lépjünk tovább.

Mivel az $LR(k)$ táblák halmaza (vagyis $\mathcal{T}$) véges, a fenti algoritmus véges számú lépés után terminál.

Példa. ($LR(1)$ táblák) G_{list}:

$$\begin{aligned} S &\rightarrow L \\ L &\rightarrow L * E \mid E \\ E &\rightarrow a \mid b \end{aligned}$$

$$\begin{aligned} T_0 = V_1(\lambda) \;=\; & [S \rightarrow .L, \lambda] \\ & [L \rightarrow .L * E, \lambda / *] \\ & [L \rightarrow .E, \lambda / *] \\ & [E \rightarrow .a, \lambda / *] \\ & [E \rightarrow .b, \lambda / *] \end{aligned}$$

$$\begin{aligned} T_1 = V_1(L) \;=\; & [S \rightarrow L., \lambda] \\ & [L \rightarrow L. * E, \lambda / *] \end{aligned}$$

$$T_2 = V_1(E) \;=\; [L \rightarrow E., \lambda / *]$$

$$T_3 = V_1(a) \;=\; [E \rightarrow a., \lambda / *]$$

$$T_4 = V_1(b) \;=\; [E \rightarrow b., \lambda / *]$$

$$\begin{aligned} T_5 = V_1(L*) \;=\; & [L \rightarrow L * .E, \lambda / *] \\ & [E \rightarrow .a, \lambda / *] \\ & [E \rightarrow .b, \lambda / *] \end{aligned}$$

$$T_6 = V_1(L * E) \;=\; [L \rightarrow L * E., \lambda / *]$$

Az $LR(k)$ tétel átfogalmazása.

Definíció. Tetszőleges α szóra
$EFF_k(\alpha) = \{FI_k(x) \mid \alpha \Rightarrow_r^* \beta \Rightarrow_r x$ úgy, hogy $\beta \neq Ax$ semmilyen $A \in N$-re$\}$.

Tétel. ($LR(k)$ tétel.) G akkor és csakis akkor $LR(k)$, ha minden $T \in \mathcal{T}$ $LR(k)$ táblára igaz, hogy ha T-ben van $[A \rightarrow \beta., u]$ alakú elem, akkor nincs benne egy olyan másik $[B \rightarrow \beta_1.\beta_2, v]$ elem, melyre $u \in EFF_k(\beta_2 v)$. (Ezt röviden úgy fejezzük ki, hogy minden $LR(k)$ tábla konzisztens.)

Tétel. G-ről eldönthető, hogy $LR(k)$-e.

Bizonyítás. Számoljuk ki $\mathcal{T}$-t, majd vegyük sorra $\mathcal{T}$-nek minden T elemét. Amennyiben T-ben nincs $[A \to \beta., u]$ alakú elem, válasszuk a következő táblázatot.

Ha T-ben találunk $[A \to \beta., u]$ alakú elemet, akkor T minden $[B \to \beta_1.\beta_2, v]$ alakú elemére vizsgáljuk meg, hogy $u \in EFF_k(\beta_2 v)$ teljesül-e. (Itt kihasználtuk, hogy $EFF_k(\beta_2 v)$ kiszámítható.) Ha találunk ilyen elemet T-ben, akkor G nem $LR(k)$. Ha egyetlen T táblázat esetén sem találunk két olyan elemet, amelyek kielégítik ezt a feltételt, akkor G $LR(k)$.

Példa. G_{list} $LR(1)$:

$$T_0 = V_1(\lambda) \;=\; \begin{array}{l} [S \to .L, \lambda] \\ [L \to .L * E, \lambda \,/\, *] \\ [L \to .E, \lambda \,/\, *] \\ [E \to .a, \lambda \,/\, *] \\ [E \to .b, \lambda \,/\, *] \end{array}$$

Ebben nincs $[A \to \beta., u]$ alakú elem, tehát nem lehet konfliktus.

Példa. G_{list} $LR(1)$:

$$T_1 = V_1(L) \;=\; \begin{array}{l} [S \to L., \lambda] \\ [L \to L. * E, \lambda \,/\, *] \end{array}$$

Az $[S \to L., \lambda]$-t és az $[L \to L.*E, \lambda \,/\, *]$ elemek konfliktusban lehetnek. De $EFF_1(*E\lambda) = EFF_1(*E*) = \{*\}$ és $\lambda \notin \{*\}$, így az $LR(1)$ feltétel megint csak nem sérül.

Hasonlóan látható a $T_2, \ldots, T_6$ táblák esetén is.

Az $LR(k)$ elemzés működése (Tfh G $LR(k)$)

Megkonstruáljuk az $LR(k)$ *elemző táblát*, amely két részre osztható: az f *tevékenység táblára* és a g *goto táblára*. Mindkét tábla sorai $\mathcal{T}$ elemeivel vannak címkézve.

Az f tevékenység tábla oszlopai $\Sigma^{*,k}$ elemeivel vannak címkézve. A $T \in \mathcal{T}$ sorhoz és az $u \in \Sigma^{*,k}$ oszlophoz tartozó $f(T,u)$ sor azt írja le, hogy mit kell tenni, ha a vizsgált járható prefixre érvényes $LR(k)$ elemek halmaza T, az előrenézett szó pedig u.

A g goto tábla oszlopai $N \cup \Sigma$ elemeivel vannak címkézve. A $T = V_k(\gamma)$ sor és az X oszlop által meghatározott elem $V_k(\gamma X)$, tehát $g(T,X) = V_k(\gamma X)$.

$f(T, u)$ pontos definíciója.

(1) Ha T-ben van $[A \to \beta., u]$ alakú elem, akkor legyen $f(T, u) =$(**redukálás**, j), ahol j az $A \to \beta$ szabály sorszáma.
(2) ha T-ben van $[A \to \beta_1.\beta_2, v]$ alakú elem, ahol $\beta_2 \neq \lambda$ és $u \in EFF_k(\beta_2 v)$, akkor legyen $f(T, u) =$**shift**.
(3) Ha $u = \lambda$ és T-ben van $[S \to \alpha., \lambda]$ alakú elem, akkor $f(T, u) =$**elfogadás**.
(4) Minden más esetben legyen $f(T, u) =$**hiba**.

A g goto tábla oszlopai $N \cup \Sigma$ elemeivel vannak címkézve. A $T = V_k(\gamma)$ sor és az X oszlop által meghatározott elem $V_k(\gamma X)$, tehát $g(T, X) = V_k(\gamma X)$.

Példa. G_{list}: 1: $S \to L$
2: $L \to L * E$ 3: $L \to E$
4: $E \to a$ 4: $E \to b$ tevékenység táblája:

$f:$	a	b	$*$	λ
T_0	**s**	**s**	**h**	**h**
T_1	**h**	**h**	**s**	**e**
T_2	**h**	**h**	**r** 3	**r** 3
T_3	**h**	**h**	**r** 4	**r** 4
T_4	**h**	**h**	**r** 5	**r** 5
T_5	**s**	**s**	**h**	**h**
T_6	**h**	**h**	**r** 2	**r** 2

$g:$	S	L	E	a	b	$*$
T_0		T_1	T_2	T_3	T_4	
T_1						T_5
T_2						
T_3						
T_4						
T_4			T_6	T_3	T_4	
T_6						

$LR(k)$ **elemzési algoritmus**

Input G ($LR(k)$) nyelvtan $LR(k)$ elemző (tevékenység és goto) táblája és $w \in \Sigma^*$ szó.

Output *Igen* ha $w \in L(G)$, *Nem* különben.

Módszer

- Az elemzés $(x, \alpha T, \pi)$ alakú konfigurációk sorozata, ahol

- $x \in \Sigma^*$ egy szó, az elemzendő szó hátralévő (vagyis a járható prefixen túli) része,

- $\alpha T \in (\mathcal{T} \cup N \cup \Sigma)^*$, ahol α tartalmazza a járható prefixet úgy, hogy ennek minden prefixe után α-ban benne van a rá érvényes $LR(k)$ tábla is. (Így T érvényes α'-re, ahol α'-t úgy kapjuk α-ból, hogy töröljük belőle a $\mathcal{T}$-beli szimbólumokat.) Egy veremnek tekintjük, melynek teteje a jobb végén van.

- π egy, a szabályok sorszámaiból alkotott sorozat. Egy sornak tekintjük, melynek teteje a bal végén van.

Módszer

- Kezdő konfiguráció: (w, T_0, λ), ahol $T_0 = V_k(\lambda)$ (vagyis a λ járható prefixre érvényes $LR(k)$ tábla). A befejező konfigurációk $(\lambda, T_0\alpha T, \pi)$ alakúak.

A konfigurációk közötti átmenetet $\vdash$-val jelöljük.

Az átmeneti reláció úgy van definiálva, hogy $w \in L(G)$ akkor és csakis akkor teljesül, ha a kezdő konfigurációból elértünk egy olyan $(\lambda, T_0\alpha T, \pi)$ konfigurációt, melyre $f(T, \lambda) =$**elfogadás**. (Ekkor α-ból elhagyva az $LR(k)$ tábla szimbólumokat, az S egyetlen alternatíváját kapjuk.)

Az elemzési algoritmus

1. $C := (w, T_0, \lambda)$;
2. Amíg van olyan C', melyre $C \vdash C'$, legyen $C := C'$;
3. Ha $C = (\lambda, T_0\alpha T, \pi)$ alakú, valamely π-re és $f(T, \lambda) =$**elfogadás**, akkor output: *Igen* különben output: *Nem*.

Amennyiben az output *Igen*, akkor 1π adja w egy jobb oldali levezetését, ahol 1 az egyetlen S bal oldalú szabály sorszáma.

Átmeneti reláció Tfh az $(x, \alpha T, \pi)$ konfigurációban vagyunk. Képezzük $u = FI_k(x)$-et.

(1) Ha $f(T, u) = ($**redukálás**$, j)$, akkor αT veremből törlünk $2 \cdot |\beta|$ darab szimbólumot, ahol $A \to \beta$ a j-edik szabály. Tegyük fel, hogy a veremben marad $\gamma T'$. Akkor $(x, \alpha T, \pi) \vdash (x, \gamma T' A T'', j\pi)$ ahol $T'' = g(T', A)$.

(2) Ha $f(T, u) =$**shift**, akkor $(x, \alpha T, \pi) \vdash (x', \alpha T a T', \pi)$, ahol $x = ax'$ és $T' = g(T, a)$.

(3) Ha $f(T, u) =$**elfogadás** (ami csak úgy lehet, ha T-ben van $[S \to \alpha., \lambda]$ alakú elem és $u = \lambda$ egyidejűleg teljesülnek) akkor nincs átmenet.

(4) Ha $f(T, u)$=**hiba**, akkor nincs átmenet.

Tekintsük a G_{list} nyelvtant és a $w = a * b * a$ szót. Adjuk meg szó elemzését.

$$
\begin{aligned}
&(a * b * a, T_0, \lambda) \vdash (*b * a, T_0 a T_3, \lambda) \vdash (*b * a, T_0 E T_2, 4) \vdash \\
&(*b * a, T_0 L T_1, 34) \vdash (b * a, T_0 L T_1 * T_5, 34) \vdash \\
&(*a, T_0 L T_1 * T_5 b T_4, 34) \vdash (*a, T_0 L T_1 * T_5 E T_6, 534) \vdash \\
&(*a, T_0 L T_1, 2534) \vdash (a, T_0 L T_1 * T_5, 2534) \vdash \\
&(\lambda, T_0 L T_1 * T_5 a T_3, 2534) \vdash \\
&(\lambda, T_0 L T_1 * T_5 E T_6, 42534) \vdash (\lambda, T_0 L T_1, 242534)
\end{aligned}
$$

Mivel $f(T_1, \lambda) =$**elfogadás**, az $a * b * a$ szó eleme $L(G_{list})$ nyelvnek. Továbbá, az 1242534 szabály sorszám sorozat ezen szó jobb oldali levezetését adja.

Összefüggések $LL(k)$ és $LR(k)$ nyelvek között

Tétel. Ha egy nyelvtan $LL(k)$, akkor az $LR(k)$ is.

Egy L nyelvet $LL(k)$-nak (hasonlóan $LR(k)$-nak) mondunk, ha van olyan G $LL(k)$ (hasonlóan $LR(k)$) nyelvtan, melyre $L = L(G)$.

Tétel. Minden $LR(k)$ nyelv generálható $LR(1)$ nyelvtannal is. Továbbá, az $LR(1)$ nyelvek osztálya megegyezik a determinisztikus nyelvek osztályával.

Precedencia nyelvtanok és elemzésük

Definíció. Egy $G = (N, \Sigma, P, S)$ nyelvtan *egyértelműen redukálható*, ha valahányszor $A \to \beta$ és $B \to \beta$ P-beli szabályok, mindannyiszor $A = B$ teljesül.

Egyszerű precedencia nyelvtanok

Egyértelműen redukálható, továbbá a nyelvtani szimbólumok (terminálisok, nemterminálisok) között fennállnak a $\prec, \doteq$ és $\succ$ *precedencia relációk* melyek intuitív jelentése a következő. Legyen $\alpha\beta w$ egy jobb oldali mondatforma, melynek a nyele β. Akkor

- α bármely két betűje között $\prec$ vagy $\doteq$ reláció áll fenn,
- α utolsó és β első betűje között a $\prec$ reláció áll fenn,
- a nyél, vagyis β bármely két betűje között az $\doteq$ reláció áll fenn,
- β utolsó és w első betűje között a $\succ$ reláció áll fenn.

Egy jobb mondatforma nyelét úgy találhatjuk meg, hogy addig shifteljük az inputot (még akkor is, ha az már redukálható), amíg azt nem találjuk, hogy a legutoljára beolvasott betű és a hátralévő input első betűje között a $\succ$ reláció áll fenn.

Ha ez teljesül, akkor megtaláltuk a nyél jobb oldali végét.

Ezután a beolvasott szimbólumok között addig olvasunk visszafelé, amíg két betű között a $\prec$ relációt nem találjuk. Ahol ez áll fenn, ott van a nyél bal oldali vége (vagyis az eleje).

Ha a nyelet megtaláltuk, akkor a jobb mondatforma redukálása egyértelműen elvégezhető (tehát nem lesz redukálás-redukálás konfliktus sem).

Az egyszerű precedencia nyelvtanok esetében is a következő két kérdés megválaszolására koncentrálunk:

(1) Ha adott egy G nyelvtan, akkor eldönthető-e, hogy G egyszerű precedencia nyelvtan-e?

(2) Hogyan működik az egyszerű precedencia elemzés?

Legyen $G = (N, \Sigma, P, S)$ egy környezetfüggetlen nyelvtan. A $\prec, \doteq$ és $\succ$ relációk az $N \cup \Sigma \cup \{\$\}$ halmazon értelmezett legszűkebb olyan relációk, melyekre az alábbi feltételek teljesülnek:

- minden $X \in (N \cup \Sigma)$-ra, ha $S \Rightarrow^{+} X\alpha$, valamely α-ra, akkor $\$ \prec X$;
- minden $X, Y \in (N \cup \Sigma)$-ra, ha van olyan $A \rightarrow \alpha X B \beta$ szabály, melyre $B \Rightarrow^{+} Y\gamma$, valamely γ-ra, akkor $X \prec Y$;
- minden $X, Y \in (N \cup \Sigma)$-ra, ha van $A \rightarrow \alpha XY\beta$ alakú szabály, akkor $X \doteq Y$;

- minden $X \in (N \cup \Sigma)$-ra és $a \in \Sigma$-ra, ha van olyan $A \rightarrow \alpha BY\beta$ szabály, melyre $B \Rightarrow^{+} \gamma X$ és $Y \Rightarrow^{*} a\delta$, valamely γ-ra és δ-ra, akkor $X \succ a$ (ahol az $Y \Rightarrow^{*} a\delta$ levezetés nulla lépésben is teljesülhet, tehát úgy, hogy $Y = a$);
- minden $X \in (N \cup \Sigma)$-ra, ha $S \Rightarrow^{+} \alpha X$, valamely α-ra, akkor $X \succ \$$.

Az előbbi precedencia relációk könnyem meghatározhatók. Pl. $\prec$ esetén: ha van olyan $A \rightarrow \alpha X B \beta$ szabály, melyre $B \Rightarrow^{+} Y\gamma$, valamely γ-ra, akkor $X \prec Y$.
$H_B = \{Y \in (N \cup \Sigma) \mid B \Rightarrow^{+} Y\gamma\}$ halmaz kiszámítása:

1. Legyen $i = 0$ és $H_i = \{Y \in (N \cup \Sigma) \mid B \rightarrow Y\alpha \in P\}$;
2. Legyen $H_{i+1} = H_i \cup \{Y \in (N \cup \Sigma) \mid \exists (A \in H_i) : A \rightarrow Y\alpha \in P\}$;
3. Ha $H_{i+1} = H_i$ akkor álljunk meg, különben legyen $i = i+1$, és menjünk 2-re;

Nyilvánvaló annak igazolása, hogy az algoritmus véges lépés után $H_i = H_B$ mellett terminál.

Definíció. Egy $G = (N, \Sigma, P, S)$ cf nyelvtan egyszerű precedencia nyelvtan, ha rá a következők teljesülnek:

- G ciklusmentes és λ-mentes,
- G egyértelműen redukálható,
- minden $X, Y \in (N \cup \Sigma)$-ra a $\prec, \doteq$ és $\succ$ relációk közül *legfeljebb egy* áll fenn X és Y között. $\square$

Tétel. Tetszőleges G környezetfüggetlen nyelvtanról eldönthető, hogy egyszerű precedencia nyelvtan-e.

Az egyszerű precedencia elemzés

Tétel. Legyen $G = (N, \Sigma, P, S)$ egy λ-mentes cf nyelvtan. Továbbá tegyük fel, hogy

$$\begin{aligned} \$S\$ &\Rightarrow_r^n X_1 \dots X_k A a_1 \dots a_l \\ &\Rightarrow_r X_1 \dots X_k Y_1 \dots Y_m a_1 \dots a_l. \end{aligned}$$

Akkor fennállnak a következő relációk:

(1) minden $1 \le i < k$ esetén $X_i \prec X_{i+1}$ vagy $X_i \doteq X_{i+1}$,

(2) $X_k \prec Y_1$,

(3) minden $1 \le i < m$ esetén $Y_i \doteq Y_{i+1}$,

(4) $Y_m \succ a_1$.

Megjegyzés. Azt nem kell feltenni, hogy G egyszerű precedencia nyelvtan, mivel az állítás csak azt mondja ki, hogy az (1)-(4) pontokban szereplő relációk fennállnak de azt nem, hogy csak ezek állnak fenn.

Következmény. Az előző tétel feltételei mellett tegyük fel még azt is, hogy G egyszerű precedencia nyelvtan. Ekkor az (1) állításban a vagy feltétel kizáró vagyot jelent, továbbá az (1)-(4) állítások mindegyike kiegészíthető azzal, hogy a megfelelő helyeken más reláció pedig nem áll fenn.

Egyszerű precedencia elemzési algoritmus.

Input G (egyszerű precedencia) nyelvtan, precedencia mátrix és $w \in \Sigma^*$ szó. Feltesszük továbbá, hogy G szabályai meg vannak számozva.

Output *Igen* ha $w \in L(G)$, *Nem* különben.

Módszer

- Az elemzés $(\$\alpha, x\$, \pi)$ alakú konfigurációk sorozata, ahol
  - $\alpha \in (N \cup \Sigma)^*$, az elemzendő szó már beolvasott részéből redukált járható prefix. Egy veremnek tekintjük, melynek teteje a jobb végén van.

- $x \in \Sigma^*$ egy szó, az elemzendő szó hátralévő része,
- teljesül továbbá, hogy $\alpha x \Rightarrow_r^* w$
- π szabályok sorszámaiból alkotott sorozat. Egy sornak tekintjük, melynek teteje a bal végén van.

- Kezdő konfiguráció: $(\$, w\$, \lambda)$. A befejező konfigurációk $(\$S, \$, \pi)$ alakúak. A konfigurációk közötti átmenetet $\vdash$-val jelöljük. Az átmeneti reláció úgy van definiálva, hogy $w \in L(G)$ akkor és csakis akkor teljesül, ha $(\$, w\$, \lambda) \vdash^* (\$S, \$, \pi)$ valamely π-re és ekkor π adja a w egy jobb oldali levezetését.

Az elemzési algoritmus

1. $C := (\$, w\$, \lambda)$;
2. Amíg van olyan C', melyre $C \vdash C'$, legyen $C := C'$;
3. Ha $C = (\$S, \$, \pi)$ alakú, valamely π-re, akkor output: *Igen*, különben output: *Nem*.

Amennyiben az output *Igen*, akkor π adja w egy jobb oldali levezetését.

Átmeneti reláció Tegyük fel, hogy a $C = (\$\alpha, x\$, \pi)$ konfigurációban vagyunk, ahol $\alpha \in (N \cup \Sigma)^*$ és $x \in \Sigma^*$. Legyen X a $\$\alpha$ verem tetején lévő szimbólum (vagyis $\$\alpha = \alpha' X$) és legyen a az $x\$$ első betűje (vagyis $ax' = x\$$). (Megjegyezzük, hogy $\alpha = \lambda$ esetén $X = \$$, $x = \lambda$ esetén $a = \$$.)

(1) Ha (a precedencia mátrix szerint) $X \prec a$ vagy $X \doteq a$, akkor $C \vdash C'$, ahol $C' = (\$\alpha a, x', \pi)$.

(2) Ha (a precedencia mátrix szerint) $X \succ a$, akkor keressük meg a $\$\alpha$ veremben fölülről lefelé az első olyan helyet, ahol két veremszimbólum között $\prec$ áll fenn, vagyis legyen β a legrövidebb szó, melyre $\$\alpha = \alpha'\beta$ és α' utolsó és β első betűje között $\prec$ áll fenn. (Előfordulhat, hogy $\alpha' = \$$.) Ekkor
   - ha β egy $A \rightarrow \beta$ szabály (mondjuk a j-edik) jobb oldala, akkor redukáljunk, vagyis $C \vdash C'$, ahol $C' = (\alpha' A, x\$, j\pi)$,
   - különben, vagyis ha β egyetlen szabálynak sem jobb oldala, akkor nincs átmenet.

(3) Különben nincs átmenet.

V É G E

Írásbeli vizsga kérdések a „Formális nyelvek és szintaktikus elemzésük” c. tárgyhoz

a) Bevezető formális nyelvi rész

1. Generatív nyelvtan definíciója, levezetés, nyelvtan által generált nyelv fogalma. Chomsky nyelvosztályok. [11–16]
2. Véges automata fogalma, nemdeterminisztikus és determinisztikus automaták ekvivalenciája. [17–22]
3. Reguláris kifejezés, általa meghatározott nyelv. A reguláris nyelvek 3 típusúak. [22–26]

4. A 3 típusú nyelvek automatával való felismerhetősége. [26–29]
5. Az automatával felismerhető nyelvek regulárisak (Kleene tétele). [29–31]
6. Pumpáló lemma reguláris nyelvekre. [31–33]
7. A reguláris nyelvek osztályának zártsági tulajdonságai (reguláris műveletek, Boole műveletek). [4.13 tétel + 34–35]
8. Környezetfüggetlen nyelvek levezetési módjai (általános, bal– és jobb oldali) és ezek kapcsolata. [37 + 5.8 lemma]
9. Derivációs fa fogalma, levezetések és derivációs fák közötti kapcsolatok. [38–41]

10. Veremautomata fogalma, felismerés végállapottal és üres veremmel, ezek ekvivalenciája. [43–48]
11. A környezetfüggetlen nyelvek veremautomatákkal való felismerhetősége. [48–49]
12. A veremautomatákkal felismerhető nyelvek környezetfüggetlenek. [50–51]
13. Pumpáló lemma környezetfüggetlen nyelvekre (Bar–Hillel lemma). [5.7 lemma, 5.9 következmény kimondása + 64 – 66]
14. A környezetfüggetlen nyelvek osztályának zártsági tulajdonságai (reguláris műveletek, Boole műveletek). [66–67]

b) Nyelvek elemzése rész

1. Szintaxis megadása környezetfüggetlen nyelvtannal, az elemzés alapfeladata. [1–6, 69–70]
2. Az általános felülről lefelé haladó elemzési algoritmus. [73–75]
3. A FI_k és a FO_k halmazok definíciója és kiszámításának algoritmusa. [77, 7.6 def., 81, 86, 88]
4. Az LL(k) és az erősen LL(k) nyelvek definíciója, és ekvivalanciájuk a $k = 1$ esetben. Az erősen LL(k) feltétel eldönthetősége. [78, 87-89, 93–95]

5. Az erősen LL(k) nyelvek elemzése: elemző tábla kosntukciója és az elemzési algoritmus. [90–92]
6. Az általános alulról felfelé haladó elemzési algoritmus. [99–101]
7. Az LR(k) nyelvtan definíciója és az LR(k) elemzéshez szükséges fogalmak bevezetése (nyél, járható prefix, LR(k) elem, érvényesség, LR(k) táblák és azok kiszámítása). [8.1, 8.2, 103–104, 108–110]
8. Az LR(k) feltétel átfogalmazása (LR(k) tétel), a tevékenység és a goto táblák definíciója. Az LR(k) elemzési algoritmus. [8.9, 111-114]

9. Egyszerű precedencia nyelvtanok definíciója, eldönthetősége. Egyszerű precedencia elemzési algoritmus. [117–118, 8.8, 121–122]

A példák nem kellenek (legfeljebb jó ha vannak).