

Záróvizsga tételsor

11. Formális nyelvek

Dobreff András

Formális nyelvek

Formális nyelvtanok és a Chomsky-féle nyelvosztályok. Automaták: véges automata, veremautomata. Reguláris nyelvek tulajdonságai és alkalmazásai. Környezetfüggetlen nyelvek tulajdonságai és elemzésük.

1 Formális nyelvtanok és a Chomsky-féle nyelvosztályok

1.1 Alapfogalmak

Ábécé, szó

Szimbólumok véges nemüres halmazát ábécének nevezzük.

Egy V ábécé elemeiből képzett véges sorozatokat V feletti szavaknak vagy sztringeknek nevezzük. A 0 hosszúságú sorozatot üres szónak nevezzük és ε -nal jelöljük.

A V ábécé feletti szavak halmazát (beleértve az üres szót is) V^* -gal, a nemüres szavak halmazát V^+ -szal jelöljük.

Konkatenáció

Az $x = uv$ szót az $u, v \in V$ szavak konkatenációjának nevezzük.

A konkatenáció asszociatív, de (általában) nem kommutatív.

V^* zárt a konkatenációra. Azaz:

$$u, v \in V^* \Rightarrow uv \in V^*$$

Továbbá ε egységelemnek tekinthető V^* -gal és a konkatenációval. Azaz:

$$u \in V^* \Rightarrow u\varepsilon \in V^*, \text{ és } \varepsilon u \in V^*$$

Egyéb definíciók

- $u, v \in V$. Az u szót a v részsavának nevezzük, ha $v = xuy$, ($x, y \in V$) teljesül. Ha még $xy \neq \varepsilon$, akkor u valódi részsav.
- $v = xuy$ ($x, y, u, v \in V$). Ekkor:
 - Ha $x = \varepsilon$, akkor u -t a v szó prefixének nevezzük
 - Ha $y = \varepsilon$, akkor u -t a v szó szuffixének nevezzük
- Egy $u \in V$ szó tükörképe alatt a szimbólumai fordított sorrendben való felírását értjük. (Jele: u^{-1})

1.2 Formális nyelvtanok

Nyelv

Ha $L \subset V^*$, akkor L -et V feletti nyelvnek tekintjük.

Az üres nyelv (egy szót sem tartalmaz) jele: \emptyset

L nyelv véges nyelv, ha véges számú szót tartalmaz, különben végtelen nyelv.

Nyelvekre vonatkozó műveletek:

- $L_1 \cup L_2 = \{u \mid u \in L_1 \text{ vagy } u \in L_2\}$: az L_1 és L_2 nyelv uniója

- $L_1 \cap L_2 = \{u \mid u \in L_1 \text{ és } u \in L_2\}$: az L_1 és L_2 nyelv metszete
- $L_1 - L_2 = \{u \mid u \in L_1 \text{ és } u \notin L_2\}$: az L_1 és L_2 nyelv különbsége
- $\bar{L} = V^* - L$: az $L \subseteq V^*$ komplementere
- $L_1 L_2 = \{u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2\}$: az L_1 és L_2 nyelv konkatenációja
Minden nyelvre fennáll: $\emptyset L = L \emptyset = \emptyset$ illetve $\{\varepsilon\} L = L \{\varepsilon\} = L$
- L^i : az L nyelv i -edik ($i \geq 1$) iterációja (a konkatenációra nézve), és $L^0 = \{\varepsilon\}$
- $L^* = \bigcup_{i \geq 1} L^i$: az L nyelv iteratív lezártja

Az unió, konkatenáció és iteráció lezárása műveleteket reguláris műveleteknek nevezzük.

Grammatika

Nyelvek sokféle módon előállíthatók. A produkciós rendszerekkel való előállítás egyik módja a nyelvek generálása grammatikával.

A G generatív grammatikán egy (N, T, P, S) négyest értünk, ahol:

- N és T diszjunkt ábécék, a nemterminális (N) és terminális (T) szimbólumok ábécéi.
- $S \in N$ a kezdőszimbólum.
- P az (x, y) rendezett párok halmaza, ahol $x, y \in (N \cup T)^*$ és x legalább egy nemterminális szimbólumot tartalmaz.

A P halmaz elemeit átírási szabályoknak nevezzük.

Az (x, y) jelölés helyett az $x \rightarrow y$ jelölést alkalmazzuk. (Ha $\rightarrow \notin (N \cup T)$)

Levezetés

$G = (N, T, P, S)$, és $u, v \in (N \cup T)^*$. A v szó közvetlenül (egy lépésben) levezethető u szóból G -ben, ha

$$u = u_1 x u_2, v = u_1 y u_2 \text{ és } x \rightarrow y \in P \quad (u_1, u_2 \in (N \cup T)^*)$$

Ezt $u \Rightarrow_G v$ jelöljük.

Azt mondjuk, hogy a v szó k (≥ 1) lépésben levezethető az u szóból G -ben, ha

$$\exists u_1, \dots, u_{k+1} \in (N \cup T)^* : u = u_1, v = u_{k+1} \text{ és } u_i \Rightarrow_G u_{i+1} \quad (1 \leq i \leq k)$$

A v szó levezethető az u szóból G -ben, ha $u = v$ vagy $\exists k \geq 1$ szám, hogy v k lépésben levezethető u -ból.

Másképp: A v szó levezethető az u szóból G -ben (jele: $u \Rightarrow_G^* v$), ha $u = v$ vagy $\exists z \in (N \cup T)^*$ szó, hogy $u \Rightarrow_G^* z$ és $z \Rightarrow_G v$

Generált nyelv

$L(G)$ a $G = (N, T, P, S)$ grammatika által generált nyelv, ha:

$$L(G) = \{w \mid S \Rightarrow_G^* w, w \in T^*\}$$

1.3 Chomsky-féle hierarchia

$G = (N, T, P, S)$ generatív grammatika i -típusú ($i = 0, 1, 2, 3$), ha P szabályhalmazára a következők teljesülnek:

Mondatszerkezetű grammatika (i=0)

Nincs korlátozás

Környezetfüggő grammatika (i=1)

P minden szabálya $u_1 A u_2 \rightarrow u_1 v u_2$ alakú, ahol, $A \in N$ és $v \neq \varepsilon$, ($u_1, u_2, v \in (N \cup T)^*$). Kivéve az $S \rightarrow \varepsilon$ szabályt (ha létezik). Ekkor S nem fordul elő egyetlen szabály jobb oldalán sem.

Környezetfüggetlen grammatika (i=2)

P minden szabálya $A \rightarrow v$ alakú, ahol, $A \in N, v \in (N \cup T)^*$.

Reguláris grammatika (i=3)

P minden szabálya $A \rightarrow vB$ vagy $A \rightarrow v$ alakú, ahol, $A, B \in N, v \in T^*$.

L nyelv i -típusú, ha i -típusú grammatikával generálható. Az i -típusú nyelvek osztályát \mathcal{L}_i jelöljük. Az i -típusú nyelvosztályok a következő tulajdonságokkal rendelkeznek:

- $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$
- Az \mathcal{L}_i ($i = 0, 1, 2, 3$) nyelvosztályok mindegyike zárt a reguláris műveletekre.

2 Automaták

Formális nyelvek megadása nemcsak generatív, hanem felismerő eszközökkel is lehetséges, azaz olyan számítási eszközök segítségével, amelyek szavak feldolgozására és azonosítására alkalmasak. Ilyen eszköz például az automata, amely egy szó, mint input hatására kétféleképpen viselkedhet: vagy elfogadja, vagy elutasítja.

2.1 Véges Automata

Definíció

A véges automata egy rendezett ötös,

$$A = (Q, T, \delta, q_0, F)$$

ahol:

- Q - állapotok véges nemüres halmaza
- T - input szimbólumok ábécéje
- $\delta : Q \times T \rightarrow Q$ - állapot-átmeneti függvény
- $q_0 \in Q$ - kezdőállapot
- $F \subseteq Q$ - elfogadó állapotok halmaza

Működés:

A véges automata diszkrét időintervallumokban végrehajtott lépések sorozata által működik. Minden egyes lépés során az automata elolvassa a következő input szimbólumot és átmegy egy olyan állapotba, amelyet az állapotátmeneti függvény meghatároz (az aktuális állapot és input szimbólum alapján).

Kezdetben az A véges automata a q_0 kezdőállapotban van és az olvasófej az input szalagon levő $u \in T^*$ szó első betűjét dolgozza fel. Ezután a véges automata lépések sorozatát végrehajtva elolvassa az input u szót; betűről betűre haladva olvas és új állapotba kerül.

Miután az u input szó utolsó betűjét is elolvasta a véges automata, vagy $q \in F$, azaz elfogadó állapotba kerül, és akkor az u szót az automata elfogadja, vagy az új állapot nem lesz eleme F -nek, és ekkor az automata a szót nem fogadja el.

VDA - Véges determinisztikus automata

A δ függvény egyértékű, ezért minden egyes (q, a) párra, ahol $(q, a) \in Q \times T$ egyetlen olyan s állapot létezik, amelyre $\delta(q, a) = s$ teljesül. Ezért ezt a véges automatát determinisztikusnak nevezzük.

VNDA - Véges nemdeterminisztikus automata

Ha többértékű állapot-átmeneti függvényt is megengedünk, azaz $\delta : Q \times T \rightarrow 2^Q$, akkor nemdeterminisztikus véges automatáról beszélünk. (Ebben az esetben aktuális állapotnak egy állapot-halmaz valamely elemét, mintsem egyetlen állapotot tekinthetünk.)

Ez azt jelenti, hogy a kezdeti állapot helyettesíthető egy $Q_0 \subseteq Q$ kezdőállapot halmazzal. (És az is előfordulhat, hogy egy a input szimbólum esetén $\delta(q, a)$ üres az aktuális állapotok mindegyikére.)

Tulajdonságok

Az állapot-átmeneteket

$$qa \rightarrow p$$

alakú szabályok formájában is írhatjuk $p \in \delta(q, a)$ esetén. Jelöljük M_δ -val az $A = (Q, T, \delta, Q_0, F)$ nemdeterminisztikus véges automata δ állapot-átmenet függvénye által az előbbi módon származó szabályok halmazát.

Ha minden egyes (q, a) párra egyetlen $qa \rightarrow p$ szabály van M_δ -ban, akkor a véges automata determinisztikus, egyébként nemdeterminisztikus.

- **Közvetlen redukció:**
Legyen $A = (Q, T, \delta, q_0, F)$ egy véges automata és legyenek $u, v \in QT^*$ szavak. Azt mondjuk, hogy az A automata az u szót a v szóra redukálja egy lépésben/közvetlenül. Ha van olyan $qa \rightarrow p$ szabály M_δ -ban, és van olyan $w \in T^*$ szó, amelyre $u = qaw$ és $v = pw$ teljesül.
- **Redukció:**
Az $A = (Q, T, \delta, q_0, F)$ véges automata az $u \in QT^*$ szót a $v \in QT^*$ szóra redukálja ($u \Rightarrow_A^* v$), ha $u = v$, vagy $\exists z \in QT^*$, amelyre $u \Rightarrow_A^* z$ és $z \Rightarrow_A v$ teljesül.
- **Az automata által elfogadott nyelv:**
Az $A = (Q, T, \delta, q_0, F)$ véges automata által elfogadott/felismert nyelv alatt az

$$L(A) = \{u \in T^* \mid q_0 u \Rightarrow_A^* p, \quad q_0 \in Q_0 \text{ és } p \in F\}$$

szavak halmazát értjük. (Az üres szó, akkor és csak akkor van benne az automata által elfogadott $L(A)$ nyelvben, ha $Q_0 \cap F \neq \emptyset$).

- **Tétel:**
Minden A nemdeterminisztikus véges automatához meg tudunk adni egy 3-típusú G grammatikát úgy, hogy $L(G) = L(A)$ teljesül.
- **Tétel:**
Minden 3-típusú G grammatikához meg tudunk adni egy A véges automatát úgy, hogy $L(A) = L(G)$ teljesül.

Ezek után fenáll a kérdés: Létezik-e olyan reguláris nyelv, amely VNDA-val felismerhető, de nem ismerhető fel VDA-val?

Válasz: Nincs.

- **Tétel:**
Minden $A = (Q, T, \delta, Q_0, F)$ VNDA-hoz meg tudunk konstruálni egy $A' = (Q', T, \delta', q'_0, F')$ VDA-t úgy, hogy $L(A) = L(A')$ teljesül.

2.2 Veremautomata

Definíció

A veremautomata egy rendezett hetes

$$A = (Z, Q, T, \delta, z_0, q_0, F)$$

ahol

- Z - a veremszimbólumok véges halmaza,
- Q - az állapotok véges halmaza,
- T - az inputszimbólumok véges halmaza,
- $\delta : Z \times Q \times (T \cup \{\varepsilon\}) \rightarrow 2^{Z^* \times Q}$ - átmeneti függvény
- $z_0 \in Z$ - a kezdeti veremszimbólum,
- $q_0 \in Q$ - a kezdeti állapot,
- $F \subseteq Q$ - az elfogadó állapotok halmaza

A veremautomata konfigurációja alatt egy uq alakú szót értünk, ahol $u \in Z^*$ a verem aktuális tartalma és $q \in Q$ az aktuális állapot.

A kezdeti konfiguráció $z_0 q_0$.

Működés:

Tegyük fel, hogy az A veremautomata olvasófeje az a inputszimbólumon áll, a veremautomata q

állapotban van, valamint a verem tetején levő szimbólum z . Legyen $\delta(z, q, a) = (u_1, r_1), \dots, (u_n, r_n)$, ahol $u_i \in Z^*$ és $r_i \in Q$, $1 \leq i \leq n$. Ekkor A következő állapota valamely r_i lesz és egyidejűleg z -t helyettesíti az u_i szóval, továbbá az olvasófej egy cellával jobbra lép az input szalagon.

Ha $\delta(z, q, \varepsilon)$ nem üres, akkor ún. ε -átmenet hajtható végre.

Ha az input szalag a $w \in T^*$ szót tartalmazza és a z_0q_0 kezdeti konfigurációból kiindulva a lépések sorozatát végrehajtva az A veremautomata egy up konfigurációba ér, ahol p elfogadó állapot, akkor azt mondjuk, hogy A elfogadta a w szó.

Tulajdonságok

- Közvetlen redukció:

$$\alpha, \beta \in Z^*QT^*$$

Az A veremautomata az α szót a β szóra redukálja egy lépésben ($\alpha \Rightarrow_A \beta$), ha:

$$\exists z \in Z, p, q \in Q, a \in T \cup \{\varepsilon\}, r, u \in Z^* \text{ és } w \in T^*$$

hogy:

$$(u, p) \in \delta(z, q, a) \text{ és } \alpha = rzqaw \text{ és } \beta = rupw$$

- Redukció:

Az A veremautomata az α szót a β szóra redukálja ($\alpha \Rightarrow_A^* \beta$), ha vagy $\alpha = \beta$, vagy $\exists \gamma_1, \dots, \gamma_n \in Z^*QT^*$ szavakból álló véges sorozat, hogy $\alpha = \gamma_1$, $\beta = \gamma_n$, és $\gamma_i \Rightarrow_A \gamma_{i+1}$ ($i = 1, \dots, n-1$)

- A veremautomata által elfogadott nyelv:

Az A veremautomata által (elfogadó állapottal) elfogadott nyelv:

$$L(A) = \{w \in T^* \mid z_0q_0w \Rightarrow_A^* up, \text{ ahol } u \in Z^*, p \in F\}$$

- Determinizmus:

A δ leképezést szabályok formájában is megadhatjuk. Az így nyert szabályhalmazt M_δ -val jelöljük. Tehát

1. $zqa \rightarrow up \in M_\delta$ ha $(u, p) \in \delta(z, q, a)$
2. $zq \rightarrow up \in M_\delta$ ha $(u, p) \in \delta(z, q, \varepsilon)$

Az $A = (Z, Q, T, \delta, z_0, q_0, F)$ veremautomatát determinisztikusnak mondjuk, ha minden $(z, q) \in Z \times Q$ pár esetén:

1. $\forall a \in T : |\delta(z, q, a)| = 1$ és $\delta(z, q, \varepsilon) = \emptyset$
vagy
2. $|(z, q, \varepsilon)| = 1$ és $\forall a \in T : \delta(z, q, a) = \emptyset$

- Üres veremmel elfogadott nyelv:

Az $N(A)$ nyelvet az A veremautomata üres veremmel fogadja el, ha

$$N(A) = \{w \in T^* \mid z_0q_0w \Rightarrow_A^* p, \text{ ahol } p \in Q\}$$

- Tétel:

Bármely G környezetfüggetlen grammatikához meg tudunk adni egy olyan A veremautomatát, amelyre $L(A) = L(G)$ teljesül.

- Tétel:

Minden A veremautomatához meg tudunk adni egy környezetfüggetlen G grammatikát úgy, hogy $L(G) = N(A)$ teljesül.

3 Reguláris nyelvek tulajdonságai és alkalmazásai

3.1 3-típusú grammatikák normálformája

Minden 3-típusú, azaz reguláris nyelv generálható egy olyan grammatikával, amelynek szabályai:

- $X \rightarrow aY$, ahol $X, Y \in N$ és $a \in T$
- $X \rightarrow \varepsilon$, ahol $X \in N$

3.2 Reguláris kifejezések

Motiváció

Ismeretes, hogy minden véges nyelv reguláris. Tudjuk továbbá, hogy az \mathcal{L}_3 nyelvosztály (a reguláris nyelvek osztálya) zárt az unió, a konkatenáció és az iteráció lezártja műveletekre nézve.

Következésképpen, kiindulva véges számú véges nyelvből és az előzőekben felsorolt, ún. reguláris műveleteket véges sokszor alkalmazva reguláris nyelvet kapunk.

Kérdés az, hogy vajon ezzel az eljárással minden reguláris nyelvet elő tudunk-e állítani, azaz, ez a módszer elégséges-e az \mathcal{L}_3 nyelvosztály leírására?

Definíció

Legyenek V és $V' = \{\varepsilon, \cdot, +, *, (,)\}$ diszjunkt ábécék. A V ábécé feletti reguláris kifejezéseket rekurzív módon a következőképpen definiáljuk:

1. ε reguláris kifejezés V felett.
2. Minden $a \in V$ reguláris kifejezés V felett.
3. Ha R reguláris kifejezés V felett, akkor $(R)^*$ is reguláris kifejezés V felett. [iteratív lezáras]
4. Ha Q és R reguláris kifejezések V felett, akkor $(Q) \cdot (R)$ [konkatenáció] és $(Q) + (R)$ [unió] is reguláris kifejezés V felett.

Megjegyzés: Minden reguláris kifejezés jelöl (meghatároz) valamely reguláris nyelvet. (Pl.: ε a $\{\varepsilon\}$ nyelvet, $a + b$ az $\{a\} \cup \{b\} = \{a, b\}$ és $a \cdot b$ az $\{a\}\{b\} = \{ab\}$ nyelvet.) A reguláris kifejezés a szintaxis, az, hogy hogyan értelmezzük, a szemantika.

Axiómák

P, Q, R reguláris kifejezések. Ekkor fennállnak a következő tulajdonságok:

- Asszociativitás:

$$\begin{aligned} P + (Q + R) &= (P + Q) + R \\ P \cdot (Q \cdot R) &= (P \cdot Q) \cdot R \end{aligned}$$

- Kommutativitás:

$$P + Q = Q + P$$

- Disztributivitás:

$$\begin{aligned} P \cdot (Q + R) &= P \cdot Q + P \cdot R \\ (P + Q) \cdot R &= P \cdot R + Q \cdot R \end{aligned}$$

- Egységelem:

$$\begin{aligned} \varepsilon \cdot P &= P \cdot \varepsilon = P \\ P^* &= \varepsilon + P \cdot P^* \\ P^* &= (\varepsilon + P)^* \end{aligned}$$

A fenti axiómák azonban még önmagukban nem elegendőek az összes reguláris kifejezés előállítására (helyettesítés segítségével). Szükség van még az alábbi inferencia szabályra:

$$P = R + P \cdot Q \quad \wedge \quad \varepsilon \notin Q \quad \implies \quad P = R \cdot Q^*$$

Vegyük még hozzá az \emptyset szimbólumot a reguláris kifejezések halmazához, amely az üres nyelvet jelöli. (Ebben az esetben nincs szükségünk az ε szimbólumra, mivel $\emptyset^* = \{\varepsilon\}$). Így, a definícióban helyettesíthetjük az ε szimbólumot az \emptyset szimbólummal. Ekkor helyettesítjük ε -t a megelőző axióma rendszerben $(\emptyset)^*$ -gal és még egy további axiómát tekintünk:

$$\emptyset \cdot P = P \cdot \emptyset = \emptyset$$

A fenti szabályok elégségesek ahhoz, hogy levezessünk minden érvényes egyenlőséget reguláris kifejezések között.

Reguláris kifejezések és reguláris nyelvek

Minden reguláris kifejezés egy reguláris (3-típusú) nyelvet jelöl, és megfordítva, minden reguláris nyelvhez megadható egy, ezen nyelvet jelölő reguláris kifejezés. (Ezzel választ adtunk a motivációban feltett kérdésre.)

3.3 Lineáris grammatikák és nyelvek

Definíció

Egy $G = (N, T, P, S)$ környezetfüggetlen grammatikát lineárisnak nevezünk, ha minden szabálya:

1. $A \rightarrow u$, $A \in N, u \in T^*$
2. $A \rightarrow u_1 B u_2$, $A, B \in N, u_1, u_2 \in T^*$

Továbbá G -t bal-lineárisnak, illetve jobb-lineárisnak mondjuk, ha $u_1 = \varepsilon$, illetve $u_2 = \varepsilon$ minden 2. alakú szabályra.

Egy L nyelvet lineárisnak, bal-lineárisnak, illetve jobb-lineárisnak mondunk, ha van olyan G lineáris, bal-lineáris, illetve jobb-lineáris grammatika, amelyre $L = L(G)$ teljesül.

Lineáris és reguláris grammatikák, nyelvek

A jobb-lineáris grammatikák azonosak a reguláris grammatikákkal (3-típusúak).

Tétel:

Minden bal-lineáris grammatika reguláris (3-típusú) nyelvet generál.

4 Környezetfüggetlen nyelvek tulajdonságai és elemzésük

4.1 Környezetfüggetlen grammatikák normálformái

Környezetfüggetlen grammatikák normálformái olyan grammatikai transzformációval előállított grammatikák, melyek:

- bizonyos szintaktikai feltételeknek/tulajdonságoknak tesznek eleget
- (általában) valamilyen szempontból egyszerűbbek, mint az eredeti grammatikák
- ugyanazt a nyelvet generálják (így ugyanazon típusba tartoznak)

Tétel:

Minden $G = (N, T, P, S)$ környezetfüggetlen grammatikához meg tudunk konstruálni egy vele ekvivalens $G' = (N', T, P', S')$ környezetfüggetlen grammatikát úgy, hogy: G' minden szabályának jobboldala nemüres szó [kivéve azt az esetet, mikor $\varepsilon \in L(G)$, ekkor $S' \rightarrow \varepsilon$ az egyetlen szabály, melynek jobboldala az üres szó és ekkor S' nem fordul elő G' egyetlen szabályának jobboldalán sem.]

ε -mentes grammatika

A G grammatika ε -mentes, ha egyetlen szabályának jobboldala sem az üres szó.

Tétel:

Minden környezetfüggetlen G grammatikához meg tudunk konstruálni egy G' ε -mentes környezetfüggetlen grammatikát, amelyre $L(G') = L(G) - \{\varepsilon\}$ teljesül.

Chomsky normálforma

A $G = (N, T, P, S)$ környezetfüggetlen grammatikát Chomsky-normálformájúnak mondjuk, ha minden egyes szabálya:

- $X \rightarrow a$, ahol $X \in N, a \in T$
- $X \rightarrow YZ$, ahol $X, Y, Z \in N$

Tétel:

Minden ε -mentes $G = (N, T, P, S)$ környezetfüggetlen grammatikához meg tudunk konstruálni egy vele ekvivalens $G' = (N', T, P', S)$ Chomsky-normálformájú környezetfüggetlen grammatikát.

Tétel (az előző következménye):

Minden G környezetfüggetlen grammatika esetében eldönthető, hogy egy u szó benne van-e G grammatika által generált nyelvben.

Redukált grammatika

- A környezetfüggetlen grammatika egy nemterminálisát inaktívnak/nem aktívnak nevezzük, ha nem vezethető le belőle terminális szó.

- A környezetfüggetlen grammatika egy nemterminálisát nem elérhetőnek nevezzük, ha nem fordul elő egyetlen olyan szóban sem, amely a kezdőszimbólumból levezethető.
- Egy nemterminálist nem hasznosnak mondunk, ha vagy inaktív, és/vagy nem elérhető.
- Az, hogy egy A nemterminális elérhető-e vagy aktív-e, az eldönthető.
- Egy környezetfüggetlen grammatika redukált, ha minden nemterminálisa aktív és elérhető.

Tétel:

Minden környezetfüggetlen grammatikához meg tudunk konstruálni egy vele ekvivalens redukált környezetfüggetlen grammatikát.

4.2 Levezetési fa

A környezetfüggetlen grammatikák levezetéseit fákkal is jellemezhetjük. A levezetési fa egy szó előállításának lehetőségeiről ad információkat. A levezetési fa egy irányított gráf, amely speciális tulajdonságoknak tesz eleget:

- A gyökér címkéje: S
- A többi csúcs címkéje $(N \cup T)$ valamely eleme

A levezetési fa nem minden esetben adja meg a levezetés során alkalmazott szabályok sorrendjét. Két levezetés lényegében azonos, ha csak a szabályok alkalmazásának sorrendjében különbözik.

Egy környezetfüggetlen grammatika minden levezetési fája egy egyértelmű (egyetlen) legbaloldalibb levezetést határoz meg. A legbaloldalibb levezetés során minden levezetési lépésben a legbaloldalibb nemterminálist kell helyettesítenünk.

Tétel:

Minden környezetfüggetlen grammatikáról eldönthető, hogy az általa generált nyelv az üres nyelv-e vagy sem.

4.3 Bar-Hillel Lemma

Minden L környezetfüggetlen nyelvhez meg tudunk adni két p és q természetes számot úgy, hogy minden olyan szó L -ben, amely hosszabb, mint p

$$uxwyz$$

alakú, ahol $|xwy| \leq q$, $xy \neq \varepsilon$, és minden

$$ux^iwy^iv$$

szó is benne van az L nyelvben minden $i \geq 0$ egész számra $(u, x, w, y, v \in T^*)$.

Tétel:

Eldönthető, hogy egy környezetfüggetlen grammatika végtelen nyelvet generál-e vagy sem.