Formális nyelvek II.
Környezetfüggetlen nyelvek és veremautomaták

Fülöp Zoltán

SZTE TTIK Informatikai Inézet
Számítástudomány Alapjai Tanszék
6720 Szeged, Árpád tér 2.

Környezetfüggetlen nyelvek

Ismétlés: Egy $G = (N, \Sigma, P, S)$ nyelvtan környezetfüggetlen, ha minden szabálya $A \to \alpha$ alakú. Egy L nyelv környezetfüggetlen, ha van olyan G környezetfüggetlen nyelvtan, amelyre $L = L(G)$. Az összes környezetfüggetlen nyelvek osztályát CF-fel jelöljük.

Példák környezetfüggetlen nyelvtanokra és nyelvekre:

1) Az $S \to aSb \mid \varepsilon$ nyelvtan, amely az $\{a^n b^n \mid n \geq 0\}$ nyelvet generálja.

2) A G_{ar} nyelvtan, melynek szabályai

- $K \to K + T \mid T$,
- $T \to T * F \mid F$,
- $F \to (K) \mid a$.

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

Környezetfüggetlen nyelvtan

3) Egy további példa a szabályos zárójelezéseket generáló nyelvtan:

$S \to SS \mid (S) \mid (\,)$

Például:
$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((\,))S \Rightarrow ((\,))(\,)$

A környezetfüggetlen nyelvtanok két legfontosabb alkalmazása:

- természetes nyelvek feldolgozása (NLP),
- programozási nyelvek szintaxisának megadása.

Az alkalmazástól függően kisebb kiterjesztések szükségesek ("majdnem környezetfüggetlen" nyelvtan).

Rövidítés: cf nyelv(tan) := környezetfüggetlen nyelv(tan)

Backus-Naur forma

A gyakorlatban (pl programozási nyelvek megadásakor) a környezetfüggetlen nyelvtan egy makró-szerű változatát használják. A neve Backus-Naur forma, röviden BNF. A jelölés tömörebb lesz, a nyelv generáló kapacitás nem változik.

1) A nemterminálisokat szögletes zárójelbe tett szavakkal adjuk meg, pl: $\langle$program$\rangle$, $\langle$ut.lista$\rangle$, a terminálisokat kövér kisbetűkkel: **if**, **while**.

2) A $\to$ helyett $::=$ jelet írunk.

3) "egy vagy több" makró: $\alpha \ldots$

pl $\langle \text{integer} \rangle ::= \langle \text{digit} \rangle \ldots$
$\langle \text{digit} \rangle ::= 0|1|2|3|4|5|6|7|8|9$

A mi jelölésünkkel: $\alpha \ldots$ helyett A-t írunk és felvesszük az $A \to A\alpha | \alpha$ szabályokat.

Backus-Naur forma

4) "egy vagy egy sem" (opcionális) makró: $[\alpha]$

pl $\langle$ifut$\rangle$ ::= **if** $\langle$relacio$\rangle$ **then** $\langle$ut$\rangle$ [**else**$\langle$ut$\rangle$]

A mi jelölésünkkel: $[\alpha]$ helyett A-t írunk és felvesszük az $A \to \alpha|\varepsilon$ szabályokat.

5) "több szimbólum egy egység" makró: $\{\alpha\}$

pl $\langle$ut.lista$\rangle$ ::= $\langle$ut$\rangle$ $[\{;\langle$ut$\rangle\}\ldots]$

A mi jelölésünkkel: $\{\alpha\}$ helyett A-t írunk és felvesszük az $A \to \alpha$ szabályt.

Backus-Naur forma

Fordítsuk le a sztenderd jelölésre az $L ::= S\,[\{;S\}\ldots]$ definíciót.

a)

$$L \to S[A\ldots] \\ A \to ;S$$

b)

$$L \to SB \\ B \to A\ldots|\varepsilon \\ A \to ;S$$

c)

$$L \to SB \\ B \to C|\varepsilon \\ C \to AC|A \\ A \to ;S$$

Korlátozás nélküli, bal- és jobb oldali derivációk

Korlátozás nélküli deriváció (levezetés):
$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$

Bal oldali deriváció
(α_i legbaloldalibb nemterminálisát helyettesítjük):
$\alpha_0 \Rightarrow_l \alpha_1 \Rightarrow_l \ldots \Rightarrow_l \alpha_n$

Jobb oldali deriváció
(α_i legjobboldalibb nemterminálisát helyettesítjük):
$\alpha_0 \Rightarrow_r \alpha_1 \Rightarrow_r \ldots \Rightarrow_r \alpha_n$

Korlátozás nélküli, bal- és jobb oldali derivációk

Példák.

Korlátozás nélküli, bal oldali és jobb oldali levezetések az aritmetikai kifejezéseket generáló G_{ar} nyelvtanban. (Az aláhúzott nemterminálist helyettesítjük.)

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

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

Korlátozás nélküli, bal- és jobb oldali derivációk

<u>A különböző levezetési módok kapcsolata:</u>

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 az állítás már megfordítható, vagyis igaz lesz, hogy bal oldali (jobb oldali) levezetésekkel ugyanazok a terminális szavak kaphatók meg mint korlátozás nélküli levezetésekkel. Lásd a következő dián.

Korlátozás nélküli, bal- és jobb oldali derivációk

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 levezetésre bizonyítjuk (szimmetria).

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

(b) Tfh $X \Rightarrow^* w$, vagyis $X \Rightarrow^n w$ valamely $n \geq 0$-ra.
n szerinti indukcióval megmutatjuk, hogy $X \Rightarrow_l^n w$.

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

Korlátozás nélküli, bal- és jobb oldali derivációk

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

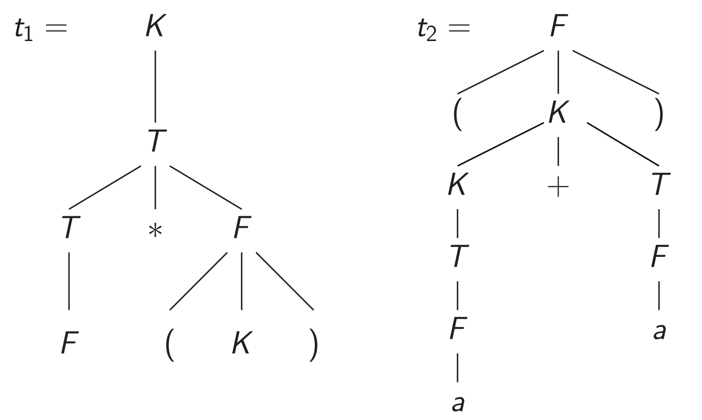
$$X \Rightarrow X_1 \ldots X_k \Rightarrow^n w_1 w_2 \ldots w_k = w.$$

1) Egyrészt, $X \rightarrow X_1 \ldots X_k \in P$.
2) Másrészt, minden $1 \leq i \leq k$-ra teljesül $X_i \Rightarrow^{n_i} w_i$, $n_i \leq n$.
Tehát, az I. F. szerint, minden $1 \leq i \leq k$-ra $X_i \Rightarrow_l^{n_i} w_i$ is fennáll. A kettőt összerakva kapjuk, hogy

$$\begin{array}{lll} X & \Rightarrow_l & X_1 X_2 \ldots X_k \\ & \Rightarrow_l^{n_1} & w_1 X_2 \ldots X_k \\ & \Rightarrow_l^{n_2} & w_1 w_2 \ldots X_k \\ & \ldots & \\ & \Rightarrow_l^{n_k} & w_1 w_2 \ldots w_k = w, \ \text{vagyis} \ \ X \Rightarrow_l^* w. \end{array}$$

Erősen kihasználjuk, hogy $w_1, \ldots, w_k \in \Sigma^*$. Különben a fenti levezetés nem bal oldali levezetés lenne!

Derivációs fák, kapcsolatuk a derivációkkal



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

Derivációs fák, kapcsolatuk a derivációkkal

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

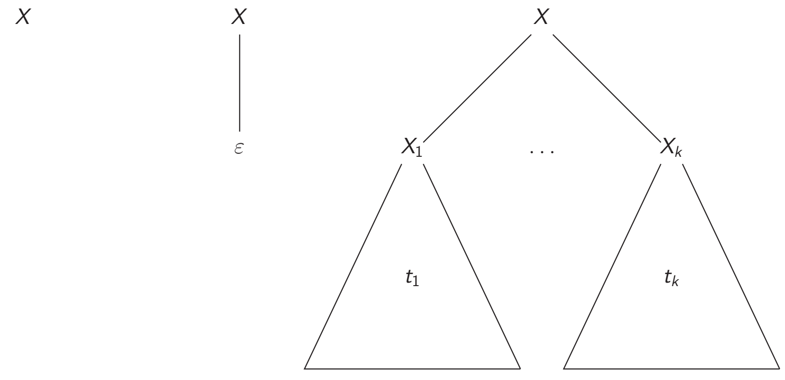
(i) Az a fa, amelynek egyetlen szögpontja (vagyis csak gyökere) az X, eleme D_X-nek.

(ii) Ha $X \to \varepsilon \in P$, akkor az a fa, amelynek gyökere X, a gyökerének egyetlen leszármazottja az ε, eleme D_X-nek.

(iii) Ha $X \to X_1 \dots X_k \in P$, továbbá $t_1 \in D_{X_1}, \dots, t_k \in D_{X_k}$, akkor az a fa, amelynek gyökere X, a gyökérből k él indul rendre a $t_1, \dots, t_k$ fák gyökeréhez, eleme D_X-nek.

Megjegyzés: ha $X \in \Sigma$, akkor az (ii) és (iii) pontok nem eredményeznek további fákat, tehát ekkor $D_X = \{X\}$.

Derivációs fák, kapcsolatuk a derivációkkal



X $\qquad$ $X[\varepsilon]$ $\qquad$ $X[t_1, \dots, t_k]$

Derivációs fák, kapcsolatuk a derivációkkal

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

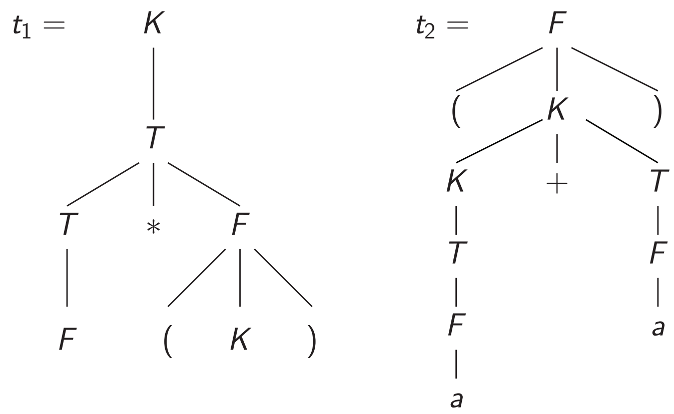
(i) Ha t az egyetlen X szögpontból álló fa, akkor $h(t) = 0$ és $fr(t) = X$.

(ii) Ha t gyökere X, aminek egyetlen leszármazottja ε, akkor $h(t) = 1$ és $fr(t) = \varepsilon$.

(iii) Ha t gyökere X, amiből k él indul rendre a $t_1, \dots, t_k$ közvetlen részfák gyökeréhez, akkor $h(t) = 1 + max\{h(t_i) \mid 1 \le i \le k\}$ és $fr(t) = fr(t_1) \dots fr(t_k)$.

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

Derivációs fák, kapcsolatuk a derivációkkal



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

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

Derivációs fák, kapcsolatuk a derivációkkal

A derivációs fák és a derivációk között a következő kapcsolat áll fenn:

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

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

1) $X \to X_1 \dots X_k \in P$
2) Minden $1 \leq i \leq k$ esetén $X_i \Rightarrow^{n_i} \alpha_i$, ahol $n_i \leq n$. (Mellesleg: $n = n_1 + \dots + n_k$.)

Derivációs fák, kapcsolatuk a derivációkkal

I. F: Minden $1 \leq i \leq k$-ra van olyan $t_i \in D_{X_i}$, hogy $fr(t_i) = \alpha_i$.
Legyen t az a fa, melynek gyökere X, amiből k él vezet rendre a $t_1, \dots, t_k$ közvetlen részfák gyökeréhez. Ekkor $t \in D_X$, továbbá

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

(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 $X \Rightarrow^* \alpha(= X)$.

Derivációs fák, kapcsolatuk a derivációkkal

(ii) $\underline{h(t) = n + 1}$: Ekkor t egy olyan fa, melynek gyökere X, amiből $k \geq 1$ él vezet rendre a $t_1, \dots, t_k$ közvetlen részfák gyökeréhez, ahol $t_1 \in D_{X_1}, \dots, t_k \in D_{X_k}$.

1) $X \to X_1 \dots X_k \in P$
2) Legyen minden $1 \leq i \leq k$-re $\alpha_i = fr(t_i)$.

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

Akkor

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

Derivációs fák, kapcsolatuk a derivációkkal

Legyen $G = (N, \Sigma, P, S)$ egy környezetfüggetlen nyelvtan. Az előző tételből azonnal kapjuk:

Következmény. Tetszőleges $w \in \Sigma^*$ esetén $S \Rightarrow^* w$ akkor és csak akkor, ha van olyan S gyökerű derivációs fa amelynek határa w.

Ebből és a terminális szóban végződő különböző levezetési módok ekvivalenciájából pedig:

Következmény. Tetszőleges $w \in \Sigma^*$ esetén a következő állítások ekvivalensek:

- $w \in L(G)$,
- $S \Rightarrow^* w$,
- $S \Rightarrow_l^* w$,
- van olyan S gyökerű derivációs fa amelynek határa w.

Derivációs fák, kapcsolatuk a derivációkkal

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

$$\begin{aligned} L(G) &= \{w \in \Sigma^* \mid S \Rightarrow^* w\}, \\ &= \{w \in \Sigma^* \mid S \Rightarrow_l^* w\}, \\ &= \{fr(t) \mid t \in D_S,\ fr(t) \in \Sigma^*\}. \end{aligned}$$

Derivációs fák, kapcsolatuk a derivációkkal

Definíció. Tetszőleges $x \in L(G)$ esetén x derivációs fájának nevezünk egy olyan S gyökerű derivációs fát, melynek határa x.

Egy $x \in L(G)$ szónak általában nem csak egy derivációs fája van. Tekintsük pl a szabályos zárójelezéseket generáló

$$S \rightarrow SS \mid (S) \mid ()$$

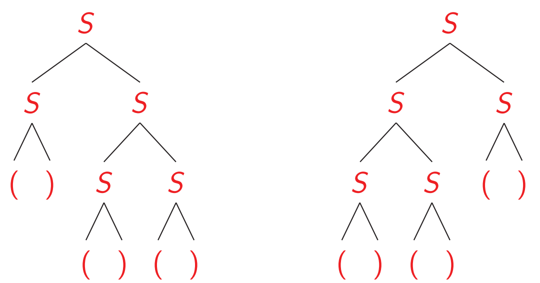
nyelvtant és az alábbi bal oldali derivációkat:

$$S \Rightarrow_l SS \Rightarrow_l ()S \Rightarrow_l ()SS \Rightarrow_l ()()S \Rightarrow_l ()()()$$

$$S \Rightarrow_l SS \Rightarrow_l SSS \Rightarrow_l ()SS \Rightarrow_l ()()S \Rightarrow_l ()()()$$

Derivációs fák, kapcsolatuk a derivációkkal

Ha megkonstruáljuk ezen derivációkhoz tartozó derivációs fákat, akkor a $()()()$ szó két különböző derivációs fáját kapjuk.



Derivációs fák, kapcsolatuk a derivációkkal

A derivációs fák és a bal oldali levezetések kapcsolata:

Vegyünk egy $x \in L(G)$ szót.

Az x szó minden derivációs fája egyértelműen meghatározza az x egy bal oldali levezetését. Ha két derivációs fa különböző, akkor a bal oldali levezetések is különbözőek lesznek.

Fordítva, x minden bal oldali levezetése egyértelműen meghatározza az x egy derivációs fáját. Különböző bal oldali levezetések különböző derivációs fákat eredményeznek.

Tehát x derivációs fái és bal oldali levezetései között bijekció áll fenn!

Egyértelmű nyelvtanok és nyelvek

Definíció. Egy G nyelvtan *egyértelmű*, ha minden $x \in L(G)$ szónak csak egy derivációs fája van.

A feltétel ekvivalens azzal, hogy minden $x \in L(G)$ szónak csak egy bal oldali levezetése van.

A természetes nyelvek nem egyértelműek. Pl "Láttam Ferit a távcsővel."

Programozási nyelveknek egyértelműeknek kell lenniük. Különben egy programnak több derivációs fája van, amiből problémák adódnak a szemantika szintjén.

Egyértelmű nyelvtanok és nyelvek

Példa. A szabályos zárójelezéseket generáló

$$S \to SS \mid (S) \mid ()$$

nyelvtan nem egyértelmű. Mint láttuk, például a $()()()$ szónak két bal oldali levezetése van:

$$S \Rightarrow_l SS \Rightarrow_l ()S \Rightarrow_l ()SS \Rightarrow_l ()()S \Rightarrow_l ()()()$$

$$S \Rightarrow_l SS \Rightarrow_l SSS \Rightarrow_l ()SS \Rightarrow_l ()()S \Rightarrow_l ()()()$$

Egyértelmű nyelvtanok és nyelvek

Példa. Tekintsük a következő, G'_{ar} nyelvtant:

- $K \to K + K$,
- $K \to K * K$,
- $K \to (K)$, $K \to a$.

Ez a nyelvtan is a már ismert aritmetikai kifejezéseket generálja. Nem egyértelmű, mert például az $a + a + a$ szónak két bal oldali levezetése is van:

$$\begin{array}{llllllll} K & \Rightarrow_l & K+K & \Rightarrow_l & a+K & \Rightarrow_l & a+K+K \\ & & & & & \Rightarrow_l^2 & a+a+a \\ K & \Rightarrow_l & K+K & \Rightarrow_l & K+K+K & \Rightarrow_l & a+K+K \\ & & & & & \Rightarrow_l^2 & a+a+a \end{array}$$

Egyértelmű nyelvtanok és nyelvek

Egy negatív eredmény:

Tétel. Nem létezik olyan algoritmus, amely tetszőleges cf nyelvtanról eldönti, hogy egyértelmű-e.

Más szóval, nem tudunk olyan programot írni, melynek inputja egy cf nyelvtan, outputja pedig **igen** vagy **nem**, attól függően, hogy az input nyelvtan egyértelmű vagy nem.

Lesznek további rossz hírek is...

Egyértelmű nyelvtanok és nyelvek

Mindkét, eddig megismert, nem egyértelmű nyelvtanhoz adható vele ekvivalens (ugyanazon nyelvet generáló) egyértelmű nyelvtan.

Az aritmetikai kifejezésekhez a már ismert G_{ar} nyelvtan.

A szabályos zárójelezésekhez pedig a következő nyelvtan:

$$\begin{array}{lcl} B & \to & (RB \mid \varepsilon \\ R & \to &) \mid (RR \end{array}$$

Pl a $()()()$ egyetlen bal oldali derivációja a következő:

$$B \Rightarrow_l (RB \Rightarrow_l ()B \Rightarrow_l ()(RB \Rightarrow_l ()()B \Rightarrow_l ()()(RB \Rightarrow_l ()()()B \Rightarrow_l ()()()$$

Egyértelmű nyelvtanok és nyelvek

Kérdés, hogy ez igaz-e általában is: megadható-e minden nem egyértelmű nyelvtanhoz vele ekvivalens egyértelmű nyelvtan.

Látni fogjuk, hogy nem: van olyan környezetfüggetlen nyelv, melyhez nem adható meg őt generáló egyértelmű nyelvtan.

Definíció. Egy L környezetfüggetlen nyelv *egyértelmű*, ha van olyan egyértelmű környezetfüggetlen G nyelvtan, melyre $L = L(G)$.

Például az $a, +, *, (,)$ szimbólumokból képezhető aritmetikai kifejezésekből álló nyelv egyértelmű, mert a G_{ar} nyelvtan egyértelmű. A szabályos zárójelezések halmaza is egyértelmű.

Az előbbi kérdés átfogalmazása: van-e nem egyértelmű környezetfüggetlen nyelv?

Egyértelmű nyelvtanok és nyelvek

Tétel. Létezik nem egyértelmű környezetfüggetlen nyelv. Például, az

$$L = \{a^i b^j c^k \mid i = j \text{ vagy } j = k\}$$

nyelv nem egyértelmű.

Próbáljuk megérteni, miért.

Egyértelmű nyelvtanok és nyelvek

$L = \{a^i b^j c^k \mid i = j \text{ vagy } j = k\}$. Tekintsük az

$$\begin{array}{lcl} S & \to & AB \mid CD \\ A & \to & aAb \mid ab \\ B & \to & cB \mid c \\ C & \to & aC \mid a \\ D & \to & bDc \mid bc \end{array}$$

nyelvtant, ami L-et generálja.

A biztosítja, hogy "a-k száma" $=$ "b-k száma". Ehhez B valamennyi c betűt hozzátesz. Hasonlóan, D biztosítja, hogy "b-k száma" $=$ "c-k száma". Ehhez C valamennyi a betűt hozzátesz. A nyelvtan tehát az $\{a^i b^j c^k \mid i = j \text{ vagy } j = k\}$ nyelvet generálja. Az $a^n b^n c^n$ alakú szavak viszont "duplán jönnek létre".

Egyértelmű nyelvtanok és nyelvek

Az $a^n b^n c^n$ alakú szavak viszont "duplán jönnek létre". Például

$S \Rightarrow_l AB \Rightarrow_l abB \Rightarrow_l abc$ és $S \Rightarrow_l CD \Rightarrow_l aD \Rightarrow_l abc$

Tehát nekünk sem sikerült egyértelmű nyelvtant megadni ;

Ez persze nem bizonyítás.

Bizonyítás. Lásd: J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison.Wesley, 1979, 4.7. fejezet (2nd. ed. 2001, 3rd. ed. 2006).
Megadnak egy (másik) cf nyelvet és megmutatják, hogy végtelen sok olyan szó van, melynek legalább két különböző derivációs fája van.

A fölösleges szimbólumok elhagyása

Definíció. Legyen $G = (N, \Sigma, P, S)$ egy környezetfüggetlen nyelvtan. Az $X \in (N \cup \Sigma)$ szimbólumról azt mondjuk, hogy *használható*, ha vannak olyan $x, y, z \in \Sigma^*$ szavak, amelyekre

$$S \Rightarrow^* xXz \Rightarrow^* xyz.$$

(Ha $X \in \Sigma$, akkor $X = y$.) Ha X nem használható, akkor *fölösleges*.

Tehát X (terminális és nemterminális) szimbólum fölösleges, ha nem jelenik megy egyetlen "sikeres" (terminálisokban végződő) derivációban sem.

A fölösleges szimbólumok elhagyása

Például, a

$$\begin{aligned} G:\ & S \to a \mid aB, \\ & A \to b, \\ & B \to Bc \end{aligned}$$

nyelvtanban

- B fölösleges, mert egyetlen B-ből kiinduló deriváció sem végződik terminális szóban (B nem "produktív")
- A is fölösleges, mert egyetlen S-ből kiinduló derivációval sem "érhető el".

A fölösleges szimbólumok elhagyása

$$\begin{aligned} G:\ & S \to a \mid aB, \\ & A \to b, \\ & B \to Bc \end{aligned}$$

Ezért

- a B-t tartalmazó $S \to aB$ és $B \to Bc$ szabályok elhagyhatók, és
- az A-t tartalmazó $A \to b$ szabály is elhagyható.

Marad:

$$G' :\ S \to a,$$

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

A fölösleges szimbólumok elhagyása

Definíció. Egy $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtanban az $A \in N$ nemterminális *produktív*, ha van olyan $x \in \Sigma^*$ szó, amelyre $A \Rightarrow^* x$.

A produktív nemterminálisok U halmazát kiszámító algoritmus:

(i) Legyen $U_1 = \{A \in N \mid \exists (A \to x \in P)$ valamely $x \in \Sigma^*$-ra$\}$ és legyen $i = 1$.

(ii) Legyen $U_{i+1} = U_i \cup \{A \in N \mid \exists (A \to \alpha \in P)$ úgy, hogy $\alpha \in (U_i \cup \Sigma)^*\}$.

(iii) Ha $U_{i+1} = U_i$, akkor álljunk meg (és ekkor $U = U_i$), különben legyen $i = i + 1$ és hajtsuk végre az (ii) lépést.

Mellesleg: $L(G) \neq \emptyset$ akkor és csak akkor, ha $S \in U$, tehát tetszőleges G cf nyelvtanról eldönthető, hogy $L(G)$ üres-e.

A fölösleges szimbólumok elhagyása

Példa Legyen G az alábbi nyelvtan:

$$\begin{array}{lcl} S & \to & A \mid B \\ A & \to & aB \mid bS \mid b \\ B & \to & AB \mid Ba \\ C & \to & AS \mid b. \end{array}$$

Ekkor:

$$U_1 = \{A, C\}, U_2 = \{S, A, C\}, U_3 = \{S, A, C\},$$

tehát B nem produktív G-ben. Az alábbi

$$\begin{array}{lcl} S & \to & A \\ A & \to & bS \mid b \\ C & \to & AS \mid b. \end{array}$$

G_1 nyelvtanra $L(G_1) = L(G)$.

A fölösleges szimbólumok elhagyása

Lemma. Ha $L(G) \neq \emptyset$, akkor megadható olyan $G_1 = (N_1, \Sigma, P_1, S)$ környezetfüggetlen nyelvtan melyre $L(G) = L(G_1)$ és minden N_1-beli nemterminális produktív.

Bizonyítás. Kiszámoljuk a produktív nemterminálisok U halmazát. Legyen $N_1 = U$ majd elhagyunk minden olyan szabályt, amelyik tartalmaz nem U-beli (vagyis nem produktív) nemterminálist.

Ez történt az előbbi példában.

A fölösleges szimbólumok elhagyása

Definíció. Egy $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtanban az $X \in (N \cup \Sigma)$ szimbólum *elérhető*, ha vannak olyan $\alpha, \beta \in (N \cup \Sigma)^*$-beli szavak, amelyekre $S \Rightarrow^* \alpha X \beta$.

Az elérhető szimbólumok H halmazát kiszámító algoritmus:

(i) Legyen $H_0 = \{S\}$ és legyen $i = 0$.

(ii) Legyen
$H_{i+1} = H_i \cup \{X \in (N \cup \Sigma) \mid$
$\exists (A \to \alpha X \beta \in P)$ úgy, hogy $A \in H_i\}$.

(iii) Ha $H_{i+1} = H_i$, akkor álljunk meg (és ekkor $H = H_i$), különben legyen $i = i + 1$ és hajtsuk végre az (ii) lépést.

A fölösleges szimbólumok elhagyása

Példa Az előbbi G_1 nyelvtanra:

$$\begin{array}{lcl} S & \to & A \\ A & \to & bS \mid b \\ C & \to & AS \mid b. \end{array}$$

G_1 nyelvtanra:

$$H_0 = \{S\}, H_1 = \{S, A\}, H_2 = \{S, A, b\}, H_3 = \{S, A, b\},$$

tehát C nem érhető el G_1-ben. Az alábbi

$$\begin{array}{lcl} S & \to & A \\ A & \to & bS \mid b. \end{array}$$

G' nyelvtanra $L(G') = L(G_1) = L(G)$.

A fölösleges szimbólumok elhagyása

Lemma. Legyen $G = (N, \Sigma, P, S)$ egy környezetfüggetlen nyelvtan. Ha Σ-ban van legalább egy elérhető terminális, akkor megkonstruálható olyan $G' = (N', \Sigma', P', S)$ környezetfüggetlen nyelvtan melyre $L(G) = L(G')$ és G'-nek minden szimbóluma elérhető.

Bizonyítás. Kiszámoljuk az elérhető szimbólumok H halmazát. Majd legyen $N' = N \cap H$, $\Sigma' = \Sigma \cap H$ majd hagyjunk el minden olyan szabályt, amelyik tartalmaz nem H-beli (vagyis nem elérhető szimbólumot).

A fölösleges szimbólumok elhagyása

Tétel. Legyen $G = (N, \Sigma, P, S)$ egy környezetfüggetlen nyelvtan. Ha $L(G) \neq \emptyset$ és Σ-ban van elérhető terminális, akkor megkonstruálható olyan $G' = (N', \Sigma', P', S)$ környezetfüggetlen nyelvtan melyre $L(G) = L(G')$ és G'-nek minden szimbóluma használható.
Bizonyítás. 1) Először számoljuk ki a (G-ben) produktív nemterminálisok U halmazát. Ha $L(G) \neq \emptyset$, akkor konstruáljuk meg a G-vel ekvivalens $G_1 = (N_1, \Sigma, P_1, S)$ nyelvtant, melynek minden nemterminálisa produktív.

2) Ezután számoljuk ki a (G_1-ben) elérhető szimbólumok H halmazát. Ha Σ-ban van legalább egy elérhető terminális (vagyis, ha $\Sigma \cap H \neq \emptyset$), akkor konstruáljuk meg $G' = (N', \Sigma', P', S)$ nyelvtant.

A kapott G' minden szimbóluma használható és $L(G') = L(G_1) = L(G)$.

Láncszabály mentesítés

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

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

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

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

halmazt.

Láncszabály mentesítés

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

$P' = \emptyset$;
minden $A \in N$-re,
minden $B \in N_A$-ra, (ahol $N_A = \{B \in N \mid A \Rightarrow^* B\}$)
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. Ha G 3-típusú, akkor G' is.

Láncszabály mentesítés

Példa láncszabálymentesítésre.

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

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

G':

- $S \to ab \mid bA \mid bB \mid a \mid bb$
- $A \to bA \mid bB \mid a \mid bb$
- $B \to bB \mid bb \mid a$
- $C \to bb$

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

Cf nyelvtanok ε-mentesítése

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

(Tehát ha egy cf nyelvtan ε-mentes, akkor az egyben környezetfüggő is. Erre még visszatérünk.)

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

Bizonyítás. G'-t két lépésben adjuk meg.

Cf nyelvtanok ε-mentesítése

Bizonyítás.
1. lépés: Először megadunk egy olyan ε-mentes $G_1 = (N, \Sigma, P_1, S)$ nyelvtant, melyre $L(G_1) = (L(G) - \{\varepsilon\})$.

Számoljuk ki a következő halmazt:

$$H = \{A \in N \mid A \Rightarrow^* \varepsilon\}.$$

(Azon nemterminálisok halmaza, amelyekből levezethető ε.)

Legyen P_1 a legszűkebb olyan szabály-halmaz, melyre teljesül, hogy

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

Cf nyelvtanok ε-mentesítése

Például, ha $A, B \in H$ és $A \to aCBbAB \in P$, akkor az $A \to aCBbAB$, $A \to aCbAB$, $A \to aCBbB$, $A \to aCBbA$, $A \to aCbB$, $A \to aCbA$, $A \to aCBb$, $A \to aCb$ szabályok mindegyike P_1-ben lesz.

Egy H-beli nemterminális elhagyásával "megelőlegezzük" azt, hogy belőle ε-t vezetünk le. Pl az $A \to aCbB$ új szabály, az

$$A \Rightarrow aCBbAB \Rightarrow^* aCbAB \Rightarrow^* aCbB$$

levezetést reprezentálja.

Hasonlóan, ha $C \to AB \in P$, akkor a $C \to A$, $C \to B \in P_1$, de $C \to \varepsilon \notin P_1$.

Cf nyelvtanok ε-mentesítése

Az új szabályokkal az eredeti G nyelvtan levezetéseit szimuláljuk, ezen kívül egyéb levezetéseket nem kapunk. Ezért

$$L(G_1) = (L(G) - \{\varepsilon\})$$

.

2. lépés: A G_1 ismeretében a következőképpen adjuk meg G'-t

Ha $\varepsilon \notin L(G)$ (vagyis, ha $S \notin H$), akkor legyen $G' = G_1$. Különben pedig legyen

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

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

Cf nyelvtanok ε-mentesítése

Példa. ε-mentesítsük az alábbi G nyelvtant:

$$\begin{array}{lcl} S & \to & ABC \\ A & \to & BB \mid \varepsilon \\ B & \to & CC \mid a \\ C & \to & AA \mid b \end{array}$$

Látható, hogy

$$H = \{S, A, C, B\}$$

és így $\varepsilon \in L(G)$.

Cf nyelvtanok ε-mentesítése

Példa (folytatás).

1. Lépés, G_1:

$$\begin{array}{lcl} S & \to & ABC \mid AB \mid BC \mid AC \mid A \mid B \mid C \\ A & \to & BB \mid B \\ B & \to & CC \mid C \mid a \\ C & \to & AA \mid A \mid b \end{array}$$

2. Lépés, G':

Mivel $\varepsilon \in L(G)$, új kezdőszimbólum: S'
Szabályok: G_1 szabályai és még $S' \to S \mid \varepsilon$.

Cf nyelvtanok ε-mentesítése

Egy fontos észrevétel:

Ha egy cf nyelvtant ε-mentesítünk, majd a kapott nyelvtant láncszabály-mentesítjük, akkor eredményül egy ε-mentes és láncszabály-mentes nyelvtant kapunk.

(Ha az algoritmusokat nem a fenti sorrendben alkalmazzuk, akkor az eredmény általában nem láncszabály-mentes nyelvtan lesz, mint ahogy a példából is látható.)

A Chomsky normálforma

Definíció. Egy $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtan *Chomsky-normálformájú* (vagy Chomsky normálalakban van), ha ε-mentes és P-ben csak $S \to \varepsilon$, $A \to BC$ és $A \to a$ alakú szabályok vannak.

Tétel. Minden $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtanhoz van vele ekvivalens Chomsky-normálformájú $G' = (N', \Sigma, P', S)$ környezetfüggetlen nyelvtan.

Bizonyítás. Feltehetjük, hogy a G nyelvtan ε- és láncszabály mentes. Ezek után G'-t két lépésben konstruáljuk meg.

A Chomsky normálforma

1. lépés: Megadunk egy $G_1 = (N_1, \Sigma, P_1, S)$ nyelvtant, mely ekvivalens G-vel, és csak $S \to \varepsilon$, $A \to a$ és $A \to A_1 \dots A_n$ alakú szabályokat tartalmaz, ahol $a \in \Sigma$, $n \geq 2$ és $A, A_1, \dots, A_n \in N_1$.

Evégett legyen

- $N_1 = N \cup \{a' \mid a \in \Sigma\}$
- P_1 a legszűkebb halmaz, amire a következők teljesülnek:
  - ha $S \to \varepsilon \in P$, akkor $S \to \varepsilon \in P_1$
  - ha $A \to a \in P$, akkor $A \to a \in P_1$
  - ha $A \to X_1 \dots X_n \in P$, ahol $n \geq 2$, akkor $A \to X_1' \dots X_n' \in P_1$, ahol
    $$X_i' = \begin{cases} X_i, \text{ ha } X_i \in N \\ a', \text{ ha } X_i = a \in \Sigma \end{cases}$$
  - minden $a \in \Sigma$-ra legyen $a' \to a \in P_1$.

Nyilvánvaló, hogy $L(G) = L(G_1)$.

A Chomsky normálforma

2. lépés: Most megkonstruáljuk G_1-ből a kívánt $G' = (N', \Sigma, P', S)$ nyelvtant. Legyen P' a legszűkebb halmaz, amire az alábbiak teljesülnek:

- ha $S \to \varepsilon \in P_1$, akkor $S \to \varepsilon \in P'$,
- ha $A \to a \in P_1$, akkor $A \to a \in P'$,
- ha $A \to BC \in P_1$, akkor $A \to BC \in P'$,
- az $A \to A_1 \dots A_n$ $(n > 2)$ alakú szabályokat pedig "szétszedjük", ld a következő diát.

A Chomsky normálforma

- ha $A \to A_1 \dots A_n$ $(n > 2)$ $\in P_1$, akkor az

$$\begin{aligned} A &\to A_1\langle A_2 \dots A_n\rangle, \\ \langle A_2 \dots A_n\rangle &\to A_2\langle A_3 \dots A_n\rangle, \\ &\vdots \\ \langle A_{n-1}A_n\rangle &\to A_{n-1}A_n \end{aligned}$$

szabályok P'-ben vannak, ahol $\langle A_2 \dots A_n\rangle, \dots, \langle A_{n-1}A_n\rangle$ új nemterminálisok.

Legyen továbbá $N' = N_1 \cup \{$ új nemterminálisok $\}$.
Nyilvánvaló, hogy G' Chomsky-normálalakban van. A bevezetett új szabályokkal csak az eredeti szabályokat tudjuk szimulálni, ezért $L(G') = L(G_1)$.
Akinek nem tetszik $\langle A_2 \dots A_n\rangle, \langle A_3 \dots A_n\rangle, \dots, \langle A_{n-1}A_n\rangle$, írhat helyette $B_2, B_3, \dots, B_{n-1}$-et.

A Chomsky normálforma

Példa: hozzuk Chomsky normálformára az $S \to aSb \mid ab$ nyelvtant.

1. lépés:

$$\begin{aligned} S &\to a'Sb' \mid a'b' \\ a' &\to a \\ b' &\to b \end{aligned}$$

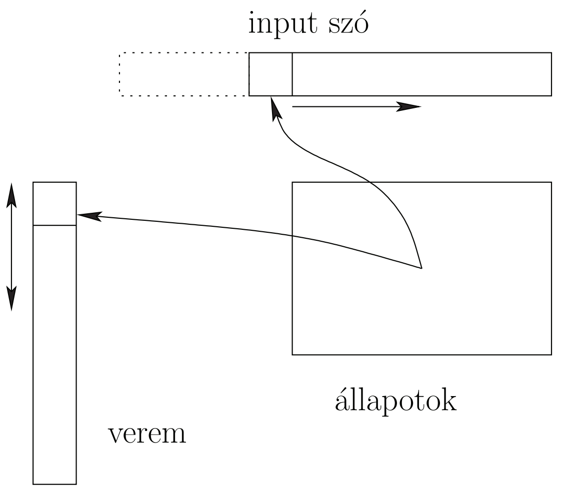
ahol a' és b' új nemterminálisok.

2. lépés:

$$\begin{aligned} S &\to a'\langle Sb'\rangle \mid a'b' \\ \langle Sb'\rangle &\to Sb' \\ a' &\to a \\ b' &\to b \end{aligned}$$

Ez már Chomsky normálforma. $\langle Sb'\rangle$ helyett írható például A.

Veremautomaták



Veremautomaták

Veremautomatának (vagy *pushdown automatának*) 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*;
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_f(Q \times \Gamma^*)$ az *átmenet függvény*.

Rövidítés: pda:= veremautomata.

Veremautomaták

Tetszőleges $q \in Q$, $a \in (\Sigma \cup \{\varepsilon\})$ 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 $\alpha_1, \ldots, \alpha_n \in \Gamma^*$. (Ha $n = 0$, akkor $\delta(q, a, Z) = \emptyset$.)
Azt jelenti, hogy ha a veremautomata q állapotban van, a inputot olvas és Z van a verem tetején, akkor átmegy egy (nemdeterminisztikusan választott) q_i állapotba és a veremben Z helyére α_i-t ír.

Két észrevétel:

1) Az alap modell nemdeterminisztikus.
2) $a = \varepsilon$ is lehetséges, ami azt fogja jelenteni, hogy a veremautomata nem minden lépésben "fogyasztja az inputot". Az ilyen mozgást ε-mozgásnak nevezzük.

Veremautomaták

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 vermet egy szónak tekintjük. A veremben felülről számított első, második, stb betű a szó első, második, stb betűje.
Például: AAZ_0 az a verem, melyben három betű van, a legfelső betű A.

A konfigurációt $(q, aw, Z\gamma)$ alakban is megadhatjuk: az input szó aw, a következő input a, (ami lehet ε is), a verem tartalma $Z\gamma$, a verem tetején Z betű van.

Veremautomaták

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

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

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

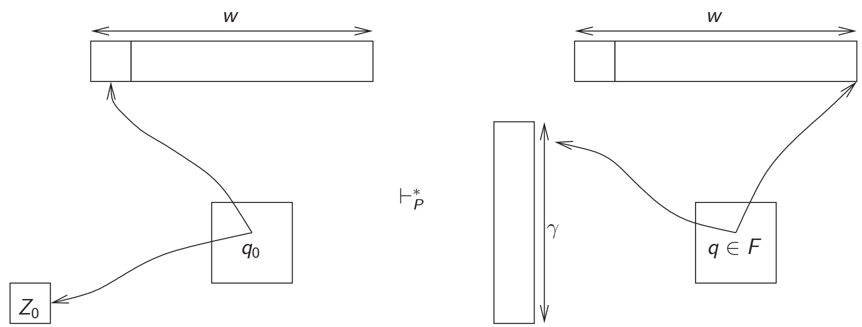
A $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ által *végállapotokkal felismert nyelv*:

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

A P által *üres veremmel felismert nyelv*:

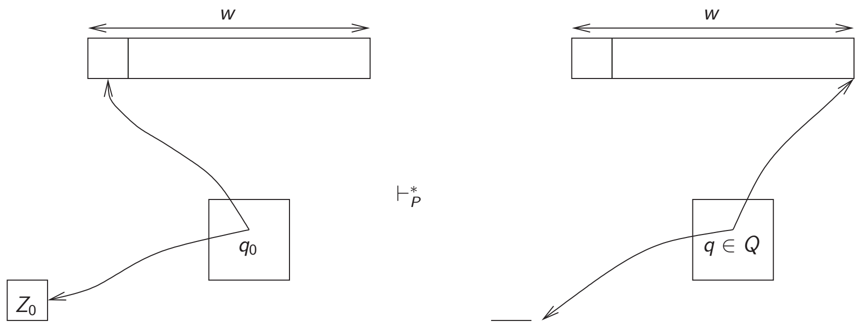
$$L_\emptyset(P) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon), \text{ ahol } q \in Q\}.$$

Veremautomaták



felismerés végállapotokkal

Veremautomaták



felismerés üres veremmel

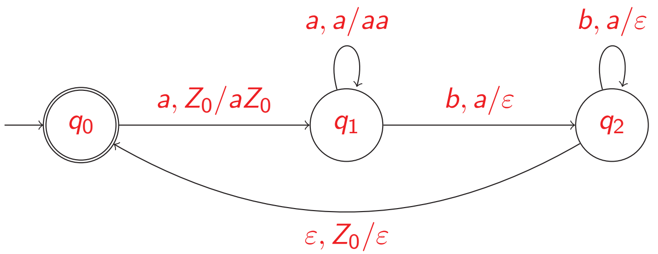
Veremautomaták

Példa $P_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, ahol

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

Veremautomaták

Ugyanez gráfként ábrázolva:



Veremautomaták

Az $aabb$ input szóra

$$\begin{array}{llllll} (q_0, aabb, Z_0) & \vdash & (q_1, abb, aZ_0) & \vdash & (q_1, bb, aaZ_0) & \vdash \\ (q_2, b, aZ_0) & \vdash & (q_2, \varepsilon, Z_0) & \vdash & (q_0, \varepsilon, \varepsilon), & \end{array}$$

tehát $aabb \in L_f(P_1)$. Az abb input szóra

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

mely utóbbi konfigurációból nem lehet tovább menni, tehát $abb \notin L_f(P_1)$. A $babb$ input szóra a

$$(q_0, babb, Z_0)$$

konfigurációból el sem lehet indulni, tehát $babb \notin L_f(P_1)$. Be lehet látni, hogy

$$L_f(P_1) = \{a^n b^n \mid n \geq 0\} \text{ és } L_\emptyset(P_1) = \{a^n b^n \mid n \geq 1\}.$$

Veremautomaták

Másik példa $P_2 = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, ahol

- $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, Z\}$, $F = \{q_2\}$,
- δ pedig a következő átmenetfüggvény:
  - $\delta(q_0, a, Z) = \{(q_0, aZ)\}$,
  - $\delta(q_0, b, Z) = \{(q_0, bZ)\}$,
  - $\delta(q_0, a, b) = \{(q_0, ab)\}$,
  - $\delta(q_0, b, a) = \{(q_0, ba)\}$,
  - $\delta(q_0, a, a) = \{(q_0, aa), (q_1, \varepsilon)\}$,
  - $\delta(q_0, b, b) = \{(q_0, bb), (q_1, \varepsilon)\}$,
  - $\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$,
  - $\delta(q_1, b, b) = \{(q_1, \varepsilon)\}$,
  - $\delta(q_1, \varepsilon, Z) = \{(q_2, \varepsilon)\}$.

Veremautomaták

Az $abba$ input szóra

$$\begin{array}{lllll} (q_0, abba, Z) & \vdash & (q_0, bba, aZ) & \vdash & (q_0, ba, baZ) \vdash \\ (q_1, a, aZ) & \vdash & (q_1, \varepsilon, Z) & \vdash & (q_2, \varepsilon, \varepsilon), \end{array}$$

tehát $abba \in L_f(P_2)$. Általában is: P_2 akkor és csak akkor ismer fel egy szót a q_2 végállapotban, ha az ww^R alakú (ahol w^R a w tükörképe), vagyis

$$L_f(P_2) = \{ww^R \mid w \in \Sigma^+\}.$$

Az is igaz, hogy a q_2 állapotba lépve kiürül a verem, tehát $L_f(P_2) = L_\emptyset(P_2)$.
Ha viszont az utolsó átmenetet megváltoztatjuk $\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$-re, akkor $L_f(P_2)$ marad ugyanaz, de $L_\emptyset(P_2) = \emptyset$ lesz, mivel a verem soha nem ürül ki.

Veremautomaták

Egy konkrét P veremautomatára általában $L_f(P) \neq L_\emptyset(P)$, lásd az előző példákat is.

Ugyanakkor érvényes a következő tétel.

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

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

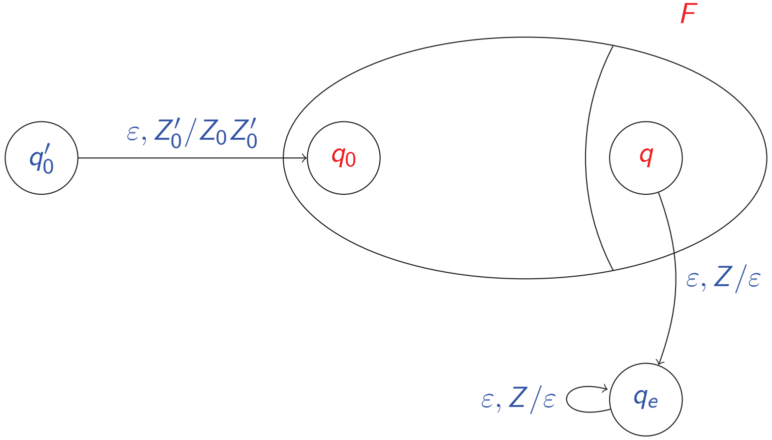
Veremautomaták

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

Konstrukció: legyen q_0' egy új kezdőállapot, q_e egy további új állapot, Z_0' pedig egy új verem kezdőszimbólum.
P' az első lépésben egy ε-mozgással átmegy az eredeti q_0 kezdőállapotba és berakja a verembe az eredeti Z_0 kezdőszimbólumot. Ezután P' utánozza P működését. Ha P végállapotba kerül, akkor P' egy ε-mozgással átmegy a q_e állapotba, amiben kiüríti a vermet. Ha P felismer, akkor P' is felismer.
Z_0'-re azért van szükség, mert ha P verme kiürül, (ami P esetében nem jelent felismerést) akkor P' vermében bennmarad Z_0', tehát P' sem ismer fel.

Veremautomaták

Ugyanez lerajzolva:



Veremautomaták

Az előző formálisan. Legyen

- $Q' = Q \cup \{q_0', q_e\}$, 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) $\delta'(q_0', \varepsilon, Z_0') = \{(q_0, Z_0Z_0')\}$,
  (2) minden $q \in Q, a \in (\Sigma \cup \{\varepsilon\}), Z \in \Gamma$ esetén

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

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

Veremautomaták

Az $L_\emptyset(P') = L_f(P)$ bizonyítása:

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

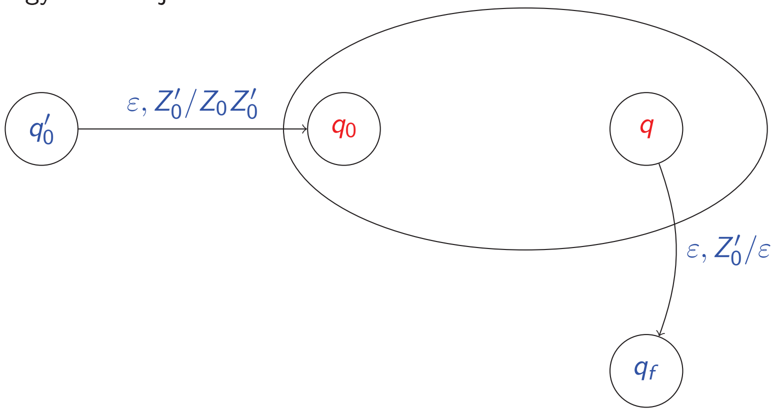
Veremautomaták

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

Konstrukció: legyen q_0' egy új kezdőállapot, q_f egy új végállapot, Z_0' pedig egy új verem kezdőszimbólum.
P' az első lépésben egy ε-mozgással átmegy az eredeti q_0 kezdőállapotba és berakja a verembe az eredeti Z_0 kezdőszimbólumot. Ezután P' utánozza P működését.
Ha P verme kiürül, akkor P' vermében bennmarad Z_0'. Ekkor P' egy ε-mozgással átmegy a q_f végállapotba. Ha P felismer, akkor P' is felismer.
Z_0'-re azért van szükség, mert ha ő van a verem tetején, akkor tudjuk, hogy P verme kiürült.

Veremautomaták

Ugyanez lerajzolva:



Veremautomaták

Az előző formálisan. Legyen

- $Q' = Q \cup \{q_0', q_f\}$, 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) $\delta'(q_0', \varepsilon, Z_0') = \{(q_0, Z_0Z_0')\}$,
  (2) minden $q \in Q, a \in (\Sigma \cup \{\varepsilon\}), Z \in \Gamma$ esetén
  $\delta'(q, a, Z) = \delta(q, a, Z)$,
  (3) minden $q \in Q$ esetén, $\delta'(q, \varepsilon, Z_0') = \{(q_f, \varepsilon)\}$.

Veremautomaták

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

$$
\begin{array}{lllll}
 & w \in L_\emptyset(P) & & & \\
\Longleftrightarrow & (q_0, w, Z_0) & \vdash_P^* & (q, \varepsilon, \varepsilon) \text{ ahol } q \in Q & \text{(def. szerint)} \\
\Longleftrightarrow & (q_0', w, Z_0') & \vdash_{P'} & (q_0, w, Z_0Z_0') & \text{((1) miatt)} \\
 & & \vdash_{P'}^* & (q, \varepsilon, Z_0') \text{ ahol } q \in Q & \text{((2) miatt)} \\
 & & \vdash_{P'} & (q_f, \varepsilon, \varepsilon) & \text{((3) miatt)} \\
\Longleftrightarrow & & & & \\
 & w \in L_f(P') & & & \text{(def. szerint).}
\end{array}
$$

A cf nyelvtanok és a pda-k ekvivalenciája

Tétel. Minden cf nyelv felismerhető pda-val.

Bizonyítás. Vegyünk egy $G = (N, \Sigma, R, S)$ környezetfüggetlen nyelvtant. Megadunk egy P pda-t, amelyre $L_\emptyset(P) = L(G)$. Legyen ez $P = (\{q\}, \Sigma, \Gamma, \delta, q, Z_0, \emptyset)$, ahol

- $\Gamma = N \cup \Sigma$,
- $Z_0 = S$,
- δ átmenetfüggvény pedig a következő módon van definiálva:
  (1) minden $A \in N$-re $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \in R\}$,
  (2) minden $a \in \Sigma$-ra $\delta(q, a, a) = \{(q, \varepsilon)\}$.

Észrevétel: egyetlen állapot elegendő!

A cf nyelvtanok és a pda-k ekvivalenciája

Az ötlet az, hogy a kapott veremautomatával a nyelvtan bal oldali levezetéseit szimuláljuk.

Ha a verem tetején egy A nemterminális van, akkor az inputban nem olvasunk tovább (ε-mozgás) és a verembe az A helyére betesszük egy A bal oldalú szabály jobb oldalát. (Pl, ha $A \to \alpha$ egy szabály, akkor A helyére α-t.) Ha a verem tetején egy a terminális van és az input első betűje is a, akkor az inputban elolvassuk, a veremből pedig töröljük a-t.

Legyen w egy input szó. Ha $w \in L(G)$, akkor, mivel a veremben kezdetben S van és a nemdeterminizmus miatt minden lehetőséget kipróbálunk, a veremautomata "meg fogja találni" w-nek egy bal oldali levezetését és ezzel kiüríti a vermet. Ha $w \notin L(G)$, akkor, S-ből nem vezethető le w, ezért e verem soha nem ürül ki.

A cf nyelvtanok és a pda-k ekvivalenciája

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

$$X \Rightarrow^* w \text{ akkor és csakis akkor, ha } (q, w, X) \vdash^* (q, \varepsilon, \varepsilon)$$

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

(i) $\underline{n = 0}$: $X = w \in \Sigma$, ezért a (2) pont szerint

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

(ii) $\underline{n \Longrightarrow n+1}$: $X \Rightarrow X_1 \dots X_k \Rightarrow^n w_1 \dots w_k = w$

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

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

A cf nyelvtanok és a pda-k ekvivalenciája

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

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

A felismerés során P a w-nek egy bal oldali levezetését szimulálja.
(b) Tfh $(q, w, X) \vdash^* (q, \varepsilon, \varepsilon)$. Az átmenetek során alkalmazott (1) típusú átmenetek száma szerinti indukcióval igazolhatjuk, hogy $X \Rightarrow^* w$.

A cf nyelvtanok és a pda-k ekvivalenciája

Példa. Tekintsük a G_{ar} nyelvtant:

- $K \to K + T \mid T$,
- $T \to T * F \mid F$,
- $F \to (K) \mid a$.

A következő $P_{ar} = (\{q\}, \Sigma, \Gamma, \delta, q, K, \emptyset)$ pda-t konstruáljuk:

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

A cf nyelvtanok és a pda-k ekvivalenciája

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

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

Ez a

$$K \Rightarrow_l K + T \Rightarrow_l^2 F + T \Rightarrow_l a + T \Rightarrow_l a + F \Rightarrow_l a + a$$

bal oldali levezetésnek felel meg.

A cf nyelvtanok és a pda-k ekvivalenciája

Tétel. Minden pda-val felismerhető nyelv környezetfüggetlen.

Bizonyítás. Legyen $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ egy pda. Megadunk egy G cf nyelvtant, amelyre $L(G) = L_\emptyset(P)$.

Legyen az $G = (N, \Sigma, R, S)$, ahol

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

Egy $[qZr]$ hármas intuitív jelentése az, hogy ha a pda q állapotban van, a verem legfelső szimbóluma Z, akkor az r állapotba jutva tudja kivenni (törölni) Z-t a veremből.

A cf nyelvtanok és a pda-k ekvivalenciája

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

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

Megmutatjuk, hogy $L(G) = L_\emptyset(P)$.

A cf nyelvtanok és a pda-k ekvivalenciája

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

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

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

$$\begin{array}{cl} & w \in L_\emptyset(P) \\ \Leftrightarrow & (q_0, w, Z_0) \vdash^* (r, \varepsilon, \varepsilon) \\ \Leftrightarrow & [q_0 Z_0 r] \Rightarrow^* w \text{ (ld. a fenti állítást)} \\ \Leftrightarrow & S \Rightarrow [q_0 Z_0 r] \Rightarrow^* w \\ \Leftrightarrow & w \in L(G) \end{array}$$

A cf nyelvtanok és a pda-k ekvivalenciája

(a) Tfh $[qZr] \Rightarrow^n w$ valamilyen $n \geq 1$-re.
(i) $\underline{n = 1}$: $[qZr] \Rightarrow w$, ez csak úgy lehet, ha $w = a \in (\Sigma \cup \{\varepsilon\})$ és $(r, \varepsilon) \in \delta(q, a, Z)$. Tehát $(q, w, Z) = (q, a, Z) \vdash (r, \varepsilon, \varepsilon)$.
(ii) $\underline{n \Longrightarrow n+1}$:

$$[qZr] \Rightarrow a[s_0 Z_1 s_1] \ldots [s_{k-1} Z_k s_k] \Rightarrow^n aw_1 \ldots w_k = w$$

1) $[qZr] \to a[s_0 Z_1 s_1] \ldots [s_{k-1} Z_k s_k] \in R$, $r = s_k$, 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 $n_i \leq n$.

A cf nyelvtanok és a pda-k ekvivalenciája

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

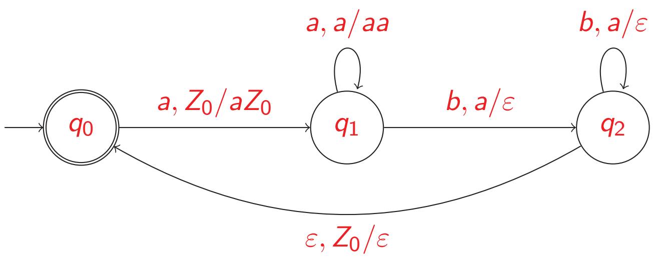
$$\begin{array}{rcl} (q, w, Z) & = & (q, aw_1 \ldots w_k, Z) \\ & \vdash & (s_0, w_1 \ldots w_k, Z_1 \ldots Z_k) \\ & \vdash^* & (s_1, w_2 \ldots w_k, Z_2 \ldots Z_k) \\ & & \ldots \\ & \vdash^* & (s_{k-1}, w_k, Z_k) \\ & \vdash^* & (s_k, \varepsilon, \varepsilon) \\ & = & (r, \varepsilon, \varepsilon) \end{array}$$

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

A cf nyelvtanok és a pda-k ekvivalenciája

Példa.

Alkalmazzuk az előző lemmában szereplő konstrukciót az $\{a^n b^n \mid n \geq 1\}$ nyelvet üres veremmel felismerő veremautomatára:



A cf nyelvtanok és a pda-k ekvivalenciája

Példa (folyt).

Egy olyan G nyelvtant kapunk, melynek szabályai a következők:

- $S \to [q_0 Z_0 q_i]$, minden $0 \leq i \leq 2$-re
- $\delta(q_0, a, Z_0) = \{(q_1, aZ_0)\}$ miatt

$$[q_0 Z_0 q_j] \to a[q_1 a q_i][q_i Z_0 q_j],$$

  minden $0 \leq i, j \leq 2$-re
- $\delta(q_1, a, a) = \{(q_1, aa)\}$ miatt

$$[q_1 a q_j] \to a[q_1 a q_i][q_i a q_j],$$

  minden $0 \leq i, j \leq 2$-re

A cf nyelvtanok és a pda-k ekvivalenciája

Példa (folyt).

- $\delta(q_1, b, a) = \{(q_2, \varepsilon)\}$ miatt $[q_1 a q_2] \to b$
- $\delta(q_2, b, a) = \{(q_2, \varepsilon)\}$ miatt $[q_2 a q_2] \to b$
- $\delta(q_2, \varepsilon, Z_0) = \{(q_0, \varepsilon)\}$ miatt $[q_2 Z_0 q_0] \to \varepsilon$.

Összesen 24 szabály!

A cf nyelvtanok és a pda-k ekvivalenciája

Példa (folyt).

Azonban könnyen belátható, hogy a fenti 24 szabály közül csak az alábbi 6 szerepel valamely $\{a, b\}^*$-beli szó levezetésében:

- $S \to [q_0 Z_0 q_0]$
- $[q_0 Z_0 q_0] \to a[q_1 a q_2][q_2 Z_0 q_0]$
- $[q_1 a q_2] \to a[q_1 a q_2][q_2 a q_2]$
- $[q_1 a q_2] \to b$
- $[q_2 a q_2] \to b$
- $[q_2 Z_0 q_0] \to \varepsilon$.

Erről a nyelvtanról már könnyen belátható, hogy az $\{a^n b^n \mid n \geq 1\}$ nyelvet generálja.

A cf nyelvtanok és a pda-k ekvivalenciája

Összefoglalás:

A következő eszközök mindegyikével pontosan a környezetfüggetlen nyelveket reprezentálhatjuk:

- környezetfüggetlen nyelvtanok,
- végállapotokkal felismerő veremautomaták,
- üres veremmel felismerő veremautomaták.

Determinisztikus veremautomaták

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

(1) $\delta(q, \varepsilon, Z) = \emptyset$ 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 $\delta(q, \varepsilon, Z)$ halmaz legfeljebb egy elemű.

A feltétel garantálja, hogy minden $(q, aw, Z\gamma)$ konfigurációhoz legfeljebb egy olyan $(p, w, \alpha\gamma)$ konfiguráció van, amelyre

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

Példa: az $\{a^n b^n \mid n \geq 0\}$ nyelvet felismerő veremautomata determinisztikus.

Determinisztikus veremautomaták

Felismerő kapacitás?

Tétel. Minden reguláris nyelv felismerhető determinisztikus veremautomatával végállapotokkal.
Bizonyítás. Legyen $M = (Q, \Sigma, \delta, q_0, F)$ egy (determinisztikus) automata. Konstruáljuk meg a $P = (Q, \Sigma, \{Z_0\}, \delta', q_0, Z_0, F)$ veremautomatát, melyre minden $p, q \in Q$ és $a \in \Sigma$ esetén

$$\delta(q, a) = p \iff \delta'(q, a, Z_0) = (p, Z_0).$$

Nyilvánvaló, hogy P determinisztikus és $L_f(P) = L(M)$.

Determinisztikus veremautomaták

Felismerő kapacitás?

Tétel. Van olyan véges nyelv, amely nem ismerhető fel determinisztikus veremautomatával üres veremmel.

Bizonyítás. Legyen például $L = \{a, ab\}$ és tegyük fel, hogy $L = L_\emptyset(P)$ valamely $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ determinisztikus veremautomatára.

Akkor $(q_0, a, Z_0) \vdash_P^* (p, \varepsilon, \varepsilon)$ és $(q_0, ab, Z_0) \vdash_P^* (q, \varepsilon, \varepsilon)$ valamely $p, q \in Q$-ra. De akkor a determinisztikusság miatt

$$(q_0, ab, Z_0) \vdash_P^* (p, b, \varepsilon) \vdash_P^* (q, \varepsilon, \varepsilon),$$

ami lehetetlen, mert üres verem esetén nincs rákövetkező konfiguráció.

Determinisztikus veremautomaták

Felismerő kapacitás?

Tehát, a determinisztikus veremautomaták esetében csak a végállapotokkal való felismerés hatékony. Ezért csak ezt vizsgáljuk. A determinisztikus veremautomatákkal végállapotokkal felismerhető nyelveket *determinisztikus környezetfüggetlen nyelveknek*, röviden csak determinisztikus nyelveknek nevezzük. A determinisztikus nyelvek osztályát DCF-fel jelöljük.

Tétel. $\mathrm{REG} \subset \mathrm{DCF}$.

Bizonyítás. Mint láttuk, minden reguláris nyelv determinisztikus, tehát $\mathrm{REG} \subseteq \mathrm{DCF}$. Az $\{a^n b^n \mid n \geq 0\}$ nyelv pedig determinisztikus, de nem reguláris, tehát $\mathrm{DCF} - \mathrm{REG} \neq \emptyset$.

Determinisztikus veremautomaták

Bizonyítás nélkül:

Tétel. Minden determinisztikus nyelv egyértelmű (vagyis generálható egyértelmű cf nyelvtannal).

Bizonyítás. (Vázlat.) Konstruáljuk meg a nyelvet felismerő determinisztikus pda-ból az ekvivalens cf nyelvtant a megismert módon. Meg lehet mutatni, hogy az így kapott nyelvtan egyértelmű.

A környezetfüggetlen és a reguláris nyelvek gépi reprezentáció

Gépi reprezentáció:

Nyelvosztály	Gépi repr.	Det. vs nemdet.
REG (reguláris nyelvek)	véges automaták	det. = nemdet. = nemdet. ε átmenetekkel
CF (k. független nyelvek)	veremautomaták	det. $\subset$ nemdet.

A valódi tartalmazást később bizonyítjuk.

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Tétel. Minden $L \subseteq \Sigma^*$ környezetfüggetlen nyelvhez

- megadható olyan (L-től függő) $k > 0$ egész szám, hogy
- minden $w \in L$ esetén,
- ha $|w| \geq k$, akkor van olyan $w = w_1 w_2 w_3 w_4 w_5$ felbontás, melyre:

1) $|w_2 w_3 w_4| \leq k$,
2) $w_2 w_4 \neq \varepsilon$,
3) 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.)

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Bizonyítás. Legyen $L = L(G)$, ahol $G = (N, \Sigma, P, S)$ Chomsky-normálformában lévő cf nyelvtan.

Legyen $k = 2^m$, ahol $m = ||N||$. Megmutatjuk, hogy ez a k megfelelő lesz.

Vegyünk egy $w \in L$ szót, melyre $|w| \geq k$. Akkor van olyan $t \in D_S$ derivációs fa, melyre $fr(t) = w$ és $h(t) \geq m + 1$.

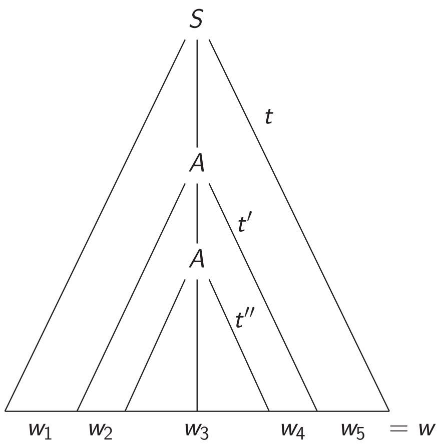
Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Ha ugyanis $h(t) \leq m$ lenne, akkor a t derivációs fa mindegyik útján legfeljebb m nemterminális lenne (a legalsó csúcsponton terminális van).

Mivel (Chomsky-normálforma miatt) a fa minden nemterminális csúcspontban legfeljebb kétfelé ágazik, a legalsó nemterminálisnál pedig már nem ágazik el, a w szó hossza legfeljebb 2^{m-1} lenne.

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

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma



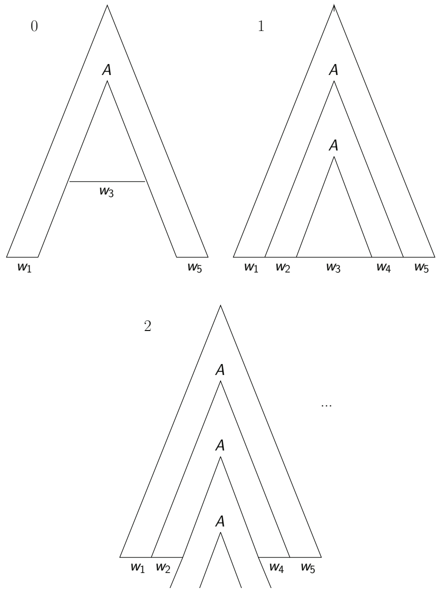
Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Mivel $m = ||N||$, van olyan nemterminális, ami ezen az úton legalább kétszer előfordul. Jelölje A azt a nemterminálist, amelyik 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 nyilván $w = w_1 w_2 w_3 w_4 w_5$, mert t határát osztottuk fel. Továbbá:

1) $|w_2 w_3 w_4| \leq k$, mert a legelső ismétlődést vettük,
2) $w_2 w_4 \neq \varepsilon$, mert az A "felső" előfordulásánál a fa kétfelé ágazik és egyik ágon sem vezethető le ε (Chomsky-normálforma!),
3) minden $n \geq 0$-ra, $w_1 w_2^n w_3 w_4^n w_5 \in L$, mert a t' fa iterálható.

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma



Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Egy következmény:

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

Bizonyítás. Tfh igen. A Bar Hillel lemma szerint van olyan $k > 0$ szám, hogy minden $w \in L$ szóra, ha $|w| \geq k$, akkor teljesülnek ezen lemmában szereplő 1) – 3) feltételek.

Vegyük az $a^k b^k c^k \in L$ szót, aminek a hossza $3k \geq k$. A pumpáló lemma szerint létezik $a^k b^k c^k = w_1 w_2 w_3 w_4 w_5$ felbontás, melyre a 2) és 3) feltétel szerint: $w_2 w_4 \neq \varepsilon$, és
minden $n \geq 0$-ra $w_1 w_2^n w_3 w_4^n w_5 \in L$.

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Tehát $a^k b^k c^k = w_1 w_2 w_3 w_4 w_5$, $w_2 w_4 \neq \varepsilon$, és
minden $n \geq 0$-ra $w_1 w_2^n w_3 w_4^n w_5 \in L$.

Hogyan helyezkedhet el w_2 és w_4 az $a^k b^k c^k$ szóban?

Sem w_2 sem w_4 nem tartalmazhat két különböző betűt, mert ekkor például a $w_1 w_2^2 w_3 w_4^2 w_5$ szóban a betűk sorrendje nem $a - b - c$ lenne!
Tehát mind w_2 mind w_4 legfeljebb egy fajta betűt tartalmaz.
Akkor viszont a $w_1 w_2^2 w_3 w_4^2 w_5$ szóban legalább egy és legfeljebb két fajta betűnek a száma több, mint a harmadik fajta betűé.
Tehát $w_1 w_2^2 w_3 w_4^2 w_5 \notin L$. ↯

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Egy informális megjegyzés:

Az $\{a^n b^n \mid n \geq 1\}$ nyelv környezetfüggetlen, de az $\{a^n b^n c^n \mid n \geq 1\}$ nyelv nem.

A környezetfüggetlen nyelvtan képes számolni "két valamit", de nem képes számolni "három valamit".

Pumpáló lemma környezetfüggetlen nyelvekre, vagy Bar–Hillel lemma

Megjegyzés: van olyan általános nyelvtan, amelyik az $L = \{a^n b^n c^n \mid n \geq 1\}$ nyelvet generálja. Szabályai a következők:

$$
\begin{array}{lll}
S \to SC_1 & S \to AC_1 & A \to aB_1 \\
A \to aAB_1 & B_1C_1 \to BC & B_1B \to BB_1 \\
CC_1 \to C_1C & B \to b & C \to c
\end{array}
$$

Egy "jó" deriváció:
$S \Rightarrow SC_1 \Rightarrow SC_1C_1 \Rightarrow AC_1C_1 \Rightarrow aAB_1C_1C_1 \Rightarrow aaB_1B_1C_1C_1 \Rightarrow aaB_1BCC_1 \Rightarrow aaBB_1CC_1 \Rightarrow aaBB_1C_1C \Rightarrow aaBBCC \Rightarrow^4 aabbcc$

Egy "rossz" deriváció:
$S \Rightarrow SC_1 \Rightarrow SC_1C_1 \Rightarrow AC_1C_1 \Rightarrow aB_1C_1C_1 \Rightarrow aBCC_1 \Rightarrow ???$

Környezetfüggetlen nyelvek zártsági tulajdonságai

Reguláris műveletek
Tétel. CF zárt a reguláris műveletekre nézve.
Bizonyítás. Legyen $L_1, L_2 \in \mathrm{CF}$, azaz legyen $G_1 = (N_1, \Sigma, P_1, S_1)$ és $G_2 = (N_2, \Sigma, P_2, S_2)$ olyan cf nyelvtan, hogy $L_1 = L(G_1)$ és $L_2 = L(G_2)$.

a) egyesítés: 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 $L(G) = L_1 \cup L_2$, tehát $L_1 \cup L_2 \in \mathrm{CF}$.

b) konkatenáció: Legyen
$G = (N_1 \cup N_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1S_2\}, S)$. Akkor $L(G) = L_1L_2$, tehát $L_1L_2 \in \mathrm{CF}$.

c) iteráció: Legyen $G = (N_1 \cup \{S\}, \Sigma, P_1 \cup \{S \to S_1S, S \to \varepsilon\}, S)$.
Akkor $L(G) = L_1^*$, tehát $L_1^* \in \mathrm{CF}$.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Boole műveletek

Tétel. CF nem zárt sem a metszetre sem a komplementerre (ugyanakkor zárt az egyesítésre).

Bizonyítás.
a) metszet: Megadunk $L_1, L_2 \in \mathrm{CF}$ nyelveket, melyekre $L_1 \cap L_2 \notin \mathrm{CF}$: az $L_1 = \{a^m b^m c^n \mid m, n \geq 0\}$ nyelv cf, mert generálható az alábbi környezetfüggetlen nyelvtannal:

$$\begin{array}{lcl} S & \to & AB \\ A & \to & aAb|\varepsilon \\ B & \to & cB|\varepsilon \end{array}$$

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.

Környezetfüggetlen nyelvek zártsági tulajdonságai

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.

Következmény. DCF nem zárt a metszetre.
Bizonyítás. Az előző bizonyításban szereplő L_1 és L_2 nyelvek valójában determinisztikus cf nyelvek és $L_1 \cap L_2 \notin \mathrm{CF}$. De akkor a $\mathrm{DCF} \subseteq \mathrm{CF}$ tartalmazás miatt $L_1 \cap L_2 \notin \mathrm{DCF}$.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Metszet reguláris nyelvvel

Tétel. CF zárt a reguláris nyelvekkel való metszésre. (Ha L cf nyelv, L' pedig reguláris nyelv, akkor $L \cap L'$ is cf nyelv.)

Bizonyítás. Legyen $L = L_f(P)$ egy környezetfüggetlen nyelv, ahol $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ egy veremautomata és $L' = L(M)$ egy reguláris nyelv, ahol $M = (Q', \Sigma, \delta', q_0', F')$ egy determinisztikus automata.

Megkonstruáljuk a $\bar{P} = (Q \times Q', \Sigma, \Gamma, \bar{\delta}, (q_0, q_0'), Z_0, F \times F')$ veremautomatát, és megmutatjuk, hogy $L \cap L' = L_f(\bar{P})$.

A $\bar{\delta} : (Q \times Q') \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_f((Q \times Q') \times \Gamma^*)$ átmenetfüggvényt a következőképpen definiáljuk:

Környezetfüggetlen nyelvek zártsági tulajdonságai

A $\bar{\delta} : (Q \times Q') \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_f((Q \times Q') \times \Gamma^*)$ átmenetfüggvényt a következőképpen definiáljuk:

- Minden $p, q \in Q$, $p', q' \in Q'$, $a \in \Sigma$, $Z \in \Gamma$ és $\gamma \in \Gamma^*$ esetén,

$$((p, p'), \gamma) \in \bar{\delta}((q, q'), a, Z)$$

  akkor és csak akkor, ha $(p, \gamma) \in \delta(q, a, Z)$ és $p' = \delta'(q', a)$,
- Minden $p, q \in Q$, $q' \in Q'$, $Z \in \Gamma$ és $\gamma \in \Gamma^*$ esetén,

$$((p, q'), \gamma) \in \bar{\delta}((q, q'), \varepsilon, Z)$$

  akkor és csak akkor, ha $(p, \gamma) \in \delta(q, \varepsilon, Z)$.

A konstrukció hasonló az automaták direkt szorzata konstrukcióhoz.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Az $L_f(\bar{P}) = L \cap L'$ bizonyítása (vázlat). Elegendő megmutatni, hogy minden $p, q \in Q$, $p', q' \in Q'$, $Z \in \Gamma$, $\gamma \in \Gamma^*$ és $x \in \Sigma^*$ esetén,

$$((q, q'), x, Z) \vdash_{\bar{P}}^* ((p, p'), \varepsilon, \gamma)$$

akkor és csak akkor, ha

$$(q, x, Z) \vdash_P^* (p, \varepsilon, \gamma) \text{ és } (q', x) \vdash_M^* (p', \varepsilon).$$

Így

$$\begin{aligned}
& x \in L(\bar{P}) \\
\iff\quad & ((q_0, q_0'), x, Z_0) \vdash_{\bar{P}}^* ((p, p'), \varepsilon, \gamma), \text{ v.mely } (p, p') \in (F \times F')\text{-re} \\
\iff\quad & (q_0, x, Z_0) \vdash_P^* (p, \varepsilon, \gamma) \text{ valamely } p \in F\text{-re} \\
& \text{és } (q_0', x) \vdash_M^* (p', \varepsilon) \text{ valamely } p' \in F'\text{-re} \\
\iff\quad & x \in L \cap L'.
\end{aligned}$$

Környezetfüggetlen nyelvek zártsági tulajdonságai

Ha P determinisztikus, akkor $\bar{P}$ is determinisztikus, tehát:

Következmény. DCF is zárt reguláris nyelvvel való metszetre.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Tétel. DCF zárt a komplementer képzésre.

Bizonyítás. (Vázlat.)

1. LÉPÉS. Egy adott $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ determinisztikus pda-hoz megkonstruálunk egy olyan $P' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F)$ determinisztikus pda-t, melyre $L_f(P) = L_f(P')$ és amely soha nem ,,akad el'', mivel minden konfigurációjára van rákövetkező: minden $q \in Q$ és $Z \in \Gamma$ esetén,

(a) vagy minden $a \in \Sigma$-ra $||\delta'(q, a, Z)|| = 1$ és $\delta'(q, \varepsilon, Z) = \emptyset$,

(b) vagy minden $a \in \Sigma$-ra $\delta'(q, a, Z) = \emptyset$ és $||\delta'(q, \varepsilon, Z)|| = 1$.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Ha egy determinisztikus pda-nak megvan a fenti tulajdonsága, még nem biztos, hogy minden input szót végig tud olvasni. Az input olvasása során kerülhet ugyanis ún. hurok-konfigurációba:

$$(q, \varepsilon, A) \vdash (q_1, \varepsilon, \alpha_1) \vdash (q_2, \varepsilon, \alpha_2) \vdash \dots$$

azaz

$$(q, v, A) \vdash (q_1, v, \alpha_1) \vdash (q_2, v, \alpha_2) \vdash \dots$$

Környezetfüggetlen nyelvek zártsági tulajdonságai

2. LÉPÉS. A hurok-konfigurációk halmaza megkonstruálható és ennek ismretében a $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ determinisztikus pda átalakítható *végig olvasóvá*:
minden $x \in \Sigma^*$ esetén van olyan $q \in Q$ állapot és $\alpha \in \Gamma^*$ szó, hogy $(q_0, x, Z_0) \vdash^* (q, \varepsilon, \alpha)$.

3. LÉPÉS. Egy végig olvasó determinisztikus $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ pda-hoz pedig már tudunk olyan P' determinisztikus pda-t konstruálni, amely az $L_f(P)$ nyelv komplementerét ismeri fel.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Ötlet: $Q' = Q \times \{0, 1, 2\}$ és P' szimulálja P-t. Továbbá:

A $(\ , 1)$ és $(\ , 0)$ alakú állapotok második komponensében P' azt az információt tárolja, hogy P volt vagy nem volt végállapotban az utolsó input elolvasása óta. A $(\ , 2)$ alakú állapotok lesznek a végállapotok.

Ha még nem volt és P már nem végezhet több ε-mozgást, akkor P' egy ε-mozgással átvált $(\ , 0)$ alakú állapotból $(\ , 2)$ alakú végállapotba. Következésképpen, ha ez a szó végén történt, akkor P' elfogadja a szót.

Ha nem a szó végén történt, akkor a következő input betű hatására P' visszavált $(\ , 1)$ vagy $(\ , 0)$ alakú állapotba attól függően, hogy P végállapotba vagy nem végállapotba érkezett.

Ily módon P' éppen $\overline{L_f(P)}$-t ismeri fel.

Környezetfüggetlen nyelvek zártsági tulajdonságai

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

- $Q' = Q \times \{0, 1, 2\}$,
- $q_0' = \begin{cases} (q_0, 0) & \text{ha } q_0 \notin F \\ (q_0, 1) & \text{ha } q_0 \in F \end{cases}$
- $F' = \{(q, 2) \mid q \in Q\}$,
- δ' a következő:

Környezetfüggetlen nyelvek zártsági tulajdonságai

- δ' a következő:
  - minden $q \in Q$, $a \in \Sigma$ és $Z \in \Gamma$-ra, ha $\delta(q, a, Z) = (p, \gamma)$, akkor
    $$\delta'((q,1), a, Z) = \delta'((q,2), a, Z) = \begin{cases} ((p,1), \gamma) & \text{ha } p \in F \\ ((p,0), \gamma) & \text{ha } p \notin F \end{cases}$$
  - minden $q \in Q$ és $Z \in \Gamma$-ra, ha $\delta(q, \varepsilon, Z) = (p, \gamma)$, akkor
    - $\delta'((q,1), \varepsilon, Z) = ((p,1), \gamma)$,
    - $\delta'((q,0), \varepsilon, Z) = \begin{cases} ((p,1), \gamma) & \text{ha } p \in F \\ ((p,0), \gamma) & \text{ha } p \notin F, \end{cases}$
  - minden $q \in Q$ és $Z \in \Gamma$-ra, ha $\delta(q, \varepsilon, Z) = \emptyset$, akkor legyen $\delta'((q,0), \varepsilon, Z) = ((q,2), Z)$.

A fentiek miatt $L_F(P') = \overline{L_f(P)}$.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Következmény. $\mathrm{DCF} \subset \mathrm{CF}$. (A determinisztikus nyelvek valódi részét képezik a környezetfüggetlen nyelveknek.)

Bizonyítás. Azt tudjuk, hogy – definíció szerint – $\mathrm{DCF} \subseteq \mathrm{CF}$. A két osztály nem lehet ugyanaz, mert – mint láttuk –, DCF zárt a komplementer képzésre, CF pedig nem.

Bizonyítás nélkül:

Tétel. DCF nem zárt sem az egyesítésre, sem a konkatenációra, sem az iterációra.

Környezetfüggetlen nyelvek zártsági tulajdonságai

Összefoglalás

Művelet	Zártság CF	Zártság DCF
$\cup$	igen	nem
konkatenáció	igen	nem
$*$	igen	nem
$\cap$	nem	nem
$\cap$ reg. nyelv	igen	igen
komplementer	nem	igen

Eldöntési kérdések cf nyelvekre

Ebben a részben a környezefüggetlen nyelvek CF osztályára vonatkozó eldöntési kérdésekkel foglalkozunk.

A környezefüggetlen nyelveket Chomsky normálalakban lévő környezefüggetlen nyelvtanokkal reprezentáljuk.

Eldöntési kérdések cf nyelvekre

Eleme-e probléma

Probléma: Eldönthető-e tetszőleges w szó és L cf nyelv esetén, hogy teljesül-e $w \in L$?
Tétel. Az eleme-e probléma cf nyelvekre eldönthető.

Input: Egy w szó és egy Chomsky normálformában lévő $G = (N, \Sigma, P, S)$ cf nyelvtannal megadott L cf nyelv (tehát $L = L(G)$).
Output: "Igen" ha $w \in L$, különben "Nem".

Algoritmus: CYK = Cocke-Younger-Kasami algoritmus, melynek lépésszáma $O(n^3)$, ahol $n = |w|$.

Eldöntési kérdések cf nyelvekre

Előkészítés: Tfh, $|w| = n \geq 1$ és legyen w_{ij} a w i-edik pozícióján kezdődő j hosszúságú rész-szava.

Minden $j = 1, \ldots, n$-re és $i = 1, \ldots, n - j + 1$-re meghatározzuk a $V_{ij} = \{A \in N \mid A \Rightarrow^* w_{ij}\}$ halmazt, a következő elv alapján:

$j = 1$: $A \in V_{i1} \iff A \to w_{i1} \in P$.

$j > 1$: $A \in V_{ij} \iff \exists 1 \leq k < j$ és $A \to BC \in P$, hogy $B \in V_{ik}$ és $C \in V_{i+k,j-k}$. (Dinamikus programozás.)

Ezután $w \in L(G)$, akkor és csak akkor, ha $S \in V_{1n}$.

Eldöntési kérdések cf nyelvekre

(CYK) Algoritmus:

begin
  for $i := 1$ **to** n **do** $V_{i1} = \{A \mid A \to w_{i1} \in P\}$;
    for $j := 2$ **to** n **do**
      for $i := 1$ **to** $n - j + 1$ **do**
        begin $V_{ij} = \emptyset$;
          for $k := 1$ **to** $j - 1$ **do**
            $V_{ij} = V_{ij} \cup \{A \mid A \to BC \in P, B \in V_{ik}, C \in V_{i+k,j-k}\}$;
        end
end

Ha $S \in V_{1n}$, akkor a válasz "Igen", különben "Nem".

Eldöntési kérdések cf nyelvekre

Példa. Tekintsük a következő nyelvtant:

(1) $S \to AA$
(2) $S \to AS$
(3) $S \to b$
(4) $S \to SA$
(5) $A \to AS$
(6) $A \to a$ és legyen $w = abaab$

(Chomsky normálalakban van és nem egyértelmű, de ez utóbbi most nem érdekes.)

Számoljuk ki a V_{ij} halmazokat ($j = 1, \ldots, 5$, $i = 1, \ldots, 6 - j$)!

Eldöntési kérdések cf nyelvekre

Példa folytatása, $w = abaab$.

$V_{11} = \{A\}$	$V_{21} = \{S\}$	$V_{31} = \{A\}$	$V_{41} = \{A\}$	$V_{51} = \{S\}$
$V_{12} = \{A, S\}$	$V_{22} = \{S\}$	$V_{32} = \{S\}$	$V_{42} = \{A, S\}$	
$V_{13} = \{A, S\}$	$V_{23} = \{S\}$	$V_{33} = \{A, S\}$		
$V_{14} = \{A, S\}$	$V_{24} = \{S\}$			
$V_{15} = \{A, S\}$				

$S \in V_{15}$, tehát $w \in L(G)$.

Eldöntési kérdések cf nyelvekre

Ürességi probléma

Probléma: Eldönthető-e tetszőleges L cf nyelv esetén, hogy teljesül-e $L = \emptyset$?
Tétel. Az ürességi probléma környezetfüggetlen nyelvekre eldönthető.

Input: Egy $G = (N, \Sigma, P, S)$, Chomsky normálformában lévő nyelvtannal megadott L cf nyelv.

Output: "Igen", ha $L = \emptyset$, különben "Nem".

Eldöntési kérdések cf nyelvekre

Algoritmus: Legyen k a környezetfüggetlen nyelvekre vonatkozó pumpáló lemmában (Bar-Hillel lemma) szereplő szám.

Könnyen látható, hogy $L \neq \emptyset$ akkor és csak akkor, ha van olyan $x \in L$, melyre $|x| < k$. (Ha $|x| \geq k$, akkor alkalmazzuk a pumpáló lemmát és "rövidítsük" x-et.)

Ez alapján egy eldöntési algoritmus a következő. Minden olyan $x \in \Sigma^*$ szóra, melyre $|x| < k$, kérdezzük le, hogy $x \in L$ teljesül-e (az eleme-e probléma eldönthető). Ha valamely x-re a válasz "Igen" (vagyis $x \in L$), akkor a válasz "Nem", különben "Igen".

Az algoritmus terminál, mivel csak véges számú olyan x szó van, melyre $|x| < k$.

Eldöntési kérdések cf nyelvekre

Végtelen-e probléma

Probléma: Eldönthető-e tetszőleges L cf nyelv esetén, hogy L végtelen-e?
Tétel. A végtelen-e probléma környezetfüggetlen nyelvekre eldönthető.

Input: Egy $G = (N, \Sigma, P, S)$, Chomsky normálformában lévő nyelvtannal megadott L cf nyelv.

Output: "Igen", ha L végtelen, különben "Nem".

Eldöntési kérdések cf nyelvekre

Algoritmus: Legyen k a környezetfüggetlen nyelvekre vonatkozó pumpáló lemmában (Bar-Hillel lemma) szereplő szám.

Könnyen látható, hogy L akkor és csak akkor végtelen, ha van olyan $x \in L$, melyre $k \leq |x| < 2k$. (A bizonyítás hasonló a reguláris nyelvek esetében alkalmazott bizonyításához.)

Ez alapján egy eldöntési algoritmus a következő. Minden olyan $x \in \Sigma^*$ szóra, melyre $k \leq |x| < 2k$, kérdezzük le, hogy $x \in L$ teljesül-e (az eleme-e probléma eldönthető). Ha valamely x-re a válasz "Igen" (vagyis $x \in L$), akkor a válasz "Igen", különben "Nem".

Eldöntési kérdések cf nyelvekre

Ugyanakkor az ekvivalencia probéma környezetfüggetlen nyelvek estén eldönthetetlen.

Tétel. Nem létezik olyan algoritmus, amely tetszőleges L_1, L_2 környezetfüggetlen nyelvekről eldönti, hogy $L_1 = L_2$ teljesül-e.
Bizonyítás. A kérdés visszavezethető a Turing gépek megállási problémájára.

Következmény. Nem létezik olyan algoritmus, amely tetszőleges L_1, L_2 környezetfüggetlen nyelvekről eldönti, hogy $L_1 \subseteq L_2$ teljesül-e.
Bizonyítás. Ha a tartalmazás eldönthető lenne, akkor az ekvivalencia is eldönthető lenne.

Eldöntési kérdések cf nyelvekre

Összefoglalás

Kérdés	Eldönthető-e?
Eleme-e	igen
Üresség	igen
Végtelen-e	igen
Tartalmazás	nem
Ekvivalencia	nem

Eldöntési kérdések cf nyelvekre

Egy megjegyzés:

A környezetfüggetlen nyelvekre vonatkozóan már számos olyan kérdés van, amelynek eldöntésére nem létezik algoritmus.
Például, az előbbiek mellett nem dönthető el az

$L = \Sigma^*$?
$L_1 \cap L_2 = \emptyset$?

kérdések egyike sem.

(Ugyanezen kérdések reguláris nyelvekre eldönthetők.)