Formális nyelvek és automaták II.

Egyetemi segédanyag

Dr. Hunyadvári László docens előadásait lejegyezte Zalán András

ELTE 2003

Tartalomjegyzék

1. Bevezetés

2. Első előadás

Célunk az, hogy a 2-es-típusú nyelvtanokat kiterjesszük/általánosítsuk, ezáltal nagyobb hatóerejű nyelvtanokat nyerjünk, de az eredeti nyelvtanok jó tulajdonságait is megőrizzük.

2.1. Homomorfizmus lemma

Legyen X és Y tetszőleges ábécé. Ekkor a $h : X^* \to Y^*$ típusú, konkatenáció tartó ($h(uv) = h(u)h(v)$) leképezésekre h *homomorfizmus.*

h-t elég az X ábécé elemein megadni, az összes többi szó előáll ezen elemek konkatenációjaként.

2.1. LEMMA. *Legyenek $G_i = \langle T_i, N_i, \mathcal{P}_i, S_i \rangle$, $i = 1, 2$ tetszőleges nyelvtanok. Legyen $h : (T_1 \cup N_1)^* \to (T_2 \cup N_2)^*$ homomorfizmus, melyre ha*

1. $h(S_1) = S_2$,
2. $h(T_1) \subseteq T_2^*$,
3. *tetszőleges $p \to q \in \mathcal{P}_1$ szabály esetén $h(p) \xrightarrow[G_2]{*} h(q)$,*

akkor $h(L(G_1)) \subseteq L(G_2)$.

A lemma bizonyításához szükségünk van egy állításra:

2.2. ÁLLÍTÁS. $\alpha \xrightarrow[G_1]{*} \beta \Longrightarrow h(\alpha) \xrightarrow[G_2]{*} h(\beta)$.

BIZONYÍTÁS. A levezetés hossza szerinti indukcióval bizonyítjuk az állítást.

$l = 0$-ra teljesül, reflexív.

$l = 1$-re tekintsünk egy $\alpha \xrightarrow[G_1]{} \beta$ szabályt, $\alpha = \alpha_1 p \alpha_2$, $\beta = \alpha_1 q \alpha_2$, $p \to q \in \mathcal{P}_1$.

$h(\alpha) = h(\alpha_1)h(p)h(\alpha_2)$, $h(\beta) = h(\alpha_1)h(q)h(\alpha_2)$.

Tehát ha $p \to q \in \mathcal{P}_1$, akkor $h(p) \xrightarrow[G_2]{*} h(q)$ teljesül h-nak a feltételben leírt tulajdonsága miatt. Sőt ebből az is következik, hogy $h(\alpha_1)h(p)h(\alpha_2) \xrightarrow[G_2]{*} h(\alpha_1)h(q)h(\alpha_2)$ és $h(\alpha) \xrightarrow[G]{*} h(\beta)$ a konkatenáció tartás miatt.

Innen indukcióval következik az állítás.

BIZONYÍTÁS. (lemma) Az állítás felhasználásával tetszőleges u szóra teljesül, hogy $u \in L(G_1) \Longleftrightarrow S_1 \xrightarrow[G_1]{*} u \wedge u \in T_1^* \Longrightarrow h(S_1) \xrightarrow[G_2]{*} h(u) \wedge u \in T_1^* \Longrightarrow S_2 \xrightarrow[G_2]{*} h(u) \wedge h(u) \in T_2^*$, azaz $h(u) \in L(G_2)$. Tehát $h(L(G_1)) \subseteq L(G_2)$ és evvel beláttuk a lemma állítását.

Legyen G egy tetszőleges nyelvtan és D pedig egy tetszőleges $\alpha \xrightarrow[G]{*} \beta$ alakú levezetés.

DEFINÍCIÓ. *Egy D levezetés* munkaterületének a hossz*ának nevezzük és* $\mathrm{WS}(D)$*-vel jelöljük a levezetésben előforduló mondatformák hosszának a maximumát.*

DEFINÍCIÓ. *Egy u szó* módosított hossz*át a következő módon definiáljuk:*

$$\tilde{\mathrm{l}}(u) = \begin{cases} \mathrm{l}(u) & \text{ha } u \neq \varepsilon, \\ 1 & \text{ha } u = \varepsilon. \end{cases}$$

Legyen $K \geq 1$ tetszőleges egész szám.

DEFINÍCIÓ. *A G nyelvtan K-*korlátolt *akkor és csak akkor, ha $\forall u \in L(G)$ szóhoz $\exists D : S \xrightarrow[G]{*} u$ levezetés, melyre* $\mathrm{WS}(D) \leq K\tilde{\mathrm{l}}(u)$.

Azaz a levezetés közbeni szavak nem lehetnek akármilyen hosszúságúak.
Speciális eset, ha $K = 1$, ekkor a nyelvtan 1-*korlátolt*.

MEGJEGYZÉS. Ez az 1-es típusú nyelvtanok általánosítása: hossznemccsökkentés. Minden kiterjesztett 1-es nyelvtan 1-korlátolt is: hossznemcsökkentés és a KES-szabály teljesül.

JELÖLÉS. *Jelölje $\mathcal{L}_{Klb}$ a K-korlátolt nyelvtannal generálható nyelvek osztályát.*

Ekkor teljesül, hogy $\mathcal{L}_1 \subseteq \mathcal{L}_{1lb} \subseteq \mathcal{L}_{2lb} \subseteq \ldots$, hiszen minden K-korlátolt nyelvtan $K+1$ korlátolt is.

2.3. ÁLLÍTÁS. *Tetszőleges $K \geq 1$ esetén $\mathcal{L}_{Klb} \subseteq \mathcal{L}_{Rek}$.*

$\mathcal{L}_{Rek}$ jelöli a rekurzív nyelvek osztályát, vagyis azon nyelvek osztályát, melyhez létezik teljes eldöntő eljárás.

Az összes levezetések fájának szintfolytonos bejárásával megtaláltuk a levezetést, ami ha 1-es típusú, akkor rekurzív is. Elég volt minden olyan levezetést megnézni, melyben az összes szó különböző volt. Most a szóhosszak K-szorosával lehet vágni, ez is véges sok. Tehát a bizonyítás menete megegyezik az 1-es nyelvtanok rekurzivitásának bizonyításával.

2.4. TÉTEL. $\forall K \geq 1$ *esetén* $\mathcal{L}_{Klb} = \mathcal{L}_1$.

BIZONYÍTÁS. • Egyrészt azt kell belátni, hogy $\forall K \geq 2$ esetén $\mathcal{L}_{Klb} = \mathcal{L}_{1lb}$.

Elég belátni, hogy tetszőleges G K-korlátolt nyelvtanhoz létezik olyan G' 1-korlátolt nyelvtan, melyre $L(G') = L(G)$. K darab egymást követő jelet a továbbiakban egy jelnek tekintünk (K-szoros összenyomás). Meg kell adnunk, hogy ezekre a blokkokra milyen szabályokat alkalmazunk – nem csak a jeleket, hanem a szabályokat is át kell alakítanunk a megfelelő módon. Legyen G olyan nyelvtan, hogy $G = \langle T, N, \mathcal{P}, S\rangle$. Legyen $\alpha \in (T \cup N)^*$ tetszőleges mondatforma. Lesz egy olyan blokk, amelyben általában K-nál kevesebb jel van. Ezt a blokkot ne egy vonással, hanem hullámmal jelöljük.

Legyen $X = \overline{(T \cup N)^K}$ az összes lehetséges K hosszú blokk halmaza. Legyen $X = \widetilde{(T \cup N)}^{1 \leq l \leq K}$ az összes lehetséges legfeljebb K hosszú blokk halmaza.

A hullám azt jelöli egy blokk felett, hogy hasonlóan a vonáshoz, blokkról van szó. A hullámmal jelölt blokk legfeljebb K hosszúságú, de azt szeretnénk, hogy ilyen blokk mindenképpen szerepeljen a mondatformában.

Az új nyelvtanban a blokkok lesznek az új nyelvtani jelek, a terminálisok voltak a régiek. Ekkor a G' nyelvtan legyen a következő alakú: $G' = \langle T, X \cup Y, \mathcal{P}', \tilde{S}\rangle$.

Konstruáljuk meg $\mathcal{P}'$-t, az új nyelvtan szabályainak halmazát úgy, hogy úgy viselkedjen blokkokon, mint az eredeti nyelvtanban.

Nézzünk egy egylépéses levezetést. Az $\alpha \xrightarrow[G]{} \beta$ levezetést tekintjük, azaz $\alpha_1 p \alpha_2 \xrightarrow[G]{} \alpha_1 q \alpha_2$ és $p \to q \in \mathcal{P}$.

α_1 elejéből és α_2 végéből levágunk annyi K méretű blokkot, amennyit csak lehetséges. A középen megmaradó rész, mely tartalmazza p-t is, összességében több blokkból is állhat (α_1 vége, p és α_2 eleje). Ezt a p-t helyettesítjük q-ra. Elég ezt a részt vizsgálni, hogy itten hogyan viselkednek a blokkok. Kérdés, hogy felmérve a K méretű blokkokat hol keletkezik a hullámos blokk. Ezt kell helyettesíteni egy hasonló alakúra.

Az összes felülvonásos blokk belemetsz p-be, hiszen legfeljebb K hosszúak, így megfogalmazhatjuk az új szabályokat.

p és a két oldali maradék $= \overline{z_1}\,\overline{z_2}\dots\overline{z_j}\,\overline{z_{j+1}}\dots\overline{z_h} \to \overline{w_1}\dots\widetilde{w_{j'}}\dots\overline{w_{h'}} \in \mathcal{P}'$ alakúak a szabályok $\Longleftrightarrow \exists p \to q \in \mathcal{P} : z_1 z_2 \dots z_j \dots z_h \xrightarrow[G]{p\to q} w_1 \dots w_{j'} \dots w_{h'}$, azaz egy lépésben levezethető és minden felülvonásos blokk belemetsz p-be, ha létezik felülvonásos blokk. Ezt nevezzük a szabályok blokkosított formájának.

$z_1 z_2 \dots z_j \dots z_h$ hossza megbecsülhető: legfeljebb $K-1$ rossz lehet balra és jobbra, továbbá a hullámosak, tehát ez p hosszánál csak valamivel hosszabb, $3K-2$, tehát az ilyen szabályok száma véges.

A hullámos blokkot kell tudni mozgatni, tehát szükségünk van az alábbi típusú átalakító szabályokra is:

$$\begin{array}{rcl}
\overline{\alpha}\tilde{\beta} \to \tilde{\gamma}\overline{\delta} & \Longleftrightarrow & \alpha\beta = \gamma\delta \\
\tilde{\alpha}\overline{\beta} \to \overline{\gamma}\tilde{\delta} & \Longleftrightarrow & \alpha\beta = \gamma\delta \\
\tilde{\alpha} \to \alpha & \Longleftrightarrow & \alpha \in T^* \\
t\overline{\alpha} \to t\alpha & & \alpha \in T^*, t \in T \\
\overline{\alpha}t \to \alpha t & & \alpha \in T^*, t \in T
\end{array}$$

Az utóbbi két szabályt akkor alkalmazzuk, ha már a hullámos blokkból bejött a terminális.

Alapvetően három szabályt alkalmazunk: eredeti szabály blokkosítása, hullám áthelyezése és terminális visszaírás. Amit eredetileg le tudtunk vezetni, azt blokkosított formában is le tudjuk vezetni. $L(G') = L(G)$.

Nincs ε jobboldalú szabály! Ha van, azt az esetet külön vizsgáljuk meg.

Be kell látni, hogy G' 1-korlátolt. $u \neq \varepsilon \in L(G') = L(G)$. Be kell látni, hogy $\exists D' : \tilde{S} \xrightarrow[G']{} u$ levezetés, melyre $\mathrm{WS}(D') \leq 1 \cdot \mathrm{l}(u)$.

Tudjuk, hogy $\exists D : S \xrightarrow[G]{*} u$ levezetés, melyre $\mathrm{WS}(D) \leq K \cdot \mathrm{l}(u)$.

Legyen D' D blokkosított formája.

$D' : \tilde{S} = \varphi_0 \xrightarrow[G']{} \varphi_1 \longrightarrow \varphi_2 \dots \varphi_j = \varphi_m = u.$

Létezik φ_j, hogy onnantól kezdve már csak a blokkokat írjuk vissza terminálisokra. A terminális visszaírók hosszúságnemcsökkentő szabályok, tehát csak a φ_0, φ_j részen kell vizsgálni, hogy kisebb-e, mint a szóhossz. Belátjuk, hogy minden olyan i-re, melyre $0 \leq i \leq j$: $\mathrm{l}(\varphi_i') \leq \mathrm{l}(u)$. A köztes φ_i-k között van egy hullámos blokk és közönséges blokkok. $1 + (\mathrm{l}(\varphi_i) - 1)K \leq \mathrm{l}(\underline{\varphi_i})$. Blokkokat visszaírtuk sorozattá. $\underline{\varphi_i}$ az eredeti levezetés egy mondatformája, ezért $\mathrm{l}(\underline{\varphi(i)}) \leq K \cdot \mathrm{l}(u)$, mert blokkstruktúra.

$\mathrm{l}(\varphi_i) - 1 + \frac{1}{K} \leq \mathrm{l}(u) \Longrightarrow \mathrm{l}(\varphi_i) \leq \mathrm{l}(u)$, mert egészek.

Ha lenne benne ε, akkor 0-ás típusú ε-mentesítést kellene alkalmazni. A levezetés munkaterületének hossza ugyanaz marad, üres szó bármikor hozzávehető – 1-korlátolt levezetést kapunk.

- $\mathcal{L}_1 = \mathcal{L}_{1lb}$

  A $\subseteq$ irány triviális.

  Elegendő belátni, hogy tetszőleges G 1-korlátolt nyelvtanhoz $\exists G' \in \mathcal{G}_{kit1}$ nyelvtan, hogy $L(G') = L(G)$. Elég az $\varepsilon \notin L(G)$ esettel foglalkozni, hiszen ez csak a KES-szabály hozzávételét, illetve elhagyását jelenti.

  Ha G minden szabálya hosszúságot nem csökkentő, akkor kész vagyunk. Egyébként: $G = \langle T, N, \mathcal{P}, S\rangle$ és van egy $p \to q \in \mathcal{P}$ szabály, melyre $\mathrm{l}(p) \leq \mathrm{l}(q)$. Ebből hosszúságot nem csökkentő szabályt szeretnénk készíteni, megkonstruáljuk G'-t. Legyen $B \in N \cup T$ egy új nyelvtani jel és a fenti „rossz” szabályok helyett az új, G' nyelvtanban, ahol $G' = \langle T, N \cup \{B\}, \mathcal{P}', S\rangle$ alakú, a szabályok pedig legyenek a következőek:

1. $p \to qB^{\mathrm{l}(p)-\mathrm{l}(q)} \in \mathcal{P}' \iff p \to q \in \mathcal{P} \wedge \mathrm{l}(p) \geq \mathrm{l}(q)$ és az így kapott szabály hosszúságot nem csökkentő, amennyivel csökken a hossz, annyi B. Ha ugyanazt akarjuk levezetni, el kell tüntetni a B-ket, az olyan szabályoknál, ahol határozottan csökkent a hossz, azaz a 2-es típusú szabályoknál.
2. $pB^i \to q \in \mathcal{P}' \iff p \to q \in \mathcal{P} \wedge \mathrm{l}(p) \leq \mathrm{l}(q) \wedge 0 \leq i \leq \mathrm{l}(q) - \mathrm{l}(p)$ és így a második típusú szabály is hosszúság nemcsökkentő.
3. B-t mozgatni kell tudni, hogy „el lehessen nyeletni". $xB \to Bx, Bx \to xB$, $x \in T \cup N$.

G' kiterjesztett 1-es nyelvtan, mert mindenhol hosszúságcsökkentő. Tehát bizonyítani kell, hogy $L(G') = L(G)$.

$\subseteq$ eset: definiáljuk a $h : (T \cup N \cup B)^* \longrightarrow (T \cup N)^*$ homomorfizmust a következőképpen:

$$h(z) = \begin{cases} z & z \in T \cup N \\ \varepsilon & z = B \end{cases}$$

A homomorfizmus szavakhoz szavakat rendel, de elég a betűkön megadni.

$G' = G_1$, $G = G_2$-re teljesülnek-e a homomorfizmus lemma feltételei?

$h(L(G')) \subseteq L(G) \Longrightarrow L(G') \subseteq (G)$.

$h(L(G')) = L(G')$, mert csupa terminális szerepel és azt helyben hagyja.

$L(G') \supseteq L(G)$ bizonyítása: ha $u \in L(G) \Longrightarrow u \in L(G')$

G 1-korlátoltsága miatt $\exists$ u-nak $D : S = \alpha_0 \to \alpha_1 \to \alpha_2 \ldots \alpha_m = u$ levezetése, hogy $\forall j = 0, \ldots, m :$ $\mathrm{l}(\alpha_j) \leq \mathrm{l}(u)$.

Készítsük el D-ből a következő G'-beli D' levezetést:

$D' : S = \alpha_0 B^{r_0} \xrightarrow[G']{*} \alpha_1 B^{r_1} \xrightarrow[G']{*} \alpha_2 B^{r_2} \xrightarrow{*} \ldots \xrightarrow{*} \alpha_m B^{r_m}$, ahol $r_0 = 0$. Minden lépésben hozzáad vagy elvesz B-ket.

$D \to D'$

(a) ha D-ben hosszúságcsökkentő $p \to q$ szabály volt, akkor ugyanazok az α-k szerepelnek, B-k bejönnek a mondatformába. A levezetés lehet, hogy több lépéses, mert lehet, hogy szükség van a B-k mozgatására.

(b) Hosszúságnövelő $p \to q$ szabály esetén maximális számú B-t melléírva nyelessük el. A $p \to q$ szabály helyett D'-ben a $pB^i \to q$ szabályt alkalmazzuk a lehető legnagyobb i-vel, miután odamozgattam a B-ket.

i maximuma: p és q hosszkülönbségétől függ, ha van egyáltalán ennyi vagy a mondatformában levő B-k számának a maximuma.

2.5. Állítás. $\forall j = 0, 1, \ldots, m : \quad \mathrm{l}(\alpha_j) + r_j \leq \mathrm{l}(u)$.

Bizonyítás. Az állítást indukcióval bizonyítjuk.

$j = 0$ esetén $\mathrm{l}(s) + 0 \leq \mathrm{l}(u)$ az $u \neq \varepsilon$ kikötés miatt. $1 \leq \mathrm{l}(u)$.

Tegyük fel, hogy az állítás $0, \ldots, j-1$-re teljesül, bizonyítsuk be j-re is. Teljesül, hogy $\mathrm{l}(\alpha_{j-1}) + r_{j-1} \leq \mathrm{l}(u)$.

$\alpha_{j-1} B^{r_{j-1}} \xrightarrow[G']{*} \alpha_j B^{r_j}$

Az (a) esetben a $p \to qB^{\mathrm{l}(p)-\mathrm{l}(q)}$ szabály nem változtatja a hosszt, a B-ket mozgató szabály sem változtatja a hosszt. Tehát $\mathrm{l}(\alpha_{j-1} B^{j-1}) = \mathrm{l}(\alpha_j B^{r_j})$. Az $\mathrm{l}(\alpha_{j-1}) + r_{j-1} = \mathrm{l}(\alpha_j) + r_j$ egyenlőség teljesül. A bal oldal az indukció miatt $\leq \mathrm{l}(u)$, tehát $\mathrm{l}(\alpha_j) + r_j \leq \mathrm{l}(u)$.

A (b) esetben a $p \to q$ hossznemcsökkentő szabály helyett a $pB^i \to q$ szabályt használjuk i maximumára.

1. maximális i lehet az összes B száma, r_{j-1}. Ekkor $r_j = 0$. $\mathrm{l}(\alpha_j) + r_j = \mathrm{l}(\alpha_j) \leq \mathrm{l}(u)$. Teljesül D első tulajdonsága miatt.

2. $\max i = \mathrm{l}(q) - \mathrm{l}(p)$ esetén nem változik a hossz, mint az (a) esetben.

$\mathrm{l}(\alpha_m) + r_m \leq \mathrm{l}(u)$, illetve mert $\alpha_m = u$, ezért $\mathrm{l}(u) + r_m \leq \mathrm{l}(u)$ és mert $r_m = 0$, ezért D' az u levezetése.

2.2. Egy alkalmazás

Vizsgáljuk meg a metszetképzés és a Chomsky-féle nyelvosztályok kapcsolatát!

2.6. TÉTEL. *$\mathcal{L}_3, \mathcal{L}_1, \mathcal{L}_0$ zárt, $\mathcal{L}_2$ nem zárt a metszetképzésre.*

BIZONYÍTÁS.

- $\mathcal{L}_2$-re megadunk két olyan nyelvet, melyek metszete nem 2-es: $\{a^n b^n c^m; n, m \geq 0\} \in L_2$ és $\{a^m b^n c^n; n, m \geq 0\} \in L_2$, metszetük pedig $\{a^n b^n c^n; n \geq 0\} \notin L_2$ a nagy Bar-Hillel lemma miatt.
- $\mathcal{L}_3$-ra a direktszorzat automata $\mathcal{F}_1 \times \mathcal{F}_2$ végállapotával éppen a metszetüket lehet elfogadni.
- Bizonyítsuk be $i = 0$-ra és 1-re is a tétel állítását, legyen $G_i = \langle T_i, N_i, \mathcal{P}_i, S_i \rangle$, $i = 1, 2$, $G_i \in \mathcal{G}_j$, $j = 0, 1$. Ekkor $\exists G_\cap \in G_j$ és $L(G_1) \cap L(G_2) = L(G_\cap)$. Feltehető, hogy terminális csak az $A \to t$ alakú szabályokban fordul elő. Továbbá feltehető az is, hogy $N_1 \cap (T_2 \cup N_2) = \emptyset$, $N_2 \cap (T_1 \cup N_1) = \emptyset$ új nyelvtani jelek bevezetésével.

  Vezessük be $\forall t \in T_1$-re az X_t új nyelvtani jeleket, $\forall t \in T_2$-re az Y_t jeleket.

  $$G_\cap = \langle T_1 \cup T_2, N_1 \cup N_2 \cup \{X_t\}_{t \in T_1} \cup \{Y_t\}_{t \in T_2} \cup \{S, B\}, \mathcal{P}, S \rangle,$$

  ahol S jelöli az új kezdőjelet és B az új végjelet.

  $$S \to BS_1S_2 \qquad S \to BX_{u_1}X_{u_2}$$

  $$\mathcal{P}_1' \qquad \mathcal{P}_2'$$

  Az $X_{t_1}Y_{t_2} \to Y_{t_2}X_{t_1}$ szabállyal a második nyelvtan elemeit vihetem balra, a $BY_tX_t \to tB$ szabállyal ha elért balra a jel, akkor összehasonlítom. Így mindig az első két jelet tudjuk csak eltüntetni. $B \to \varepsilon$ szabály szükséges azért, hogy a végén a B-t el tudjuk tüntetni.

  Ami benne van a metszetben, azt az új nyelvtanban le lehet vezetni, ez látszik a konstrukció alapján. $L(G_1) \cap L(G_2) \subseteq L(G_\cap)$.

  A másik irány a homomorfizmus lemmával könnyen belátható.

  Tehát $L(G_\cap) = L(G_1) \cap L(G_2)$. $j = 0$ esetén teljesül, nincs megkötés. $j = 1$-re $G_\cap$ 1-típusú-e, G_1, G_2 1-típusú? DE $G_\cap$ nem is hosszúságcsökkentő!

  $S \to BS_1S_2 \to \cdots \to B\alpha_1'\alpha_2' \to u$

  Egy közbeeső mondatforma maximális hosszúságú, ha α_1', α_2' maximális hosszúságú. Külön-külön legfeljebb olyan hosszúak, mint u, illetve B hossza $\tilde{\mathrm{l}}(u)$-nál nem hosszabb. Tehát $B\alpha_1'\alpha_2' \leq 3\,\tilde{\mathrm{l}}(u)$, mert $B \leq \tilde{\mathrm{l}}(u)$ és $\alpha_1, \alpha_2 \leq \mathrm{l}(u)$.

  Azaz $G_\cap$-ben a közbülső mondatformák hossza $\leq 3\,\tilde{\mathrm{l}}(u) \Longrightarrow G_\cap$ 3-korlátolt $\Longrightarrow$ az előző tétel alapján $L(G_1) \in \mathcal{L}_1$.

3. Speciális nyelvtanok

3.1. Mátrix-nyelvtanok

Ebben a részben olyan speciális nyelvtanokat szeretnénk megadni, ahol csak bizonyos szabályszekvenciák alkalmazása megengedett.

PÉLDA. A nyelv álljon az $\{a^n b^n c^n; n \geq 0\}$ alakú szavakból. Készítsünk rá egy 2-es típusú nyelvtant. Ekkor a szabályok a következőek:

$$[S \to ABC], [A \to aA, B \to bB, C \to cC], [A \to \varepsilon, B \to \varepsilon, C \to \varepsilon].$$

Definíció. *G mátrixnyelvtan*on *a következő négyest értjük: $G = \langle T, N, M, S\rangle$, ahol T jelöli a terminális jeleket, N a nyelvtani jeleket, M mátrixok egy véges halmazát és S jelöli a nyelvtan kezdőszimbólumát.*

Az $m \in M$ mátrixok közönséges szabályok véges sorozata. Az így kapott nyelvtan hasonló a közönséges értelemben vett nyelvtanhoz, csak itt szabályok helyett szabályok sorozata, szabályszekvenciák szerepelnek.

Ezek után definiáljuk a levezetésfogalmat is a mátrix-nyelvtanokban:

$$\alpha, \beta \in (T \cup N)^*, m \in M, \qquad m = \left[p_1 \to q_1, \ldots, p_{|m|} \to q_{|m|}\right]$$

$$\alpha \xrightarrow[G]{m} \beta \iff \exists \gamma_0, \gamma_1, \ldots, \gamma_{|m|} \in (T \cup N)^*$$

$$\alpha = \gamma_0, \beta = \gamma_{|m|}, \qquad \gamma_{j-1} \xrightarrow{p_j \to q_j} \gamma_j, \forall j = 1, 2, \ldots, |m|$$

$$\alpha \xrightarrow[G]{} \beta \iff \exists m \in M : \alpha \xrightarrow[G]{m} \beta$$

$\xrightarrow[G]{*}$ legyen $\xrightarrow[G]{}$ reflexív, tranzitív lezártja.

$L(G) = \{u; u \in T^* \land S \xrightarrow[G]{*} u\}$.

Definíció. *Egy mátrix-nyelvtant* i típusú mátrix-nyelvtan*nak nevezünk, ha a benne lévő összes szabály mint közönséges szabály kiterjesztett i típusú.*

Az 1-es típusú szabályoknál módosításra van szükség: KES-szabály önmagában van, $[S \to \varepsilon]$ fog szerepelni a mátrix-nyelvtanban.

Definíció. *$\mathcal{M}_i$ az i-típusú mátrix-nyelvtanok által generált nyelvek halmaza.*

Kérdés, hogy mátrix-nyelvtanok alkalmazásával mennyivel nő meg az egyes nyelvosztályok generálási hatékonysága?

3.1. tétel. *$i = 0, 1, 3$ esetén $\mathcal{M}_i = \mathcal{L}_i$, továbbá $\mathcal{M}_2 \supsetneq \mathcal{L}_2$.*

Bizonyítás.

- Minden közönséges nyelvtan tekinthető 1 hosszú mátrix-nyelvtannak, azaz $\forall i = 0, 1, 2, 3 :$ $\mathcal{M}_i \supseteq \mathcal{L}_i$. Láttuk, hogy $\{a^n b^n c^n\} \in \mathcal{M}_2$, tehát $\mathcal{M}_2 \supsetneq \mathcal{L}_2$.
- Másrészt elég belátni, hogy ha G i-típusú mátrixnyelvtan ($i = 0, 1, 3$), akkor $\exists G'$ közönséges i-típusú nyelvtan, melyre $L(G') = L(G)$.
- $i = 3$ esetén $G = \langle T, N, M, S\rangle$ 3-as típusú mátrix-nyelvtan.

  $$m = \left[X_1 \to u_1 Y_1, X_2 \to u_2 Y_2, \ldots, X_{|m|} \to u_{|m|} Y_{|m|}\right],$$

  ahol $j = 1, \ldots, |m|$ $u_j \in T^*$, $X_j \in N$, $Y_j \in N \cup \{\varepsilon\}$.
  m konzisztens $\iff j = 1, \ldots, |m| - 1 : Y_j = X_{j+1}$. Csak az ilyen mátrixok hajthatók végre.

  Ha m konzisztens, akkor hatása a következő közönséges, kiterjesztett 3-as típusú szabállyal egyezik meg: $X_1 \to u_1 u_2 \ldots u_{|m|} Y_{|m|}$. Tehát $G' = \langle T, N, \mathcal{P}', S\rangle$, ahol $\mathcal{P}'$ a konzisztens mátrixokból képzett, hatásukat leíró kiterjesztett 3-as típusú szabályokból áll. Tehát minden G-beli levezetés szimulálható G'-beli levezetéssel. G'-ben csak szimulált levezetések vannak, mást nem lehet csinálni bennük.

  $L(G') = L(G)$, $L(G')$ 3-as típusú nyelv.
- $i = 0, 1$ esetén $G = \langle T, N, M, S\rangle$ i-típusú mátrix-nyelvtan. Most is G'-ben egy m mátrix hatását próbáljuk szimulálni i típusú közönséges szabályokkal.

  $$m = \left[p_1 \to q_1, \ldots, p_{|m|} \to q_{|m|}\right]$$

$$\alpha \xrightarrow[G]{m} \beta$$

Bevezetjük új nyelvtani jelek egy családját: X_j^m, $j = 0, 1, \ldots, |m|$, ahol a felső index jelöli, hogy melyik mátrixra vonatkozik, az alsó index pedig, hogy már hány darab szabályt hajtottunk végre.

$S' \to SX_0^m \quad \forall m \in M$ mátrixra elkészítünk egy ilyen szimulációs szabályt, mely kijelöli, hogyan használhatom ezt a mátrixot.

$X_{j-1}^m p_j \to X_j^m q_j \quad j = 1, \ldots, |m|$ szabályok jelentik a tényleges használatot.

$X_l^m Z \to ZX_l^m \quad \forall m \in M, \forall l = 0, \ldots, |m|,\ Z \in T \cup N$

$ZX_l^m \to X_l^m Z \quad \forall m \in M, \forall l = 0, \ldots, |m|,\ Z \in T \cup N$

Ezen szabályokkal tudjuk mozgatni X-et.

$X_{|m|}^m \to X_0^{m'} \quad \forall m, m' \in M$ alakú szabályok biztosítják, hogy egy mátrixot befejezve áttérhessek egy következő mátrixra.

Tehát egy eredeti mátrix-nyelvtanban elkészített levezetést el tudok végezni ebben az új nyelvtanban, csak az $X_{|m|}^m$ nyelvtani jel van benne. Ennek eltüntetéséhez be kell vezetni még egy új szabályt:

$X_{|m|}^m \to \varepsilon$.

$L(G') = L(G)$ biztosan teljesül, még G' típusát kell megvizsgálni a továbbiakban.

Ha G 0-ás típusú, akkor G' kiterjesztett 0-ás típusú, $L(G') = L(G) \in \mathcal{L}_0$.

Ha G 1-es típusú, akkor G 1-korlátolt is (az $S \to \varepsilon$ szabálynál majd a hullám miatt fog teljesülni). $u \in L(G)$ G' egy mondatformája: G egy mondatformája + X jelző nyelvtani jel. G' minden mondatformája $\leq G$ megfelelő mondatformája $+1 \leq \tilde{\mathrm{l}}(u) + \tilde{\mathrm{l}}(u) \leq 2\,\tilde{\mathrm{l}}(u)$, tagonként nézve a felső korlátot, a hossznemcsökkentés miatt. Tehát G' 2-korlátolt, amelyek megegyeznek az 1-esekkel. $L(G') = L(G) \in \mathcal{L}_1$. (Vegyük észre, hogy G' nem lett 1-es típusú, csak 2-korlátolt, de most ez is elég).

3.2. Előfordulás-ellenőrzéses mátrix-nyelvtanok

Az eddigiekhez képest egy további kérdést teszünk fel a levezetéseknél: bizonyos megjelölt szabályokat bizonyos esetekben át szabad ugrani. Ezeket a megjelölt szabályokat nevezzük pontozott szabályoknak, jelölésben egy pontot teszünk a levezetést jelölő nyíl fölé. A mátrixokban a pontozott szabályokat - ha a bal oldaluk nem fordul elő a mondatformában - át lehet ugrani.

Példa. $L = \{a^{2^n}; n \geq 0\}, \quad L \notin \mathcal{L}_2$

Periodikusan beiterálható egy kellően hosszú szóba a nagy Bar-Hillel lemma miatt.

Itt n tetszőleges – a közök egyre nagyobbak, nem lesznek egy idő után K-nál kisebbek $\Longrightarrow$ nem eleme $\mathcal{L}_2$-nek.

Az előfordulás-ellenőrzés segítségével a 2-es nyelvtannal a 0-ás is leírható lesz.

Példa. $S \to ZZ$, $Z \to S$, $S \to A$.

$n = 0$ esetén S, $n = i$ esetén 2^i darab S az előállított szó.

A második szabályt csak akkor szabad alkalmazni, ha már minden S-et átírtam Z-re. Ennek megvalósításához egy új nyelvtant kell készíteni.

$S \to ZZ$, $Z \to Y$, $Y \to S$,

$\left[Y \stackrel{\cdot}{\longrightarrow} U, A \stackrel{\cdot}{\longrightarrow} U, S \longrightarrow ZZ\right]$, $\left[S \stackrel{\cdot}{\longrightarrow} U, A \stackrel{\cdot}{\longrightarrow} U, Z \longrightarrow Y\right]$, $\left[Z \stackrel{\cdot}{\longrightarrow} U, A \stackrel{\cdot}{\longrightarrow} U, Y \longrightarrow S\right]$, $\left[Y \stackrel{\cdot}{\longrightarrow} U, Z \stackrel{\cdot}{\longrightarrow} U, S \longrightarrow A\right]$, $\left[S \stackrel{\cdot}{\longrightarrow} U, Y \stackrel{\cdot}{\longrightarrow} U, Z \stackrel{\cdot}{\longrightarrow} U, A \longrightarrow a\right]$.

$U \in N$, de nem vonatkozik rá semmilyen szabály – ha egyszer alkalmaztuk, örökre zsákutca.

Ha egyszer átírtam egy S-et A-ra akkor már nem lehet mást csinálni.

A rossz levezetések zsákutcák lesznek, csak a jó levezetések állítják elő azt, amit szeretnénk.

Definíció. G előfordulás-ellenőrzéses *mátrix-nyelvtan, ha olyan közönséges mátrix-nyelvtan, melyek mátrixaiban bizonyos szabályok megjelöltek, azaz pontozottak. Ekkor a levezetésfogalom:* $m = \Big[p_1 \xrightarrow{(\cdot)} q_1, \dots, p_{|m|} \xrightarrow{(\cdot)}$

ahol $\xrightarrow{(\cdot)}$ *azt jelenti, hogy a szabály lehet pontozott is és nem pontozott (hagyományos szabály) is.*

$\alpha \underset{G,ac}{\xrightarrow{m}} \beta \iff \exists \alpha_0, \alpha_1, \dots, \alpha_{|m|}$ *szavak, hogy* $\alpha_0 = \alpha_1, \alpha_{|m|} = \beta, \forall j = 1, 2, \dots, |m| \Big((p_j \subseteq \alpha_{j-1} \wedge \alpha_{j-1} \xrightarrow{p_j \to q_j} \alpha_j)$

$\alpha \underset{G,ac}{\longrightarrow} \beta$ *azt jelenti, hogy előfordulás-ellenőrzést alkalmazunk a levezetésben,* $\alpha \underset{G,ac}{\xrightarrow{*}} \beta$*-t pedig az eddigiekhez hasonlóan mint többszöri alkalmazást értelmezzük.*

$L^{ac}(G) = \{u; u \in T^* \wedge S \underset{G,ac}{\xrightarrow{*}} u\}.$

$\mathcal{M}_i^{ac}$ az i-típusú mátrix-nyelvtanokban előfordulásellenőrzéssel előállítható nyelvek halmaza. Mindenhol kiterjesztett i-típusú közönséges szabályok szerepelnek.

$\mathcal{M}_{2-\varepsilon}^{ac}$ az olyan 2-es típusú nyelvek halmaza, melyben nincs ε-os szabály (nem kiterjesztett 2-es, hanem alap 2-es).

3.2. Tétel. *$i = 0, 1, 3$ esetén $\mathcal{M}_i^{ac} = \mathcal{L}_i$, továbbá $\mathcal{M}_2^{ac} = \mathcal{L}_0$.*

Ez utóbbi állítás a kiterjesztettség miatt nehéz, majd később bizonyítjuk. Kiterjesztett 2-es nyelvtan nem kiterjesztett 1-es is!

A tétel állításának bizonyításához felhasználunk egy állítást:

3.3. Állítás. $\mathcal{L}_2 \subsetneq \mathcal{M}_{2-\varepsilon}^{ac} \subseteq \mathcal{M}_1^{ac} = \mathcal{L}_1.$

Az első helyen valódi tartalmazás áll fenn az előbbi ellenpélda miatt. Egyébként az állítás triviális, a tétel bizonyításához fogjuk felhasználni.

Bizonyítás. $i = 3$ G előfordulás-ellenőrzéses 3-as típusú mátrix-nyelvtan, G' kiterjesztett 3-as típusú.

$$m = \Big[X_1 \xrightarrow{(\cdot)} u_1 Y_1, \dots, X_{|m|} \xrightarrow{(\cdot)} u_{|m|} X_{|m|}\Big]$$

Ha végignéznénk, hogy hol van pont és hol nincsen, az 2^m különböző eset lenne.

$VA \underset{G,ac}{\xrightarrow{m}} VuB \quad V \in T^*$. A hatása: ha m alkalmazható A-ra, akkor uB lesz a hatása.

Az m-et szimuláló szabályhalmazban $A \to uB$ benne van $\iff A \underset{G,ac}{\xrightarrow{m}} uB$. Tehát legfeljebb $|N|$ szabály van. Ezek kiterjesztett 3-as típusú szabályok. Az új nyelvtanban pontosan egy mátrix hatásai vezethetők le, azaz $L(G') = L(G)$.

$i = 0, 1$ Ha p_j nem részszava semelyik mondatformának sem és $p_j \xrightarrow{\cdot} q_j$ pontozott szabály, akkor egy $X_{j-1}^m \to X_j^m$ szabályt kell csinálni. Csak a $p_j \subsetneq$ feltétel mellett szabad alkalmazni a szabályt. Hogyan lehet ellenőrizni az illeszkedést? Alkalmazzuk a Knuth–Morris–Pratt féle mintaillesztő automatát. A nyelvtan egy ilyen automatát szimulál.

$$m = \Big[p_1^m \xrightarrow{(\cdot)} q_1^m, \dots, p_{|m|}^m \xrightarrow{(\cdot)} q_{|m|}^m\Big] \quad X_i^m \quad i = 0, 1, \dots, |m|.$$

Legyen $p \in Z^*$ egy minta. Ekkor $L^p = \{u, u \in T^* \wedge p \subseteq u\}$ azon szavak összessége, melyben előfordul a minta. A mintaillesztés feladata annak eldöntése, hogy az adott szó eleme-e a nyelvnek.

A p mintához konstruáljuk meg az $\mathcal{A}^p = \langle \{q_0, \dots, q_{|p|}\}, Z, \delta, q_0, q_{|p|} \rangle$ automatát. Az állapot indexe az, hogy az eddig olvasott rész legfeljebb hány jelre egyezett meg. Az állapotátmenetfüggvényre pedig teljesüljön a következő:

$\delta(q_{|p|}, z) = q_{|p|}$

$\delta(q_j, z) = q_{f(j,z)}$, ahol $f(j,z) = \max |v|$ és a maximumot olyan v-kre értjük, melyekre teljesül, hogy p-nek kezdőszelete és $y_1 \dots y_j z$-nek vége, $j = 0, 1, \dots, |p| - 1$, $p = y_1 y_2 \dots y_{|p|}$. Azaz hozzáfűzve egy z-t milyen hosszan fog megegyezni a minta elejével.

Példa. $p = ababb$, $Z = \{a, b\}$.

	a	b
0	1	0
1	1	2
2	3	0
3	1	4
4	3	5

$p_i^m \stackrel{\cdot}{\longrightarrow} q_i^m$ szabályokat veszünk hozzá.

Ellenőrizni kell, hogy p_i^m előfordul-e a mondatformában. A p_i^m Knuth–Morris–Pratt automatájának működését szimuláljuk nyelvtannal.

$S' \to LSX_0^m R$, L és R „left és right marker” bevezetése, mellyel a szavak két végét jelöljük.

X_j^m-ek mozgatása ugyanúgy történi, mint korábban, mindkét irányba.

$X_{j-1}^m p_i \to X_j^m q_i^m$ jelenti az m mátrix i-edik szabályának az alkalmazását.

$X_{|m|}^m \to X_0^{m'}$ áttérés egy másik mátrixra.

$X_{|m|}^m \to \varepsilon$

Ha $p_i^m \stackrel{\cdot}{\longrightarrow} q_i^m$ pontozott szabály, akkor aktiváljuk a KMP automatát a következő szabályokkal:

$LX_{j-1}^m \to La_0^{m,j} \qquad a_k^{m,j} \qquad k = 0, 1, \dots, |p_k^m|$. Ezek állapotok, de nyelvtani jelként értelmezzük őket.

A szó elejéről akarom elkezdeni a feldolgozást, csak ott tehetem meg: $La_0^{m,j} z_1 z_2 \dots R$, akkor csere.

$a_k^{m,j} z \to z a^{m,j}_{f^{p_j^m}(k,z)} \qquad k = 0, 1, \dots, |p_j^m| - 1$.

Ha már előfordult a minta, akkor a !!!! szabály azt már kezelte.

$a_k^{m,j} R \to X_j^m R$, ahol j olyan, mint az előbb.

$L \to \varepsilon$, $R \to \varepsilon$ Ezek a szabályok kellenek a markerek eltüntetéséhez.

Világos módon erre a nyelvtanra $L(G') = L(G)$ teljesül.

Ha G 0-ás típusú nyelvtan volt, akkor kész vagyunk.

Ha G 1-es típusú nyelvtan volt, akkor minden szabály hossznemcsökkentő, azaz minden levezetés 1-korlátolt volt. Tehát G'-ben minden szó levezetésére adható egy felső korlát. Az új munkaterület mérete a régi munkaterület méreténél 3-mal több az S, L és R jelek miatt. $u \neq \varepsilon$ D'-ben. Tehát $\mathrm{WS}(D') \leq \mathrm{l}(u) + 3 \leq 4\,\mathrm{l}(u)$ teljesül.

$u = \varepsilon$ esetén az $S \to \varepsilon$ szabály egy szabályként szerepel a mátrix-nyelvtanban, KES-szabály.

ε levezetésénél 4 a munkaterület mérete és ekkor igaz is a 4-es korlát a hullám tulajdonság miatt. Tehát G' 4-korlátolt nyelvtan $\Longrightarrow L(G')$ 1-es típusú.

3.3. Időváltozós nyelvtanok

Eddig a nyelvtanokat az egész levezetés során állandó szabályhalmaznak tekintettük. Ebben az alfejezetben a nyelvtanokat időben változó objektumok lesznek, minden pillanatban más szabályok érvényesek, de összességében feltesszük, hogy csak véges sok szabály van.

PÉLDA. Az $\{a^nb^nc^n; n \geq 1\}$ nyelvtanra szeretnénk egy időváltozós nyelvtant készíteni. Jelölje $\varphi(i)$ az i-edik ütemben érvényes szabályokat, $i = 1, 2, \ldots$.

$$\begin{aligned}
\varphi(1) &= \{S \longrightarrow ABC\}\\
\varphi(2) &= \{A \longrightarrow aA, A \longrightarrow a, X \longrightarrow b\}\\
\varphi(3) &= \{B \longrightarrow bB, B \longrightarrow X\}\\
\varphi(4) &= \{C \longrightarrow cC, C \longrightarrow c\}\\
\varphi(5) &= \{A \longrightarrow aA, A \longrightarrow a, X \longrightarrow b\}
\end{aligned}$$

és így ismétlődik tovább periodikusan, tehát $\varphi(k) = \varphi(k+3), \forall k \geq 2$.

Időváltozós nyelvtanok esetén minden ütemben csinálni kell valamit, különben megszakad a végrehajtás. Egy adott ütem elvégzése után a következő ütemből kell egy alkalmazható szabályt kiválasztani és végrehajtani.

A periodikusság miatt elég a periódus elemeivel foglalkozni. Ekkor $L(G) = \{a^nb^nc^n; n \geq 1\} \cup \{a^{n+1}b^nc^n; n \geq 1\} \cup \{a^{n+1}b^{n+1}c^n; n \geq 1\}$.

$\varphi(2)$-höz vegyük hozzá az $X \to b$ szabályt is, $\varphi(3)$-ban a $B \to b$ szabályt cseréljük ki a $B \to X$ szabályra. $\varphi(2)$-ben lehet csak először eltérni a regularitástól, mert ennek kétszer kell sorra kerülni ahhoz, hogy a kívánt eredményt kapjuk.

Az $A \to a$ szabály alkalmazása után mindenképpen a $B \to X$ szabály végrehajtásának kell következnie, hogy $X \to b$ is végrehajtható legyen. A $C \to c$ szabályt is meg kell csinálni, hogy jó legyen.

A nyelvtan kvázi-periodikus, 2-es típusú szabályokkal, de 1-es típusú nyelvet tudunk vele generálni.

DEFINÍCIÓ. G Időváltozós nyelvtan *a következő:*

$$G = \langle T, N, \mathcal{P}, S, \varphi\rangle$$

ahol T, N és S mint korábban, $\mathcal{P}$ esetleg pontozott szabályok véges halmaza. Ilyen értelemben $p \to q$ és $p \overset{\cdot}{\longrightarrow} q$ különböző szabályoknak számítanak. $\varphi : U \to 2^{\mathcal{P}}$, ahol $U = \{1, 2, 3, \ldots\}$ az ütemek halmaza. 1 lépéses levezetésekre igaz a következő: $(i-1, \alpha) \underset{G(,ac)}{\longrightarrow} (i, \beta) \iff \exists p \overset{(\cdot)}{\longrightarrow} q \in \varphi(i) : \alpha \overset{p \to q}{\longrightarrow} \beta \Big(\vee$előfordul p baloldala a mintában $\vee \exists p \overset{\cdot}{\longrightarrow} q \in \varphi(i), p \notin \alpha, \alpha = \beta\Big)$.

Több lépésben: $(i-1, \alpha) \underset{G,ac}{\overset{*}{\longrightarrow}} (j, \beta) \iff \exists \alpha_{i-1}, \alpha_i, \ldots, \alpha_j$ szavak, $j \geq i-1 : \alpha_{i-1} = \alpha, \alpha_j = \beta \wedge (l-1, \alpha_{l-1}) \underset{G(,ac)}{\longrightarrow} (l, \alpha_l)$, ahol $l = i, \ldots, j$.

DEFINÍCIÓ. $L^{(ac)}(G) = \{u; (0, S) \underset{G(,ac)}{\overset{*}{\longrightarrow}} (j, u), u \in T^*, j \in U\}$.

Ideiglenesen vezessük be a típusokat: ha minden szabály kiterjesztett i típusú, akkor kiterjesztett i típusú időváltozós nyelvtanról beszélünk.

DEFINÍCIÓ. $\mathcal{T}_i^{(ac)}$ *i típusú időváltozós nyelvtanokban esetleg előfordulás-ellenőrzéses nyelvtanok.*

3.4. TÉTEL. *Tetszőleges L nyelvhez létezik olyan G, 3-as típusú időváltozós nyelvtan, melyre $L(G) = L$.*

Mit is mond a Church-tézis? Csak $\mathcal{L}_0$-ra – a mi definíciónk nem konstruktív.

Feltehető, hogy $\varepsilon \notin L$, KES hozzávételével megcsinálható. $T = \{t_1, \ldots, t_m\}$, $L \subset T^*$ $L = \{u_1, u_2, u_3, \ldots\}$ felsoroljuk (nem algoritmikus, hanem analízises értelemben). Azaz létezik ilyen függvény, de nem konstruktív a felsorolás.

Írjuk ki a szavait betűnként. t_i^j jelöli az i-edik szó j-edik betűjét. $t_1^1.t_1^2.\,.t_1^{|u_1|}, t_2^1.t_2^2.\ldots, \ldots$ ahol a szavak között vessző van, a betűk között pedig pont. Ekkor L szavai a következő alakba írhatók:

$$L = z_1.z_2.\ldots, z_{|u_1|+1}.\ldots, z_{|u_1|+|u_2|+1}.\ldots, \ldots.$$

Így egy végtelen sorozatot kapunk. Ha L véges, akkor bármilyen nyelvtan készíthető. Minden közönséges nyelvtan időváltozós is.

$i = 1, 2, \ldots$

$$\varphi(i) = \begin{cases} S \longrightarrow S & \\ S \longrightarrow z_i S' & z_i \text{ előtt vessző, mögötte pont,} \\ S \longrightarrow z_i & z_i \text{ előtt és mögött vessző, azaz egy betűs a szó,} \\ S' \longrightarrow z_i S' & z_i \text{ előtt és mögött pont,} \\ S' \longrightarrow z_i & z_i \text{ előtt pont és mögötte vessző,} \end{cases}$$

Vessző után elkezdi az adott ütemben érvényes szót generálni és a vessző előtt befejezi.

Tehát $L(G) = L$.

Definíció. Kvázi-periodikus*nak nevezünk egy φ függvényt, ha teljesül rá, hogy $\exists K \geq 0$ egész és $d \geq 1$ egész $\forall l \geq 1 : \varphi(K + l) = \varphi(K + l + d)$. K jelöli a periódus előtti részt, d a periódust. Speciálisan $K = 0$ esetén periodikusságról beszélünk.*

A továbbiakban csak kvázi-periodikus φ-ket engedünk meg, $\mathcal{T}$-ben is csak ezek vannak benne.

Jelölés. $\mathcal{T}_i^{(ac)}$ *jelöli az esetleg előfordulás-ellenőrzéses, i-típusú időváltozós nyelvtanokat.*

$\mathcal{T}_{2-\varepsilon}^{(ac)}$ *az olyan kettes típusú nyelvtanokat jelöli, ahol az ε-os szabályok nem megengedettek.*

3.5. tétel. $\mathcal{T}_i = \mathcal{L}_i$, *ha* $i = 0, 1, 3$.

$\mathcal{T}_i^{ac} = \mathcal{L}_i$, *ha* $i = 0, 1, 3$.

$\mathcal{L}_2 \subsetneq \tau_2 \subseteq \tau_2^{ac} = \mathcal{L}_0$.

Bizonyítás. Az állítás harmadik részének a bizonyítása: az első tartalmazásnál az $a^n b^n c^n$ példa volt arra, hogy valódi tartalmazásról van szó. A második tartalmazás triviális, míg az egyenlőséget később bizonyítjuk.

$i = 3$ $G = \langle T, N, \mathcal{P}, S, \varphi \rangle$ 3-as típusú, időváltozós nyelvtan. Keresünk egy olyan G', kiterjesztett 3-as közönséges nyelvtant, melyre $L(G') = L(G)$.

$\exists K \geq 0, l \geq 1 : \varphi(1), \varphi(2), \ldots, \varphi(K), \varphi(K+1), \ldots, \varphi(K+d)$ a különböző szabályrendszerek. Az eredeti G nyelvtan nyelvtani jeleit megindexeljük, hogy melyik ütemet hajtottam már végre. Minden $A \in N$ nyelvtani jelre A_j lesz az új nyelvtani jel, ahol j jelenti azt, hogy hányadik ütem van már kész, $j = 0, 1, \ldots, K + d$.

Ezek után a G' nyelvtan szabályai: $A_j \to uB_{j+1} \Longleftrightarrow A \to uB \in \varphi(j+1), \quad j = 0, 1, \ldots, K + d - 1$. Ez a szabály elvégzi a szabályátalakításokat és az indexnövelést is,

$B_{K+d} \to B_K$ biztosítja a kvázi-periodikus végrehajtást,

$A_j \to u \Longleftrightarrow A \to u \in \varphi(j+1)$ a termináló szabály.

Ezek a szabályok elégségesek, ha nincs előfordulás-ellenőrzés. Ha van, akkor még hozzá kell venni egy szabályt a korábbi szabályokhoz. A j-edik ütemben eljutunk egy A nyelvtani jelhez és azt szeretnénk, hogy a $j+1$-edik ütemben ne változzon meg az A. Ehhez az kell, hogy legyen olyan pontozott szabály, melyben szerepel az A.

$A_j \to A_{j+1} \Longleftrightarrow \exists B \overset{\cdot}{\longrightarrow} \cdots \in \varphi(j+1), A \neq B, \quad j = 0, 1, \ldots, K + d - 1$.

$i = 0, 1$ Új nyelvtani jeleket vezetünk be ebben az esetben, melyek az idő nyilvántartására szolgálnak: X_j, ahol $j = 0, 1, \ldots, K + d$.

Az előfordulás-ellenőrzés nélküli esetben a szabályok:

$S' \to SX_0$

$X_jp \to X_{j+1}q \iff p \to q \in \varphi(j+1), \quad j = 0, 1, \ldots, K+d-1$

$X_{K+d} \to X_K$

$X_j \to \varepsilon \quad j = 0, 1, \ldots, K+d$

$X_jZ \to ZX_j, ZX_j \to X_jZ \quad Z \in T \cup N, \quad j = 0, 1, \ldots, K+d$

Ha van előfordulás-ellenőrzés, akkor még az alábbi szabályt kell hozzávenni a fentiekhez:

$X_j \to A_0^{j+1,p} \iff \exists p \stackrel{\cdot}{\longrightarrow} q \in \varphi(j+1)$.

Az A felső indexe azt jelöli, hogy hányadik ütemben vagyunk és melyik a bal oldal, az alsó index pedig megadja, hogy hányadik illeszkedésig terjedő darabot vizsgáljuk.

A bal oldalt egy KMP mintafelismerő automatával vizsgáljuk. Ezt a korábbiakhoz hasonlóan kell megkonstruálni.

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

Ha G 0-ás típusú időváltozós nyelvtan, akkor G' kiterjesztett 0-ás típusú.

Ha G 1-es típusú időváltozós nyelvtan, akkor G' nem előfordulás-ellenőrzéses esetben 2-korlátolt, előfordulás-ellenőrzéses esetben 4-korlátolt (a fentieken kívül még L-et és R-et is el kell vinni ε-ba).

3.4. Programozott nyelvtanok

Az így megadott nyelvtanok nem *go-to* mentesek, amelyek Dijkstra óta nem számítanak helyes programnak.

Példa. Az $\{a^nb^nc^n; n \geq 1\}$ nyelvet szeretnénk generálni egy programozott nyelvtan segítségével. Minden szabályhoz tartozik egy címke, mely az első oszlopban található, illetve egy címke, hogy az adott szabály végrehajtása után mely szabályra kell továbblépni. Ahol több címke szerepel, ott bármelyiket választhatom. Tehát az így kapott nyelv nem determinisztikus, de mint tudjuk, a program sem determinisztikus.

A szabályok:

$$\begin{array}{lll} f_1: & S \to ABC & f_2, f_5 \\ f_2: & A \to aA & f_3 \\ f_3: & B \to bB & f_4 \\ f_4: & C \to cC & f_2, f_5 \\ f_5: & A \to a & f_6 \\ f_6: & B \to b & f_7 \\ f_7: & C \to c & \texttt{halt} \end{array}$$

A `halt` címke jelöli, hogy egy olyan állapotba került a program, ahonnan már nem tud továbblépni.

Példa. Az $\{a^{2^n}; n \geq 0\}$ nyelv generálási szabályait adjuk meg:

$$\begin{array}{lll|l} f_1: & S \to ZZ & f_1 & f_2 \\ f_2: & Z \to S & f_2 & \{f_1, f_3\} \\ f_3: & S \to a & f_3 & \texttt{halt} \end{array}$$

A Z nyelvtani jelet akkor írom át S-re, ha már az első szabály nem alkalmazható. Ezért szükség van az előfordulás-ellenőrzés alkalmazására. Tehát a harmadik oszlopban álló szabályt alkalmazom, amíg lehet. Amikor ez már nem lehetséges, akkor a negyedik oszlopban levő szabályt alkalmazom.

A fenti példákban több belépési pont is lehet, bármelyik szabállyal el lehet kezdeni a szabályok végrehajtását. Bizonyos esetekben azonban ez értelmetlen esetekhez vezet.

DEFINÍCIÓ. *G programozott nyelvtan alatt a következőt értjük: $G = \langle T, N, \mathcal{P}, S, F, \mathrm{pr}, \varphi, \sigma \rangle$. $T, N, \mathcal{P}, S$ mint a korábbiakban, F egy véges címkehalmazt jelöl, $\mathrm{pr} : F \to \mathcal{P}$ egy projekciós függvény. $\varphi : F \to 2^F$ a pozitív rákövetkezési függvény, mely azt fogalmazza meg, hogy mit csináljunk akkor, ha tudjuk alkalmazni a szabályt. Hasonlóan $\sigma : F \to 2^F$ a negatív rákövetkezési függvény, mely akkor határozza meg, hogy mit csinálunk, ha nem lehet alkalmazni a szabályt. Vegyük észre, hogy 2^F üres is lehet, lehet, hogy semmit nem kell csinálni, ha egy szabályt tudunk alkalmazni/nem tudunk alkalmazni.*

Minden utasítás egy négyesből áll, az utasítás címkéjéből, egy $p \to q$ szabályból, egy pozitív és egy negatív rákövetkezési függvényből. Ezek után definiáljuk a levezetésfogalmat. Most nem az időpillanatot, hanem a címkét kell majd hozzárendelni, evvel fogunk indexelni.

$(f, \alpha) \underset{G}{\longrightarrow} (g, \beta) \iff \alpha \overset{\mathrm{pr}(f)}{\longrightarrow} \beta \wedge g \in \varphi(f)$,

$(f, \alpha) \underset{G,ac}{\longrightarrow} (g, \beta) \iff \alpha \overset{\mathrm{pr}(f)}{\longrightarrow} \beta \wedge g \in \varphi(f)$ vagy $\mathrm{bo}(\mathrm{pr}(f)) \nsubseteq \alpha$ és $\alpha = \beta$ és $g \in \sigma(f)$, ahol $\mathrm{bo}(\mathrm{pr}(f))$ jelöli $\mathrm{pr}(f)$ bal oldalát.

Ennek a levezetésnek elkészíthető a reflexív, tranzitív lezártja a szokásos módon, ezt jelölje $\underset{G(,ac)}{\overset{*}{\longrightarrow}}$.

A generált nyelv pedig:

$$L^{(ac)}(G) = \{u; u \in T^* \wedge \exists f \in F, h \in F : (f, S) \underset{G(,ac)}{\overset{*}{\longrightarrow}} (h, u)\}.$$

A projekciós függvény a `halt` címkére nincs definiálva.

DEFINÍCIÓ. *$\mathcal{P}_i^{(ac)}$ jelölje az i típusú programozott nyelvtanokban, esetleg előfordulás-ellenőrzéssel, elfogadott nyelvek osztályát.*

$\mathcal{P}_{2-\varepsilon}^{(ac)}$ jelöli most is azt az esetet, amikor az üres szabályok nem megengedettek a korábbiakhoz hasonlóan.

3.6. TÉTEL. *$\mathcal{P}_i = \mathcal{L}_i$, ha $i = 0, 1, 3$.*
$\mathcal{P}_i^{ac} = \mathcal{L}_i$, ha $i = 0, 1, 3$.
$\mathcal{L}_2 \subsetneq \mathcal{P}_2 \subseteq \mathcal{P}_2^{ac} = \mathcal{L}_0$.

BIZONYÍTÁS. Az utóbbi állítás három részére hasonlóan az előzőhöz az alábbiak igazak: $a^n b^n c^n$ eleme a különbségnek, tehát a tartalmazás valódi; előfordulás-ellenőrzéssel triviálisan bővebb nyelvet kapunk, míg az egyenlőséget szintén később látjuk majd be.

$i = 3$ G 3-típusú programozott nyelvtan, esetleg előfordulás-ellenőrzéssel. G' közönséges kiterjesztett 3-as típusú nyelvtan, $L(G) = L(G')$.

$\forall A \in N, f \in F$-re elkészítjük az A_f nyelvtani jeleket.

S_f $f \in F$ bármely S_f-ből el lehet indulni, S' új kezdőszimbólum, $S' \to S_f$ szabály $f \in F$.

- Az előfordulás-ellenőrzés nélküli esetben:

$$\begin{aligned} A_f \to UB_g &\iff \mathrm{pr}(f) = A \to uB \text{ és } g \in \varphi(f) \text{ és } g \in \varphi(f) \\ A_f \to u &\iff \mathrm{pr}(f) = A \to u \end{aligned}$$

- ha van előfordulás-ellenőrzés, akkor még kell az alábbi is:

$$A_f \to A_g \iff \mathrm{bo}(\mathrm{pr}(f)) \neq A \text{ és } g \in \sigma(f)$$

$i = 0, 1$ Bevezetjük az $\{X_f\}_{f \in F}$ nyelvtani jeleket, melyek azt fogják megadni, hogy melyik jelen van a vezérlés. Ebben az esetben a G' szimuláló nyelvtan szabályai az alábbiak, ahol S' az új kezdőszimbólum.

$S' \to X_f S \quad \forall f \in F$, ha nincs előfordulás-ellenőrzés.
$S' \to LX_f R \quad$ abban az esetben, ha van előfordulás-ellenőrzés.

Egy $f : p \to q \quad \varphi(f) \quad \sigma(f)$ G-beli szabály szimulációja a G' nyelvtanban a következő:
ha nincs előfordulás-ellenőrzés, akkor $X_f p \to X_g q \quad \forall g \in \varphi(f)$,
ha van előfordulás-ellenőrzés, akkor $X_f p \to X_g q \quad \forall g \in \varphi(f)$ pozitív kimenet esetén, $X_f \to X_g \quad \forall g \in \sigma(f)$, ha pr$(f)$ bal oldala nem fordul elő a mondatformában, ennek kell a hatásának lennie.
$LX_f \to LY^f$, ahol Y^f jelöli az f címkéjű szabály automatájának kezdőállapotát. A KMP automatát ugyanúgy működtetjük, mint a mátrix-nyelvtanoknál működtettük. $Y_0^f, Y_1^f, \ldots, Y^f_{|\,\mathrm{bo}(\mathrm{pr}(f))|}$ lesz pr(f) bal oldalának a KMP automatája.
Az $Y_j^f R \to X_g R$ szabályt kell alkalmazni, ha az automata eljutott a végére, $0 \leq j \leq |\,\mathrm{bo}(\mathrm{pr}(f))| - 1$ és minden $g \in \sigma(f)$-re elkészítjük ezt a szabályt.
$X_f Z \to Z X_f \quad \forall Z \in T \cup N$
$X X_f \to X_f Z \quad \forall f \in F$
$X_f \to \varepsilon$
Előfordulás-ellenőrzés esetén még a bal és a jobb véget jelző jelet el kell tüntetni, tehát az $L \to \varepsilon, R \to \varepsilon$ szabályokat kell még hozzávennünk a fentiekhez.

Az így megkonstruált nyelvtanra teljesül, hogy $L(G') = L^{(ac)}(G)$, mert minden szabályt szimulálhatunk egy megfelelő szabállyal.

Típusellenőrzés: $i = 0$ esetén G' is 0-ás típusú. $i = 1$ esetén G' vagy 2-korlátolt, ha nincs előfordulás-ellenőrzés, vagy 4-korlátolt, ha van előfordulás-ellenőrzés.

$L(G') \in \mathcal{L}_1$, ezért $L(G) \in \mathcal{L}_1$ és $L(G') = L(G)$.

3.5. Kontroll-nyelvtanok

Ebben az alfejezetben olyan nyelvtanokat szeretnénk megadni, hogy a levezetéshez tartozó címkesorozat egy speciális nyelvben van benne.

Példa. $\{a^n b^n c^n; n \geq 1\}$

$$\begin{aligned}
f_1 &: \; S \longrightarrow ABC \\
f_2 &: \; A \longrightarrow aA \\
f_3 &: \; B \longrightarrow bB \\
f_4 &: \; C \longrightarrow cC \\
f_5 &: \; A \longrightarrow a \\
f_6 &: \; B \longrightarrow b \\
f_7 &: \; C \longrightarrow c
\end{aligned}$$

A kívánt szavakat előállító ó levezetés például az $f_1 f_2 f_3 f_4 f_2 f_3 f_4 f_5 f_6 f_7$. Az összes helyes levezetés $C := f_1(f_2 f_3 f_4)^* f_5 f_6 f_7$ módon adható meg. Ezzel a kontroll nyelvvel kontrolláljuk az összes levezetést.

Példa. Tekintsük az $\{a^{2^n}; n \geq 0\}$ szavakból álló nyelvet. A szabályok legyenek az alábbiak:

$$\begin{aligned}
f_1 &: \; S \longrightarrow ZZ \\
f_1' &: \; S \stackrel{\cdot}{\longrightarrow} U \\
f_2 &: \; Z \longrightarrow S \\
f_2' &: \; Z \stackrel{\cdot}{\longrightarrow} U \\
f_3 &: \; S \longrightarrow a
\end{aligned}$$

Ha csak az f_1, f_2, f_3 szabályokat tekintjük, akkor a $(f_1^* f_2^*)^* f_3^*$-gal leírt szavak között az összes a^{2^n} alakú megtalálható, de nem csak azok. Ezért van szükség az f_1' és f_2' szabályok hozzávételére, az előfordulás-ellenőrzésre. A kontrollnyelv tehát $C := (f_1^* f_1' f_2^* f_2')^* f_3^*$.

DEFINÍCIÓ. G kontroll-nyelvtan: *$G = \langle T, N, \mathcal{P}, S, F, \mathrm{pr}, C\rangle$, ahol $\mathcal{P}$ esetleg pontozott szabályok véges halmaza,* $\mathrm{pr} : F \longrightarrow \mathcal{P}$ *a címkékhez szabályokat rendelő függvény,* $C \subseteq F^*$ *a kontroll-nyelv, reguláris.*

A levezetésfogalom: $\alpha \xrightarrow[G]{f} \beta \Longleftrightarrow \alpha \xrightarrow[G]{\mathrm{pr}(f)} \beta$

$\alpha \xrightarrow[G,ac]{f} \beta \Longleftrightarrow \alpha \xrightarrow[G]{\mathrm{pr}(f)} \beta$ vagy $(\mathrm{bo}(\mathrm{pr}(f)) \nsubseteq \alpha \wedge \mathrm{pr}(f)$ *pontozott* $\wedge\, \alpha = \beta)$

Ismét készítsük el ennek a levezetésnek a reflexív, tranzitív lezártját: címkesorozatra a levezetés $f_1 \dots f_k$ sorozat szerinti működés, $k \geq 0$.

$\alpha \xrightarrow[G(,ac)]{f_1 \dots f_k} \beta$, $\sigma = f_1 \dots f_k$, $\alpha \xrightarrow[G(,ac)]{\sigma} \beta$ *a σ kontrollszó szerinti levezetés.*

Ekkor definiáljuk az esetleg előfordulás ellenőrzéssel elfogadott kontroll-nyelvek halmazát a következő módon: $L^{(ac)}(G) = \{u; u \in T^* \wedge \sigma \in C \quad S \xrightarrow[G(,ac)]{\sigma} u\}$

DEFINÍCIÓ. i-típusú kontroll-nyelvtan: *szabályai közönséges értelemben kiterjesztett i típusúak.*

3.7. ÁLLÍTÁS. *Minden nyelv generálható 3-as típusú kontroll-nyelvtannal.*

BIZONYÍTÁS. Legyen $L \subseteq T^*$ tetszőleges nyelv, a G nyelvtant elfogadó kontroll-nyelvtan. $\forall t \in T$ jelre elkészítjük az f_t címkét. $F := \{f_t\}_{t \in T} \cup \{\#\}$. Ezek után a címkéhez tartozó szabály a következő: $f_t : S \longrightarrow tS$ az f_t címke generálja az $S \longrightarrow tS$ szabályt.

$S \xrightarrow[G]{f_{t_1} \dots f_{t_l}} t_1 \dots t_l S$

$u = t_1 \dots t_l$, $f_u := f_{t_1} \dots f_{t_l}$ módon rövidítjük, tehát $S \xrightarrow[G]{f_u} uS$.

Tegyük fel, hogy $u \in L$. $\# : S \longrightarrow \varepsilon$, $S \xrightarrow[G]{f_u} uS \xrightarrow{\#} u$, u-t az $f_u\#$ kontrollszó állítja elő.

Tehát L-et a következő kontroll-nyelv állítja elő: $C := \{f_u\}_{u \in L}\#$. Így $L(G) = L$ ismét teljesül. Magát az L-et is megengedtem kontroll-nyelvként – célszerű rá megszorítást tenni: csak 3-as típusú kontroll-nyelveket engedünk meg a kontroll-nyelvtanban.

JELÖLÉS. *$\mathcal{C}_i$ azt jelöli, hogy a kontrollnyelvtant i típusú nyelvtannal generáltuk, $\mathcal{C}_i^{ac}$ azt jelenti, hogy még előfordulás-ellenőrzést is alkalmaztunk. $\mathcal{C}_{2-\varepsilon}^{(ac)}$ ε-mentes 2-es nyelvtan esetleg előfordulás-ellenőrzéssel.*

3.8. TÉTEL. *$i = 0, 1, 3$ esetén $\mathcal{C}_i = \mathcal{C}_i^{ac} = \mathcal{L}_i$, $\quad \mathcal{L}_2 \subsetneq \mathcal{C}_2 \subseteq \mathcal{C}_2^{ac} = \mathcal{L}_0$.*

3.9. KÖVETKEZMÉNY. $\mathcal{C}_{2-\varepsilon} \subseteq \mathcal{C}_{2-\varepsilon}^{ac} \subseteq \mathcal{L}_1$.

BIZONYÍTÁS. i=2 Az első tartalmazás valódiságát a példa mutatta, a harmadik egyenlőséget pedig majd később bizonyítjuk.

$i = 3$ Minden közönséges nyelvtan tekinthető kontroll-nyelvtannak. A szabályokat akárhogyan megcímkézzük és ekkor a kontroll-nyelv az ezen címkékből képzett összes szó. $\mathcal{C}_3 \subseteq \mathcal{L}_3$-at és $\mathcal{C}_3^{ac} \subseteq \mathcal{L}_3$-at kell belátni.

$G = \langle T, N, \mathcal{P}, S, F, \mathrm{pr}, C\rangle$ 3-as típusú kontroll-nyelvtan, szimuláljuk kiterjesztett 3-as típusú közönséges nyelvtannal.

- Ha nincs előfordulás-ellenőrzés, akkor $uA \xrightarrow[G]{f} uvB \Longleftrightarrow f : A \longrightarrow vB$. $u \in L(G)$ estén $S \xrightarrow[G]{\sigma} u \wedge u \in T^* \wedge \sigma \in C$. 3-as típusú kontrollsorozat. $\sigma \in C$ ellenőrzése a legkönnyebben egy őt elfogadó véges determinisztikus automatával lehetséges.

  $\mathcal{A} = \langle Q, F, \delta, q_0, V\rangle$ véges determinisztikus automata, erre $L(\mathcal{A}) = C$.

  A G' nyelvtanban valahogy nyilvántartjuk az $\mathcal{A}$ állapotát.

G' konstrukciója: a nyelvtani jelek legyenek $N' := \{B_q\}, B \in N, q \in Q$ minden nyelvtani jelnek tetszőleges állapottal címkézett változata előfordul. A kezdőjel G'-ben S_{q_0}. Léptetés: $f : A \to vB \iff A_q \to vB_{\delta(q,f)}$; $f : A \to v \iff A_q \to v \wedge \delta(q,f) \in V \quad \forall q$ esetén, melyre teljesül az utóbbi feltétel.

- Ha van előfordulás-ellenőrzés, akkor:

  az $f : A \longrightarrow vB$ szabály nem pontozott, ugyanaz.

  $f : A \stackrel{\cdot}{\longrightarrow} vB \iff \forall q \quad (A_q \to vB_{\delta(q,f)}$ vagy $\forall C \neq A \quad C_q \to C_{\delta(q,f)})$

  $f : A \longrightarrow v$ esetén mint az előbb.

  $f : A \stackrel{\cdot}{\longrightarrow} v \quad \forall q : \delta(q,f) \in V : A_q \to v$ szabály elkészítése vagy $\forall q \in Q$ és $\forall C \neq A : C_q \to C_{\delta(q,f)}$.

  Ez a 3-as típusú nyelv szimulálja.

$i = 0, 1$ Megkonstruáljuk a megfelelő G' nyelvtant abban az esetben, ha nincs előfordulás-ellenőrzés. Azt kell nézni, hogy az adott címke az automatát milyen állapotban tartja. A nyelvtan legyen a következő alakú:

$G' = \langle T, N \cup \{X_q\}_{q \in Q} \cup \{S'\}, \mathcal{P}', S' \rangle$

$S' \to X_{q_0} S$

$X_q Z \to Z X_q, Z X_q \to X_q Z \quad \forall q \in Q, \quad \forall Z \in T \cup N$

$pX_q \to rX_{\delta(q,f)} \iff f : p \to r$. Ennek segítségével a címkének megfelelő állapotba kerül az automata is.

$X_q \to \varepsilon \iff q \in V$

Ha van előfordulás-ellenőrzés, akkor egy Knuth–Morris–Pratt féle automatát működtetünk a szabályok bal oldalaira.

Jelölés. $\mathcal{A}_f$ a bo(pr(f))-et *felismerő KMP automata*, $Y_j^{f,q}$ *pedig ennek az automatának az állapota, ahol* $f \in F, q \in Q, 0 \leq j \leq |\operatorname{bo}(\operatorname{pr}(f))|$, j *jelöli, hogy a KMP automata melyik állapotában van,* q *pedig azt, hogy a nagy automata melyik állapotban van.*

$G' = \langle T, N \cup \{X_q\} \cup \{Y_j^{f,q}\} \cup \{S'\} \cup \{L, R\}, \mathcal{P}', S' \rangle$

Ekkor a szabályok a következőképpen módosulnak:

$S' \longrightarrow L X_{q_0} S R$

$X_q Z \longrightarrow Z X_q, Z X_q \longrightarrow X_q Z$

$pX_q \to rX_{\delta(q,f)} \iff f : p \xrightarrow{(\cdot)} r$

$LX_q \to LY_0^{f,q}, \quad f : p \xrightarrow{(\cdot)} r$ ellenőrizni kell a kezdőállapottól kezdve.

$Y_j^{f,q} Z \to Z Y_{\delta+(j,Z)}^{f,q}$ a KMP automata működésének egy lépését írja le.
$Y_j^{f,q} R \to X_{\delta(q,f)} R$, $j \leq |\operatorname{bo}(\operatorname{pr}(f))|$, azaz nem találta meg.
$X_q \to \varepsilon, \quad q \in V$
$L \to \varepsilon, R \to \varepsilon.$

Ha $i = 0$, akkor G' 0-adik típusú.
Ha $i = 1$, akkor G' 4-korlátolt nyelvtan.
Tehát $L(G') = L(G)$, azaz $\mathcal{C}_3 = \mathcal{L}_3$.

3.10. Tétel. $\mathcal{M}_2^{ac} = \mathcal{T}_2^{ac} = \mathcal{P}_2^{ac} = \mathcal{C}_2^{ac} = \mathcal{L}_0$.

Bizonyítás. Legyen $\mathcal{R} \in \mathcal{M}, \mathcal{T}, \mathcal{P}, \mathcal{C}$. $\mathcal{R}_2^{ac} \subseteq R_0^{ac}$, mert minden 2-es típusú nyelvtan 0-ás típusú is egyben és $\mathcal{R}_0^{ac} = \mathcal{L}_0$ mindegyikre teljesül, így mindegyik része $\mathcal{L}_0$-nak is.

Tehát elég a másik irányt belátni. Ezt két lépésben fogjuk megtenni. Egyrészt $\mathcal{P}_2^{ac} \supseteq \mathcal{L}_0$, másrészt $\mathcal{M}_2^{ac}, \mathcal{T}_2^{ac}, \mathcal{C}_2^{ac} \supseteq \mathcal{P}_2^{ac}$.

Előbb a másodikat látjuk be: $G = \langle T, N, \mathcal{P}, S, F, \mathrm{pr}, \sigma, \varphi \rangle$ 2-es típusú, előfordulás-ellenőrzéses, programozott nyelvtan. Célunk: előfordulás-ellenőrzéses 2-es típusú mátrix, időváltozós, programozott, illetve korlátolt nyelvtannal szimulálni.

$F = \{f_1, f_2, \ldots, f_n\}$. Új jelek: $X_f, Y_f, Z_f \quad \forall f \in F$, továbbá U jelöli a zsákutcát. X_f a programozott nyelvtannak az f címkéjével jelölt programlépést kell végrehajtania $\forall f \in F$. S' az új kezdőjel.

3.5.1. Mátrixnyelvtan

$\left[S' \longrightarrow SX_f\right]$, $\forall f \in F$i-re előállítottuk a régi nyelvtan kezdőszimbólumát, és meghatározzuk, hogy melyik programlépést kell végrehajtani.

$\left[X_f \longrightarrow \varepsilon\right]$, $\forall f \in F$

Az f címkéhez tartozó programlépés végrehajtása a következő: $\left[\mathrm{pr}(f), X_f \longrightarrow X_g\right]$, $\forall g \in \sigma(f)$ a pozitív végrehajtás. Adott az f címkéjű szabály, csak akkor hajtjuk végre, amikor az f címkénél tart. $X_f \to X_g$ nem pontozott.

$\left[A \stackrel{\cdot}{\longrightarrow} U, X_f \longrightarrow X_g\right]$, $\forall g \in \varphi(f)$ a negatív végrehajtás. Ha A nincs benne, akkor végrehajtom. Ha benne van, akkor a fenti, vagy zsákutca.

3.5.2. Időváltozós nyelvtan

Az eddigi nyelvtanok kvázi-periodikusak voltak, K volt a periódus előtti rész, d pedig a periódus. Egy teljes d hosszú ciklussal egy programlépést szimulálunk.

$K = 1$, $d = 4|F|$ módon megválasztva a paramétereket egy f-hez 4 ütem tartozik.

$f_1 : 2, 3, 4, 5; \ldots; \quad f_j : 4j-2, 4j-1, 4j, 4j+1; \quad \ldots j = 1, \ldots, n$

$4n+1$-ről visszaugrunk 2-re. Az első lépés: $\widetilde{\varphi(1)} = \{S' \longrightarrow SX_f\}_{\forall f \in F}$. Ez a lépés jelenti az f_j címkéhez tartozó utasítás végrehajtásának szimulálását.

$\widetilde{\varphi(4j-2)} = \{\mathrm{pr}(f_j)\} \cup \{X_f \to Y_f\}_{\forall f \in F} \cup \{X_{f_j} \to \varepsilon\}$

$\widetilde{\varphi(4j-1)} = \{X_{f_j} \to Z_g\}_{\forall g \in \sigma(f_j)} \cup \{Y_f \to X_f\}_{\forall f \in F}$.

A lehetséges lefutása a két ütemnek együttesen: ha az első sorban az unió második tagját választom, akkor visszaírom utána a második lépésben, így összességében semmi sem változik. Ha viszont az f_j címkéjű szabályt alkalmazzunk, akkor léptetjük is. Csak akkor működik jól, ha nincs benne f_j bal oldala.

$\widetilde{\varphi(4j)} = \{\mathrm{bo}(\mathrm{pr}(f_j)) \stackrel{\cdot}{\longrightarrow} U\} \cup \{X_f \to Y_f\}_{\forall f \in F} \cup \{Z_g \to Y_g\}_{\forall g \in \sigma(f_j)}$.

$\widetilde{\varphi(4j+1)} = \{X_{f_j} \to X_g\}_{g \in \varphi(f_j)} \cup \{Y_f \to X_f\}_{f \in F}$. Ennek a két szabálynak az együttes hatása az lehet, hogy nem csináltunk semmit (azaz $\mathrm{pr}(f_j)$-t nem alkalmaztunk), vagy ha alkalmaztunk és ekkor Z indexében a következő alkalmazandó címke van.

Ha az első két lépésben alkalmaztuk a szabályt, akkor a harmadikból

- az elsőt választva zsákutcába jutunk vagy semmi nem történik, a negyedikben nem tudok mit választani,
- a másodikat választva nem alkalmazható,
- a harmadikat választva pozitívan és negatívan is nem alkalmazható.

A harmadikból a harmadikat, a negyedikből a másodikat kell ekkor alkalmaznunk.

3.5.3. Kontroll-nyelvtan

$f : \mathrm{pr}(f)\ \sigma(f)\ \varphi(f)$ módon vegyük fel az f-et.
Készítsük el az alábbi uniót: $F^+ \cup F^-$. Azaz megduplázzuk a címkéket úgy, hogy mindegyikhez hozzáveszünk egy $+$, illetve $-$ címkét.

Az új szabályok legyenek a következők: f^+ és $\mathrm{pr}(f)$, ezt $\sigma(f)^+$ és $\sigma(f)^-$-beli címkéjű szabályok követhetik, illetve f^- és $\mathrm{bo}(\mathrm{pr}(f))$, ezt $\varphi(f)^+$ és $\varphi(f)^-$-beli címkéjű szabályok követhetik.

$C := \{\alpha; \alpha \in (F^+ \cup F^-)^* \wedge \forall f \in F(f^+$-t csak $\sigma(f)^+$ és $\sigma(f)^-$-beli követheti, f^--t csak $\varphi(f)^+$ és $\varphi(f)^-$-beli követheti$\}$.

Ezzel a kontroll-nyelvvel csak a program szerinti végrehajtás megengedett, azaz $L^{ac}(G') = L^{ac}(G)$, ahol G' kontroll-, G programozott nyelvtan.

Még be kell látni, hogy C 3-as típusú nyelvtan és akkor készen is vagyunk.

Ehhez általánosabbat bizonyítunk: $L \subseteq T^*$; $\quad L = \{u; u \in T^* \wedge \mathrm{pre}(u,1) \in K, \mathrm{post}(u,1) \in V$ és $\forall t, t' : (t' \notin \mathrm{kov}(t) \to tt' \nsubseteq u)\}$

3.11. Következmény. $T \to 2^T \quad K \subseteq T, U \subseteq T$ *a rákövetkezési reláció.*

3.12. Állítás. $L \in \mathcal{L}_3$.

$L \in \mathcal{L}_{Reg}$.

Véges determinisztikus automatát készítünk L-hez. $\mathcal{A} = \langle \{q_t\}_{t\in T} \cup \{q_0, q_{\mathrm{hiba}}\}, T, \delta, q_0, \{q_t\}_{t\in V}\rangle$.

$$\delta(q_0, t) = \begin{cases} q_{\mathrm{hiba}} & t \notin K \\ q_t & t \in K \end{cases}$$

$$\delta(q_t, t') = \begin{cases} q_{t'} & t' \in \mathrm{kov}(t) \\ q_{\mathrm{hiba}} & t' \notin \mathrm{kov}(t) \end{cases}$$

$$\delta(q_{\mathrm{hiba}}, t) = q_{\mathrm{hiba}}.$$

$L(\mathcal{A}) = L$, ezzel a tétel egyik felét beláttuk.

Az $\mathcal{L}_0 \subseteq \mathcal{P}_2^{ac}$ irány belátása előtt rövid kitérőt teszünk, a bizonyítás előtt megvizsgáljuk a programozott nyelvtanok képességeit. Eddig csak szógenerálásra használtuk, de bármilyen más feladatmegoldásra is lehet használni.

Legyen $\mathcal{P}$ egy programozott nyelvtan, $G = \langle T, N, \mathcal{P}, S, F, \mathrm{pr}, \delta, \varphi\rangle$.

$$(f, \alpha) \xrightarrow[G,ac]{*} (g, \beta) \qquad \alpha, \beta \in (T \cup N)^*$$

Az alábbiakban a programok utáni első oszlopban azt tüntetjük fel, hogy mi a teendő, ha van X, illetve a másodikban azt, hogy mi a teendő, ha nincs X. Természetesen bármelyik oszlop állhat üresen.

Példa. Készítsük el az M-mel való szorzás programját!
Előfeltétel: $l_X(\alpha) = n'$.
Utófeltétel: $l_X(\beta) = Mn'$.

A feladatot megvalósító program:

$$\begin{array}{lll} f_1 : X \to Z & f_1 & f_2 \\ f_2 : Z \to X^M & f_2 & f_{ki} \end{array}$$

A programot a következő makróval hívhatjuk meg: $\mathrm{Szoroz}_M(X, f_{be}, f_{ki})$

$$\begin{array}{lll} f_{be} : X \to Z & f_{be} & f_2 \\ f_{ki} : Z \to X^M & f_2 & f_{ki} \end{array}$$

Az X-ek számát szorozzuk M-mel. A makrónak két formális paramétere van, melyek a híváskor bemásolódnak az aktuális paraméterek. f_2 lokális, egyedi címke. Minden hívásnál új, egyedi címke generálódik hozzá.

A többi változó mindig globális (ami aktuális paraméter).

PÉLDA. Készítsük el az M-mel való osztást megvalósító makrót!

$\mathrm{Oszt}_m(X, Y, f_{be}, f_{ki})$, $l_X(\alpha) = n'$.

Előfeltétel: $l_X(\beta) = \left[\frac{n'}{M}\right]$

Utófeltétel: $l_Y(\beta) = \left\{\frac{n'}{M}\right\}$ az egészrész és maradékrész szorzása M-mel.

$$
\begin{array}{lrlll}
 & f_{be}: & E \to E & g_0 & \\
 & g_0: & X \to \varepsilon & g_1 & h_0 \\
 & g_1: & X \to \varepsilon & g_2 & h_1 \\
 & & \vdots & & \\
 & g_{M-2}: & X \to \varepsilon & g_{M-1} & h_{M-2} \\
 & g_{M-1}: & X \to Q & g_0 h_{M-1} & \\
 & & \vdots & & \\
ß = 0, \ldots, M-1 & h_i: & E \to EY^i & k & \\
 & k: & Q \to X & k & f_{ki}
\end{array}
$$

M darab X levonása után egy Q hozzávétele történik meg mindig. Az, hogy a h_i halmaz eleme, az azt jelenti, hogy i-vel kongruens modulo M.

A h_i szabályoknál nem kerül semmi a második oszlopba, mert az E mindig benne van a mondatformában, hogy semmiből ne kelljen generálni.

PÉLDA. Készítsük el az összeadás műveletét megvalósító makrót, jelölje $\mathrm{Add}(X, Y, f_{be}, f_{ki})$.
$Q : l_X(\alpha) = n', l_Y(\alpha) = m'$
$R : l_X(\beta) = n' + m', l_Y(\beta) = 0$

Az egyetlen szabály, mely a megvalósításhoz kell:

$$f_{be} : Y \to X \qquad f_{be} \qquad f_{ki}$$

PÉLDA. Legyen $K \geq 0$ egész szám. Készítsük el az egyenlőségvizsgálatot megvalósító makrót, ezt jelöljük a következő módon: $\mathrm{Egyenlo}_K(X, f_{be}, f_{ki}^+, f_{ki}^-)$.

A levezetéshez használt szabályok legyenek a következők:

$$(f_{be}, \alpha) \xrightarrow[G,ac]{*} (f_{ki}^+, \beta) \Longleftrightarrow l_X(\alpha) = K \wedge \forall A \in N : l_A(\alpha) = l_A(\beta)$$

$$(f_{be}, \alpha) \xrightarrow[G,ac]{*} (f_{ki}^-, \beta) \Longleftrightarrow l_X(\alpha) \neq K \wedge \forall A \in N : l_A(\alpha) = l_A(\beta)$$

$$
\begin{array}{lrlll}
 & f_{be}: & E \to E & f_0 & \\
 & f_0: & X \to \varepsilon & f_1 & g_0 \\
 & f_1: & X \to \varepsilon & f_2 & g_1 \\
 & & \vdots & & \\
 & f_K: & X \to \varepsilon & f_{K+1}^- & g_k \\
i = 0, \ldots, K-1 & g_i: & E \to EX^i & f_{ki}^- & \\
 & g_k: & E \to EX^K & f_{ki}^+ & \\
 & f_{k+1}: & E \to EX^{K+1} & f_{ki}^- &
\end{array}
$$

Hasonlóan készíthető el a nem-egyenlőséget megvizsgáló makró is.

Ezek alapján a probléma $G \in \mathcal{G}_{kit0}$ eldöntése. Konstruáljuk meg a G' programozott nyelvtant, melyre teljesül, hogy $L(G') = L(G)$. $G = \langle T, N, \mathcal{P}, S \rangle$ alakú legyen a G nyelvtan és $S \xrightarrow[G]{*} u$, $\alpha_0 = S \xrightarrow[G]{} \alpha_1 \xrightarrow[G]{} \alpha_2 \ldots \xrightarrow[G]{} \alpha_k = u$ a levezetés.

Az $\alpha \xrightarrow[G]{} \beta$ alakú levezetés hogyan szimulálható a G' nyelvtanban?

$\gamma_1 p \gamma_2 \to \gamma_1 q \gamma_2 \qquad (\alpha = \gamma_1 p \gamma_2, \beta = \gamma_1 q \gamma_2)$, $p \to q \in \mathcal{P}$

Azt tudjuk megvizsgálni, hogy valami előfordul-e a nyelvtanban, vagy nem. Legyen $p = z_1 \dots z_r$.

A fenti szabályrendszer környezetfüggetlen! Számolásra kellene visszavezetni a levezetést! Kódoljuk a levezetéseket számokkal!

Ezek után a jeleket M-áris számrendszerben felírt számoknak tekintjük és így kódoljuk! A kódolás: $M = |T \cup N| + 1$, $\alpha \in (T \cup N)^*$, ahol a plusz 1 azért szerepel, hogy a 0-át ne kelljen hozzárendelni semmihez.

Bevezetjük a jegy függvényt is, mely megadja, hogy melyik jel melyik számjegynek felel meg:

$\text{jegy} : (T \cup N) \to \{1, \dots, M\}$.

Definiáljuk a kod függvényt az alábbi módon:

$\text{kod}(\varepsilon) = \emptyset$

$\text{kod}(uZ) = M\,\text{kod}(u) + \text{jegy}(Z)$ a rekurzív definíció.

Például $\text{kod}(S) = \text{jegy}(S)$.

Legyenek u és Z olyanok, hogy teljesül $u \in (T \cup N)^*$ és $Z \in (T \cup N)$.

Definiáljuk a kódot meghatározó kod függvény inverzét a következő módon: $\text{kod}^{-1}(x)$, ahol x nemnegatív szám:

$\text{kod}^{-1}(0) = \varepsilon$, $\text{kod}^{-1}(x) = \text{kod}^{-1}([\frac{X}{M}])\,\text{jegy}^{-1}(\{\frac{X}{M}\})$.

A kérdés az, hogy egy G-beli levezetésnek hogyan feleltetünk meg egy számolást.

$\alpha \underset{G}{\longrightarrow} \beta$ esetén $\gamma_1 p \gamma_2 \longrightarrow \gamma_1 q \gamma_2 \quad Y := \gamma_2^{-1}$. γ_2-t betűnként kell áttenni egy másik szóba, annak a végére.

Elemi lépésekre bontjuk. A legvégéről leválasztunk egy jegyet és egy másik szó végére tesszük.

X tartalmazza $\text{kod}(\alpha)$-t, $Y = \text{kod}(\gamma_2^{-1})$, $B = \text{kod}(p^{-1})$, $J = \text{kod}(q^{-1})$.

Marad $\gamma_1 p$. Legyen $B := p^{-1}$. Azért a végére került a szónak, mert a számolásnál a egy jegyet könnyű hozzátenni vagy elvenni.

Marad γ_1. A kérdés az, hogy B egyenlő-e valamelyik bal oldallal, azaz ha $p_1, \dots, p_n$ jelöli a bal oldalakat, akkor $B \stackrel{?}{=} p_1^{-1}, \dots, p_n^{-1}$. Ha igen, akkor ki kellene cserélni j-re, $j = q_1^{-1} \dots q_n^{-1}$. A q-kat a γ_1 után, az Y-okat a végéről kezdve rakjuk vissza.

Felbontjuk elemi lépésekre: a legvégéről egy jegyet (betűt) leválasztunk és egy másik szó végére tesszük.

X szám: $\text{kod}(\alpha)$-t tartalmazza.

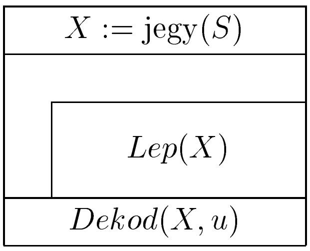
Y: $\text{kod}(\gamma_1^{-1})$

B: $\text{kod}(p^{-1})$

J: $\text{kod}(q^{-1})$.

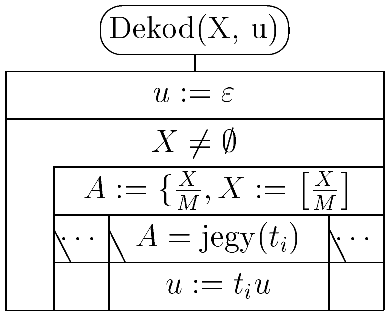
Egy lépés a kódokon:

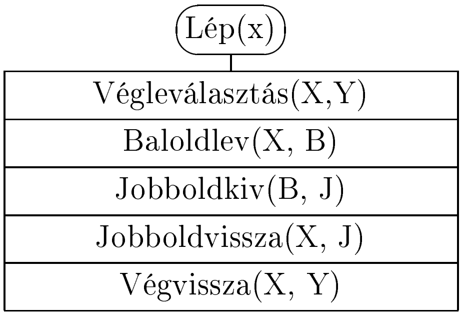
A teljes program:



X jelöli az α mondatforma kódját.

$Lep(X)$ előfeltétele, hogy $X = \text{kod}(\alpha')$, utófeltétele, hogy $X = \text{kod}(\beta')$ és $\alpha' \underset{G}{\longrightarrow} \beta'$. A ciklusinvariáns legyen az, hogy $\text{kod}^{-1}(x)$ levezethető S-ből.





A számolást egy G' 2-es típusú, előfordulás-ellenőrzéses programozott nyelvtanban is szeretnénk megvalósítani:

G legyen δ aktuális mondatformája.

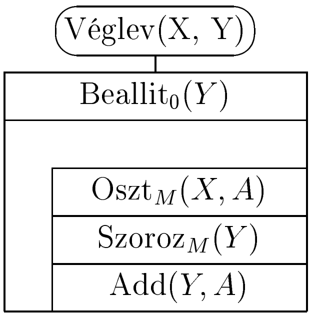
X változó reprezentációja $\Longleftrightarrow$ X nyelvtani jel G'-ben és $_X(\delta)$ az X aktuális értéke.

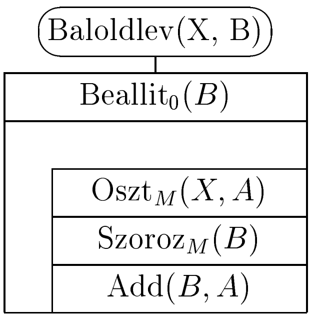
Hasonló feltételeket teszünk a többi változóra is.

Az Y, A, B, J, illetve a segédváltozóknak megfelelő nyelvtani jeleket ugyanilyen módon reprezentáljuk.

S' G' kezdőjele.

E nyelvtani jel, mely „mindig” benne van σ-ban, az első utasítás hozza be és az utolsó tünteti el.





!!! Itt még egy fél oldal hiányzik!!!

3.6. Indexelt nyelvtanok

PÉLDA. $< kif > \to < tag > \mid < tag >< addop >< tag >$

$< tag > \to < tenyezo > \mid < tenyezo >< multop >< tenyezo >$

$< tenyezo > \rightarrow < azonosito > \mid < konstans > \mid (< kif >)$

WHILE log. kif. DO .... bizonyos esetekben nem lehet akármi

FOR I=<egkif><egkif><egkif> ... END

A példákban addop jelöli az additív operátort, multop a multiplikatív operátort, logtag a logikai tagot, logaddop a logikai additív operátort.

egkif jelöli az egészkifejezést stb.

$< logkif > \rightarrow < logtag > \mid < logtag >< logaddop >< logkif >$

$< egkif > \rightarrow < egtag > \mid < egtag >< egaddop >< egkif >$

$< kif >_{log} < kif >_{val} < kif >_{eg}$ módon jelölhetjük az egyes kifejezésfajtákat, hogy ne kelljen minden szabályt és műveletet is megháromszorozni.

Ezek után meg kell adnunk azt is, hogy hogyan kell egy kifejezést kifejteni.

Az index a kifejtésnél minden nyelvtani jelre öröklődjön (későbbi szóhasználattal: teljes öröklődés):

$< kif >_{log} \rightarrow < tag >_{log} < addop >_{log} < kif >_{log}$

$< addop >_{log} \rightarrow \vee$

$< addop >_{eg} \rightarrow + \mid -$

A mondatforma minden elemének akár több indexe is lehet.

Lehet, hogy csak az első index öröklődik, azaz a legbal.

Definíció. *G indexelt nyelvtan a következő: $G = \langle T, N, \mathcal{P}, S, F \rangle$. $T, N, \mathcal{P}, S$, mint korábban. F most az indexek véges halmazát jelöli. Továbbá megköveteljük azt is, hogy $F \cap T = \emptyset = F \cap N$. A tényleges nyelvtani jelek NF^* elemei. $\mathcal{P}$ 2-es típusú szabályokból áll, a nyelvtani jelek bennük NF^* elemei.*

A levezetés menete a következő: $\alpha \in (T \cup NF^*)^*$ mondatforma G-ben. $\sigma \in F^*$. Ekkor bevezetjük az alábbi jelöléseket:

Definíció. *$\alpha^{\leftarrow\sigma}$ jelöli az α-nak σ-val való jobbról szorzását teljes öröklődéssel.*

$\alpha^{\overset{\leftarrow\sigma}{lb}}$ jelöli az α-nek σ-val való jobbról szorzását legbal öröklődéssel.

Legyen $\alpha = Z_1 Z_2 \ldots Z_k$ alakú, ahol $Z_i \in T \cup NF^*$ mindkét esetben.

$\alpha^{\leftarrow\sigma} = Z_1 \Theta_1 Z_2 \Theta_2 \ldots Z_k \Theta_k$,

$$\Theta_i = \begin{cases} \sigma & Z_i \in NF^* \\ \varepsilon & Z_i \in T \end{cases}$$

Minden nyelvtani jel mögé egy σ-t írunk. Ezt nevezzük teljes öröklődés*nek.*

$\alpha^{\overset{\leftarrow\sigma}{lb}} = Z_1 \Theta_1 Z_2 \Theta_2 \ldots Z_k \Theta_k$,

$$\Theta_i = \begin{cases} \sigma & Z_i \in NF^* \wedge \nexists j \leq i : Z_j \in NF^* \\ \varepsilon & \textit{egyébként} \end{cases}$$

Ezek után definiáljuk a levezetésfogalmat:

$\alpha \underset{G}{\longrightarrow} \beta \iff \alpha = \alpha_1 A \sigma \alpha_2 \qquad \alpha_1, \alpha_2 \in (T \cup NF^*)^*, A \in NF^*, \sigma \in F^*$, továbbá $\beta = \alpha_1 q^{\leftarrow\sigma} \alpha_2$ és $A \rightarrow q \in \mathcal{P}$.

Minden, az A nyelvtani jel helyére helyettesített jel örökli a σ-kat.

$\alpha \underset{G,lb}{\longrightarrow} \beta \iff \alpha = \alpha_1 A \sigma \alpha_2 \qquad \alpha_1 \in T^*, \alpha_2 \in (T \cup NF^*)^*, A \in NF^*, \sigma \in F^*$, továbbá $\beta = \alpha_1 (q\alpha_2)^{\overset{\leftarrow\sigma}{lb}} \alpha_2$ és $A \rightarrow q \in \mathcal{P}$.

Ebben az esetben a $q\alpha_2$-be öröklődik a σ. Az az érdekes eset, ha q-ban nincs nyelvtani jel. Ekkor ez a sorozat át tud menni α_2-be.

Az egész fogalomból az a leglényegesebb, hogy nem csak lefelé, hanem vízszintesen is lehet öröklődni a fában.

A szokásos módon definiáljuk a lezártját is ezeknek a levezetésfogalmaknak. Ezeket jelölje $\overset{*}{\underset{G}{\longrightarrow}}$ és $\overset{*}{\underset{G,lb}{\longrightarrow}}$

A nyelvtanok által elfogadott nyelv a következő alakú:

$$L(G) = \{u; u \in T^* \wedge S \overset{*}{\underset{G}{\longrightarrow}} u\}$$

$$L^{lb}(G) = \{u; u \in T^* \wedge S \xrightarrow[G,lb]{*} u\}$$

JELÖLÉS. $\mathcal{L}_{Ind}$ *jelöli az indexelt nyelvtanokban a teljes öröklődéssel elfogadott nyelvek osztályát.*
$\mathcal{L}_{Ind}^{lb}$ *jelöli az indexelt nyelvtanokban a legbal öröklődéssel elfogadott nyelvek osztályát.*

PÉLDA. Vizsgáljuk meg azt a nyelvet, mely a következő szavakat fogadja el: $\{a^n b^n c^n; n \geq 1\}$ és továbbá azt is feltesszük, hogy teljes öröklődést alkalmazunk.

$F = \{f, g\}$

Ekkor a szabályok a következőek:

$S \to S'fg$
$S' \to S'f$ a két szabály az indexek számát növeli 1-gyel
$S' \to ABC$

$S \to S'fg \cdots \to S'f^n g \to \cdots \to Af^n gBf^n gCf^n g$, ahol $n \geq 1$. Tehát a teljes öröklődéssel bekerül minden nyelvtani jel mögé ugyanaz az index.

$Af \to aA$ mindex index az A-ra öröklődik
$Bf \to bB$ mindex index az B-re öröklődik
$Cf \to cC$ mindex index az C-re öröklődik

Az $Ag \to \varepsilon, Bg \to \varepsilon, Cg \to \varepsilon$ szabályokat a végén alkalmazzuk.

PÉLDA. Vizsgáljuk meg az $\{a^{2^n}; n \geq 0\}$ nyelvet legbal öröklődés esetén.
!!! Részletesen kidolgozva 17/2.

3.13. TÉTEL. $\mathcal{L}_2 \subsetneq \mathcal{L}_{Ind} \subseteq \mathcal{L}_{ind}^{lb} = \mathcal{L}_0$.

BIZONYÍTÁS. Az első tartalmazás triviális, mert minden kettes típusú nyelvtan 2-es típusú indexelt nyelvtan is az üres indexhalmazzal. Továbbá valódi részhalmaza, mert $a^n b^n c^n$ eleme a különbségnek.

Csak $\mathcal{L}_0 \subseteq \mathcal{L}_{Ind}^{lb}$-t kell megmutatni, mert Turing-géppel mindkettő szimulálható.

Legyen $G \in \mathcal{G}_{kit}$ tetszőleges nyelvtan, $G = \langle T, N, \mathcal{P}, S\rangle$. A G' indexelt nyelvtanra teljesül, hogy $L(G') = L(G)$.

$\forall z \in T \cup N$ elemnek feleltessük meg egy f_Z módon jelölt indexet. Ezek alapján G' konstrukciója az alábbi: $F = \{f_Z\}_{Z \in T \cup N} \cup \{g\}$. Legyen $\alpha \in (T \cup N)^*$, $\alpha = z_1 \ldots z_k$-ra $f_\alpha = f_{z_1} \ldots f_{z_k}$.

A szimuláció általános mondatformája: $If_\alpha g \Longleftrightarrow S \xrightarrow[G]{*} \alpha$.

A kezdőlépés legyen $S' \to If_S g$. Ezek után $I \to DI$, ahol D jelöli, hogy egy levezetési lépést csinálunk, illetve $I \to V$, ahol V jelöli a visszaírást.

D működése az legyen az alábbi: $\alpha \xrightarrow[G]{} \beta$, ahol a $\alpha_1 p \alpha_2$ és $\alpha_1 q \alpha_2$ szavak között kell elvégezni a megfeleltetést. Ehhez az elejéről kellene levenni a jeleket, a programozott nyelvtannál a végéről lehet. Alkalmazzuk a következő lépést $Df_\alpha gI$, $Df_Z \to DBf_Z$ szabály esetén minden $Z \in T \cup N$-re: α elejéről el tudunk vinni jeleket B végére és a maradék ekkor pontosan p-vel kezdődik.

$Df_p \to Lf_q$ esetén B-re az előbbihez hasonló előállító szabály kell.

$L \to \varepsilon$

$B \to \varepsilon$

$V_f \to zV \quad \forall z \in T$ terminálisvisszaírás.

$V_g \to \varepsilon$

Így bebizonyítottuk a tétel állítását.

3.7. Attribútum nyelvtanok

Ebben a fejezetben az a célunk, hogy az indexelt nyelvtanok egy további általánosítását adjuk meg.

Programozási nyelvekben szerzett tapasztalatainkból jól ismert az öröklődés fogalma. Ha magunk elé képzeljük a szintaxisfát, akkor például a legbalság azt jelenti, hogy a legbal, vagy a következő utáni jel örökli majd a megfelelő attribútumot.

PÉLDA. Egy példán keresztül szemléltessük egy program felépítését! Jelölje P a programot, D a deklarációs részt, mely azonosítók egy listája, melyek egymástól pontosvesszővel vannak elválasztva, V pedig a végrehajtható részt jelentse. A jelöli az azonosítót, B pedig az azonosítóvéget. Ekkor a szabályok a következők:

$$\begin{array}{ll}
P \to D\#V & \\
D \to A & D \to A; D \\
V \to \varepsilon & V \to A; V \\
A \to < betu > & A \to < betu > B \\
B \to < betu > B & B \to < szamjegy > B \\
B \to < betu > & B \to < szamjegy > \\
< betu > \to a & \text{minden ábécé beli elemre} \\
< szamjegy > \to 0 & \text{minden számjegyre}
\end{array}$$

A nyelvtani jelek most rekord típusok nevei lesznek.

P, D, V egy mezőből álló rekordok, ennek típusa *sztabla* (a szimbólumtábla szóból). *sztabla* egy olyan speciális halmaztípus, mely lehetséges azonosítók véges részhalmazainak halmazát tartalmazza. Hivatkozás rájuk: P.sztabla, D.sztabla, V.sztabla. Továbbá megköveteljük azt is, hogy P.sztabla=D.sztabla=V.sztabla is teljesüljön.

$< betu >$ nyelvtani jelek tulajdonsága: egy mezőből áll, típusa $\{a, b, \dots\}$. Mezőneve: bet. Tehát a $< betu > \to a$ szabály alakja a következő lesz: $< betu > .bet = a$.

$< szjegy >$ szintén egy mezőből áll, típusa $\{0, \dots, 9\}$, mezőneve *jegy*. A $< szjegy > \to 0$ szabály megfelelője: $< szjegy > .jegy = 0$.

A: egy mezőből állt, típusa alfa-numerikus sorozat, mezőneve *szo*. $A \to < betu >$ $A.szo =< betu > .bet$
$A \to < betu > B$ $A.szo =< betu > .betB.szo$

B: egy mezőből állt, típusa alfa-numerikus sorozat, mezőneve szo.

$$\begin{array}{ll}
B \to < betu > B & B^1.szo =< betu > .betB^2.szo \\
B \to < betu > & B.szo =< betu > .bet \\
B \to < szjegy > & B.szo =< szjegy > .jegy \\
B \to < szjegy > B & B^1.szo =< szjegy > .jegyB^2.jegy
\end{array}$$

$$\begin{array}{ll}
D \to A & D.sztabla = A.szo \\
D \to A; D & a.szo \notin D^1.sztabla \wedge D^1.sztabla = D^2.sztabla \cup \{A.szo\} \\
V \to \varepsilon & \\
V \to A; V & V^1.sztabla = V^2.sztabla \wedge A.szo \in V^1.sztabla
\end{array}$$

PÉLDA. Nézzük meg egy konkrét példán is!

DEFINÍCIÓ. G *attribútumnyelvtan a következő:* $G = \langle T, N, \mathcal{P}, S \rangle$. *$N$ most is a nyelvtani jeleket jelöli, mely rekordtípus nevekből áll. E_i a $mezo_i$ attribútumhoz tartozó attribútum tartomány. Az $E_1 \times \cdots \times E_l$ direktszorzat elemei, a hozzájuk tartozó megfelelő attribútumnevek pedig $mezo_1, \dots, mezo_l$. $\mathcal{P}$ szabályok véges halmaza. Minden szabály egy párból (két részből) áll: a pár első tagja egy közönséges 2-es típusú szabály, a pár második tagja egy elsőrendű logikai formula. Tekintsük az $A \to BAt\dots$ szabályt. A formulában a szabad változók szerepét $A.mezo_i$ alakú jelek töltik be. Ha több azonos jel van egy szabályban, akkor a jeleket megsorszámozzuk. Ekkor a fenti szabály helyett $A^1 \to BA^2t\dots$ szerepel és a formulában $A^j.mezo_i$ módon hivatkozhatunk rá.*

Az így megadott nyelvtan hogyan fogad el nyelvet?

szintaxis fa: közönséges szabályok segítségével építhető fel, ahogy azt korábban tanultuk

szemantikus fa: veszünk egy szintaxis fát, majd a nyelvtani jellel címkézett pontjait, ami egy rekordtípus neve, kiértékeljük: minden nyelvtani jel mellé odaírjuk egy lehetséges értékét a szintaxisfában. Tehát szemantikus fa = szintaxis fa + kiértékelés.

helyes szemantikus fa: minden olyan részfára, mely egy belső pont és annak gyerekei, azaz egy szabályalkalmazás teljesül, hogy véve a szabályhoz tartozó logikai formulát, azt kiértékeljük a fából vett értékekkel és ez minden logikai formulára igazra értékelődik ki, azaz minden hozzárendelés jó a fában.

$L(G) = \{u; u \in T^* \wedge \exists t$ teljes szemantikus fa, hogy $gy(t) = S, front(t) = u\}$.

Példa. $a^n b^n c^n$

$$\begin{array}{ll}
S \to ABC & A.szam = B.szam + C.szam \\
A^1 \to aA^2 & A^1.szam = A^2.szam + 1 \\
B^1 \to bB^2 & B^1.szam = B^2.szam + 1 \\
C^1 \to cC^2 & C^1.szam = C^2.szam + 1 \\
A \to a & A.szam = 1 \\
B \to a & B.szam = 1 \\
C \to a & C.szam = 1
\end{array}$$

2-es típusú nyelvtannal nem fog előállni az $a^n b^n c^n$ alakú szavak összessége, bejönnek az attribútumok. Minden nyelvtani jelnek legyen egy *szam* mezőneve is, egy komponense a $[0, \ldots, \infty]$ egészek. Továbbá minden szabályhoz meg kell adni, hogy milyen formulák legyenek benne.

Adjuk meg az ua alakú szavakat elfogadó nyelvtant: $t \in \{a, b, c\}$

$$\begin{array}{ll}
B \to t & B.szo = t \\
A \to tA & A^1.szo = tA^2.szo \\
A \to t & A.szo = t \\
S \to tB & B^1.szo = tB^2.szo \\
B \to t & B.szo = t
\end{array}$$

szo $\{a, b, c, \}^*$ az attribútum.

Példa. !!!

3.14. tétel. *Ha minden attribútum tartomány véges, akkor ezt csak a kettes típusú nyelvtan tudja. Mert ekkor a nyelvtani jelhez rendeljünk hozzá annyi nyelvtani jelet, amennyi az összes lehetséges kiértékelése.*

$A \Rightarrow \{A_a\}_{a \in E_1 \times \cdots \times E_l}$

$A \to B_1 B_2 \ldots B_l$

$\Phi(\ldots A.mezo_i)$

$A_a \to B_{1b_1} \ldots B_{lb_l} \in \mathcal{P}'$. Nagyon sok ilyen szabályt csinálhatok, azokat veszem be, melyekre a $b_1, \ldots, b_l$ teljesíti Φ-t.

Csak szintaxissal ki tudjuk fejezni a szemantikus szabályt úgy, hogy benne van a kiértékelés is.

3.8. Kétszintű nyelvtanok

Példa. $< azonositosorozat > \to < azonosito >$

$< azonositosorozat > \to < azonosito >< azonositosorozat >$

$< betusorozat > \to < betu >$

$< betusorozat > \to < betu >< betusorozat >$

$< szamsorozat > \to < szam >$
$< szamsorozat > \to < szam >< szamsorozat >$

Sémát szeretnénk hozzá definiálni, hogy ne kelljen minden egyes lehetséges típusra egy hasonló szabálysorozatot megadni.

Ehhez definiálunk a hiper szinten szabályokat, melyek azt adják meg, hogy mit lehet behelyettesíteni a sémákba:
$\overline{valami} \to szam$
$\overline{valami} \to betu$
$\overline{valami} \to azonosito$

Ezek után pedig szeretnénk megadni a séma szintet is, azaz a metanyelvtant, mely a sémákat tartalmazza:
$< \overline{valami}sorozat > \to < \overline{valami} >$
$< \overline{valami}sorozat > \to < \overline{valami} >< \overline{valami}sorozat >$

A sémából a konkrét szabályt példányosítással kapjuk vissza úgy, hogy kiértékeljük a hipernyelvtant jeleket ($\overline{valami}$), és az eredményül kapott valamit írjuk minden előfordulása helyére.

Gyakran rögzített hosszú sorozatokat kell nézni. Ekkor például az alábbi szabályrendszerrel kezelhetjük az öt hosszúságú sorozatokat:
<öt hosszú azonosító sorozat> $\to$ <azonosító><négy hosszú azonosító sorozat>
<négy hosszú azonosító sorozat> $\to$ <azonosító><három hosszú azonosító sorozat>
<három hosszú azonosító sorozat> $\to$ <azonosító><kettő hosszú azonosító sorozat>
<kettő hosszú azonosító sorozat> $\to$ <azonosító><egy hosszú azonosító sorozat>
<egy hosszú azonosító sorozat> $\to$ <azonosító>

Ezt a következő módon általánosíthatom, ahol $\overline{n}$-et n darab pálcikának tekintem:
$<\overline{n}|$ hosszú $\overline{valami}$ sorozat$>$ $\to$ $<\overline{valami}><\overline{n}$ hosszú $\overline{valami}$ sorozat$>$

$<|$ hosszú $\overline{valami}$ sorozat$>$ $\to$ $<\overline{valami}>$

Így már akármit le tudunk írni ezzel a sorozattal, és tetszőleges hosszú is lehet ez.

A két szabály ekkor:
$\overline{n} \to |\overline{n}$
$\overline{n} \to |$
ahol a hiperszinten $\overline{n}$ a nyelvtani jel.

MEGJEGYZÉS. A `CDL, Compiler Description Language nyelv leírására találták ki.`

A hiperszinteken csak 3-as, a metaszinteken csak 2-es típusú szabályokat engedünk meg.

DEFINÍCIÓ. *Egy G* kétszintű nyelvtan *alatt a következő párt értjük: $G = \langle G_1, G_2 \rangle$, ahol G_1 és G_2 is egy-egy nyelvtan.*

G_1-et nevezzük a hipernyelvtannak, $G_1 = \langle HT, HN, H\mathcal{P} \rangle$ és $G_1 \in \mathcal{G}_{kit3}$. A hipernyelvtan mindig kiterjesztett 3-as típusú, hiperterminálisokból, hiper nyelvtani jelekből és hiperszabályokból áll, de nincs kezdőjele. Továbbá teljesül az is, hogy $\forall A \in HN : L(G_1, A) = \{\alpha; \alpha \in HT^* \wedge A \xrightarrow[G]{*} \alpha\}$, minden hipernyelvtani jelre, mint kezdőszimbólumra.

$G_2 = \langle T, N, \mathcal{P}, S \rangle$, T a terminális ábécé, $N \subseteq (HT \cup HN)^*$ véges halmaz, ahol $T \cap N = \emptyset$, $\mathcal{P}$ közönséges 2-es típusú szabályokat jelöl, $S \in HT^* \cap N$ a kezdőjel, megkötéssel.

Ezek után vezessük be a példányosítás fogalmát, a nyelvtani jelek lesznek a változók.

DEFINÍCIÓ. Példányosítás *alatt egy $\sigma : (HT \cup HN)^* \to (HT)^*$ értünk, melyre $\sigma(A) \in L(G_1, A)$ $\quad \forall A \in HN$ jelre és $\sigma(ht) = ht \quad \forall ht \in HT$. σ-t kiterjesztjük $(T \cup N)^*$-ra úgy, hogy $\sigma(t) = t$ minden $t \in T$-re.*

Ekkor már a $p \to q$ szabályokra is kiterjeszthető $\sigma(\to) = \to$ módon, $\sigma(p \to q) = \sigma(p) \to \sigma(q)$ a $p \to q$ szabály σ szerinti példányosítása. Ezek után az egész szabályrendszerre, $\mathcal{P}$-re kiterjeszthető. Vesszük az összes lehetséges példányosítást, ezek összességéből készítünk egy új, $\overline{G}$ nyelvtant.

Definíció. $\overline{G} = \langle T, \{\sigma(N)\}_{\sigma \text{péld.}} \{\sigma(\mathcal{P})\}_{\sigma \text{péld.}}, S\rangle$ *a G nyelvtan* teljes példányosítása.

Ez nem nyelvtan, mert a nyelvtani jelek és a szabályok száma is lehet végtelen, de a levezetésfogalomnak van értelme. Ennek definíciója: $L(G) = L(\overline{G}) = \{u; u \in T^* \wedge S \xrightarrow[\overline{G}]{*} u\}$.

A végtelen sok jelből és szabályból mindig csak azt a véges sokat használjuk, ami az adott szóhoz kell.

Definíció. $\mathcal{L}_{\text{kétsz}}$ *jelöli a fenti értelemben levezethető nyelvtanokat.*

Példa. $\{a^n b^n c^n; n \geq 0\}$
!!! 21/2

3.15. tétel. $\mathcal{L}_{\text{kétsz.}} = \mathcal{L}_0$.

Bizonyítás. Elég $\mathcal{L}_0 \subseteq \mathcal{L}_{\text{kétsz.}}$ belátása, mert a másik irány következik a Church-tézisből.