

Formális nyelvek III. Elemzők

Fülöp Zoltán

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

1/83

A szintaktikus elemzés, mint feladat

Mint láttuk, a programozási nyelvek szintaxisa megadható környezetfüggetlen nyelvtannal. Ezért a feladat így fogalmazható meg:

Adott: A PL programozási nyelv szintaxisa:

$$G_{PL} = (N, \Sigma, P, S)$$

cf nyelvtan és egy (felhasználó által megírt) PL nyelvű w program (vagyis egy $w \in \Sigma^*$ szó).

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

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

Tehát a kérdés: Teljesül-e, hogy $w \in L(G_{PL})$?

3/83

A szintaktikus elemzés, mint feladat

Kérdés:

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

2/83

A szintaktikus elemzés, mint feladat

Az elemzés definíciója:

Definíció. Legyen adott egy $G = (N, \Sigma, P, S)$ környezetfüggetlen nyelvtan, melynek szabályai meg vannak számozva 1-től p -ig. Egy $\alpha \in (N \cup \Sigma)^*$ szó egy bal oldali (jobb oldali) elemzésének egy olyan $n_1 \dots n_k \in [1, p]^*$ sorozatot nevezünk, amely megadja α egy bal oldali (jobb oldali) levezetését.

Például, a

- (1) $K \rightarrow K + K$,
- (2) $K \rightarrow K * K$,
- (3) $K \rightarrow (K)$,
- (4) $K \rightarrow a$.

nyelvtan estén 11444 és 14144 az $a + a + a$ bal oldali elemzései is és jobb oldali elemzései is.

4/83

Felülről lefelé haladó elemzés

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

Ha sikerül, akkor $w \in L(G)$, különben $w \notin L(G)$.

Csak balrekurziómentes nyelvtanokkal foglalkozunk: nincs olyan $A \in N$ és α , melyekre $A \Rightarrow^+ A\alpha$.

Felülről lefelé haladó elemzés

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

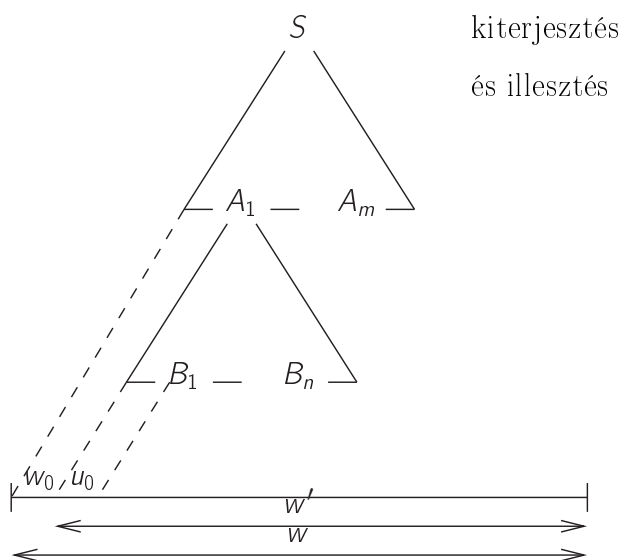
Kiterjesztés: A derivációs fában egy nemterminálist helyettesítünk valamely alternatívájával.

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

5/83

6/83

Felülről lefelé haladó elemzés



7/83

Általános felülről lefelé haladó elemzés

Az általános felülről lefelé haladó elemzés alapötlete:

- ▶ Minden $A \in N$ -re rögzítjük le az A alternatíváinak egy $A \rightarrow \gamma_1 \mid \dots \mid \gamma_k$ sorrendjét.
- ▶ Legyenek S alternatívái, $S \rightarrow \alpha_1 \mid \dots \mid \alpha_k$.
- ▶ Keressük meg az első olyan $\alpha_i = w_0 A_1 w_1 \dots A_m w_m$ alternatívát, melyre w_0 illeszkedik-e w elejéhez, azaz fennáll $w = w_0 w'$. ($w_0 = \lambda$ mindig jó!)
- ▶ Ha nincs ilyen α_i , akkor azt mondhatjuk, hogy $w \notin L(G)$ és az algoritmus leáll.
- ▶ ...

8/83

Általános felülről lefelé haladó elemzés

Az általános felülről lefelé haladó elemzés alapötlete (folyt):

- ▶ ...
- ▶ Ha van ilyen i , azaz $\alpha_i = w_0 A_1 w_1 \dots A_m w_m$ és $w = w_0 w'$, akkor A_1 -et terjesztjük ki. Az $A_1 \rightarrow \beta_1 \mid \dots \mid \beta_l$ alternatívák között megkeressük a legelső olyat, ami illeszkedik w' elejéhez: $\beta_j = u_0 B_1 u_1 \dots B_n u_n$ és $w' = u_0 w''$.
- ▶ Ha egy nemterminális kiterjesztése esetén (pl. most A_1) nem találunk olyan alternatívát ami illeszkedik w megmaradt részéhez, akkor nem mondhatjuk azt, hogy $w \notin L(G)$, mivel valamennyi előző kiterjesztés esetében a legelső olyan alternatívát választottuk ami illeszkedett.
- ▶ Ezért, ha egy szinten nem találunk illeszkedő alternatívát, akkor egy szintet visszamegyünk (ún. *backtrack*-et hajtunk végre) és az ott választott alternatíva után következő első olyan alternatívát kell választanunk ami illeszkedik.

9/83

LL(k) elemzés

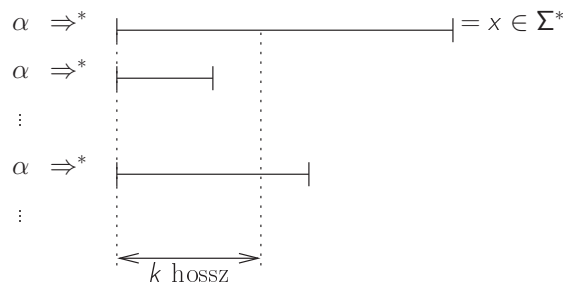
Definíció. Legyen $k \geq 0$ egy egész szám.

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

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

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

$$\text{FIRST}_k(L) = \bigcup_{\alpha \in L} \text{FIRST}_k(\alpha)$$



11/83

LL(k) elemzés

Az LL(k) elemzés (ahol $k \geq 0$ egy szám) speciális felülről lefelé elemzés.

Csak azokat nyelvtanokat lehet LL(k) módon elemezni, amelyek kielégítik az ún. LL(k) feltételt! (Definíciót lásd később.)

Amikor egy A nemterminális kiterjesztését keressük akkor pótlólagos információként "előre nézzük" az input még feldolgozatlan részének k hosszúságú prefixét és ennek ismeretében választjuk ki A -nak azt az alternatíváját amely szerint ki kell őt terjesztenünk.

Ha van ilyen alternatíva, akkor az egyértelműen meghatározható. Ha nincs ilyen alternatíva, akkor az elemzett szó nem eleme $L(G)$ -nek.

Nincs backtrack!

10/83

LL(k) elemzés

Észrevételek:

1. $\text{FIRST}_k(\alpha), \text{FIRST}_k(L) \subseteq \Sigma^{*,k}$ (és így véges halmazok).
2. $u \in \Sigma^{*,k}$ -ra

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

Ilyenkor $\text{FIRST}_k(u) = \{u\}$ és $\text{FIRST}_k(u) = \{w\}$ helyett rendre $\text{FIRST}_k(u) = u$ és $\text{FIRST}_k(u) = w$ -t írunk.

3. $\text{FI}_1(aba) = a$, $\text{FI}_2(aba) = ab$ és $\text{FI}_k(aba) = aba$ minden $k \geq 3$ -ra.

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

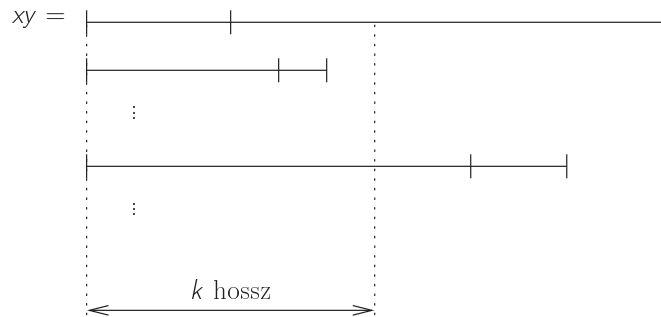
12/83

LL(k) elemzés

FI $_k(\alpha)$ -t kiszámító algoritmus előkészítése:

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

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



13/83

LL(k) elemzés

FI $_k(\alpha)$ kiszámítása

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

$$\text{FI}_k(\alpha) = \text{FI}_k(X_1) \oplus_k \dots \oplus_k \text{FI}_k(X_n).$$

Tehát, elegendő csak FI $_k(X_i)$ -t kiszámítani, ahol $X_i \in (N \cup \Sigma)$.

Ha $X_i \in \Sigma$, akkor FI $_k(X_i) = X_i$ (ld. FI $_k(\alpha)$ definíciója).

Tehát elegendő FI $_k(A)$ -t kiszámolni minden $A \in N$ -re.

15/83

LL(k) elemzés

FI $_k(\alpha)$ -t kiszámító algoritmus előkészítése:

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

Megjegyzés. $L_1 \oplus_k L_2$ mindig véges, mivel $L_1 \oplus_k L_2 \subseteq \Sigma^{*,k}$.

Továbbá a $\oplus_k$ művelet asszociatív és könnyen igazolható, hogy minden α, β -ra

$$\text{FI}_k(\alpha\beta) = \text{FI}_k(\alpha) \oplus_k \text{FI}_k(\beta).$$

14/83

LL(k) elemzés

FI $_k(A)$ halmazok kiszámítása:

Input Egy G környezetfüggetlen nyelvtan.

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

Módszer FI $_k(A)$ -t iteráljuk a $H_0(A), H_1(A), \dots$ sorozattal, amíg minden $A \in N$ -re $H_i(A) = H_{i+1}(A)$. Ekkor FI $_k(A) = H_i(A)$.

16/83

LL(k) elemzés

FI $_k$ (A) halmazok kiszámítása:

Algoritmus

- (1) Legyen minden $a \in \Sigma$ és $i \geq 0$ esetén $H_i(a) = \{a\}$.
- (2) Legyen minden $A \in N$ -re $H_0(A) = \{x \in \Sigma^* \mid A \rightarrow x\alpha \in P \text{ ahol } (|x| = k) \text{ vagy } (|x| < k \text{ és } \alpha = \lambda)\}$ és legyen $i = 0$.
- (3) Minden $A \in N$ -re $H_0(A), \dots, H_i(A)$ már ismertek. Legyen

$$H_{i+1}(A) = H_i(A) \cup \{x \in \Sigma^* \mid x \in H_i(X_1) \oplus_k \dots \oplus_k H_i(X_n) \text{ valamely } A \rightarrow X_1 \dots X_n \in P \text{ esetén}\}.$$
- (4) Ha minden $A \in N$ -re $H_i(A) = H_{i+1}(A)$, akkor álljunk meg, különben legyen $i = i + 1$ és menjünk vissza (3)-ra.

17/83

LL(k) elemzés

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

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

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

19/83

LL(k) elemzés

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

Számítsuk ki a $FI_1(K), FI_1(T)$ és $FI_1(F)$ halmazokat.

A $H_0, H_1, \dots$ közelítések egy táblázat soraiban ábrázolhatók a következő módon:

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

Tehát $FI_1(K) = FI_1(T) = FI_1(F) = \{(\cdot, a)\}$.

18/83

LL(k) elemzés

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

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

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

20/83

LL(k) elemzés

Az LL(k) nyelvtan definíciója:

Definíció.

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

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

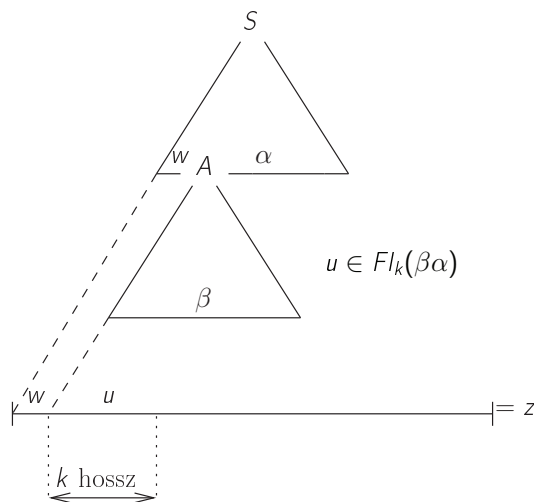
feltételek, mindannyiszor $\beta = \gamma$.

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

21/83

LL(k) elemzés

Az LL(k) elemzés alapötlete rajzolva:



23/83

LL(k) elemzés

Az LL(k) elemzés alapötlete:

Adott: G nyelvtan, ami LL(k) és $z \in \Sigma^*$.

Kérdés: $z \in L(G)$?

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

Ha A -t az $A \rightarrow \beta$ alternatívával terjesztjük ki, akkor $S \Rightarrow_I^* wA\alpha \Rightarrow_I^* w\beta\alpha$. Tehát teljesülni kell, hogy

$$u \in FI_k(\beta\alpha).$$

Az LL(k) feltétel miatt legfeljebb egy ilyen szabály van. Ha nincs ilyen szabály, akkor $z \notin L(G)$.

22/83

LL(k) elemzés

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

Nem LL(1), mert

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

($w = \varepsilon$, $\alpha = \varepsilon$, $x = a + a$, és $y = a$) mégis $K + T \neq T$.

24/83

LL(k) elemzés

Az elemzési algoritmust csak az *erősen* LL(k) nyelvtanokra adjuk meg. Ehhez szükség lesz a következő definícióra:

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

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

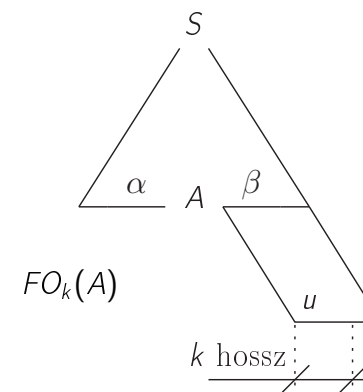
Tehát $\text{FOLLOW}_k(A)$ az olyan terminális szavak FI_k -inak halmaza, amelyek az A -t tartalmazó mondatformák (mint $\alpha A \beta$) A -t követő részéből (vagyis β -ből) vezethetők le.

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

25/83

LL(k) elemzés

Az ábrán $u \in \text{FO}_k(A)$:



26/83

LL(k) elemzés

A $\text{FO}_k(A)$ halmazok kiszámítása.

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

Output Minden $A \in N$ -re $\text{FO}_k(A)$.

Módszer $\text{FO}_k(A)$ -t iteráljuk a $H_0(A), H_1(A), \dots$ sorozattal, amíg minden $A \in N$ -re $H_i(A) = H_{i+1}(A)$. Ekkor $\text{FO}_k(A) = H_i(A)$.

27/83

LL(k) elemzés

Algoritmus

- (1) Legyen $i = 0$, $H_0(S) = \{\lambda\}$ és minden $A \neq S$ -re legyen $H_0(A) = \emptyset$.
- (2) Minden $A \in N$ -re $H_0(A), \dots, H_i(A)$ már ismertek. Legyen

$$H_{i+1}(A) = H_i(A) \cup \{x \in \Sigma^* \mid x \in \text{FI}_k(\beta H_i(B))\}$$
 valamely $B \rightarrow \alpha A \beta$ szabály esetén }.
- (3) Ha minden $A \in N$ -re $H_i(A) = H_{i+1}(A)$, akkor álljunk meg, különben legyen $i = i + 1$ és menjünk (2)-re.

Minden A -ra $\text{FO}_k(A) = H_i(A)$.

28/83

LL(k) elemzés

Példa. Vegyük a G'_{ar} nyelvtant, melynek szabályai:

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

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

LL(k) elemzés

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

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

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

29/83

30/83

LL(k) elemzés

Az erősen LL(k) nyelvtan definíciója:

Definíció. Legyen $k \geq 1$. Azt mondjuk, hogy G erősen LL(k), ha tetszőleges $A \in N$ és

$$A \rightarrow \beta, A \rightarrow \gamma$$

különböző szabályok esetén teljesül, hogy

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

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

Ha G erősen LL(k), akkor LL(k) (bizonyítás a jegyzetben).

31/83

LL(k) elemzés

A következő két kérdést vizsgáljuk meg:

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

32/83

LL(k) elemzés

Az erősen LL(k) feltétel eldöntése.

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

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

Algoritmus

- (1) Válasszunk egy olyan A nemterminálist melynek legalább két alternatívája van.
- (2) Válasszunk két különböző $A \rightarrow \beta$ és $A \rightarrow \gamma$ szabályt és számoljuk ki a $(FI_k(\beta FO_k(A))) \cap (FI_k(\gamma FO_k(A)))$ halmazt. Ha ez nemüres, akkor G nem erősen LL(k), tehát álljunk meg és output: *Nem*.
- (3) Ha választható A -nak újabb két alternatíva-párja, akkor ismételjük (2)-t.
- (4) Ha választható újabb nemterminális, akkor ismételjük (1)-et.
- (5) Adjunk outputra *Igen* jelzést.

33/83

LL(k) elemzés

Példa. Döntsük el, hogy G'_{ar} LL(1)-e!

Két olyan szabálpár van, melyeknek bal oldala ugyanaz:

2) $F' \rightarrow *FF'$ és $F' \rightarrow \lambda$:

$$FI_1(*FF'FO_1(F')) = \{*\}$$
 és

$$FI_1(\lambda FO_1(F')) = FO_1(F') = \{+,), \lambda\}.$$

Tehát

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

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

35/83

LL(k) elemzés

Példa. Döntsük el, hogy G'_{ar} LL(1)-e!

Két olyan szabálpár van, melyeknek bal oldala ugyanaz:

1) $T' \rightarrow +TT'$ és $T' \rightarrow \lambda$:

$$FI_1(+TT'FO_1(T')) = \{+\}$$
 és

$$FI_1(\lambda FO_1(T')) = FO_1(T') = \{\}, \lambda\}.$$

Tehát

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

34/83

LL(k) elemzés

Az erősen LL(k) elemzés alapötlete: Kérdés: $z \in L(G)$?

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

Ha A -t az $A \rightarrow \beta$ alternatívával terjesztjük ki, akkor $S \Rightarrow_j^* wA\alpha \Rightarrow_j^* w\beta\alpha$ miatt $u \in FI_k(\beta\alpha)$ -nak teljesülnie kell.

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

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

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

36/83

LL(k) elemzés

Példa. A G'_{ar} nyelvtant esetében (a már kiszámolt $FI_1()$ és $FO_1()$ halmazok alapján:

- 1: $K \rightarrow TT'$ $FI_1(TT'FO_1(K)) = \{(\cdot, a)\}$
- 2: $T' \rightarrow +TT'$ $FI_1(+TT'FO_1(T')) = \{+\}$
- 3: $T' \rightarrow \lambda$ $FI_1(\lambda FO_1(T')) = \{(\cdot), \lambda\}$
- 4: $T \rightarrow FF'$ $FI_1(FF'FO_1(T)) = \{(\cdot, a)\}$
- 5: $F' \rightarrow *FF'$ $FI_1(*FF'FO_1(F')) = \{*\}$
- 6: $F' \rightarrow \lambda$ $FI_1(\lambda FO_1(F')) = \{+, \cdot, \lambda\}$
- 7: $F \rightarrow (K)$ $FI_1((K)FO_1(F)) = \{(\cdot)\}$
- 8: $F \rightarrow a$ $FI_1(aFO_1(F)) = \{a\}$

37/83

LL(k) elemzés

Az $\mathcal{M}$ elemzőtábla definíciója:

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

$$\mathcal{M}(A, u) = (\beta, j),$$

- (2) Minden $a \in \Sigma$ és $u \in \Sigma^{*,k}$ esetén, ha $u = av$, akkor legyen $\mathcal{M}(a, u) = \text{pop}$.
- (3) Legyen $\mathcal{M}(\$, \lambda) = \text{elfogadás}$,
- (4) Minden más $X \in (N \cup \Sigma \cup \{\$, \lambda\})$ és $u \in \Sigma^{*,k}$ esetén legyen $\mathcal{M}(X, u) = \text{hiba}$.

39/83

LL(k) elemzés

Az erősen LL(k) elemzés alapötlete:

Erősen LL(k) nyelvtanok esetében egy $\mathcal{M}$ elemző táblát konstruálunk a következőképpen.

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

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

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

38/83

LL(k) elemzés

Példa. G'_{ar} elemzőtáblája:

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

40/83

LL(k) elemzés

Erősen LL(k) elemzési algoritmus.

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

Output $lgen$ és z egy bal oldali levezetése, ha $z \in L(G)$, *Nem* különben.

41/83

LL(k) elemzés

Az elemzési algoritmus

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

Amennyiben az output $lgen$, π adja z egy bal oldali levezetését.

Itt $\vdash$ a konfigurációk közötti átmeneti reláció, amelyet a következőképpen definiálunk:...

43/83

LL(k) elemzés

Módszer

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

42/83

LL(k) elemzés

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

- (1) Ha $X = A \in N$ és $\mathcal{M}(A, u) = (\beta, j)$, akkor $(x, X\alpha, \pi) \vdash (x, \beta\alpha, \pi j)$. (A verem tetején egy nemterminális állt amelyet kiterjesztettünk.)
- (2) Ha $X = a \in \Sigma$ és $\mathcal{M}(a, u) = \text{pop}$, akkor $(x, X\alpha, \pi) \vdash (x', \alpha, \pi)$, ahol $x = ax'$. (Az illesztendő terminális illeszkedik az elemzendő szóhoz.)
- (3) Ha $\mathcal{M}(X, u) = \text{elfogadás}$ (ami csak úgy lehet, ha $X = \$, \alpha = \lambda$ és $u = \lambda$), akkor nincs átmenet.
- (4) Ha $\mathcal{M}(X, u) = \text{hiba}$, akkor nincs átmenet.

44/83

LL(k) elemzés

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

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

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

45/83

LL(k) elemzés

Faktorizálás:

Egy módszer az LL(1) konfliktus megszüntetésére.

Tekintsük az if-then-else nyelvtant:

$$\begin{aligned} S &\rightarrow iEtSeS \mid iEtS \mid x \\ E &\rightarrow y, \end{aligned}$$

ami nyilvánvalóan nem LL(1) az $S \rightarrow iEtSeS \mid iEtS$ szabályok miatt:

$$FI_1(iEtSeS \text{ FO}_1(S)) \cap FI_1(iEtS \text{ FO}_1(S)) = \{i\} \neq \emptyset.$$

47/83

LL(k) elemzés

Mivel az LL(k) nyelvtanok jól elemezhetők, kíváncsian lenne ha minden G cf nyelvtant át lehetne alakítani egy vele ekvivalens LL(k) nyelvtanná valamely k -ra.

Ez sajnos nem lehetséges, mert van olyan L cf nyelv, amelyet generáló bármely nyelvtan nem LL(k) semmilyen k -ra.

A *faktorizálás* nevű módszer, azonban néha segít.

46/83

LL(k) elemzés

Faktorizálás:

Kiemeljük a közös prefixet és bevezetünk egy új változót:

$$\begin{aligned} S &\rightarrow iEtSeS \mid iEtS \mid x \\ E &\rightarrow y \end{aligned}$$

helyett:

$$\begin{aligned} S &\rightarrow iEtSA \mid x \\ A &\rightarrow eS \mid \lambda \\ E &\rightarrow y. \end{aligned}$$

Ez utóbbi nyelvtan már LL(1).

(Persze a módszer nem működik minden nyelvtan esetén.)

48/83

LL(k) elemzés

Faktorizálás:

FACTOR(G)

Input: $G = (N, \Sigma, P, S)$ cf nyelvtan.

1) Minden $A \in N$ -re:

2) Keressük meg A -szabályok jobb oldalainak leghosszabb $\alpha \neq \lambda$ közös prefixét.

- helyettesítsük az $A \rightarrow \alpha\beta_1 \mid \dots \mid \alpha\beta_m \mid \gamma_1 \mid \dots \mid \gamma_m$ szabályokat
- az $A \rightarrow \alpha A' \mid \gamma_1 \mid \dots \mid \gamma_m$ és $A' \rightarrow \beta_1 \mid \dots \mid \beta_m$ szabályokkal

3) Ismételjük 2)-t, amíg már nincsenek A -szabályok, melyek jobb oldalainak közös a prefixe.

49/83

Alulról felfelé haladó elemzés

Alulról felfelé (bottom-up) haladó elemzési algoritmusok:

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

A w szóból kiindulva próbálunk meg felépíteni egy olyan derivációs fát aminek a gyökere S , a határa pedig w .

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

Csak a λ -mentes és az ún. ciklusmentes nyelvtanokra alkalmazható (nincs olyan $A \in N$, melyre $A \Rightarrow^+ A$).

51/83

LL(k) elemzés

Faktorizálás:

Példa. A

- ▶ $K \rightarrow T + K \mid T$
- ▶ $T \rightarrow F * T \mid F$
- ▶ $F \rightarrow (K) \mid a$

nyelvtanból kapjuk a következő LL(1) nyelvtanokat:

$K \rightarrow TT'$	$K \rightarrow TT'$
$T' \rightarrow +K \mid \lambda$	$T' \rightarrow +TT' \mid \lambda$
$T \rightarrow FF'$	$T \rightarrow FF'$
$F' \rightarrow *T \mid \lambda$	$F' \rightarrow *FF' \mid \lambda$
$F \rightarrow (K) \mid a$	$F \rightarrow (K) \mid a$

50/83

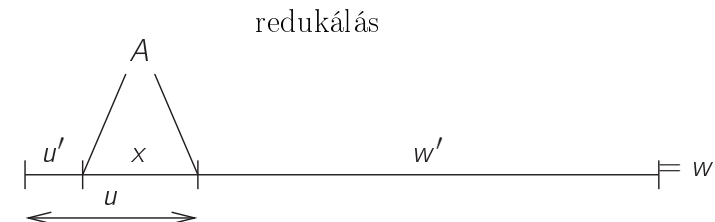
Alulról felfelé haladó elemzés

Általános alulról felfelé elemzés:

Két alapművelet van:

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

Redukálás: egy szabály jobb oldalát a bal oldalával helyettesítjük.



Redukálás az $A \rightarrow x$ szabály szerint.

52/83

Alulról felfelé haladó elemzés

Két probléma adódik:

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

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

Keresünk egy olyan, szűkebb nyelvtan osztályt ($LR(k)$ nyelvtanok), amelyek elemzése során egyik konfliktus sem merül fel.

53/83

$LR(k)$ elemzés

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

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

55/83

$LR(k)$ elemzés

$LR(k)$ nyelvtanok és elemzésük:

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

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

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

54/83

$LR(k)$ elemzés

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

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

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

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

56/83

LR(k) elemzés

Ha egy G nyelvtan LR(k), akkor az elemzés során nem lesz redukálás-redukálás konfliktus:

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

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

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

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

57/83

LR(k) elemzés

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

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

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

59/83

LR(k) elemzés

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

- (1) Ha adott egy G nyelvtan és egy k szám, akkor eldönthető-e, hogy G nyelvtan LR(k)-e?
- (2) Hogyan működik az LR(k) elemzés?

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

58/83

LR(k) elemzés

Definíció. a) Az $[A \rightarrow \beta_1.\beta_2, u]$ alakú párokat LR(k) *elemeknek* nevezzük, ahol $A \rightarrow \beta_1\beta_2 \in P$ és $u \in \Sigma^{*,k}$.) (A $\beta_1 = \lambda$ és/vagy a $\beta_2 = \lambda$ is lehetséges.)
b) Az $[A \rightarrow \beta_1.\beta_2, u]$ LR(k) elem *érvényes az $\alpha\beta_1$ járható prefixre*, ha létezik $S \Rightarrow_r^* \alpha A w \Rightarrow \alpha\beta_1\beta_2 w$ deriváció úgy, hogy $u = FI_k(w)$.

Az érvényesség szemléletes jelentése a következő:

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

60/83

LR(k) elemzés

Jelölések. Adott γ járható prefix esetén a γ -ra érvényes LR(k) elemek halmazát $V_k(\gamma)$ -val jelöljük.

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

Bevezetjük a

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

jelölést.

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

61/83

LR(k) elemzés

Algoritmus

(a) $V_k(\lambda)$ kiszámítása

- (1) (Inicializálás.) Minden $S \rightarrow \alpha \in P$ esetén legyen $[S \rightarrow \cdot \alpha, \lambda] \in V_k(\lambda)$.
- (2) (Lezárás.) Mindaddig, amíg $V_k(\lambda)$ bővíthető, a $V_k(\lambda)$ minden $[A \rightarrow \cdot B\beta, u]$ alakú elemére és $B \rightarrow \delta$ P -beli szabályra, legyen $[B \rightarrow \cdot \delta, v] \in V_k(\lambda)$ minden $v \in Fl_k(\beta u)$ -ra.

63/83

LR(k) elemzés

$V_k(\gamma)$ meghatározása:

Input G nyelvtan és $\gamma = X_1 \dots X_n$ szó, ahol $X_1, \dots, X_n \in (N \cup \Sigma)$.

Output $V_k(\gamma)$ halmaz.

Módszer Kiszámoljuk a $V_k(\lambda)$, $V_k(X_1)$, $\dots$, $V_k(X_1 \dots X_n)$ halmazokat.

62/83

LR(k) elemzés

Algoritmus

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

- (1) (Léptetés.) Minden $[A \rightarrow \alpha X_i \beta, u]$ alakú $V_k(X_1 \dots X_{i-1})$ -beli elem esetén legyen $[A \rightarrow \alpha X_i \cdot \beta, u] \in V_k(X_1 \dots X_i)$.
- (2) (Lezárás.) Mindaddig, amíg $V_k(X_1 \dots X_i)$ bővíthető, a $V_k(X_1 \dots X_i)$ minden $[A \rightarrow \alpha B \beta, u]$ alakú elemére és $B \rightarrow \delta$ P -beli szabályra, legyen $[B \rightarrow \cdot \delta, v] \in V_k(X_1 \dots X_i)$ minden $v \in Fl_k(\beta u)$ -ra.

64/83

LR(k) elemzés

Példa. (LR(1) táblák) G_{list}

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

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

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

LR(k) elemzés

Példa. (LR(1) táblák) G_{list} :

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

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

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

LR(k) elemzés

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

Input G nyelvtan. Output $\mathcal{T} = \{V_k(\gamma) \mid \gamma \text{ járható prefix}\}$.

- (1) Legyen $\mathcal{T} = \{V_k(\lambda)\}$ ($V_k(\gamma)$ Algoritmus (a) pont).
- (2) Mindaddig amíg $\mathcal{T}$ -ben van megjelöletlen táblázat tegyük a következőt:
 - ▶ Vegyünk egy megjelöletlen $T \in \mathcal{T}$ elemet. Legyen ez $T = V_k(\gamma)$.
 - ▶ Jelöljük meg T -t.
 - ▶ Minden $X \in (N \cup \Sigma)$ -ra számoljuk ki $T' = V_k(\gamma X)$ -et ($V_k(\gamma)$ Algoritmus (b) pont). Ha $T' \notin \mathcal{T}$ akkor T' -t vegyük fel $\mathcal{T}$ -be, különben lépünk tovább.

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

65/83

66/83

LR(k) elemzés

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

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

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

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

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

67/83

68/83

LR(k) elemzés

Az LR(k) tétel átfogalmazása:

Definíció. Tetszőleges α szóra

$$\text{EFF}_k(\alpha) = \{\text{FI}_k(x) \mid \alpha \Rightarrow_r^* \beta \Rightarrow_r x \text{ úgy, hogy } \beta \neq Ax \text{ semmilyen } A \in N\text{-re}\}.$$

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

69/83

LR(k) elemzés

Példa. A G_{list} nyelvtan LR(1):

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

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

71/83

LR(k) elemzés

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

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

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

70/83

LR(k) elemzés

Példa. A G_{list} nyelvtan LR(1):

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

Az $[S \rightarrow L., \lambda]$ -t és az $[L \rightarrow L. * E, \lambda / *]$ elemek konfliktusban lehetnek. De $\text{EFF}_1(*E\lambda) = \text{EFF}_1(*E*) = \{*\}$ és $\lambda \notin \{*\}$, így az LR(1) feltétel megint csak nem sérül.

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

72/83

LR(k) elemzés

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

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

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

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

73/83

LR(k) elemzés

Példa. G_{list} : 1: $S \rightarrow L$
2: $L \rightarrow L * E$ 3: $L \rightarrow E$
4: $E \rightarrow a$ 5: $E \rightarrow b$ *tevékenység táblája:*

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

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

75/83

LR(k) elemzés

$f(T, u)$ pontos definíciója.

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

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

74/83

LR(k) elemzés

LR(k) elemzési algoritmus:

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

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

Módszer

- Az elemzés $(x, \alpha T, \pi)$ alakú konfigurációk sorozata, ahol
 - $x \in \Sigma^*$ egy szó, az elemzendő szó hátralévő (vagyis a járható prefixen túli) része,
 - $\alpha T \in (\mathcal{T} \cup N \cup \Sigma)^*$, ahol α tartalmazza a járható prefixet úgy, hogy ennek minden prefixe után α -ban benne van a rá érvényes LR(k) tábla is. (Így T érvényes α' -re, ahol α' -t úgy kapjuk α -ból, hogy töröljük belőle a $\mathcal{T}$ -beli szimbólumokat.) Egy veremnek tekintjük, melynek teteje a jobb végén van.
 - π egy, a szabályok sorszámaiból alkotott sorozat. Egy sornak tekintjük, melynek teteje a bal végén van.

76/83

LR(k) elemzés

Módszer (folyt):

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

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

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

77/83

LR(k) elemzés

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

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

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

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

(4) Ha $f(T, u) = \text{hiba}$, akkor nincs átmenet.

79/83

LR(k) elemzés

Az elemzési algoritmus:

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

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

78/83

LR(k) elemzés

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

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

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

80/83

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

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

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

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