Automaták és formális nyelvek

Jegyzet a programozó matematikus szak
IV. félévéhez

Készítette:
Dr. Hunyadvári László
Manhertz Tamás

Lektorálta: Lipták László

E kötet a programozó matematikus szak nappali tagozatán 1995-ben elhangzott IV. félévi, Dr. Hunyadvári László egyetemi docens által tartott előadássorozata alapján készült.

Küön köszönet illeti meg Csizmazia Balázst, aki támogatásával elősegítette jelen dokumentáció elkészülését.

Első kiadás, utolsó javítás dátuma: 1996. szeptember 30.

Készült Donald E. Knuth TEX szövegszedő rendszerével.

Budapest, Eötvös Loránd Tudományegyetem Természettudományi Kar, 1996.

1. Bevezetés

1.1. Történelmi háttér

A formális nyelvek elmélete *Noam Chomsky* munkásságával indult az 1950-es években. Főként a természetes nyelvek struktúrájának formális jellemzésére törekedett, azaz megpróbálta formális módszerekkel leírni a nyelvtanok szabályait. Erre háromféle modellt is kidolgozott, melyeket 1956-ban egy kiadványában meg is jelentetett. Ezek közül az első modell (finite-state grammar) nem került alkalmazásba, a második modell (transformation grammar) a természetes nyelvek tanulmányozásában vált fontossá, és a harmadik modell (phrase-structure grammar) vált mérföldkővé a formális nyelvek elméletében.

Chomsky a kifejezés rendszerű modelljében (phrase-structure grammar) a nyelvet kibővítette olyan szimbólumokkal, melyek nem tartoznak a nyelv jelei közé, és ezeket elnevezte „nemterminális" jeleknek. Ezek a jelek mindig valamilyen kifejezést, szót, esetleg teljes szöveget jelentenek. Mindezek mellett bevezetett helyettesítési szabályokat, melyek egy szimbólumsorozat valamely részét helyettesítették új szimbólumsorozattal. (Jelöljük nagybetűkkel a nemterminális jeleket.) Például:

$$ABC \longrightarrow ADEFC$$

Itt látható, hogy a B szimbólum lett helyettesítve a DEF szimbólumsorozattal.

Chomsky két nagy csoportra osztotta a modelljében konstruált nyelvtanokat; környezetfüggő és környezetfüggetlen nyelvtanokra. Mint a későbbiekben látni fogjuk, ezek fontos szerepet töltenek be mind a formális nyelvek, mind pedig a fordítóprogramok tárgykörén belül. Az előbbi példában egy környezetfüggő szabályt láthattunk, ahol B szimbólum környezete A és C.

A 60-as években Chomsky főleg a környezetfüggetlen nyelvtanok elméletén dolgozott, de később visszatért a természetes nyelvek tanulmányozásához. Eközben többen folytattak kutatást a környezetfüggetlen nyelvtanok tulajdonságainak felderítésére, melyben több problémára is rámutattak, és jónéhány megválaszolatlan kérdésre is adtak bizonyítást (Bar-Hillel, Perles, Odgen, Parikh). 1966-ban Seymour Ginsburg jelentette meg könyvét *The Mathematical Theory of Context-free Languages* címmel, melyben összegezte a környezetfüggetlen nyelvtanok vizsgálatával összefüggő eredményeket. Tulajdonképpen ez az első olyan munka, mely rámutatott a formális nyelvek és az automaták elméletének kapcsolatára.

A 60-as évek végén és a 70-es évek folyamán új fejlesztések indultak el a környezetfüggetlen nyelvtanok formális elemző módszerei irányába. Ebben a témakörben is született egy tanulmány *The Theory of Parsing, Translation, and Compiling* címmel (Aho & Ullman), mely a fordítóprogramok elméletének egyik alapműve.

1.2. Programozási nyelvek

1.3. Extended BNF (EBNF)

1.4. Szintaxis diagramm

2. Alapfogalmak

2.1. Ábécé, szavak, nyelv.

Definíció:

Ábécének nevezünk egy tetszőleges véges szimbólumhalmazt.

Ennek jelölésére az X, Y betűket fogjuk használni és a továbbiakban T az úgynevezett terminális szimbólumhalmazt fogja jelenteni. Egy tetszőleges ábécé elemeit a, b, c betűk fogják jelölni.

Definíció:

Az X ábécé elemeinek egy tetszőleges véges sorozatát az X ábécé feletti szónak nevezzük. Ha X nem lényeges vagy egyértelmű, akkor szóról beszélünk. Egy szó jelölésére az u, v, w betűket fogjuk használni és egy tetszőleges u szó ábécéjét $X(u)$-val azonosítjuk, azaz formálisan:

$$X(u) \ := \ \{u \text{ szó különböző szimbólumai}\}.$$

Az X ábécé feletti összes szó halmazát jelölje X^*. Az ε jelentse az üres szót és legyen $X^+ := X^* \setminus \{\varepsilon\}$.

Definíció:

Az X^* valamely részhalmazát az X ábécé feletti nyelvnek nevezzük (2^{X^*} eleme).

Egy nyelv jelölésére az L betűt, illetve ennek indexelt változatát fogjuk használni. Ha X nem lényeges vagy egyértelmű, akkor nyelvről beszélünk. $X(L)$ jelöli az L nyelv ábécéjét és megegyezés szerint az u szót azonosítjuk az $\{u\}$ nyelvvel.

2.2. Műveletek szavakkal

1. Konkatenáció:

   Legyenek u, v szavak egy adott L nyelvből. Ekkor a két szó konkatenációja uv. (A két szó egymás utáni leírásával kapott szó.) Ez egy asszociatív művelet, melynek ε az egységeleme, így

   $$\langle X^*, \text{konkatenáció}, \varepsilon \rangle$$

   egy egységelemes félcsoportot képez.

2. Megfordítás:

   Legyen u a szó, ekkor a megfordítása u^R.

3. Szó hossza:

Egy adott u szó hossza a benne lévő, ábécébeli jelek száma. Jelölése: $l(u)$. Egy adott $y \in X$ jel száma egy adott u szóban: $l_y(u)$. Legyen $H \subseteq X$. Jelölje u szóban lévő H-beli szimbólumok számát $l_H(u)$. Például $l_{\{a,b\}}(abcac) = 3$.

4. Homomorfizmus (átkódolás):

Legyenek X, Y tetszőleges ábécék. Ekkor a $h : X^* \longmapsto Y^*$ konkatenációtartó leképezést homomorfizmusnak hívjuk. (Tehát u, v szavak esetén $h(uv) = h(u)h(v)$, és $h(\varepsilon) = \varepsilon$.) Ekkor h konkatenációtartó tulajdonsága miatt elég h-t az X-en megadni.

Nyelvekre vonatkozó kiterjesztése:

$$h(L) := \bigcup_{u \in L} h(u).$$

2.3. Műveletek nyelvekkel

1. Halmazműveletek:

Véges sok halmazelméleti művelet eredményéhez mindig megadható egy ábécé, mely fölötti nyelv az eredmény. Legyenek $L_1, L_2, \ldots, L_k$ nyelvek. Ezekkel végzett művelet eredményének megfelelő ábécé:

$$\bigcup_{i=1}^{k} X(L_i).$$

Végtelen sok művelet esetén garantálni kell valahogy azt, hogy az eredményhez létezzen ábécé. A következő példa is ennek a fontosságát mutatja:

Legyen

$$\Delta_n := \{(_1, (_2, \ldots, (_n,)_1,)_2, \ldots,)_n\} \text{ egy ábécé, és}$$

$$D_n := HE(\Delta_n),$$

azaz a Δ_n feletti helyes zárójelezések halmaza. Ekkor

$$D = \bigcup_{i=1}^{\infty} D_i$$

már nem lesz nyelv, mert nincsen véges ábécéje.

2. Konkatenáció:

Legyenek L_1, L_2 nyelvek. Ekkor az L_1 és L_2 nyelvek konkatenációján az $L_1L_2 := \{uv \mid u \in L_1 \wedge v \in L_2\}$ nyelvet értjük. (Ez tulajdonképpen komplexusszorzás.)

A konkatenáció rendelkezik az asszociativitás tulajdonságával. Erre vonatkozó egységeleme $\{\varepsilon\}$. Az unió és a konkatenáció között mindemellett kétoldalú disztributivitás áll fenn, azaz

$L(L_1 \cup L_2) = LL_1 \cup LL_2$, valamint

$(L_1 \cup L_2)L = L_1L \cup L_2L$.

Ezekkel a tulajdonságokkal a

$$\left\langle 2^{X^*}, \{\cup, \text{konkatenáció}\}\right\rangle$$

algebra félgyűrűt ad.

3. Helyettesítés (nemdeterminisztikus átkódolás):

Legyenek X, Y ábécék, és $\Phi : 2^{X^*} \longmapsto 2^{Y^*}$. *A* Φ leképezést helyettesítésnek nevezzük, ha unió- és konkatenációtartó. (Tehát a 2^{X^*} és a 2^{Y^*} félgyűrűk közötti algebrai homomorfizmusról van szó.)

Megjegyzés: A homomorf leképezésből adódóan $\Phi(\emptyset) = \emptyset$, valamint $\Phi(\{\varepsilon\}) = \{\varepsilon\}$ is fennáll. A művelettartás miatt Φ-t elegendő X elemein megadni. Tetszőleges homomorfizmus tekinthető helyettesítésnek.

4. Lezárás, iteráció.

Legyen L egy nyelv. Ekkor L iteráltján vagy lezártján a következőt értjük:

$$L^* := \bigcup_{i=0}^{\infty} L^i, \text{ ahol } L^i := LL\ldots L, \text{ és } L^0 := \{\varepsilon\}.$$

Például:

$$\{ab, ba\}^* = \{\varepsilon, ab, ba, abab, abba, baab, baba, ababab, \ldots\}.$$

Pozitív lezárás (ε nélkül):

$$L^+ := \bigcup_{i=1}^{\infty} L.$$

2.4. Nyelvek definiálásának módszerei

Egy nyelv definiálására többféle módszer létezik, de mindegyik esetében követelmény, hogy véges leírást adjunk.

Definíció:

Legyen X egy adott ábécé. Elemi nyelveknek nevezzük azokat a L nyelveket, melyek megfelelnek az alábbi két feltételnek:

a) L elemi nyelv, ha $\{a\}, \{\varepsilon\}, \emptyset$ valamelyike, ahol $a \in X$,

b) vagy elemi nyelvekből az unió, konkatenáció, vagy a lezárás műveletek segítségével hozzuk létre.

Például legyen $X := \{a, b\}$ ábécé, ekkor $\{\{a\}\{b\} \cup \{b\}\{a\}\}^*$ elemi nyelv.

Feladat: Le lehet-e írni minden nyelvet ezen műveletek véges alkalmazásával?

Nyelvek definiálására a következő módszerek léteznek:

1° Véges nyelvek esetén felsorolhatjuk a nyelv elemeit.

2° Logikai formulával.

Például legyen $X := \{a, b\}$ és $L := \{a^n b^n \mid n \in \mathcal{N}\}$.

3° Strukturális indukcióval.

Ekkor adott egy véges elemi nyelvkészlet (véges sok véges nyelv) és adott egy véges műveletkészlet. Úgy adjuk meg a nyelvet, hogy megmondjuk, mely elemi nyelvekből és milyen megengedett műveletekkel áll elő (véges sok nyelvből véges sok művelettel).

4° Algoritmus segítségével.

Definíció:

Az L nyelv rekurzívan felsorolható $\Longleftrightarrow$ ha létezik A algoritmus, mely az elemeit felsorolja.

Rekurzív felsorolás: egy adott algoritmus az outputjára szavakat állít elő, s így a nyelv összes szavát felsorolja (esetleg végtelen sokat).

Definíció:

Az L nyelv parciálisan rekurzív $\Longleftrightarrow$ létezik A algoritmus, mely minden $u \in L$ szó esetén *igen* válasszal áll le, míg $u \notin L$ esetén nem terminál, vagy ha terminál, akkor *nem* választ ad.

Definíció:

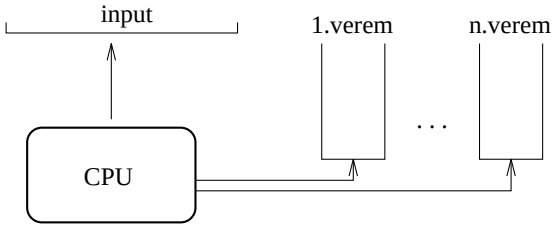
Az L nyelv rekurzív $\Longleftrightarrow$ létezik eldöntő algoritmus, mely inputjára egy tetszőleges szót helyezve eldönti, benne van-e a nyelvben (*igen* a válasz, ha eleme a nyelvnek, és *nem* a válasz ellenkező esetben).

Megjegyzés: Az első és a második definíció egyenértékű, az utolsó szűkebb.

5° Matematikai gépekkel:

Az absztrakt, matematikai gépek felépítése a következő: Adott egy potenciálisan végtelen input szalag, amelyre fel van írva az elolvasásra megadott szó. A gép ezt az input szalagot egy olvasófejjel olvassa végig. Adott továbbá n darab lista (verem), melyek mindegyikéhez tartozik egy író/olvasófej. Ezeket pedig egy úgynevezett központi egység vezérli, ami véges sok állapottal rendelkezik. Ez az állapot határozza meg a gép további működését.

Egy ilyen matematikai gép például a Turing-gép.



2.1. Ábra: A matematikai gépek felépítése

Működése:

Adott egy diszkrét időskála, melynek minden ütemére a gép végrehajt egy lépést. Egy ilyen lépés tulajdonképpen két alapvető részből áll:

1) Felderítés: Milyen jelet olvastunk, a CPU milyen állapotban van, és mi van a vermek tetején.
2) Tényleges működés: A CPU új állapotba kerül(het), vermekbe új jeleket ír(hat)unk, olvasófejet mozgatjuk.

Ha létezik olyan működéssorozat, amelyre a szekvenciális inputot elolvassa a gép, és a CPU végállapotba kerül, akkor a szó a nyelvbe tartozik, különben nem.

6° Produkciós rendszerrel.

Itt axiómák és következtetési szabályok segítségével adjuk meg a nyelvet. A produkciós rendszer olyan szavakból álló nyelvet ír le, amelyek az axiómákból levezethetők, illetve az axiómákra visszavezethetők. Speciális produkciós rendszer az úgynevezett generatív grammatika (formális nyelvtan).

3. Nyelvek megadása produkciós rendszerrel

Definíció:

A Π produkciós rendszer alatt a következő hármast értjük:

$$\Pi = \langle X, P, A_x \rangle,$$

ahol X az ábécé, $P \subseteq X^* \times X^*$ a produkciós szabályok halmaza, és $A_x \subseteq X^*$ az axiómahalmaz.

Megállapodás szerint $(p, q) \in P$ esetén az írásmód: $p \longrightarrow q$.

Definíció:

Közvetlen levezetés Π-ben: Az $a \in X^*$ szóból közvetlenül levezethető a $b \in X^*$ szó a Π produkciós rendszerben $\iff$ létezik $g_1, g_2 \in X^*$ és $p \longrightarrow q \in P$ szabály, hogy $a = g_1 p g_2$, és $b = g_1 q g_2$.

Jelölésben: $a \underset{\Pi}{\longrightarrow} b$.

Definíció:

Közvetett levezetés Π-ben: Az $a \in X^*$ szóból közvetett módon levezethető a $b \in X^*$ szó a Π produkciós rendszerben $\iff$ van olyan $k \in \mathcal{N}$ és vannak olyan $\delta_0, \delta_1, \ldots, \delta_k \in X^*$ szavak, hogy $a = \delta_0, b = \delta_k$, és minden $i \in [0, k-1]$ esetén $\delta_i \underset{\Pi}{\longrightarrow} \delta_{i+1}$.

Jelölésben: $a \overset{*}{\underset{\Pi}{\longrightarrow}} b$.

Definíció:

Az előbbi definícióban szereplő k számot a levezetés hosszának nevezzük.

Megjegyzés: Amennyiben $k = 0$, akkor $a = \delta_0 = b$. Tehát a definíció közvetlen következménye, hogy minden szó levezethető saját magából, tehát a levezetés reflexív. Vezessük be jelölésként a pontosan k hosszú levezetést a következőképpen: $a \overset{(k)}{\underset{\Pi}{\longrightarrow}} b$.

Pozitív levezetésről beszélünk, ha $k \geq 1$. Jelölésben: $a \overset{+}{\underset{\Pi}{\longrightarrow}} b$.

3.1. A Π által leírt nyelv

Definíció:

A Π produkciós rendszer által (a T ábécére relatív) generált nyelv:

$$L_T^g(\Pi) = \{u \in T^* \mid \exists a \in A_x : a \overset{*}{\underset{\Pi}{\longrightarrow}} u\}.$$

Definíció:

A Π produkciós rendszer által elfogadott nyelv (akceptált nyelv):

$$L_T^a(\Pi) = \{u \in T^* \mid \exists a \in A_x : u \overset{*}{\underset{\Pi}{\longrightarrow}} a\}.$$

Néhány példa:

Legyen $\Pi_1 := \langle \{(,)\}, \{() \longrightarrow \varepsilon\}, \{\varepsilon\} \rangle$, ekkor a Π_1 által elfogadott nyelv a helyes zárójelezések halmaza, azaz $L^a_{\{(,)\}}(\Pi_1) = HE$.

Legyen $\Pi_2 := \langle \{\mathrm{S}, (,)\}, \mathcal{P}, \{\mathrm{S}\} \rangle$, ahol $\mathcal{P} := \{\mathrm{S} \longrightarrow (\mathrm{S}), \mathrm{S} \longrightarrow \mathrm{SS}, \mathrm{S} \longrightarrow \varepsilon\}$. Ekkor a generált nyelv a helyes zárójelezések halmaza, tehát $L^g_{\{(,)\}}(\Pi_1) = HE$.

Megjegyzés: Az utóbbi példában szereplő $\mathcal{P}$ halmazt — a rövidebb írásmód kedvéért — a következő módon szokták megadni: $\mathcal{P} := \{\mathrm{S} \longrightarrow (\mathrm{S}) \mid \mathrm{SS} \mid \varepsilon\}$.

3.2. Generatív nyelvtanok

A generatív nyelvtanok speciális produkciós rendszerek, néhány feltétellel megszorítva. Négy alapvető tulajdonságuk van:

a) $X = T \cup N$ és $T \cap N = \emptyset$, ahol T a terminális jelek, N a nyelvtani jelek halmaza.

b) $p \longrightarrow q \in P$ esetén p-ben van legalább egy nyelvtani jel.

c) $A_x = \{S\}$, ahol $S \in N$ a kezdőszimbólum.

d) Csak a T-re relatív generálás megengedett.

Definíció:

Generatív nyelvtannak (formális nyelvtannak) nevezzük az alábbi négyest:

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

ahol T a terminális jelek halmaza, N a nyelvtani jelek halmaza, $\mathcal{P}$ egy véges szabályhalmaz, amely bármely szabályának bal oldalán legalább egy nyelvtani jel áll, és $S \in N$ a kezdőszimbólum.

Definíció:

A G által generált nyelv az összes — az S kezdőszimbólumból — levezethető szónak a halmaza, azaz:

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

Feladat:

Legyen $G_1 = \langle \{a, b\}, \{\mathrm{S}\}, \{\mathrm{S} \longrightarrow aSb \mid bSa \mid ab \mid ba \mid \mathrm{SS}\}, \mathrm{S} \rangle$. Bizonyítsuk be, hogy ekkor a generált nyelv:

$$L(G_1) = \{u \in \{a, b\}^* \mid l_a(u) = l_b(u) > 0\}.$$

Megoldás:

„$\supseteq$”:

Először legyen $u \in \{x \in \{a,b\}^* \mid l_a(x) = l_b(x) > 0\}$, és bizonyítani kell, hogy ekkor $u \in L(G_1)$. Tehát meg kell mutatni, hogy ekkor létezik olyan levezetés G_1-ben az S kezdőszimbólumból, mely ezt az u szót adja. Legyen $n := l_a(u) = l_b(u)$, és n-re vonatkozó teljes indukcióval igazoljuk:

$n = 1$: Ekkor $u = ab$ vagy $u = ba$, mely S kezdőszimbólumból az $S \longrightarrow ab$ vagy az $S \longrightarrow ba$ szabályok alkalmazásával kapjuk meg.

$n + 1$: Most n-re elfogadjuk és bizonyítjuk $n + 1$-re. Az alábbi eseteket kell vizsgálni u szó felépítése alapján:

$u = abw$ esetén indukciós feltevésünk alapján $S \xrightarrow[G_1]{*} w$, ezért $S \longrightarrow SS \longrightarrow abS \xrightarrow[G_1]{*} abw = u$,

$u = baw$ esetén $S \longrightarrow SS \longrightarrow baS \xrightarrow[G_1]{*} baw = u$,

$u = awb$ esetén $S \longrightarrow aSb \xrightarrow[G_1]{*} awb = u$, illetve

$u = bwa$ esetén $S \longrightarrow bSa \xrightarrow[G_1]{*} bwa = u$.

„$\subseteq$”:

Most legyen $u \in L(G_1)$ és meg kell mutatni, hogy $u \in \{x \in \{a,b\}^* \mid l_a(x) = l_b(x) > 0\}$. Ehhez nyilván elég belátni azt, hogy a szabályok tetszőleges alkalmazásával csak olyan u szavak vezethetők le, melyekben az a és b szimbólumok száma megegyezik. Ez viszont teljesül, mert $\mathcal{P}$ csak olyan szabályokat tartalmaz, melyek azonos számú a, b jeleket tartalmaznak, és a legrövidebb levezethető szó is legalább 2 hosszú.

$\triangle$

4. Nyelvtanok osztályozása

A generatív nyelvtanok osztályozását a szabályok alakja alapján tehetjük meg. Egy megkötést már ismerünk: egy nyelvtani elemnek a szabály bal oldalán mindenképpen szerepelnie kell.

A szabályok alakja szerint a következő osztályozás lehetséges. Ez egyben a nyelv típusát jelenti:

0. típus:

Itt nincs további megkötés, azaz minden nyelvtan egyben 0-ás típusú is.

1. típus:

Két szabályforma megengedett:

a) $a_1 A a_2 \longrightarrow a_1 q a_2$, ahol $a_1, a_2 \in (T \cup N)^*$ a környezet, $A \in N$, és $q \in (T \cup N)^+$ (azaz $q \neq \varepsilon$). Ez a környezetfüggő szabályforma.

b) $S \longrightarrow \varepsilon$, ugyanis az a) pont alapján nem lehet ε-ra vezetni. Erre vonatkozó megkötés: ha

van ilyen szabály, akkor S nem fordulhat elő más szabály jobb oldalán. Ekkor S csupán ε levezetésére szolgál.

2. típus:

Itt is két szabályforma lehetséges:

a) $A \longrightarrow q$, ahol $A \in N$, és $q \in (T \cup N)^+$. Ez a környezetfüggetlen szabályforma.

b) hasonlóan, mint az 1. típus b) pontja.

Megjegyzés: Ez a típus az 1. típus megszorítása: a_1, a_2 mindig ε-nal egyenlő.

3. típus:

Itt három szabályforma megengedett:

a) $A \longrightarrow aB$, ahol $A, B \in N$, és $a \in T$.

b) $A \longrightarrow q$, ahol $A \in N$ és $q \in T$.

c) $S \longrightarrow \varepsilon$, ahol S a kezdőszimbólum.

Megjegyzés: Ez meg a 2. típus megszorítása, ugyanis q csak ilyen speciális alakú lehet.

A különböző típusú nyelvtanok osztályokat alkotnak, melyeket rendre $\mathcal{G}_i$-vel jelölünk. Tehát:

$\mathcal{G}_i := \{i.$ típusú nyelvtanok$\}$.

Könnyen belátható, hogy $\mathcal{G}_3 \subseteq \mathcal{G}_2 \subseteq \mathcal{G}_1 \subseteq \mathcal{G}_0$. Hasonló módon a nyelveket is osztályozhatjuk:

$\mathcal{L}_i := \{L \mid L$ nyelv, és van olyan $G \in \mathcal{G}_i$, amelyre $L(G) = L\}$.

Az előző nyelvtanhierarchia alapján $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$. Ez az ún. **Chomsky-féle hierarchia-tétel**.

A műveleteket tekintve nem minden osztály zárt, ugyanis például L_2 nem zárt a metszetre nézve. (Lásd később a zártsági tételeket.)

Legyen T véges, nemüres ábécé. Világos, hogy 2^{T^*} a T ábécé feletti összes nyelv halmazát jelenti. Jelölje $\mathcal{L}_0^T$ azokat a T feletti nyelveket, melyek nyelvtannal leírhatók. Kérdés: Vajon minden nyelvhez megadható-e olyan nyelvtan, ami azt a nyelvet generálja? Erre ad választ a következő tétel.

Tétel:

Nem minden nyelv írható le nyelvtannal.

Bizonyítás:

Ehhez azt kell belátnunk, hogy $\mathcal{L}_0^T \subset 2^{T^*}$, ahol T^* megszámlálható számosságú, és 2^{T^*} kontinuum számosságú. Emiatt elég belátni, hogy $\mathcal{L}_0^T$ megszámlálható számosságú.

Legyen N a potenciális nyelvtani jelek halmaza, ami megszámlálható számosságú. Legyen $Ny := \{G \mid G$ nyelvtani jelei N-ből, terminális jelei pedig T-ből valók$\}$.

Állítás: Minden $L \in \mathcal{L}_0^T$ esetén létezik olyan $G \in Ny$, amelyre $L(G) = L$.

Ez az állítás könnyen belátható, hiszen minden ilyen L-hez létezik egy G' nyelvtan, melyre $L(G') = L$. Ebben a G' nyelvtanban vagy minden nyelvtani jel N-ből származik, vagy ha létezik olyan, ami nincs N-ben, akkor egyszerűen csak átcseréljük. (Biztos, hogy át tudjuk cserélni, mert rendelkezésünkre áll megfelelő mennyiségű nyelvtani jel.)

Legyen $\Phi : Ny \longmapsto \mathcal{L}_0^T$, és $\Phi(G) := L(G)$. Ekkor ez a Φ egy ráképezés, tehát a teljes $\mathcal{L}_0^T$-re képez. Tehát nyilván fennáll, hogy $|Ny| \geq |\mathcal{L}_0^T|$. Ez alapján elég belátni, hogy Ny megszámlálható számosságú.

Ekkor a G nyelvtant a következőképp írhatjuk fel:

$$G = \langle \{t_1, \ldots, t_k\}, \{A_1, \ldots, A_2\}, \{x_1 \ldots x_l \longrightarrow y_1 \ldots y_s, \ldots\}, S \rangle.$$

Viszont Φ ráképezés volta miatt minden ilyen G-hez hozzá tudjuk rendelni a generált nyelvét, ezért tekinthetjük G-t az $X := N \cup T \cup \{<, >, \longrightarrow, \{, \}\} \cup \{,\}$ halmazbeli jelek véges sorozatának. Tehát $Ny \subseteq X^*$, azaz ebből már nyilván következik, hogy $|Ny| \leq |X^*|$, ami pontosan a tétel bizonyítását jelenti, mert X^* megszámlálható számosságú.

$\triangle$

Church-tézis:

Minden valamilyen konstruktív módon megadható nyelv leírható nyelvtannal.

5. Kiterjesztési tételek

Definíció:

A G nyelvtan kiterjesztett 1-es típusú nyelvtan, ha minden szabálya megfelel a következő feltételnek:

i) $p \longrightarrow q \in \mathcal{P}$ esetén $l(p) \leq l(q)$, azaz a szabály nem csökkenti a hosszt.

ii) $S \longrightarrow \varepsilon$ alakú, ahol S a kezdőszimbólum.

Jelölés: $\mathcal{G}_{\text{kit1}}$.

Definíció:

A G nyelvtan kiterjesztett 2-es típusú, ha a szabályok alakja $A \longrightarrow q \in \mathcal{P}$, ahol $A \in N, q \in (T \cup N)^*$.

Jelölés: $\mathcal{G}_{\text{kit2}}$.

Definíció:

A G nyelvtan kiterjesztett 3-as típusú, ha a szabályok alakja $A \longrightarrow uB$ vagy $A \longrightarrow u$, ahol $A, B \in N$, és $u \in T^*$.

Jelölés: $\mathcal{G}_{\text{kit3}}$.

Ezekkel a jelölésekkel nyilván $\mathcal{G}_i \subseteq \mathcal{G}_{\text{kit}i}$, hiszen csak gyengítettünk a szabályok megkötésein. Ebből következik, hogy $\mathcal{L}_i \subseteq \mathcal{L}_{\text{kit}i}$.

Tétel:

$$\mathcal{L}_i = \mathcal{L}_{\text{kit}i}, \quad \text{ahol i= } 1, 2, 3$$

Bizonyítás:

Csak $\mathcal{L}_{\text{kit}i} \subseteq \mathcal{L}_i$-t kell belátni, mert a másik irány nyilvánvaló. Ehhez viszont elég a következő lemmát belátni:

Lemma:

Tetszőleges $G \in \mathcal{G}_{\text{kit}i}$ nyelvtanhoz található olyan $G' \in \mathcal{G}_i$, melyre $L(G') = L(G)$.

Bizonyítás:

Ezt mindhárom kiterjesztésre be kell látni.

$i = 1$:

Feltehető, hogy G-ben terminális jel csak $A \longrightarrow a \quad (a \in T, A \in N)$ alakú szabályokban fordul elő.

Ha ez nem így van, akkor G-t a következőképpen alakítjuk át: minden $t \in T$-hez bevezetjük az x_t-vel jelölt álterminálisokat, melyek új nyelvtani jelek lesznek. (Ezek különbözzenek mind egymástól, mind pedig N jeleitől.) $\mathcal{P}$-ben minden szabályban a terminális jeleket kicseréljük a megfelelő álterminális jelekre. Végül felvesszük az $x_t \longrightarrow t$ új szabályokat. Nyilván az így átalakított nyelvtan ugyanazt tudja, mint az eredeti, és terminális jel csak az $x_t \longrightarrow t$ alakú szabályokban van, ami pontosan a kívánt alakú.

Cél olyan G' nyelvtan konstrukciója, mely 1-es típusú, és $L(G') = L(G)$. Ha véletlenül G minden szabálya 1-es típusú, akkor készen vagyunk ($G' = G$). Ha nem, akkor vegyük ki G olyan szabályait, melyek nem 1-es típusúak. Milyen ezeknek az alakja?

Rossz szabály a következő alakú lehet: $X_1X_2 \ldots X_n \longrightarrow Y_1Y_2 \ldots Y_m \quad (m \geq n)$. A rossz szabályokat szimuláljuk csupa 1-es típusú szabállyal. Ehhez új nyelvtani jeleket kell bevezetni, ezek legyenek a $Z_1, Z_2, \ldots, Z_n$ jelek. Szimuláció:

$$X_1X_2 \ldots X_n \longrightarrow Z_1X_2 \ldots X_n,$$

$$Z_1X_2 \ldots X_n \longrightarrow Z_1Z_2X_3 \ldots X_n,$$

$$Z_1 Z_2 \dots Z_{n-1} X_n \longrightarrow Z_1 Z_2 \dots Z_n Y_{n+1} \dots Y_m \quad (n \leq m),$$

$$Z_1 Z_2 \dots Z_n Y_{n+1} \dots Y_m \longrightarrow Y_1 Z_2 \dots Z_n Y_{n+1} \dots Y_m,$$

$$Y_1 \dots Y_{n-1} Z_n Y_{n+1} \dots Y_m \longrightarrow Y_1 Y_2 \dots Y_m.$$

$Z_1, \dots, Z_n$ új volta miatt a szabályokat csak ebben a sorrendben lehet végrehajtani, ezért az új nyelvtan is ugyanazt a nyelvet generálja. Csináljuk meg ezt a szabálytranszformációt az összes „rossz" szabályra. Az így kapott G' nyelvtan már 1-es típusú.

$i = 2$:

Itt a problémát az jelenti, hogy a kiterjesztett 2-es nyelvek esetén megengedett, hogy üres szó álljon a szabályok jobb oldalán. Legyen G ilyen kiterjesztett 2-es típusú nyelvtan. A probléma az $A \longrightarrow \varepsilon$ alakú szabályokkal van. Hogyan viselkednek ε nélkül?

Példa: Legyen $G = \langle T, N, \mathcal{P}, \mathrm{S} \rangle$, ahol $T = \{a, b, c\}$, $N = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{S}\}$, $\mathcal{P}$ pedig a következő szabályokból áll:

$$\mathrm{S} \longrightarrow \mathrm{AB}c | \mathrm{AA},$$

$$\mathrm{B} \longrightarrow \mathrm{CC},$$

$$\mathrm{A} \longrightarrow \varepsilon | a,$$

$$\mathrm{C} \longrightarrow \varepsilon | b.$$

Vegyük például a következő levezetést:

$$\mathrm{S} \underset{G}{\longrightarrow} \mathrm{AB}c \underset{G}{\longrightarrow} \mathrm{B}c \underset{G}{\longrightarrow} \mathrm{CC}c \underset{G}{\longrightarrow} b\mathrm{C}c \underset{G}{\longrightarrow} bc.$$

Viszont nekünk az kell, hogy:

$$\mathrm{S} \underset{G'}{\longrightarrow} \mathrm{B}c \underset{G'}{\longrightarrow} \mathrm{C}c \underset{G'}{\longrightarrow} bc.$$

Azt az algoritmust, mely megoldja az ε-os szabályok ilyen módosítását, ε-mentésítési eljárásnak hívjuk.

Legyen $H := \{A \in N \mid A \overset{*}{\underset{G}{\longrightarrow}} \varepsilon\}$. Alapvető kérdés, hogyan lehet ezt a H halmazt megtalálni? Konstrukciós lépések:

$$H_1 := \{A \in N \mid A \longrightarrow \varepsilon \in \mathcal{P}\},$$

$$H_{i+1} := H_i \cup \{A \in N \mid \exists A \longrightarrow Q \in \mathcal{P} : Q \in H_i^*\}.$$

Ekkor nyilván teljesül a következő összefüggés: $H_1 \subseteq H_2 \subseteq \dots \subseteq H_i \subseteq \dots$. Mivel minden i-re $H_i \subseteq N$, és N halmaz véges, ezért egy i_0 indextől kezdődően biztosan azonosak lesznek ezek a halmazok. Legyen ez az i_0 a legkisebb ilyen index. Ekkor $H_{i_0} = H_{i_0+1}$.

Belátjuk, hogy ekkor $H_1 \subset H_2 \subset \dots \subset H_{i_0} = H_{i_0+1} = H$. Ehhez indukcióval elég azt belátni, hogy $H_j = H_{j+1}$ esetén $H_{j+1} = H_{j+2}$, hiszen ebből következik, hogy ha H_i nem bővebb H_{i-1}-nél,

akkor onnantól kezdve a halmazok ugyanazok lesznek. Tegyük fel, hogy $H_j = H_{j+1}$. Ekkor nyilván $H_j^* = H_{j+1}^*$, és a definíció alapján:

$$H_{j+2} = H_{j+1} \cup \{A \in N \mid \exists A \longrightarrow Q \in \mathcal{P} : Q \in H_{j+1}^*\} =$$
$$= H_j \cup \{A \in N \mid \exists A \longrightarrow Q \in \mathcal{P} : Q \in H_j*\} = H_{j+1}.$$

Tehát a H halmaz megkonstruálható. Következmény: $S \in H \Longleftrightarrow \varepsilon \in L(G)$.

Konstruálunk egy új $\bar{G}$ nyelvtant ezen H segítségével. Legyen $\bar{G} := \langle T, N, \bar{\mathcal{P}}, S \rangle$, ahol $A \longrightarrow \bar{q} \in \bar{\mathcal{P}}$ akkor és csak akkor, ha $\bar{q} \neq \varepsilon \ \wedge \ \exists \mathrm{A} \longrightarrow q \in \mathcal{P}$, hogy $\bar{q}$-t q-ból néhány (esetleg nulla) H-beli jel elhagyásával kapjuk.

Az előbbi példát tekintve az ε-mentesített nyelvtan szabályai:

$H_1 = \{\mathrm{A},\mathrm{C}\}$,

$H_2 = \{\mathrm{A},\mathrm{C}\} \cup \{\mathrm{B},\mathrm{S}\} = H$.

$\bar{\mathcal{P}}$ elemei:

$\mathrm{S} \longrightarrow \mathrm{AB}c \mid \mathrm{B}c \mid \mathrm{A}c \mid c \mid \mathrm{AA} \mid \mathrm{A}$,

$\mathrm{B} \longrightarrow \mathrm{CC} \mid \mathrm{C}$,

$\mathrm{A} \longrightarrow a$,

$\mathrm{C} \longrightarrow b$.

Állítás:

$L(\bar{G}) = L(G) \setminus \{\varepsilon\}$

Bizonyítás:

$L(\bar{G}) \subseteq L(G) \setminus \{\varepsilon\}$:

Legyen $u \in L(\bar{G})$ egy tetszőleges szó. Vizsgáljuk meg a levezetését:

$$S \xrightarrow[\bar{G}]{*} \alpha_1 A \alpha_2 \xrightarrow[\bar{G}]{} \alpha_1 \bar{q} \alpha_2 \xrightarrow[\bar{G}]{*} u.$$

A kiemelt szabály helyettesíthető a G nyelvtan következő levezetésével:

$$A \xrightarrow[G]{} q \xrightarrow[G]{*} \bar{q},$$

ahol a $\bar{q}$-ból elhagyott H-beli elemekből levezetjük ε-t. Ha ezt minden $\bar{G}$-beli szabályra elvégezzük, akkor u egy G-beli levezetését kapjuk. Tehát $S \xrightarrow[G]{*} u$, így $u \in L(G)$. Nyilván $u \neq \varepsilon$ miatt teljesül $u \in L(G) \setminus \{\varepsilon\}$ is.

$L(\bar{G}) \supseteq L(G) \setminus \{\varepsilon\}$:

Ehhez elég bebizonyítani, hogy ha $S \xrightarrow[G]{*} u \neq \varepsilon$, akkor $S \xrightarrow[\bar{G}]{*} u$, ezért belátjuk a következő lemmát:

Lemma:

Ha $A \xrightarrow[G]{*} \alpha \neq \varepsilon$, akkor $A \xrightarrow[\bar{G}]{*} \alpha$, bármely $A \in N$ esetén.

Bizonyítás:

A levezetés hossza szerinti indukcióval látjuk be. Legyen i a levezetés hossza.
Nyilván $i = 0$-ra az állítás teljesül, mert $A \xrightarrow[G]{(0)} \alpha \iff A = \alpha \iff A \xrightarrow[\bar{G}]{(0)} \alpha$.

Most tegyük fel, hogy minden i-nél rövidebb levezetésre az állítás igaz. Legyen $A \xrightarrow[G]{(i)} \alpha$, $(i \geq 1)$, és tekintsük a levezetés legelső lépését:

$$A \xrightarrow[G]{(1)} Z_1 \dots Z_k \xrightarrow[G]{(i-1)} \alpha \neq \varepsilon.$$

Erre könnyen bizonyítható, hogy ilyenkor α felbontható $\alpha = \alpha_1 \alpha_2 \dots \alpha_k$ alakra úgy, hogy

$$Z_1 \xrightarrow[G]{<i} \alpha_1, \ \dots \ , Z_k \xrightarrow[G]{<i} \alpha_k.$$

Viszont α_i-k között még lehetnek üres szavak, ezeket vegyük ki. Í gy kapunk $\alpha_{i_1}, \dots, \alpha_{i_p} \neq \varepsilon$ részszavakat, melyekre igaz, hogy $\alpha = \alpha_{i_1} \dots \alpha_{i_p}$. Ezekre már alkalmazható az indukciós feltétel, vagyis

$$Z_{i_1} \xrightarrow[\bar{G}]{*} \alpha_{i_1}, \ \dots \ , Z_{i_k} \xrightarrow[\bar{G}]{*} \alpha_{i_k}.$$

Ebből következik, hogy $Z_{i_1} Z_{i_2} \dots Z_{i_p} \xrightarrow[\bar{G}]{*} \alpha_{i_1} \alpha_{i_2} \dots \alpha_{i_p}$. Azt állítjuk, hogy ekkor $A \longrightarrow Z_{i_1} Z_{i_2} \dots Z_{i_p} \in \bar{\mathcal{P}}$, ugyanis láttuk, hogy $A \longrightarrow Z_1 Z_2 \dots Z_k \in \mathcal{P}$, viszont $Z_{i_1} Z_{i_2} \dots Z_{i_p}$-t $Z_1 Z_2 \dots Z_k$-ból néhány H-beli elem elhagyásával kaptuk.

$\triangle$

Ha ε nem eleme $L(G)$-nek, akkor készen vagyunk, $L(G) = L(G')$. Ha viszont $\varepsilon \in L(G)$, akkor természetesen gondoskodni kell ε levezethetőségéről. Ennek megoldására vegyünk fel egy S' új kezdőszimbólumot, és vegyük fel a következő két szabályt:

$$S' \longrightarrow S \ \mid \ \varepsilon.$$

Így már tényleg $L(G) = L(\bar{G})$.

$i = 3$:

Itt azt kell belátnunk, hogy tetszőleges $G \in \mathcal{G}_{\text{kit3}}$ esetén van olyan $G' \in \mathcal{G}_3$, melyre $L(G) = L(G')$. Ehhez a következő három transzformációt kell elvégezni:

a) ε-mentesítés:

— Ez a kiterjesztett 2-es nyelvtanok ε-mentesítő eljárása. Ha ezt kiterjesztett 3-as nyelvtanra alkalmazzuk, az eredmény is kiterjesztett 3-as típusú lesz.

b) Lineárislánc-mentesítés:

— Ezt is mentesítő algoritmussal végezzük. (Kiterjesztett 2-es típusú nyelvtanra is alkalmazható algoritmust adunk.)

c) Hosszredukció.

Lineárislánc-mentesítés:

Lineáris láncnak nevezzük az $A \longrightarrow B (A, B \in N)$ alakú szabályokat. Ezeket kell szimulálni, hogy ne legyenek fölösleges lépések a levezetés során.

Tegyük fel, hogy G-ben a következő levezetést tudjuk elvégezni:

$$\alpha_1 A \alpha_2 \underset{G}{\longrightarrow} \alpha_1 C_1 \alpha_2 \underset{G}{\longrightarrow} \alpha_1 C_2 \alpha_2 \underset{G}{\longrightarrow} \ldots \underset{G}{\longrightarrow} \alpha_1 B \alpha_2 \underset{G}{\longrightarrow} \alpha_1 q \alpha_2, \quad \text{ahol } q \notin N, \text{és} A, B \in N.$$

Ezt a levezetést tudjuk szimulálni az $A \underset{G'}{\longrightarrow} q$ új szabállyal a következő módon:

$$\alpha_1 A \alpha_2 \underset{G'}{\longrightarrow} \alpha_1 q \alpha_2$$

A megvalósításhoz meg kell határozni ezeket az új szabályokat. Legyen $A \in N$ esetén $H(A) := \{B \in N \mid A \overset{*}{\underset{G}{\longrightarrow}} B\}$. E halmazt a következő módon határozhatjuk meg:

$$H_0(A) := \{A\}$$

$$H_{i+1}(A) := H_i(A) \cup \{B \mid \exists C \in H_i(A) \wedge C \underset{G}{\longrightarrow} B\}$$

Ezek a halmazok is egy i_0 indextől kezdve azonosak lesznek. Tegyük fel, hogy $H_j(A) = H_{j+1}(A)$. Ekkor a definíció alapján:

$$H_{j+2}(A) = H_{j+1}(A) \cup \{B \mid \exists C \in H_{j+1}(A) \wedge C \underset{G}{\longrightarrow} B\} = H_j(A) \cup \{B \mid \exists C \in H_j(A) \wedge C \underset{G}{\longrightarrow} B\} = H_{j+1}.$$

Ha $G = \langle T, N, \mathcal{P}, S \rangle$, akkor $G' = \langle T, N, \mathcal{P}', S \rangle$ lesz az új nyelvtan, ahol

$$\mathcal{P}' = \{A \longrightarrow q \mid q \notin N \wedge \exists B \in H(A) : B \longrightarrow q \in \mathcal{P}\}.$$

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

Ha G kiterjesztett 3-as nyelvtan volt, akkor az eredmény is kiterjesztett 3-as típusú lesz, mert a jobb oldal nem változott. Ezenkívül ha G ε-mentes nyelvtan volt, akkor G' is az marad.

Hosszredukció:

Itt a nem megfelelő szabályok a következőképpen írhatók föl:

$$A \longrightarrow uB,$$

$$A \longrightarrow u,$$

ahol $l(u) \geq 2$. Legyen ekkor a „rossz” szabály a következő alakú: $A \longrightarrow t_1 t_2 \ldots t_k B$, ahol $k \geq 2$, és $t_i \in T$. Ezt könnyen szimulálhatjuk a következő kiterjesztett 3-as típusú szabályrendszerrel, ahol $Z_1, Z_2, \ldots, Z_k$ új nyelvtani jelek:

$$A \longrightarrow t_1 Z_1,$$

$$Z_1 \longrightarrow t_2 Z_2,$$

$$\vdots$$

$$Z_{k-1} \longrightarrow t_k B.$$

Az $A \longrightarrow u$ szabály esetén ugyanígy járunk el.

Ezzel bebizonyítottuk a kiterjesztési tételt.

△

6. Normálformák

A normálformák további megszorításokat jelentenek az egyes nyelvosztályokra. A továbbiakban az 1, 2, 3-as osztályokra adunk újabb megszorításokat.

$i = 1$:

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ egy nyelvtan. A G nyelvtan Kuroda-féle normálformájú, ha szabályai a következő alakúak:

(i) $S \longrightarrow \varepsilon$, ilyenkor S kezdőszimbólum, és szabály jobb oldalán nem szerepelhet,
(ii) $A \longrightarrow t$,
(iii) $A \longrightarrow B$,
(iv) $A \longrightarrow BC$,
(v) $AB \longrightarrow AC$,
(vi) $BA \longrightarrow CA$, ahol $A, B, C \in N$, és $t \in T$.

Tétel:

Minden $L \in \mathcal{L}_1$ nyelv esetén létezik olyan G nyelvtan, amely Kuroda-normálformájú, és $L(G) = L$.

Bizonyítás:

Elegendő belátni, hogy tetszőleges G kiterjesztett 1-es nyelvtanhoz létezik olyan G' Kuroda-normálformájú nyelvtan, melyre $L(G) = L(G')$. Legyen $G = \langle T, N, \mathcal{P}, S \rangle$. Feltehető, hogy G-

ben terminális jel csak $A \longrightarrow a$ alakú szabályban fordul elő, ahol $A \in N$, és $a \in T$.

A normálformának nem megfelelő szabályokat szimuláljuk jó szabályokkal. Hogyan néznek ki ezek a szabályok? Csak nyelvtani jelek vannak benne, és ha a bal oldal hossza 1, akkor így:

$$A \longrightarrow X_1 X_2 \ldots X_k \ (k \geq 3).$$

Vezessük be a $Z_1, \ldots, Z_{k-2}$ új nyelvtani jeleket, és következő szabályokat:

$$A \longrightarrow X_1 Z_1,$$

$$Z_1 \longrightarrow X_2 Z_2,$$

$$\vdots$$

$$Z_{k-2} \longrightarrow X_{k-1} X_k.$$

Ha a bal oldal hossza legalább 2, akkor a „rossz" szabály a következőképpen írható le:

$$X_1 X_2 \ldots X_m \longrightarrow Y_1 Y_2 \ldots Y_n \ (m \geq 2, n \geq m)$$

azaz hosszúságot nem csökkentő szabály. Szimulációja a $Z_1, Z_2, \ldots, Z_n$ új nyelvtani jelek bevezetésével:

$$X_1 \longrightarrow Z_1,$$

$$Z_1 X_2 \longrightarrow Y_1 Z_2,$$

$$Z_2 X_3 \longrightarrow Y_2 Z_3,$$

$$\vdots$$

$$Z_{m-1} X_m \longrightarrow Y_{m-1} Z_m.$$

Továbbá ha $n = m$, akkor

$$Z_m \longrightarrow Y_m,$$

egyébként $(n > m)$:

$$Z_m \longrightarrow Y_m Z_{m+1},$$

$$\vdots$$

$$Z_{n-1} \longrightarrow Y_{n-1} Y_n.$$

De ezek közül még a fentiek nem Kuroda-normálformájúak. Ezek a szabályok $AB \longrightarrow CD$ alakban vannak megadva. Megoldásként be kell vezetni egy új nyelvtani jelet (W), és ennek segítségével lehet ezt a „rossz" szabályt szimulálni:

$$AB \longrightarrow AW,$$

$AW \longrightarrow CW,$

$CW \longrightarrow CD.$

$\triangle$

$i = 2$:

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ egy nyelvtan. A G nyelvtan Chomsky-féle normálformájú, ha szabályai a következő alakúak:

(i) $S \longrightarrow \varepsilon$, és ekkor S kezdőszimbólum, és szabály jobb oldalán nem szerepelhet,

(ii) $A \longrightarrow t, \quad t \in T, A \in N,$

(iii) $A \longrightarrow BC$, ahol $A, B, C \in N$.

Tétel:

Minden $L \in \mathcal{L}_2$ esetén van olyan G Chomsky-normálformájú nyelvtan, amelyre $L(G) = L$.

Bizonyítás:

Először azt bizonyítjuk be, hogy $G \in \mathcal{G}_{\text{kit2}}$ esetén van olyan G' Chomsky-normálformájú nyelvtan, melyre $L(G) = L(G')$. A konstrukció lépései:

a) ε-mentesítés,

b) lineárislánc-mentesítés,

c) álterminálisok bevezetése,

d) hosszredukció.

Az ε-, és a lineárislánc-mentesítés ugyanúgy hajtható végre, mint Kuroda-normálforma esetében. Az álterminálisok bevezetésének célja, hogy az $A \longrightarrow q \in \mathcal{P}$ $\big(l(q) \geq 2\big)$ szabályokban a terminális jeleket álterminális (nyelvtani) jelekre cseréljük. Ezért a $\mathcal{P}$' szabályrendszerhez hozzá kell venni az $X_t \longrightarrow t$ szabályokat. Így ezen lépés után a következő szabályformák lesznek:

$S \longrightarrow \varepsilon$, ahol S a kezdőszimbólum,

$A \longrightarrow t$, ahol $A \in N$, és $t \in T$,

$A \longrightarrow$Q, ahol Q$\in N^*$.

A hosszredukció $A \longrightarrow Q$ alakú szabályra $l(Q) \geq 3$ esetén ugyanúgy végezhető el, mint a Kuroda-normálforma esetében.

$\triangle$

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ egy nyelvtan. A G nyelvtan Greibach-féle normálformájú, ha szabályai a következő alakúak:

$S \longrightarrow \varepsilon$, és ekkor S kezdőszimbólum,

$A \longrightarrow aQ$, ahol $A \in N$, $a \in T$, és $Q \in N^*$.

Definíció:

Egy nyelvtani jel rekurzív (önbeágyazott), ha van olyan legalább 1 hosszú levezetés, amelynek mindkét végén előfordul, azaz $A \xrightarrow[G]{(i)} \alpha_1 A \alpha_2$ és $i > 0$ esetén A rekurzív.

Definíció:

Egy $A \in N$ nyelvtani jel balrekurzív, ha rekurzív, és $A \xrightarrow[G]{(i)} A\alpha_2$ valamely $i > 0$ és $\alpha_2 \in (T \cup N)^*$ esetén.

Definíció:

Egy $A \in N$ nyelvtani jel jobbrekurzív, ha rekurzív, és $A \xrightarrow[G]{(i)} \alpha_1 A$ valamely $i > 0$ és $\alpha_1 \in (T \cup N)^*$ esetén.

Vegyük észre, hogy ha egy nyelvtani jel nem balrekurzív és nem jobbrekurzív, akkor attól még lehet rekurzív.

Definíció:

G nyelvtan rekurzív, ha létezik rekurzív nyelvtani jele.

A Greibach normálforma lényege, hogy kiküszöböli a balrekurziót a 2-es típusú nyelvtanokból. Kérdés: Csökken-e a második típusú nyelvtanok hatóereje, ha nem csak a bal, hanem a teljes rekurziót kizárjuk? Ezt már nem lehet megtenni a következő tétel miatt.

Tétel:

Ha a G 2-es típusú nyelvtanban nincs rekurzív nyelvtani jel, akkor $L(G)$ véges és így ekkor 3-as típusú.

Bizonyítás:

A nyelvtani jelek számára vonatkozó indukcióval. A bizonyítást az olvasóra bízzuk.

$\triangle$

$i = 3$:

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ egy nyelvtan. A G nyelvtan 3-as normálformájú, ha szabályai a következő alakúak:

$$A \longrightarrow \varepsilon,$$

$$A \longrightarrow aB.$$

Kiindulás egy tetszőleges 3-as típusú G nyelvtan. Ebben a rossz alakú szabályok $A \longrightarrow a$ alakúak. Könnyen látható, hogy ezt egy új nyelvtani jel és a következő szabályok segítségével szimuláljuk:

$$A \longrightarrow aF,$$

$$F \longrightarrow \varepsilon.$$

6.1. 2-es típusú nyelvtanok redukálása

A redukálás a felesleges nyelvtani jelek, illetve szabályok elhagyását jelenti. Milyenek ezek a felesleges nyelvtani jelek? Az első csoportot az olyan nyelvtani jelek képezik, melyek a levezetés során zsákutcába visznek, azaz belőlük nem vezethető le T^*-beli szó. A második csoportba pedig a nem elérhető nyelvtani jelek tartoznak.

Zsákutcák megkeresése:

Legyen $H := \{A \in N \mid \exists u \in T^* : A \xrightarrow[G]{*} u\}$. Ezen H halmaz konstrukciós lépései:

$$H_1 := \{A \in N \mid \exists q \in T^* : A \longrightarrow q \in \mathcal{P}\}.$$

$$H_{i+1} := H_i \cup \{A \mid \exists q \in (H_i \cup T)^* : A \longrightarrow q \in \mathcal{P}\}.$$

Nyilván igaz, hogy a H_i halmazok csak bővülhetnek, ezért egy bizonyos i_0 indextől kezdvődően mind azonosak lesznek, és ekkor $H_{i_0} = H$. Ekkor az $N \setminus H$ nyelvtani jelek jelentik a zsákutcákat, azaz a zsákutcamentes nyelvtan nyelvtani jeleit a H halmaz tartalmazza.

Nem elérhetők megkonstruálása (összefüggővé alakítás):

Itt az elérhető jeleket határozzuk meg. Legyen K halmaz az elérhető nyelvtani jelek halmaza. Konstrukciós lépései:

$$K_0 = \{S\},$$

$$K_{i+1} = K_i \cup \{A \in N \mid \exists B \in K_i : B \longrightarrow \alpha_1 \mathrm{A} \alpha_2 \in \mathcal{P}\}.$$

Nyilván itt is igaz, hogy K_i halmazok csak bővülhetnek, ezért egy bizonyos i_0 indextől kezdve mind azonosak lesznek, és ekkor $K_{i_0} = K$. A nem elérhető nyelvtani jelek halmaza az

$N \setminus K$ halmaz, azaz az összefüggő nyelvtan nyelvtani jeleit a K halmaz tartalmazza.

A zsákutcák kiszűrése és az összefüggővé alakítás után a következő nyelvtant kapjuk:

$G' = \langle T, N', \mathcal{P}', S \rangle$, ahol $N' = H \cap K$, és $\mathcal{P}' = \{A \longrightarrow \alpha \in \mathcal{P} \mid A \in N' \wedge \alpha \in (N' \cup T)^*\}$.

Már csak az a kérdés merülhet fel, hogy milyen sorrendben kell ezt a két lépést végrehajtani.

Állítás:

Csak az előbbi sorrend a megfelelő, azaz először a zsákutcákat kell kiszűrni, utána pedig a nem elérhető jeleket.

Bizonyítás:

Miért nem jó a másik sorrend? Ellenpélda:

$S \longrightarrow a \mid AB$,

$A \longrightarrow a$.

Ez összefüggő. Ha most kiszűrjük a zsákutcákat:

$S \longrightarrow a$,

$A \longrightarrow a$.

Ez pedig nem összefüggő.

$\triangle$

Állítás:

Ha a G nyelvtan zsákutcamentes, akkor összefüggővé alakítás után is zsákutcamentes lesz.

Bizonyítás:

A bizonyítást az olvasóra bízzuk.

Definíció:

A G nyelvtant rendezett nyelvtannak nevezzük, ha nyelvtani jelei megszámozhatók úgy, hogy $N = \{A_1, A_2, \ldots, A_n\}$, és ha $A_i \longrightarrow A_j q \in \mathcal{P}$, akkor $i < j,\ q \in (T \cup N)^*$.

Definíció:

A G nyelvtan kvázi Greibach-normálformájú, ha szabályai $A \longrightarrow aq$ alakúak, ahol $q \in (T \cup N)^*, a \in T$, és $A \in N$.

Tétel:

Tetszőleges $L \in \mathcal{L}_2$ nyelvtan esetén létezik olyan G Greibach-normálformájú nyelvtan, melyre $L(G) = L$.

Bizonyítás:

Elegendő bizonyítani, hogy tetszőleges $G \in \mathcal{G}_{\text{kit2}}$ nyelvtan esetén létezik olyan G' Greibach-normálformájú nyelvtan, melyre $L(G') = L(G)$. Egy $G \in \mathcal{G}_{\text{kit2}}$ Greibach-normálformájú nyelvtanná alakításának lépései:

a) ε-mentesítés,

b) lineárislánc-mentesítés,

c) redukció,

d) rendezetté alakítás,

e) kvázi Greibach-normálformájúvá alakítás, és

f) Greibach-normálformájúvá alakítás.

Az első három pont az eddig tárgyalt módszerekkel végezhetők el, a d) és e) lépéseket a következőkben adjuk meg. Az f) lépés álterminálisok bevezetésével egyszerűen megtehető.

△

Megjegyzés: Ha egy ε-mentes nyelvtan rendezett, akkor biztosan nem balrekurzív. (Ugyanis mindig nőnie kell az indexnek, tehát a Greibach-nyelvtanokat a rendezetté alakítás teszi balrekurzívmentessé.)

6.2. Elemi transzformációk

A típusú transzformáció:

Legyen G 2-es típusú nyelvtan. Tegyük fel, hogy

$s : A \longrightarrow \alpha_1 B \alpha_2 \in \mathcal{P}$, és

$B \longrightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_k$.

Ekkor ez az eljárás az s szabályt a következőkkel helyettesíti:

$A \longrightarrow \alpha_1\beta_1\alpha_2 \mid \alpha_1\beta_2\alpha_2 \mid \ldots \mid \alpha_1\beta_k\alpha_2$.

A többi szabályt változatlanul hagyjuk. Nyilván továbbra is fennáll az így kapott G' nyelvtanra, hogy $L(G) = L(G')$.

Jelölése: Atrans(s, k), ami az s szabály k-adik nyelvtani jelére végzi el a transzformációt. Ha nincs k-adik nyelvtani jel, akkor nem csinál semmit (ezt SKIP jelöli).

Megjegyzés: Ha G ε-mentes, lineárislánc-mentes és zsákutcamentes volt, akkor G' is az lesz.

B típusú transzformáció:

Legyen G 2-es típusú nyelvtan. Tegyük fel, hogy az $A \in N$ nyelvtani jele balrekurzív, azaz legyen az összes szabály A-ra:

$$A \longrightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_k \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_l,$$

ahol minden i-re β_i első jele nem A.

Hogyan néz ki egy ilyen szabály alkalmazása? Az A nyelvtani jelből a következő alakú szavak vezethetők le (mint közbülső eredmény):

$$\{\beta_1, \ldots, \beta_l\}\{\alpha_1, \ldots, \alpha_k\}^*.$$

Új nyelvtani jelet vezetünk be: Z. Az új szabályok legyenek:

$$A \longrightarrow \beta_1 \mid \ldots \mid \beta_l \mid \beta_1 Z \mid \ldots \mid \beta_l Z,$$

$$Z \longrightarrow \alpha_1 Z \mid \ldots \mid \alpha_k Z \mid \alpha_1 \mid \ldots \mid \alpha_k.$$

Ezáltal megszűnt a balrekurzió, jobbrekurzió lett belőle. Nyilván itt is teljesül az így kapott G' nyelvtanra, hogy $L(G) = L(G')$.

Jelölés: Btrans(A, Z), ahol ha az $A \in N$ nyelvtani jel nem balrekurzív, akkor maga az eljárás egy SKIP.

6.3. Rendezetté alakítás

Tegyük fel, hogy a G nyelvtan rendezetlen, és a Greibach-normálformájú nyelvtanná alakítás eddigi fázisai megvoltak. Valamint számozzuk meg az eredeti G nyelvtan jeleit: $N = \{A_1, \ldots, A_n\}$, ahol legyen A_1 a kezdőszimbólum. Legyen az algoritmus elő-, illetve utófeltétele:

$$Q : G \text{ nyelvtan } \varepsilon\text{-mentes, lineárislánc-mentes és zsákutcamentes,}$$

$$R : Q \wedge (\forall i \in [1, n] : (s : A_i \longrightarrow A_j\alpha \in \mathcal{P} \Longrightarrow i < j)).$$

Ezt nyilván egy ciklussal oldjuk meg, melynek i-edik fázisában a következő invariáns teljesül:

$$P : Q \wedge i \in [1, b] \wedge (\forall k \in [1, i] : (s : A_k \longrightarrow A_j\alpha \in \mathcal{P} \Longrightarrow k < j)).$$

Ez már $i = 1$-re teljesül, mert A_1-re, a kezdőszimbólumra nem létezik rossz szabály. Az algoritmus struktogramja:

$i := 0$	
$i < n$	
	$i := i + 1$
	$(\exists s : A_i \longrightarrow A_j\alpha \in \mathcal{P}) \wedge j < i$
	Atrans$(s, 1)$
	Btrans(A_i, Z_i)

Már csak egy dologra kell vigyáznunk. Az újonnan bevezetett Z_i nyelvtani jelekre még ellenőrizni kéne, hogy teljesül-e a rendezettség. De ezt egy egyszerű módon biztosíthatjuk. Számozzuk meg a nyelvtani jeleket a következő sorrend szerint:

$$Z_n, Z_{n-1}, \ldots, Z_1, A_1, A_2, \ldots, A_n.$$

Így már rendezett lesz az új G' nyelvtan.

6.4. Kvázi Greibach-normálformájúvá alakítás

Legyen a G nyelvtan kvázi Greibach. Vezessük be az $\mathrm{lh}(\alpha)$ jelölést az α szó legelső jelére, azaz $\mathrm{lh}(\alpha) \in N \cup T$. (Az lh az angol lefthead szó rövidítése.)

Legyen az algoritmus elő-, illetve utófeltétele:

$$Q : G \text{ } \varepsilon\text{-mentes, lineárislánc-mentes, zsákutcamentes és rendezett,}$$

$$R : Q \wedge \forall i \in [1, n] : (s : A_i \longrightarrow \alpha \in \mathcal{P} \Longrightarrow \mathrm{lh}(\alpha) \notin N).$$

Nyilván ezt ismét ciklussal oldjuk meg, melynek invariánsa:

$$P : Q \wedge i \in [1, n] \wedge \forall k \in [i, n] : (s : A_k \longrightarrow \alpha \in \mathcal{P} \Longrightarrow \mathrm{lh}(\alpha) \notin N).$$

Az algoritmus struktogramja:

$i := n$	
$i > 1$	
	$i := i - 1$
	$\exists s : A_i \longrightarrow \alpha \in \mathcal{P} \wedge \mathrm{lh}(\alpha) \in N$
	Atrans$(s, 1)$

7. Zártsági tételek

Legyen Φ n-változós nyelvi operátor. Ekkor az $L_1, \ldots, L_n$ nyelvek esetén $\Phi(L_1, \ldots, L_n)$ is egy nyelv.

Definíció:

Az $\mathcal{L}$ nyelvcsalád zárt a Φ nyelvi operátorra, ha $L_1, \ldots, L_n \in \mathcal{L}$ esetén $\Phi(L_1, \ldots, L_n) \in \mathcal{L}$.

Tétel:

Az $\mathcal{L}_i$ $(i = 0, 1, 2, 3)$ nyelvosztályok zártak az unió, konkatenáció és a lezárás műveletekre.

Bizonyítás:

Elég bizonyítani, hogy ha $G_1, G_2 \in \mathcal{G}_{\mathbf{kit}i}$, akkor léteznek olyan $G_\cup, G_{\text{konk}}, G^* \in \mathcal{G}_{\mathbf{kit}i}$ nyelvtanok, melyekre

$$L(G_\cup) = L(G_1) \cup L(G_2),$$

$$L(G_{\text{konk}}) = L(G_1)L(G_2),$$

$$L(G^*) = (L(G_1))^*.$$

Legyenek $G_1 = \langle T_1, N_1, \mathcal{P}_1, S_1 \rangle$ és $G_2 = \langle T_2, N_2, \mathcal{P}_2, S_2 \rangle$. Feltehető, hogy $N_1 \cap N_2 = \emptyset$. (Ha nem, akkor csupán átnevezzük.) Továbbá feltehető, hogy $i = 0, 1, 2$-re terminális jel csak $A \longrightarrow u$ szabályokban fordul elő. Alapkonstrukció:

$$G_\cup := \langle T_1 \cup T_2, N_1 \cup N_2 \cup \{S\}, \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{S \longrightarrow S_1 \mid S_2\}, S \rangle,$$

$$G_{\text{konk}} := \langle T_1 \cup T_2, N_1 \cup N_2 \cup \{S\}, \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{S \longrightarrow S_1 S_2\}, S \rangle,$$

$$G^* := \langle T_1, N_1 \cup \{S\}, \mathcal{P}_1 \cup \{S \longrightarrow \varepsilon \mid S_1 S\}, S \rangle.$$

Nyilván a fenti nyelvekre a nyelvtanok definíciója alapján a „$\supseteq$” tartalmazás fennáll. Bizonyítani kell, hogy a „$\subseteq$” irányú tartalmazás is teljesül, és hogy a fent definiált nyelvtanok i-típusú nyelvtanok.

1) Unió:

Bizonyítani kell, hogy $S \xrightarrow[G_\cup]{*} u \Longrightarrow S_1 \xrightarrow[G_1]{*} u \vee S_2 \xrightarrow[G_2]{*} u$. Ez nyilván teljesül, hiszen kezdésként csak két szabály közül alkalmazhatunk egyet, azaz $S \longrightarrow S_1$ vagy $S \longrightarrow S_2$ valamelyikét.

Tegyük föl, hogy az $S \longrightarrow S_1$ szabályt alkalmaztuk. Ekkor nyilván $S_1 \xrightarrow[G_1]{*} u$, mert a levezetés során csak N_1-beli nyelvtani jelek kerülnek be, és $N_1 \cap N_2 = \emptyset$ miatt csak $\mathcal{P}_1$-beli szabályok alkalmazhatók. Ugyanez az $S \longrightarrow S_2$ szabályra is igaz. Tehát unióra a „$\subseteq$” irány is teljesül.

Kérdés, hogy megmarad-e a típus? Egyetlen probléma ε levezetésénél van. Világos, hogy $i = 0, 2, 3$-ra mindenképp teljesül. $i = 1$ esetnél az alapkonstrukción módosítani kell:

$$\varepsilon \notin L(G_1) \cup L(G_2) \Longrightarrow \text{nincs módosítás},$$

$$\varepsilon \in L(G_1) \cup L(G_2) \Longrightarrow \mathcal{P}_\cup := \mathcal{P}_1 \setminus \{S_1 \longrightarrow \varepsilon\} \cup \mathcal{P}_2 \setminus \{S_2 \longrightarrow \varepsilon\} \cup \{S \longrightarrow S_1 \mid S_2 \mid \varepsilon\}.$$

2) Konkatenáció:

A „$\subseteq$” irányú tartalmazás bizonyításához be kell látni, hogy $S \xrightarrow[G_{\text{konk}}]{*} u$ esetén létezik az u szónak olyan $u = vw$ felbontása, amelyre $S_1 \xrightarrow[G_1]{*} v$, és $S_2 \xrightarrow[G_2]{*} w$ teljesül.

Tekintsünk egy tetszőleges $u \in L(G_{\text{konk}})$ szót. Erre létezik levezetés G_{konk}-ban, azaz $S \xrightarrow[G_{\text{konk}}]{*} u$. Mivel $N_1 \cap N_2 = \emptyset$, így nyilván ez a levezetés elvégezhető a következő módon is:

$$S \underset{G_{\text{konk}}}{\longrightarrow} S_1S_2 \xrightarrow[G_{\text{konk}}]{*} vS_2 \xrightarrow[G_{\text{konk}}]{*} vw,$$

ahol S_1-ből $\mathcal{P}_1$-beli szabályokkal vezettük le a v szót, S_2-ből pedig a $\mathcal{P}_2$-beli szabályokat alkalmaztuk w levezetésére. Ezekre tudjuk, hogy $vw = u$, mert az $S \xrightarrow[G_{\text{konk}}]{*} u$ levezetést részleteztük. Viszont ekkor nyilván létezik u és w szavaknak levezetése az eredeti G_1, G_2 nyelvtanokban, tehát $S_1 \xrightarrow[G_1]{*} v$ és $S_2 \xrightarrow[G_2]{*} w$. Ez viszont pontosan azt jelenti, hogy $L(G_{\text{konk}}) \subseteq L(G_1)L(G_2)$, mert $u = vw \in L(G_{\text{konk}})$, és ez megfelel a konkatenáció definíciójának.

További kérdés, hogy ugyanaz lesz-e az így kapott nyelvtan típusa?

Világos, hogy az $i = 0, 2$ esetekben biztosan, hiszen ezekben az esetekben a kiterjesztés definíciója értelmében nincs megkötés a szabályok jobb oldalán álló nyelvtani jelek maximális számára. Viszont $i = 3$ esetén csak az $A \longrightarrow u$ és az $A \longrightarrow uB (u \in T^*)$ alakú szabályok vannak megengedve, viszont az $S \longrightarrow S_1S_2$ szabály már nem, tehát szimulálnunk kell. Tegyük fel, hogy az $A \longrightarrow a \in \mathcal{P}_1$ szabállyal fejeződik be a G_1-beli levezetés. Ekkor ettől az a jeltől kell folytatnunk a második (G_2-beli) levezetést a következő szabállyal:

$$A \longrightarrow aS_2.$$

Innen már változatlan a levezetés, mint G_2-ben. Jelöljük ezt a transzformációt $\Psi(\mathcal{P}_1, S_2)$-vel. Így a módosított nyelvtan:

$$G_{\text{konk}} := \langle T_1 \cup T_2, N_1 \cup N_2, \Psi(\mathcal{P}_1, S_2) \cup \mathcal{P}_2, S_1 \rangle.$$

Ekkor már $i = 3$-ra is megtartja a típusát.

Az $i = 1$ esetben ismét az ε-levezetéses szabályokkal van probléma. Itt négy eset lehetséges. Ha ugyanis se a G_1 nyelvtanban, se a G_2 nyelvtanban nem lehet ε-ra vezetni, akkor nincs probléma, az eredeti konstrukció is 1-es típusú lesz. Ha viszont vagy a G_1, vagy a G_2 nyelvtanban levezethető ε, akkor nyilván változtatnunk kell G_{konk} szabályrendszerének definícióján az alábbi módon:

$$\mathcal{P}_{\text{konk}} := \begin{cases} (\mathcal{P}_1 \cup \mathcal{P}_2 \cup \{S \longrightarrow \varepsilon \mid S_1S_2\}) \setminus \{S_1 \longrightarrow \varepsilon\}, & \text{ha } \varepsilon \in L(G_1), \varepsilon \notin L(G_2), \\ (\mathcal{P}_1 \cup \mathcal{P}_2 \cup \{S \longrightarrow \varepsilon \mid S_1S_2\}) \setminus \{S_2 \longrightarrow \varepsilon\}, & \text{ha } \varepsilon \notin L(G_1), \varepsilon \in L(G_2), \\ (\mathcal{P}_1 \cup \mathcal{P}_2 \cup \{S \longrightarrow \varepsilon \mid S_1S_2\}) \setminus \{S_1 \longrightarrow \varepsilon, S_2 \longrightarrow \varepsilon\}, & \text{ha } \varepsilon \in L(G_1), \varepsilon \in L(G_2). \end{cases}$$

Így G_{konk} már nyilván 1-es típusú lesz.

3) Lezárás:

Bizonyítani kell, hogy $S \xrightarrow[G^*]{*} u$ esetén létezik olyan k konstans, hogy $u = u_1 u_2 \dots u_k$, ahol minden i-re $S_1 \xrightarrow[G_1]{*} u_i$.

Ha G 2-es típusú nyelvtan, akkor a konstrukciónk miatt a levezetés a következőképpen néz ki:

$$S \xrightarrow[G^*]{} S_1^{(1)} \dots S_1^{(k)} S \xrightarrow[G^*]{*} \alpha_1 \alpha_2 \dots \alpha_k \xrightarrow[G^*]{*} u_1 \dots u_k = u.$$

Nyilván α_i-k levezethetők S_1-ből. Ugyanis egy korábbi állítás alapján a környezetfüggetlen nyelvtanok esetén u felbontható $u = u_1 \dots u_k$ alakra, melyre $S_1 \xrightarrow[G^*]{} u_i$. Tehát $i = 2$-re beláttuk az állítást.

Ugyanakkor az $i = 0, 1, 3$ esetekben ugyanez már nem mondható el, az eredeti konstrukción változtatni kell. Miért nem lesz jó az előbbi levezetés? Mert a levezetés során a határokon levő jelekre is alkalmazhatnánk szabályokat. Ugyanis tegyük fel, hogy a levezetés során a következő szót kapjuk:

$$S \xrightarrow[G^*]{*} \alpha S_1 \beta S_1 \gamma.$$

Világos, hogy mivel $i \neq 2$, ezért ezek nem környezetfüggetlen nyelvtanok, s így akár olyan szabály is alkalmazható lenne, mely α és β néhány jelét is felhasználná, ami viszont nyilván az eredeti nyelvtanban még csak föl se merült.

Ezért módosítsuk az eredeti konstrukciót a következő módon:

$$\mathcal{P}^* := \mathcal{P}_1 \cup \{S \longrightarrow \varepsilon \mid S_1 S'\} \cup \{tS' \longrightarrow tS \mid \forall t \in T_1\}.$$

Itt természetesen elegendő a terminális jelekre megadni a szabályokat, hiszen a tétel elején feltettük, hogy terminális jel $i = 0, 1, 2$ esetén csak az $A \longrightarrow u$ szabályokban fordul elő.

Ekkor már $i = 0, 1, 3$ esetén is benne lesz az eredeti nyelv lezártja az általunk definiált nyelvtan generált nyelvében.

△

Definíció:

Az előbbi tétel alapján az únió, konkatenáció és a lezárás műveleteket reguláris műveleteknek nevezzük.

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle \in \mathcal{G}_1$. Ekkor $u \in T^*$ esetén legyen K a legfeljebb $l(u)$ hosszúságú szavak száma. Jelölésben: $K = K(G, u) = \left|(T \cup N)^{\leq l(u)}\right|$.

Állítás:

Minden $G \in \mathcal{G}_{\text{kit1}}$ nyelvtan bármely $u \in L(G)$ szavának van legfeljebb $K(G,u)$ hosszú levezetése.

Bizonyítás:

Két esetet különböztetünk meg.

$u = \varepsilon$:

Ekkor $K(G,\varepsilon) = 1$, mert csak egy nulla hosszú szó létezik, és ez illeszkedik az egyetlen $S \longrightarrow \varepsilon$ levezetés 1 hosszához.

$u \neq \varepsilon$:

Ha $u \in L(G)$, akkor létezik $S = \alpha_0 \underset{G}{\longrightarrow} \alpha_1 \underset{G}{\longrightarrow} \ldots \underset{G}{\longrightarrow} \alpha_s = u$ levezetés. Ekkor természetesen van olyan $S = \beta_0 \underset{G}{\longrightarrow} \ldots \underset{G}{\longrightarrow} \beta_{s'}$ levezetés is, ahol mindegyik β_i különböző. Ugyanis, ha $i < j$ esetén $\alpha_i = \alpha_j$, akkor az i és j közötti levezetéseket egyszerűen elhagyjuk. Tehát $s' \leq s$.

De a hosszúság nem csökkentése miatt $l(\beta_0) \leq l(\beta_1) \leq \ldots \leq l(\beta_{s'}) = l(u)$. Ebből pedig már következik, hogy $\beta_0, \beta_1, \ldots, \beta_{s'} \in \left(T \cup N\right)^{\leq l(u)}$, és mivel ezek mind különbözők, ezért biztosan igaz, hogy $s' \leq \left|\left(T \cup N\right)^{\leq l(u)}\right| = K(G,u)$.

$\triangle$

8. Algoritmussal definiált nyelvek

Legyen $\mathcal{L}_{\text{RekFel}}$ a rekurzívan felsorolható nyelvek osztálya, $\mathcal{L}_{\text{ParcRek}}$ a parciálisan rekurzív nyelvek osztálya, $\mathcal{L}_{\text{Rek}}$ pedig a rekurzív nyelvek osztálya.

Tétel:

$$\mathcal{L}_0 \subseteq \mathcal{L}_{\text{RekFel}} \ , \qquad \mathcal{L}_0 \subseteq \mathcal{L}_{\text{ParcRek}} \ , \qquad \mathcal{L}_1 \subseteq \mathcal{L}_{\text{Rek}} \ .$$

Bizonyítás:

Megfelelő algoritmusok konstrukciójával bizonyítjuk. Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ tetszőleges 0-ás típusú nyelvtan. Konstruáljuk meg a G-beli S-ből kiinduló összes levezetés t_G fáját.

A t_G fa definíciója: pontjai $(T \cup N)^*$ elemeivel, élei $\mathcal{N} \times \mathcal{N}$ elemeivel vannak címkézve, valamint gyökere S.

A fa megkonstruálása: egy α pontból kiinduló, β végpontú és (i,j)-vel címkézett él akkor és csak akkor lesz eleme a t_G fának, ha $\alpha \underset{G}{\longrightarrow} \beta$, a levezetés során G-nek a j-edik szabályát használjuk és a helyettesítés első jele α-nak az i-edik pozícióján van. Tehát előzőleg számozzuk meg a szabályokat.

Nyilván ez a fa G összes levezetését tartalmazni fogja. Például legyen a G nyelv-

tan szabályrendszere a következő:

1) S$\longrightarrow$SS,

2) S$\longrightarrow a$Sb,

3) S$\longrightarrow \varepsilon$.

Ekkor ennek a levezetésfája a következőképpen néz ki:

Kérdés, hogy mikor vezethető le egy adott u szó. Nyilván akkor, ha a t_G fának van olyan pontja, ami u-val van címkézve $\big(\text{cimke}(c) = u\big)$. Tehát a felsoroláshoz csak szintfolytonosan kell bejárnunk a fát. Ekkor minden pontot érinteni fogunk, mivel minden pontnak csak véges számú leágazása lehet.

Tehát a felsoroló algoritmus csúcsfeldolgozó eljárása:

cimke$(c) \in T^*$	különben
kiír(cimke(c))	SKIP

Parciális eldöntő algoritmus csúcsfeldolgozó eljárása, ahol u az input szó:

cimke$(c) = u$	különben
Exit(+, bejárás)	SKIP

Eldöntő algoritmus csúcsfeldolgozó eljárása, ha $G \in \mathcal{G}_{\text{kit1}}$, azaz hosszúságot nem csökkentő (ekkor az előző fejezet utolsó tétele szerint ha u szó levezethető, akkor legfeljebb a $K(G,u)$ szintig meg kell találni):

szintszám$(c) > K(G,u)$	szintszám$(c) \le K(G,u)$ $\wedge$ cimke$(c) = u$	különben
Exit(−, bejárás)	Exit(+, bejárás)	SKIP

Tehát minden esetben találtunk algoritmust. Az utolsó két eljárás esetén az „Exit(+, bejárás)” és az „Exit(−, bejárás)” az jelentik, hogy a fabejárást befejezzük, és pozit!v, illetve negat!v választ adunk vissza. A fa bejárása pedig a szokásos szintfolytonos fabejárás algoritmusa.

△

Ha a Church-tézist elfogadjuk, akkor $\mathcal{L}_0$=$\mathcal{L}_{\text{ParcRek}}$ =$\mathcal{L}_{\text{RekFel}}$.

9. Matematikai gépekkel jellemezhető nyelvek

Definíció:

n-verem alatt a következő $2n+5$-öst értjük:

$$\mathcal{V} = \langle A, T, \Sigma_1 \ , \ldots, \Sigma_n \ , \delta, a_0, \sigma_0^1 \ , \ldots, \sigma_0^n \ , F \rangle,$$

ahol

A az állapotok halmaza (ez legyen véges halmaz),

T egy ábécé,

Σ_i az i-edik verem ábécéje,

δ az állapotátmeneti függvény,

$a_0 \in A$ kezdőállapot,

σ_0^i a vermek kezdőszimbólumai,

$F \subseteq A$ a végállapotok halmaza.

Ekkor $\delta : A \times (T \cup \{\varepsilon\}) \times \Sigma_1 \ \times \ldots \times \Sigma_n \ \longrightarrow \ 2^{A \times \Sigma_1^* \times \ldots \times \Sigma_n^*}$ (nemdeterminisztikus modell).

Legyen adott $\mathcal{V}$ automata, aminek ábécéjére vezessük be a $T(\mathcal{V})$ jelölést.

Definíció:

Konfigurációnak nevezzük azoknak az adatoknak az összességét, amiktől a gép elkövetkezendő működése függ.

Nyilván ez már az eddig elolvasottaktól nem függ, ezért a következő $n+2$-est nevezzük konfigurációnak:

$$[a, u, \alpha_1, \ldots, \alpha_n],$$

ahol:

a az aktuális állapot,

u az input szó (elolvasatlan),

α_i az i-edik verem tartalma.

Definíció:

Közvetlen konfigurációátmenetről beszélünk, ha $\mathcal{V}$ egy lépésben eljut a megadott konfigurációba, azaz $[a, u, \alpha_1, \ldots, \alpha_n] \underset{\mathcal{A}}{\vdash} [b, v, \beta_1, \ldots, \beta_n]$ akkor és csak akkor, ha van olyan $t \in T \cup \{\varepsilon\}$, hogy $u = tv$, és minden $i \in [1, n]$ esetén van olyan $\sigma_i \in \Sigma_i$ és $\gamma_i, \tau_i \in \sigma_i^*$, amelyekre $\alpha_i = \sigma_i\gamma_i, \beta_i = \tau_i\gamma_i$, valamint $(b, \tau_1, \ldots, \tau_n) \in \delta(a, t, \sigma_1, \ldots, \sigma_n)$.

Definíció:

A közvetett konfigurációátmenet a közvetlen átmenet reflexív, tranzitív lezártja. Jelölése: $\overset{*}{\underset{\mathcal{V}}{\vdash}}$

Definíció:

Az u szóhoz tartozó kezdőkonfiguráció: $[a_0, u, \sigma_0^1, \ldots, \sigma_0^n]$.

Definíció:

Egy konfiguráció terminális konfiguráció, ha nincs rákövetkezője.

Definíció:

Elfogadó egy $[f, \varepsilon, \beta_1, \ldots, \beta_n]$ konfiguráció, ha $f \in F$.

A $\mathcal{V}$ n-verem elfogadja az u szót, ha létezik átmenet az u szóhoz tartozó kezdőkonfigurációból elfogadott konfigurációba. A $\mathcal{V}$ által elfogadott nyelv:

$$L(\mathcal{V}) = \left\{ u \in T^* \mid [a_0, u, \sigma_0^1, \ldots, \sigma_0^n] \overset{*}{\underset{\mathcal{V}}{\vdash}} [f, \varepsilon, \beta_1, \ldots, \beta_n] \text{ valamely } f \in F\text{-re} \right\}.$$

Definíció:

Egy adott $\mathcal{V}$ n-verem determinisztikus, ha minden konfigurációban legfeljebb egy darab rákövetkezője van. Ez két feltétel teljesülése esetén áll fenn:

1) Minden $(a, t, \sigma_1, \ldots, \sigma_n) \in D_\delta$ esetén $\mid \delta(a, t, \sigma_1, \ldots, \sigma_n) \mid \leq 1$.
2) $\delta(a, \varepsilon, \sigma_1, \ldots, \sigma_n) \neq \emptyset$ esetén minden $t \in T$-re $\mid \delta(a, t, \sigma_1, \ldots, \sigma_n) \mid = 0$.

A második feltétel szerint ε-os működésnél nem lehet valódi működése.

Jelölések:

$\mathcal{L}_{n\mathcal{V}}$ jelentse az n-vermek által elfogadott nyelvek osztályát,

$\mathcal{L}_{Dn\mathcal{V}}$ jelentse a determinisztikus n-vermek által elfogadott nyelvek osztályát.

10. 0-vermek

Milyen egy 0-verem? Általában állapotátmeneti diagrammal ábrázoljuk, melynek pontjai állapotokkal vannak címkézve, élei pedig $T \cup \{\varepsilon\}$-beli elemekkel. Ez így egy irányított gráf, ahol egy adott $a \in A$-val címkézett pontból a $b \in B$ pontba pontosan akkor vezet egy $t \in T$-vel címkézett él, ha $b \in \delta(a,t)$.

0-vermek osztályozása:

— Nincs megkötés (véges, ε-átmenetes, nemdeterminisztikus automata), elfogadott nyelvét jelölje $\mathcal{L}_{\varepsilon\mathrm{NDA}}$.

— ε-átmenet megtiltása (véges, nemdeterminisztikus automata), jelölés: $\mathcal{L}_{\mathrm{NDA}}$.

— Nemdeterminisztikusság és ε-átmenet megtiltása (véges, parciálisan determinisztikus automata), jelölés: $\mathcal{L}_{\mathrm{PDA}}$.

— Nemdeterminisztikusság és ε-átmenet megtiltása, azaz bármely $a \in A$ és $t \in T$ esetén $\mid \delta(a,t) \mid = 1$, és $\mid \delta(a,\varepsilon) \mid = 0$ (véges, determinisztikus automata), jelölés: $\mathcal{L}_{\mathrm{DA}}$.

Ezen osztályozás alapján nyilvánvaló, hogy $\mathcal{L}_{0\nu} = \mathcal{L}_{\varepsilon\mathrm{NDA}} \supseteq \mathcal{L}_{\mathrm{NDA}} \supseteq \mathcal{L}_{\mathrm{PDA}} \supseteq \mathcal{L}_{\mathrm{DA}}$.

Állítás:

A $\overset{*}{\underset{\nu}{\dashv}}$ reláció független a szó elolvasatlan részétől.

Bizonyítás:

Tegyük fel, hogy $[a, uw, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [b, w, \beta_1, \ldots, \beta_n]$. Ez pontosan azt jelenti, hogy w-nek nincs szerepe az automata működésében, azaz $[a, u, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [b, \varepsilon, \beta_1, \ldots, \beta_n]$. Ennek teljes indukcióval való belátását az olvasóra bízzuk.

$\triangle$

Állítás:

A konfigurációátmenetek folytathatók, azaz $[a, u, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [b, \varepsilon, \beta_1, \ldots, \beta_n]$ és $[b, v, \beta_1, \ldots, \beta_n] \overset{*}{\underset{\nu}{\dashv}} [c, \varepsilon, \gamma_1, \ldots, \gamma$ esetén $[a, uv, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [c, \varepsilon, \gamma_1, \ldots, \gamma_n]$.

Bizonyítás:

Tegyük fel, hogy

$$[a, u, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [b, \varepsilon, \beta_1, \ldots, \beta_n] \text{ és}$$

$$[b, v, \beta_1, \ldots, \beta_n] \overset{*}{\underset{\nu}{\dashv}} [c, \varepsilon, \gamma_1, \ldots, \gamma_n].$$

Ekkor az előző állítás szerint $[a, uv, \alpha_1, \ldots, \alpha_n] \overset{*}{\underset{\nu}{\dashv}} [c, v, \gamma_1, \ldots, \gamma_n]$, és a $\overset{*}{\underset{\nu}{\dashv}}$ reláció tranzitivitásából adódik az állítás.

$\triangle$

Tétel:

$\mathcal{L}_{0\mathcal{V}} = \mathcal{L}_3$.

Bizonyítás:

A kétirányú tartalmazást látjuk be.

„$\supseteq$":

Minden 3-as típusú nyelvhez létezik 3-as normálformájú nyelvtan, így tehát elég belátni, hogy tetszőleges, 3-as normálformában adott nyelvtanhoz létezik $\mathcal{V}$ 0-verem, hogy $L(\mathcal{V}) = L(G)$.

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$. Ennek a nyelvtannak a szabályai legyenek a következő alakúak:

$A \longrightarrow tB$

$A \longrightarrow \varepsilon$, ahol $A, B \in N$ és $t \in T$.

Legyen az s levezetés a következő:

$$A = B_0 \underset{G}{\longrightarrow} t_1 B_1 \underset{G}{\longrightarrow} t_1 t_2 B_2 \underset{G}{\longrightarrow} \ldots \underset{G}{\longrightarrow} t_1 t_2 \ldots t_k B_k \ (B_k := C).$$

A kérdés az, hogyan tudjuk ezt automatával szimulálni? A t_i jel generálásának feleljen meg a t_i jel olvasása. Eredmény: B_i-k lesznek a nyelvtani jelek. Ha $A \longrightarrow \varepsilon$ szabályt alkalmazunk, akkor A legyen végállapot. Ekkor a következő konfigurációátmeneteket várjuk:

$$[B_0, t_1 \ldots t_k] \underset{\mathcal{A}}{\vdash} [B_1, t_2 \ldots t_k] \underset{\mathcal{A}}{\vdash} \ \ldots \ \underset{\mathcal{A}}{\vdash} [B_k, \varepsilon].$$

Konstrukció: Legyen $\mathcal{V} = \langle N, T, \delta, S, F \rangle$, ahol

$C \in \delta(B, t) \iff B \longrightarrow tC \in \mathcal{P}$, ahol $t \in T$, és

$B \in F \iff B \longrightarrow \varepsilon \in \mathcal{P}$.

Könnyen belátható az u szó hosszára vonatkozó indukció alapján, hogy

$$[A, u] \overset{*}{\underset{\mathcal{V}}{\vdash}} [B, \varepsilon] \iff A \overset{*}{\underset{G}{\longrightarrow}} uB.$$

Ekkor már valóban $L(\mathcal{V}) = L(G)$, ugyanis $u \in L(\mathcal{V}) \iff \exists B \in F : ([S, u] \overset{*}{\underset{\mathcal{V}}{\vdash}} [B, \varepsilon]) \iff \exists B : (B \underset{G}{\longrightarrow} \varepsilon, S \overset{*}{\underset{G}{\longrightarrow}} uB) \iff S \overset{*}{\underset{G}{\longrightarrow}} u \iff u \in L(G)$.

Milyen az így kapott automata? Ez egy ε-átmenet nélküli automata, így most egy kicsit többet láttunk be, azaz $\mathcal{L}_3 \subseteq \mathcal{L}_{\text{NDA}}$.

„$\subseteq$":

Itt elég belátni, hogy tetszőleges 0-veremhez létezik olyan $G \in \mathcal{G}_{\text{kit3}}$, hogy $L(G) = L(\mathcal{V})$. Tulajdonképpen ugyanazt csináljuk, mint az ellenkező irány esetén.

Legyen $\mathcal{V} = \langle A, T, \delta, a_0, F \rangle$, és legyen $G = \langle T, A, \mathcal{P}, a_0 \rangle$ a hozzá konstruálandó nyelvtan. Milyenek legyenek $\mathcal{P}$ szabályai?

$a \longrightarrow tb \in \mathcal{P} \iff b \in \delta(a,t)$, ahol $t \in T \cup \{\varepsilon\}$,

$f \longrightarrow \varepsilon \in \mathcal{P} \iff f \in F$.

Ez kiterjesztett 3-as nyelvtan lesz, mert t lehet ε is, ugyanis $\mathcal{V}$-ben nem volt megtiltva az ε-átmenet. Ekkor a levezetés hossza szerinti indukcióval belátható, hogy

$$a \xrightarrow[G]{*} ub \iff [a,u] \overset{*}{\underset{\mathcal{V}}{\dashv}} [b,\varepsilon].$$

Ebből már következik, hogy $L(G) = L(\mathcal{V})$.

△

10.1. Determinisztikus 0-vermek ($\mathcal{L}_{D0\mathcal{V}}$)

Tétel:

$\mathcal{L}_{D0\mathcal{V}} = \mathcal{L}_{0\mathcal{V}}$.

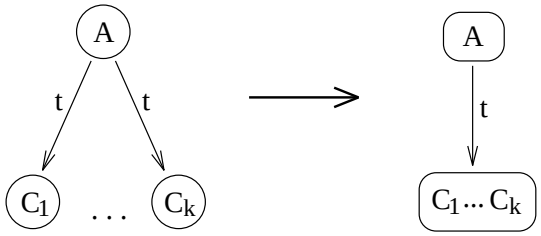
Bizonyítás:

Eddig már beláttuk, hogy $\mathcal{L}_{0\mathcal{V}} = \mathcal{L}_3 \subseteq \mathcal{L}_{\mathrm{NDA}}$. Másrészt a definícióból tudjuk, hogy $\mathcal{L}_{\mathrm{DA}} \subseteq \mathcal{L}_{D0\mathcal{V}} \subseteq \mathcal{L}_{0\mathcal{V}}$. Tehát elég bebizonyítani, hogy $\mathcal{L}_{\mathrm{NDA}} \subseteq \mathcal{L}_{\mathrm{DA}}$.

Determinisztikussá tétel algoritmusa.

Be kell látni, hogy tetszőleges $\mathcal{A}$ véges, nemdeterminisztikus automatához van olyan $\mathcal{A}'$ véges, determinisztikus automata, amelyre az elfogadott nyelvek azonosak, azaz $L(\mathcal{A}) = L(\mathcal{A}')$.

Tekintsünk egy $\mathcal{A}$-beli átmenetet. Legyen a egy csúcs, és egy adott $t \in T$ jel hatására az összes átmenet $c_1, \ldots, c_k$. Ezek mind párhuzamos működésűek. Vegyük egy állapotnak ezeknek az összevont halmazát, így már csak egy darab állapotátmenet lesz a t jel hatására. Tehát az új, determinisztikus automata állapothalmaza 2^A, ahol A a nemdeterminisztikus automata állapothalmaza.



10.1. Ábra: Nemdeterminisztikus átmenetek átalakítása

Tegyük föl, hogy $\mathcal{A} = \langle A, T, \delta, a_0, F \rangle$, és legyen $\mathcal{A}' = \langle 2^A, T, \delta', \{a_0\}, \mathcal{F} \rangle$. Definiáljuk δ'-t a

következő módon:

$$\delta'(\{b_1,\ldots,b_s\},t) := \bigcup_{i=1}^{s} \delta(b_i,t),$$
$$B \in \mathcal{F} \iff B \cap F \neq \emptyset.$$

Állítás:

$[B,u] \overset{*}{\underset{\mathcal{A}'}{\vdash}} [C,\varepsilon]$ akkor és csak akkor teljesül, ha egyrészt minden $b \in B$ és $a \in A$ esetén $[b,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [a,\varepsilon]$, akkor $a \in C$, másrészt ha minden $c \in C$-re van olyan $b \in B$, amelyre $[b,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [c,\varepsilon]$.

Bizonyítás:

Teljes indukcióval.

Ez az automata ugyanazt tudja, mint az eredeti, hiszen: $u \in L(\mathcal{A}') \iff \exists B \in \mathcal{F} : ([\{a_0\},u] \overset{*}{\underset{\mathcal{A}'}{\vdash}} [B,\varepsilon]) \iff \exists B : (B \cap F \neq \emptyset \wedge \forall b \in B : [a_0,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [b,\varepsilon]) \iff \exists b \in F : [a_0,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [b,\varepsilon] \iff u \in L(\mathcal{A})$.

Ezért tényleg $L(\mathcal{A}') = L(\mathcal{A})$, és $\mathcal{A}'$ egy véges, determinisztikus automata. Gyakorlatban általában ennek csak az összefüggő részét konstruálják meg.

△

Következmény:

$$\mathcal{L}_3 = \mathcal{L}_{0\nu} = \mathcal{L}_{\varepsilon\text{NDA}} = \mathcal{L}_{\text{NDA}} = \mathcal{L}_{\text{PDA}} = \mathcal{L}_{D\,0\nu}.$$

10.2. A 3. típusú nyelvek néhány tulajdonsága

Kis Bar-Hillel lemma:

Minden $L \in \mathcal{L}_3$ nyelvre van olyan $n = n(L) \in \mathcal{N}$ nyelvfüggő konstans, hogy minden $u \in L$ szó esetén ha $u = \alpha_1 u' \alpha_2$ $\big(l(u') \geq n\big)$, akkor van u'-nek olyan v részszava $(u' = \beta_1 v \beta_2)$, hogy $0 < l(v) \leq n$, és minden $i \geq 0$ esetén $\alpha_1\beta_1 v^i \beta_2 \alpha_2 \in L$. Tehát L minden szavának elég hosszú részszavában létezik elég rövid, nemüres beiterálható részszó.

Bizonyítás:

Tudjuk, hogy minden $L \in \mathcal{L}_3$ esetén van $\mathcal{A}$ véges, determinisztikus automata, melyre $L(\mathcal{A}) = L$. Legyen $\mathcal{A} = \langle A,T,\delta,a_0,F\rangle$, és $n := \big|A\big|\big(= n(L)\big)$. Vizsgáljunk meg egy megfelelő tulajdonságú szót, azaz legyen $u \in L(\mathcal{A}), u = \alpha_1 u' \alpha_2$, és $l(u') \geq n = \big|A\big|$.

Tegyük fel, hogy $u' = t_1 t_2 \ldots t_m$, és $m \geq n$. Az automatában a következő konfigurációátmenetek kellenek az elfogadáshoz:

$$[a_0,\alpha_1 t_1 t_2 \ldots t_m \alpha_2] \overset{*}{\underset{\mathcal{A}}{\vdash}} [c_0,t_1 t_2 \ldots t_m \alpha_2] \underset{\mathcal{A}}{\vdash} [c_1,t_2 \ldots t_m \alpha_2] \underset{\mathcal{A}}{\vdash} \ldots$$

$$\ldots \underset{\mathcal{A}}{\dashv\!\!-} [c_m, \alpha_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [f, \varepsilon] \quad (f \in F).$$

Tekintsük a $(c_0, c_1, \ldots, c_m)$ sorozatot. Ez $m+1 \geq n+1$ elemű. A skatulyaelv alapján biztosan léteznek k, j $(k \neq j)$ számok, melyekre $0 \leq j < k \leq n$ és $c_j = c_k$, ugyanis $n = |A|$. Ekkor viszont föltehető, hogy j és k között már nincs ilyen pár (mert akkor azt választottuk volna j, k-nak).

u' szó felbontása legyen $\beta_1 := t_1 \ldots t_j, v := t_{j+1} \ldots t_k$, és $\beta_2 := t_{k+1} \ldots t_m$. Tudunk adni j, k választása miatt v hosszára egy becslést: $0 < l(v) \leq n$.

Továbbá az is igaz, hogy

$$[c_0, \beta_1] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_j, \varepsilon],$$

$$[c_j, v] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_k, \varepsilon] = [c_j, \varepsilon],$$

$$[c_k, \beta_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_m, \varepsilon].$$

De ezekből már nyilván következik, hogy

$$[c_j, v^i] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_j, \varepsilon].$$

Tehát összegezve az eddigieket:

$$[a_0, \alpha_1\beta_1 v^i \beta_2\alpha_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_0, \beta_1 v^i \beta_2\alpha_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_j, v^i\beta_2\alpha_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [c_j, \beta_2\alpha_2] \overset{*}{\underset{\mathcal{A}}{\dashv\!\!-}} [f, \varepsilon].$$

és ebből már következik az állítás, hogy $\alpha_1\beta_1 v^i \beta_2\alpha_2 \in L(\mathcal{A}) = L$.

△

Most nézzünk egy példát a Kis Bar-Hillel lemma alkalmazására. Jelölje HE a helyes zárójelezések halmazát. Erre fogalmazzuk meg a következő állítást:

Állítás:

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

Bizonyítás:

Indirekt módon. Tegyük fel, hogy $HE \in \mathcal{L}_3$. A Kis Bar-Hillel lemma alapján ekkor létezik $n = n(L)$. Vegyük a következő szót: $u := ({}^n)^n$, és legyen $u' = ({}^n$. Ekkor $l(u') \geq n$, így alkalmazhatjuk a Kis Bar-Hillel lemmát.

Ekkor u'-ben létezik v nemüres, beiterálható szó. Legyen $v := ({}^d$, ahol $d > 0$. Kérdés az, hogy ekkor $({}^{n-d}({}^d)^i)^n$ eleme lesz-e a HE halmaznak. Ez viszont nem igaz, mert csak $i = 1$ esetén teljesül, ami így nyilván ellentmondást ad. △

Következmény:

A programozási nyelvek szintaxisai nem írhatók le tisztán 3-as típusú nyelvtannal. Viszont a tokenek, azonosítók leírására már használható automata, a fordítóprogramokban ez a lexikális

elemző (scanner).

Véges, determinisztikus automata esetén kiterjeszthetjük a δ függvényt nemcsak egy jelből álló u szóra, hiszen minden $a \in A$ és $u \in T^*$ esetén létezik pontosan egy $b \in A$, amelyre $[a,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [b,\varepsilon]$.

Definíció:

A kiterjesztett δ függvény legyen $\hat{\delta}(a,u) := b$, amelyre tehát $[a,u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [\hat{\delta}(a,u),\varepsilon]$.

Ezt kihasználva nyilvánvaló összefüggések az alábbiak:

$\hat{\delta}(a,\varepsilon) = a,$

$\hat{\delta}(a,uv) = \hat{\delta}(\hat{\delta}(a,u),v),$

$[a,t] \underset{\mathcal{A}}{\vdash} [\hat{\delta}(a,t),\varepsilon],\ a \in A$ és $t \in T.$

A konfigurációátmenet fogalmát felhasználva $\delta(a,t) = \hat{\delta}(a,t)$. Tehát a $\hat{\delta}$ tényleg valódi kiterjesztése δ függvénynek, így tehát a $\hat{}$ jel elhagyható.

Definíció:

Adott L nyelv $p \in T^*$-ra vonatkozó maradéknyelve L_p, ahol

$$L_p := \{v \mid pv \in L\}.$$

Tulajdonságai:

$(L_p)_q = L_{pq},$

$L_\varepsilon = L,$

$\varepsilon \in L_p \iff p\varepsilon \in L \iff p \in L.$

Definíció:

Az $\mathcal{A}$ determinisztikus automata $a \in A$ állapotra vonatkozó maradékát jelölje $L_a^{\mathcal{A}}$. Azaz legyen

$$L_a^{\mathcal{A}} := \{v \mid \delta(a,v) \in F\}.$$

Tehát az adott a állapotból a v input szó hatására az automata végállapotba kerül. Ennek tulajdonságai:

$(L_a^{\mathcal{A}})_q = L_{\delta(a,q)}^{\mathcal{A}},$

$L_{a_0}^{\mathcal{A}} = L(\mathcal{A}),$

$\varepsilon \in L_a^{\mathcal{A}} \iff a \in F.$

Myhill—Nerode tétele:

$L \in \mathcal{L}_3$ akkor és csak akkor, ha $\left|\{L_p\}_{p\in T^*}\right| < \infty$, ahol $T = T(L)$ az L nyelv ábécéje.

Bizonyítás:

„$\Longleftarrow$"

Konstruálunk egy véges, determinisztikus automatát, melynek az állapothalmaza $\{L_p\}_{p\in T^*}$. Ez a feltétel szerint véges halmaz. Ekkor L_p azt az állapotot jelöli, ahova a kezdőállapotból p input elolvasása után jutunk.

Legyen $\mathcal{A}_*^L = \langle \{L_p\}_{p\in T^*}, T, \delta, L_\varepsilon, F\rangle$, ahol $F := \{L_p \mid \varepsilon \in L_p\}$, és $\delta(L_p, t) := L_{pt}$. Ekkor $\mathcal{A}_*^L$ tulajdonságai:

1) δ jól definiált, azaz $L_p = L_q$ esetén $\delta(L_p, t) = \delta(L_q, t)$.
   — Ugyanis $L_p = L_q \implies (L_p)_t = (L_q)_t \implies L_{pt} = L_{qt} \implies \delta(L_p, t) = \delta(L_q, t)$.
2) Minden $u \in T^*$ esetén $\delta(L_p, u) = L_{pu}$.
   — A δ definíciója segítségével az u szó hosszára vonatkozó indukcióval belátható.
3) $L_p \in F \iff p \in L$.

Már csak azt kell belátni, hogy $L(\mathcal{A}_*^L) = L$. Ez pedig könnyen igazolható, hiszen $u \in L(\mathcal{A}_*^L) \iff \delta(L_\varepsilon, u) \in F \iff L_{\varepsilon u} = L_u \in F \iff u \in L$.

„$\Longrightarrow$"

Legyen $L \in \mathcal{L}_3$. Tudjuk, hogy létezik olyan $\mathcal{A} = \langle A, T, \delta, a_0, F\rangle$ véges, determinisztikus automata, amelyik az L nyelvet fogadja el, azaz $L = L(\mathcal{A})$. Ekkor biztos, hogy $|\{L_a^{\mathcal{A}}\}_{a\in A}| \le |A| < \infty$, mert $\mathcal{A}$ véges és determinisztikus. Vegyük tetszőleges $u \in T^*$-ra az L_u maradéknyelvet. Ekkor $L_u = (L_{a_0}^{\mathcal{A}})_u = L_{\delta(a_0,u)}^{\mathcal{A}}$. Így nyilván fennáll, hogy

$$\{L_u\}_{u\in T^*} \subseteq \{L_a^{\mathcal{A}}\}_{a\in A},$$

amivel a tételt beláttuk.

$\triangle$

Ekkor $\mathcal{A}_*^L$ állapothalmaza $\{L_u\}_{u\in T^*}$.

Következmény:

Az $\mathcal{A}_*^L$ automata állapotszáma kisebb vagy egyenlő, mint tetszőleges, az L nyelvhez adott automata állapotszáma.

Definíció:

Az $\mathcal{A}_*^L$ automatát minimális automatának nevezzük. (L-hez adott minimális állapotszámú automata.)

Megjegyzés: általában scanner programokhoz célszerű ezt használni, mert ezáltal jelentősen felgyorsítható a lexikális elemzés sebessége. Tehát célunk az, hogy megtaláljuk ezt a minimális automatát.

10.3. Minimális automata megkeresése

Adott egy $L \in \mathcal{L}_3$ nyelv, mely egy véges, determinisztikus automatával van megadva. Próbáljuk ebből megkonstruálni a minimális automatát, $\mathcal{A}_*^L$-et. Egyelőre állapotszám-csökkentő transzformációkat adunk.

1)Összefüggővé alakítás:

Nyilván ki lehet hagyni az automata által nem használt állapotokat.

Definíció:

Egy $\mathcal{A}$ automata összefüggő, ha minden $a \in A$ esetén létezik olyan $u \in T^*$ szó, hogy $\delta(a_0, u) = a$.

Tehát ha egy automata nem összefüggő, elhagyhatjuk a kezdőállapotból nem elérhető állapotokat, hiszen ezeknek amúgy sincs szerepük a nyelv kialakításában. Tehát az összefüggő automata állapothalmaza:

$$A_{\text{össz}} = \{a \in A \mid \exists u \in T^* : \delta(a_0, u) = a\}.$$

Konstrukció:

$$H_0 := \{a_0\},$$

$$H_{i+1} := H_i \cup \{a \mid \exists b \in H_i \wedge \exists t \in T : \delta(b, t) = a\},$$

A H_i halmazok itt is csak bővülhetnek, ezért az A állapothalmaz végessége miatt egy i_0 indextől kezdődően végig megegyeznek, és ekkor $A_{\text{össz}} = H_{i_0}$. Legyen $\mathcal{A}_{\text{össz}} = \langle A_{\text{össz}}, T, \delta \mid_{A_{\text{össz}} \times T}, a_0, F \cap A_{\text{össz}} \rangle$, erre nyilván teljesül, hogy $L(A_{\text{össz}}) = L(\mathcal{A})$.

2) Redukció:

Legyen $\mathcal{A} = \langle A, T, \delta, a_0, F \rangle$. Két állapot nem megkülönböztethető a nyelvelfogadás szempontjából, ha ugyanazon szavak hatására kerülünk belőlük végállapotba. Célunk az, hogy ezeket az állapotokat is elhagyjuk, de úgy, hogy az elfogadott nyelv ne változzon meg.

Definíció:

Legyenek $a, b \in A$ állapotok. a és b ekvivalensek, ha $L_a^{\mathcal{A}} = L_b^{\mathcal{A}}$. Jelölése: $a \sim b$.

Tulajdonságai:

— ekvivalenciareláció,

— kongruenciareláció is.

Definíció:

Egy $\rho : A \times A$ reláció jobbkongruencia, ha minden $u \in T^*$ esetén $a\rho b \Longrightarrow \delta(a,u)\rho\delta(b,u)$.

Állítás:

A $\sim$ reláció jobbkongruencia.

Bizonyítás:

Ugyanis $a \sim b \Longrightarrow \; L_a^{\mathcal{A}} = L_b^{\mathcal{A}} \; \Longrightarrow \; \forall u \in T^*$-ra $\left(L_a^{\mathcal{A}}\right)_u = \left(L_b^{\mathcal{A}}\right)_u \iff \forall u \in T^*$-ra $L^{\mathcal{A}}_{\delta(a,u)} = L^{\mathcal{A}}_{\delta(b,u)} \iff \forall u \in T^*$-ra $\delta(a,u) \sim \delta(b,u)$.

$\triangle$

Definíció:

Az „$\sim$” reláció két automata állapotaira vonatkozó kiterjesztése:

$$a_1 \sim a_2, \text{ ha} L_{a_1}^{\mathcal{A}_1} = L_{a_2}^{\mathcal{A}_2}, \text{ ahol}$$

$\mathcal{A}_1, \mathcal{A}_2$ a két automata, és $a_1 \in A_1, a_2 \in A_2$.

Definíció:

Legyenek $\mathcal{A}_1, \mathcal{A}_2$ két automata $a_0^{(1)}$ és $a_0^{(2)}$ kezdőállapotokkal. Ekkor $\mathcal{A}_1$ ekvivalens $\mathcal{A}_2$-vel, azaz

$$\mathcal{A}_1 \sim \mathcal{A}_2, \text{ ha } a_0^{(1)} \sim a_0^{(2)} \left(\text{vagyis ha } L(\mathcal{A}_1) = L(\mathcal{A}_2)\right).$$

Annak érdekében, hogy egyszerűbb automatához jussunk, vegyük a faktorstruktúrát. Ekkor magát a redukciót az jelenti, hogy képezzük az $\mathcal{A}$ automata $\sim$-ra vonatkozó faktorautomatáját. Jelölésben: $\mathcal{A}/_\sim$, azaz

$$\mathcal{A}/_\sim = \left\langle \{C_a\}_{a \in A}, T, \delta', C_{a_0}, \mathcal{F} \right\rangle,$$

ahol

C_a az a-val ekvivalens állapotok osztálya, melynek reprezentánsa a,

$\mathcal{F} = \{C_a \mid a \in F\}$,

$\delta'(C_a, t) = C_{\delta(a,t)}$, azaz egy tetszőleges reprezentánssal definiáljuk.

Az $\mathcal{A}/_{\sim}$ automata tulajdonságai:

1) δ' jól definiált. Ez nyilvánvaló, mert jobbkongruencia.

2) $\delta'(C_a, u) = C_{\delta(a,u)}$, ahol $u \in T^*$.

3) $C_b \in \mathcal{F} \iff b \in F$.

Definíció:

Egy adott $\mathcal{A}$ automata redukált automata, ha minden $a, b \in A$ esetén $a \sim b \iff a = b$, azaz nincsenek különböző ekvivalens állapotai.

Állítás:

Az $\mathcal{A}/_{\sim}$ automata redukált, és $L(\mathcal{A}/_{\sim}) = L(\mathcal{A})$.

Bizonyítás:

$\mathcal{A}/_{\sim}$ automata tulajdonságai alapján $u \in L_a^{\mathcal{A}} \iff \delta(a,u) \in F \iff C_{\delta(a,u)} \in \mathcal{F} \iff \delta'(C_a, u) \in \mathcal{F} \iff u \in L_{C_a}^{\mathcal{A}/\sim}$. Tehát beláttuk, hogy a két nyelv azonos, azaz $L(\mathcal{A}/_{\sim}) = L(\mathcal{A})$. Emiatt a definíció alapján tetszőleges $a \in A$ állapotra $a \sim C_a$, amiből következik, hogy $L_a^{\mathcal{A}} = L_{C_a}^{\mathcal{A}/\sim}$. Ezenkívül $\mathcal{A}/_{\sim}$ redukált is, ugyanis ha $C_a \sim C_b$, akkor $L_a^{\mathcal{A}} = L_{C_a}^{\mathcal{A}/\sim} = L_{C_b}^{\mathcal{A}/\sim} = L_b^{\mathcal{A}}$ miatt $a \sim b$, és emiatt ugyanazt az osztályt reprezentálják, azaz $C_a = C_b$.

$\triangle$

Megjegyzés:

Ha az automata a redukció előtt már összefüggő volt, akkor a redukció után is az lesz. Tehát ha adott egy $\mathcal{A}$ automata, akkor abból már tudunk készíteni egy összefüggő, redukált automatát, várhatóan kisebb állapotszámmal.

Kérdés, hogy ezzel mennyire közelítettük meg a minimális automatát.

Definíció:

Legyenek $\mathcal{A}_i = \langle A_i, T, \delta_i, a_0^{(i)}, F_i \rangle$, ahol $i = 1, 2$, véges, determinisztikus automaták. Ekkor $\mathcal{A}_1$ és $\mathcal{A}_2$ izomorfak, ha létezik $\phi : A_1 \longrightarrow A_2$ kölcsönösen egyértelmű ráképezés, melyre a következők teljesülnek:

1) $\phi\big(a_0^{(1)}\big) = a_0^{(2)}$,

2) $\phi\big(F_1\big) = F_2$,

3) δ-t megőrzi, azaz minden $a_1 \in A_1$ és minden $t \in T$ esetén $\phi\big(\delta_1(a_1, t)\big) = \delta_2\big(\phi(a_1), t\big)$. Könnyen belátható, hogy ekkor ez minden $u \in T^*$-ra is igaz.

Jelölés: $\mathcal{A}_1 \cong \mathcal{A}_2$.

Tétel:

Legyenek $\mathcal{A}_1$ és $\mathcal{A}_2$ összefüggő, redukált és egymással ekvivalens automaták. Ekkor $\mathcal{A}_1 \cong \mathcal{A}_2$.

Bizonyítás:

Az összefüggőségből adódik a következő két állítás:

$$A_i = \{\delta_i(a_0^{(i)}, p) \mid p \in T^*\},$$

$$F_i = \{\delta_i(a_0^{(i)}, p) \mid p \in L(\mathcal{A}_i)\}.$$

Így már könnyű a megfeleltetést megadni:

$$\phi : \delta_1(a_0^{(1)}, p) \longmapsto \delta_2(a_0^{(2)}, p) \text{ bármely } p \in T^*\text{-ra.}$$

Most már csak annak ellenőrzése van hátra, hogy ez egy bijektív függvény.

$$\delta_1(a_0^{(1)}, p) = \delta_1(a_0^{(1)}, q) \overset{\text{red, öf}}{\iff} \delta_1(a_0^{(1)}, p) \sim \delta_1(a_0^{(1)}, q) \iff L^{\mathcal{A}1}_{\delta_1(a_0^{(1)},p)} = L^{\mathcal{A}1}_{\delta_1(a_0^{(1)},q)} \iff \left(L^{\mathcal{A}1}_{a_0^{(1)}}\right)_p = \left(L^{\mathcal{A}1}_{a_0^{(1)}}\right)_q \overset{\text{ekv}}{\iff} \left(L^{\mathcal{A}2}_{a_0^{(2)}}\right)_p = \left(L^{\mathcal{A}2}_{a_0^{(2)}}\right)_q \iff L^{\mathcal{A}2}_{\delta_2(a_0^{(2)},p)} = L^{\mathcal{A}2}_{\delta_2(a_0^{(2)},q)} \iff \delta_2(a_0^{(2)}, p) \sim \delta_2(a_0^{(2)}, q) \iff \delta_2(a_0^{(2)}, p) = \delta_2(a_0^{(2)}, q).$$

Tehát ϕ jól definiált és bijektív leképezés. Továbbá meg kell vizsgálni, teljesülnek-e az 1) — 3) feltételek.

$$\phi\big(a_0^{(1)}\big) = \phi\big(\delta_1(a_0^{(1)}, \varepsilon)\big) = \delta_2(a_0^{(2)}, \varepsilon) = a_0^{(2)},$$

$\phi\big(F_1\big) = F_2$, ez F_1, F_2 előállításából következik.

Legyen $\delta_1(a_0^{(1)}, p) \in A_1$ tetszőleges.

$$\phi\big(\delta_1(\delta_1(a_0^{(1)}, p), t)\big) = \phi\big(\delta_1(a_0^{(1)}, pt)\big) = \delta_2(a_0^{(2)}, pt), \text{ és}$$

$$\delta_2\big(\phi(\delta_1(a_0^{(1)}, p)), t\big) = \delta_2\big(\delta_2(a_0^{(2)}, p), t\big) = \delta_2(a_0^{(2)}, pt).$$

△

Következmény:

Egy tetszőleges nyelv két automatájának összefüggő, redukált automatái izomorfak.

10.4. Állapotok ekvivalenciájának algoritmikus eldöntése

Két állapot ekvivalenciáját egyelőre nem tudjuk a definíció alapján eldönteni, ugyanis

$$a \sim b \iff L_a^{\mathcal{A}} = L_b^{\mathcal{A}} \iff \forall u \in T^* : \big(\delta(a, u) \in F \iff \delta(b, u) \in F\big).$$

Itt T^* végtelen, tehát algoritmikusan ez így nem megoldható. Megpróbáljuk approximálni véges relációval.

Definíció:

Legyenek ρ_1, ρ_2 bináris relációk. Ekkor ρ_1 finomabb, mint ρ_2, ha bármely $a, b \in A$ esetén $a\rho_1 b \implies a\rho_2 b$.
Jelölése: $\rho_1 \succ \rho_2$.

Definíció:

Legyen $\overset{i}{\sim}$, az i-edik ekvivalencia ($i \geq 0$) a következőképpen definiálva:

$a \overset{i}{\sim} b$ pontosan akkor, ha minden $u \in T^{\leq i}$ esetén $\big(\delta(a,u) \in F \iff \delta(b,u) \in F\big)$.

Tulajdonságai:

1) ekvivalenciareláció,
2) minden $a, b \in A$ esetén $a \overset{i+1}{\sim} b \iff a \overset{i}{\sim} b \wedge (\forall t \in T : \delta(a,t) \overset{i}{\sim} \delta(b,t))$,
3) $\overset{0}{\sim} \prec \overset{1}{\sim} \prec \overset{2}{\sim} \prec \ldots \prec \sim$. Még azt is tudjuk, hogy $a \sim b \iff \forall i \geq 0 : a \overset{i}{\sim} b$. Jelölje $|\overset{i}{\sim}|$ az $\overset{i}{\sim}$ ekvivalencia ekvivalenciaosztályainak a számát. Ekkor igaz lesz $\overset{i}{\sim}$-ra, hogy

$$|\overset{0}{\sim}| \leq |\overset{1}{\sim}| \leq |\overset{2}{\sim}| \leq \ldots \leq |\sim| \leq |A| < \infty.$$

Ekkor viszont létezik olyan $n \in \mathcal{N}$, melyre $|\overset{n}{\sim}| = |\overset{n+1}{\sim}|$. Legyen a legkisebb ilyen szám i_0. Tehát

$$i_0 := \min\{n \mid |\overset{n}{\sim}| = |\overset{n+1}{\sim}|\}.$$

Milyen becslést tudunk adni i_0-ra? Az nyilvánvaló, hogy $1 \leq |\overset{0}{\sim}|$, i_0 pedig akkor lesz a legnagyobb, ha az osztályok száma a lehető leglassabban nő, azaz egyesével. Tehát tegyük föl, hogy $|\overset{0}{\sim}| \geq 1, |\overset{1}{\sim}| \geq 2, \ldots, |\overset{i_0}{\sim}| \geq i_0 + 1$, azaz $i_0 + 1 \leq |A|$. Tehát biztos, hogy $i_0 \leq |A| - 1$.

A rekurzív összefüggés alapján nyilván minden $j \geq i_0$ esetén $\overset{j}{\sim} = \overset{i_0}{\sim}$, ugyanis ehhez azt kell belátni, hogy ha $\overset{j}{\sim} = \overset{j+1}{\sim}$, akkor $\overset{j+1}{\sim} = \overset{j+2}{\sim}$, és ez következik abból, hogy $\overset{j}{\sim} = \overset{j+1}{\sim}$ esetén minden a, b-re $a \overset{j+2}{\sim} b \iff a \overset{j+1}{\sim} b \wedge (\forall t \in T : \delta(a,t) \overset{j+1}{\sim} \delta(b,t)) \iff a \overset{j}{\sim} b \wedge (\forall t \in T : \delta(a,t) \overset{i}{\sim} \delta(b,t)) \iff a \overset{j+1}{\sim} b$.

Tehát az is fennáll, hogy minden $j \geq 0$-ra $\overset{i_0}{\sim} \succ \overset{j}{\sim}$, amiből adódik, hogy $\overset{i_0}{\sim} \succ \sim$. Azaz kölcsönösen finomabbak egymásnál, ami csak úgy lehetséges, hogy megegyeznek, tehát $\overset{i_0}{\sim} = \sim$. Í gy már algoritmikusan is eldönthető az ekvivalencia.

Következmény:

$\sim = \overset{|A|-1}{\sim}$.

11. Kleene tétele

Definíció:

Reguláris nyelveknek nevezzük az elemi nyelveket (halmazuk legyen $\mathcal{L}_{\mathrm{REG}}$). Ekkor

(i) $\mathcal{L}_{\mathrm{REG}}$ tartalmazza az elemi nyelveket,

(ii) $\mathcal{L}_{\mathrm{REG}}$ zárt az unió, konkatenáció és a lezárás műveletekre,

(iii) $\mathcal{L}_{\mathrm{REG}}$ a legszűkebb olyan nyelvosztály, mely az (i), (ii) feltételeknek megfelel.

Tétel:

$$\mathcal{L}_{\mathrm{REG}} = \mathcal{L}_3.$$

Bizonyítás:

Itt is a kétirányú tartalmazást kell belátni:

„$\subseteq$":

Elég belátni, hogy $\mathcal{L}_3$ rendelkezik az (i), (ii) tulajdonságokkal.

(i) Tudunk adni 3-as típusú szabályokat az elemi nyelveknek:

— $\mathcal{P} = \{S \longrightarrow t\}$, ha $L = \{t\}$,

— $\mathcal{P} = \{S \longrightarrow \varepsilon\}$, ha $L = \{\varepsilon\}$,

— $\mathcal{P} = \emptyset$, ha $L = \emptyset$.

(ii) korábban már láttuk, hogy $\mathcal{L}_3$ zárt a reguláris műveletekre.

„$\supseteq$"

Tudjuk, hogy $\mathcal{L}_3 = \mathcal{L}_{\mathrm{DA}}$. Elegendő tetszőleges $\mathcal{A}$ véges, determinisztikus automata esetén belátni, hogy $L(\mathcal{A}) \in \mathcal{L}_{\mathrm{REG}}$.

Számozzuk meg az automata állapotait a következőképpen:

$$\mathcal{A} = \left\langle \{a_1, \ldots, a_n\}, T, \delta, a_1, \left\{a_{i_j}\right\}_{j=1}^{s} \right\rangle.$$

Vezessük be a következő jelölést, ahol $1 \leq i, j \leq n$ és $k \geq 0$:

$L_{i,j}^k := \{u \in T^* \mid [a_i, u] \overset{*}{\underset{\mathcal{A}}{\vdash}} [a_j, \varepsilon]$, és a közben érintett állapotok indexe nem nagyobb k-nál $\}$.

A definíció alapján világos, hogy $L_{i,j}^n = L_{i,j}^{n+1} = \ldots$, hiszen nincs valódi megkötés az érintett állapotokra. Tehát ha a végállapotokra vesszük az uniót:

$$L(\mathcal{A}) = \bigcup_{j=1}^{s} L_{1,i_j}^n.$$

Tehát ha minden i, j, k-ra igaz lenne a regularitás $\left(L_{i,j}^k \in \mathcal{L}_{\mathrm{REG}}\right)$, akkor $L(\mathcal{A}) \in \mathcal{L}_{\mathrm{REG}}$, mert $\mathcal{L}_{\mathrm{REG}}$ zárt az únió műveletére.

Lemma:

Minden i, j, k esetén $L_{i,j}^k \in \mathcal{L}_{\mathrm{REG}}$.

Bizonyítás:

Teljes indukcióval k-ra vonatkozóan, minden i, j-re párhuzamosan.

$k = 0$:

Ekkor nyilván $L^0_{i,j} = \{t \in T \mid \delta(a_i, t) = a_j\} \cup \Delta_{ij}$, ahol

$$\Delta_{ij} = \begin{cases} \{\varepsilon\}, & \text{ha } i = j, \\ \emptyset, & \text{ha } i \neq j. \end{cases}$$

Tehát $L^0_{i,j}$ reguláris, mert elemi nyelvek uniója.

$k + 1$:

Tegyük fel, hogy k-ig igaz az állítás, tehát $L^k_{i,j}$ reguláris bármely i, j-re. Vizsgáljuk meg $k + 1$-re. Két eset lehetséges:

$k \geq n$:

Ebből viszont következik, hogy $L^{k+1}_{i,j} = L^k_{i,j}$, tehát $L^{k+1}_{i,j}$ is reguláris.

$k < n$:

Legyen $u \in L^{k+1}_{i,j}$ tetszőleges. Ekkor az automata a_i állapotából a_j állapotába kerül az u szó hatására úgy, hogy a közbülsőleg érintett állapotok sorszáma maximum $k + 1$. Tehát két eset lehetséges:

a) Nem volt a_{k+1} állapot. Ekkor nyilván $u \in L^k_{i,j}$ is fennáll, mely az indukciós feltétel szerint regurális nyelv.

b) Volt a_{k+1} állapot. Ekkor osszuk föl az u szót úgy, hogy ezeknél az a_{k+1} állapotoknál állapítjuk meg a részszavak határát. Tehát:

$$a_i \underbrace{\overbrace{\cdots}^{u_1} a_{k+1} \overbrace{\cdots}^{v_1} a_{k+1} \overbrace{\cdots}^{v_2} \cdots \overbrace{\cdots}^{v_l} a_{k+1} \overbrace{\cdots}^{u_2} a_j}_{u}.$$

Tudjuk, hogy minden közbülső szakaszon az állapotok sorszáma legfeljebb k, vagyis

$$u_1 \in L^k_{i,k+1}, \quad v_d \in L^k_{k+1,k+1}, \quad u_2 \in L^k_{k+1,j}.$$

Ebből már következik, hogy

$$u = u_1 v_1 \ldots v_l u_2 \in L^k_{i,k+1} \big(L^k_{k+1,k+1}\big)^* L^k_{k+1,j}.$$

Tehát azt kaptuk, hogy $L^{k+1}_{i,j} \subseteq L^k_{i,j} \cup L^k_{i,k+1} \big(L^k_{k+1,k+1}\big)^* L^k_{k+1,j}$. Itt a „$\supseteq$” tartalmazás már nyilvánvaló, tehát azonosak.

Tehát a lemma már következik mindebből, ugyanis az indukciós feltétel szerint a jobb oldalon álló nyelvek regulárisak, így $L^{k+1}_{i,j}$ is az.

$\triangle$

11.1. További zártsági tételek 3-as típusú nyelvekre

Tegyük fel, hogy adott egy $\mathcal{A} = \langle A, T, \delta, a_0, F \rangle$ véges, determinisztikus automata. Ez kiterjeszthető tetszőleges $T' \supseteq T$ ábécére. Bővítsük ki az állapothalmazt egy d zsákutca állapottal, és definiáljuk erre δ-t a következőképp:

$\delta(a, t') := d$ minden $t' \in T' \setminus T$-re és $a \in A$-ra,

$\delta(d, t') := d$ minden $t' \in T'$-re.

További feltétel, hogy $d \notin F$, így $L(\mathcal{A}') = L(\mathcal{A})$.

Tétel:

Az $\mathcal{L}_3$ nyelvosztály zárt a komplementer, a metszet, a különbség és a szimmetrikus differencia műveletekre.

Bizonyítás:

Elég belátni egy tetszőleges $\mathcal{L}_{\mathrm{DA}}$ automatára.

Komplementer:

Az $\mathcal{A}$ véges, determinisztikus automatához konstruálunk egy olyan $\overline{\mathcal{A}}$ automatát, amelyre $L(\overline{\mathcal{A}}) = \overline{L(\mathcal{A})}$. Legyen $\overline{\mathcal{A}} = \langle A, T, \delta, a_0, A \setminus F \rangle$, és erre teljesül az állítás.

Metszet, különbség, szimmetrikus differencia:

Legyen $\odot$ tetszőleges művelet a $\cap, \setminus, \triangle$ műveletek közül. Legyenek $\mathcal{A}_i = \langle A_i, T, \delta_i, a_0^{(i)}, F_i \rangle$, automaták, $i = 1, 2$, ahol feltettük, hogy az ábécék azonosak, hiszen kiterjeszthetők. Konstruálni kell egy $\mathcal{A}_\odot$ automatát, melyre fennáll, hogy

$$L(\mathcal{A}_\odot) = L(\mathcal{A}_1) \odot L(\mathcal{A}_2).$$

Legyen $\mathcal{A}_\odot = \langle A_1 \times A_2, T, \delta_1 \times \delta_2, (a_0^{(1)}, a_0^{(2)}), F_\odot \rangle$ ún. direkt szorzat automata. Ekkor $\mathcal{A}_\odot$ működése egyszerű, mert a kezdőállapotból $\delta_1 \times \delta_2$ szerint haladunk, azaz komponensenként. Tehát $(\delta_1 \times \delta_2)\big((a_1, a_2), t\big) = \big(\delta_1(a_1, t), \delta_2(a_2, t)\big)$. Ekkor még a végállapotok halmazát kell definiálni:

$F_\cap := F_1 \times F_2$,

$F_\setminus := F_1 \times (A_2 \setminus F_2)$,

$F_\triangle := \big(F_1 \times (A_2 \setminus F_2)\big) \cup \big((A_1 \setminus F_1) \times F_2\big)$.

$\triangle$

12. Algoritmikusan eldönthető problémák 3-as típusú nyelveken

Egy adott probléma algoritmikusan eldönthető, ha létezik olyan algoritmus, mely *igen* válasszal áll le, ha van megoldás, különben *nem* választ ad.

Eldöntési problémák:

I) $u \in L$, ahol u input szó, és $L \in \mathcal{L}_3$.

Tegyük fel, hogy L mint egy véges, determinisztikus automata van megadva. (Ez már véges leírás.) A döntési eljárás:

1) ha $u \notin T(\mathcal{A})^*$, akkor „nem" a válasz,
2) ha $u \in T(\mathcal{A})^*$, akkor u szót az $\mathcal{A}$ automata inputjára tesszük, és elindítjuk $\mathcal{A}$-t. Ha az $\mathcal{A}$ automata elfogadja az u szót, akkor a döntési eljárás is igent válaszol, egyébként nemet.

Itt az automatát nem kell feltétlen ismerni, csak mint „fekete doboz"-t, ugyanis maga a döntési eljárás nem használja ki az automata belső szerkezetét. Tehát orákulumként működik, azaz felteszünk egy kérdést, amire igennel vagy nemmel válaszol.

Viszont legalább három információt ismerni kell az automatáról:

1) ismerni kell az ábécéjét,
2) az állapotainak számát $\big(n(\mathcal{A})\big)$,
3) és lennie kell egy RESET kapcsolónak, amivel alapállapotba tudjuk hozni.

Ezekkel az eldöntési feladat már megoldható.

II) $L \neq \emptyset$?

Egy kézenfekvő (direkt) módszer, hogy minden T^*-ban szereplő szóra alkalmazzuk az I) eldöntési problémát. Ekkor ha valamikor *igen* választ kapunk, akkor nem üres a nyelv. Természetesen ez az út így nem járható, mert nem véges.

Állítás:

$$L \neq \emptyset \iff L \cap T^{<n(\mathcal{A})} \neq \emptyset.$$

Bizonyítás:

„$\Longleftarrow$" irány triviális. „$\Longrightarrow$" irány bizonyítása a kis Bar-Hillel-lemmával történik. Legyen u az L nyelv egy minimális hosszúságú szava. Azt kellene belátni, hogy $l(u) < n(\mathcal{A})$.

Indirekt módon tegyük fel, hogy $l(u) \geq n(\mathcal{A})$. Mivel $L \in \mathcal{L}_3$, ezért igaz a kis Bar-Hillel-lemma, ahol konstansként választhatjuk $n(\mathcal{A})$-t, hiszen bármely automata állapotszáma jó. Mivel $l(u) \geq n(\mathcal{A})$, ezért létezik $v \neq \varepsilon$ beiterálható részszava. Iteráljuk ezt $i = 0$-szor és jelölje az így kapott szót $u' \in L$. Világos, hogy $l(u') = l(u) - l(v) < l(u)$, ami nyilván ellentmond a

feltevésnek.

$\triangle$

Ezzel beláttuk, hogy mégis visszavezethető az I)-es eldöntési problémára ez a feladat úgy, hogy az összes $u \in T(\mathcal{A})^{<n(\mathcal{A})}$ szóra megkérdezzük, hogy $u \in L$ teljesül-e.

III) $|L| = \infty$?

Ha ez eldönthető, akkor nyilván $|L| < \infty$ is eldönthető.

Állítás:

$|L| = \infty \iff L \cap T(\mathcal{A})^{n(\mathcal{A}) \leq l < 2n(\mathcal{A})} \neq \emptyset$.

Bizonyítás:

Az előző állításhoz hasonlóan a kis Bar-Hillel-lemmával történhet, amit az olvasóra bízunk.

$\triangle$

IV) $L_1 \subseteq L_2$?

Világos, hogy $L_1 \subseteq L_2 \iff L_1 \setminus L_2 = \emptyset$. (A zártsági tételt felhasználva $L_1 \setminus L_2 \in \mathcal{L}_3$!)

Probléma: Az L_1, L_2 automaták külön-külön adottak, most pedig a különbségautomatára van szükségünk. Nyilván az előző zártsági tételt felhasználva megkonstruálható a direkt szorzat automata: $\mathcal{A}_\setminus$. Ennek végállapotai: $F_1 \times (A_2 \setminus F_2)$. Erre meg tudjuk, hogy $n(\mathcal{A}_\setminus) = n(\mathcal{A}_1)n(\mathcal{A}_2)$.

Kérdés: hogyan lehetne $\mathcal{A}_\setminus$-t mint orákulumot előállítani az $\mathcal{A}_1, \mathcal{A}_2$ orákulumokból? Szimulálással. Ugyanis minden szóval az $\mathcal{A}_1, \mathcal{A}_2$ orákulumokat egyszerre indítjuk el, és ha mindkettő leállt, megvizsgáljuk az eredményt. Ha $\mathcal{A}_1$ igent adott, $\mathcal{A}_2$ pedig nemet, akkor *igent* ad vissza a szimulált orákulum is, egyébként *nemet*.

V) $L_1 = L_2$?

Ez is visszavezethető, ugyanis $L_1 = L_2 \iff L_1 \triangle L_2 = \emptyset$. Ezáltal visszavezettük a II)-es problémára. Itt az $\mathcal{A}_\triangle$ automata az előző feladathoz hasonló módon szimulálható.

13. 2-es típusú nyelvek

Definíció:

Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ tetszőleges kiterjesztett 2-es típusú nyelvtan. Ekkor a t fát G feletti szintaxisfának nevezzük, ha megfelel a következő tulajdonságoknak:

1) Pontjai $T \cup N \cup \{\varepsilon\}$ elemeivel vannak címkézve.

2) Belső pontjai N elemeivel vannak címkézve.
3) Legyen egy belső pont címkéje X, a közvetlen leszármazottjainak címkéi pedig $X_1, X_2, \ldots, X_k$ balról jobbra olvasva. Ekkor $X \longrightarrow X_1 X_2 \ldots X_k \in \mathcal{P}$.
4) Az ε-nal címkézett pontoknak nincs testvére.

A szintaxisfákkal a levezetés szerkezetét ábrázoljuk. Jelölje egy adott t szintaxisfa leveleinek balról jobbra való összeolvasását front(t), a fa gyökerét pedig gy(t).

Lemma:

Ha t G feletti szintaxisfa, akkor $\text{gy}(t) \xrightarrow[G]{*} \text{front}(t)$.

Bizonyítás:

A fa magasságára vonatkozó indukcióval.

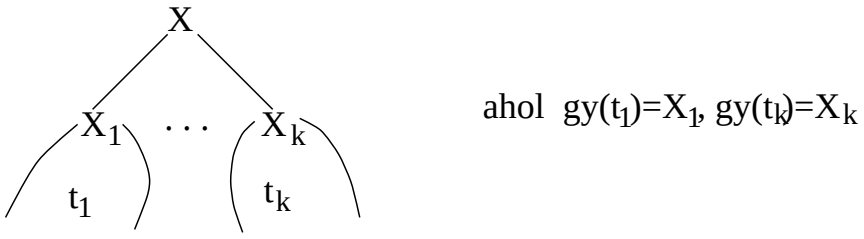
$h = 0$ magasság:

Ekkor a t szintaxisfa csak egy pontból állhat, mely esetben $\text{gy}(t) = \text{front}(t)$. (Ugyanis ekkor minden G nyelvtanra az állítás igaz lesz, mert a levezetés reflexív.)

$h + 1$ magasság:

Tegyük fel, hogy h magasságra igaz az állítás, be kell látnunk, hogy $h + 1$ magasságra is igaz lesz.

Legyen $h(t) = h+1$. Ekkor biztosan van legalább két szintje. Legyen a gyökér $X \in N$-nel cimkézve. Ha t egy szintaxisfa volt, akkor a $t_1, t_2, \ldots, t_k$ részfák is szintaxisfák, tehát igaz, hogy $h(t_1), \ldots, h(t_k) \leq h$. Alkalmazzuk az indukciós feltételt a t_i részfákra, tehát $\text{gy}(t_i) = \text{front}(t_i)$, minden $i = 1, \ldots, k$ esetén.



13.2. Ábra: Indukciós feltétel alkalmazása egy szintaxisfára

Legyenek $\text{gy}(t_i) = X_i$ nyelvtani jelek. Tudjuk a definícióból, hogy balról jobbra olvasva $X \longrightarrow X_1 \ldots X_k \in \mathcal{P}$, ezért megkonstruálhatjuk a következő levezetést:

$$\text{gy}(t) = X \xrightarrow[G]{} X_1 \ldots X_k \xrightarrow[G]{*} \text{front}(t_1) X_2 \ldots X_k \xrightarrow[G]{*} \ldots \xrightarrow[G]{*} \text{front}(t_1) \ldots \text{front}(t_k) = \text{front}(t).$$

△

Definíció:

Kaptunk egy speciális levezetést, ugyanis ha egy pozíción elvégzünk egy levezetést, az attól balra eső jelek már változatlanok maradnak a levezetés végéig. Ezt nevezzük *legbal* levezetésnek.

Jelölése: $A \underset{G,\mathrm{lb}}{\overset{*}{\longrightarrow}} \alpha$.

Ugyanígy definiálható a *legjobb* levezetés is. Jelölése: $A \underset{G,\mathrm{lj}}{\overset{*}{\longrightarrow}} \alpha$.

Következmény:

Minden G feletti t szintaxisfa esetén $\mathrm{gy}(t) \underset{G,\mathrm{lb}}{\overset{*}{\longrightarrow}} \mathrm{front}(t)$. (Természetesen legjobb levezetés is létezik.)

Lemma:

Legyen $X \in T \cup N \cup \{\varepsilon\}$ tetszőleges. Ha $X \underset{G}{\overset{*}{\longrightarrow}} \alpha$, akkor létezik G feletti t szintaxisfa, amelyre $\mathrm{gy}(t) = X$, és $\mathrm{front}(t) = \alpha$.

Bizonyítás:

Levezetés hossza szerinti indukcióval.

$h = 0$ hosszú levezetésre:

Ekkor nyilván $X \underset{G}{\overset{(0)}{\longrightarrow}} \alpha$. A definíció szerint $X = \alpha$, és ekkor egy pontból áll a fa, melynek X a címkéje.

$h + 1$ hosszú levezetésre:

Tegyük fel, hogy h hosszú levezetésre igaz az állítás, bizonyítjuk $h+1$-re. Legyen $X \underset{G}{\overset{(h+1)}{\longrightarrow}} \alpha$. Ekkor természetesen létezik legelső lépés ($h \geq 0$). Tegyük fel, hogy ez a legelső lépés a következőképpen néz ki:

$$X \underset{G}{\longrightarrow} X_1 X_2 \ldots X_k \underset{G}{\overset{(h)}{\longrightarrow}} \alpha.$$

Mivel a G nyelvtan 2-es típusú, ezért egy korábbi állítás alapján felbontható $\alpha = \alpha_1 \ldots \alpha_k$ alakra úgy, hogy $X_i \underset{G}{\overset{(\leq h)}{\longrightarrow}} \alpha_i$, $i = 1, \ldots, k$. Alkalmazva az indukciós feltételt ezekhez léteznek t_i szintaxisfák, amikre $\mathrm{gy}(t_i) = X_i$ és $\mathrm{front}(t_i) = \alpha_i$. Tehát megkonstruálható a következő fa:

a) $k = 0$ esetén az $X \longrightarrow \varepsilon$ szabályt alkalmaztuk, tehát a szintaxisfa az X nyelvtani jellel címkézett gyökérből és egy ε-nal címkézett levélből áll.

b) $k \geq 1$ esetén létrehozunk egy X jellel címkézett csúcsot, és ehhez kapcsoljuk hozzá balról jobbra a $t_1, \ldots, t_k$ szintaxisfák gyökeréit. Ezt megtehetjük, mert $X \longrightarrow X_1 X_2 \ldots X_k \in \mathcal{P}$. Világos,hogy mindkét esetben $\mathrm{gy}(t) = X$ és $\mathrm{front}(t) = \alpha$.

$\triangle$

Következmény:

$X \underset{G}{\overset{*}{\longrightarrow}} \alpha \iff X \underset{G,\mathrm{lb}}{\overset{*}{\longrightarrow}} \alpha \iff X \underset{G,\mathrm{lj}}{\overset{*}{\longrightarrow}} \alpha \iff$ létezik G feletti t szintaxisfa, amelyre $\mathrm{gy}(t) = X$ és

front$(t) = \alpha$.

Tehát szóproblémát úgy is el lehet dönteni, hogy megpróbálunk megkonstruálni egy G feletti t szintaxisfát, amire gy$(t) = S$, és front$(t) = u$. Ezt a folyamatot nevezzük elemzésnek. Például programnyelvek elemzéséhez használható, ugyanis még a szó szerkezetét is megadja.

Továbbá lényeges azt vizsgálni, hogy egy adott szónak hány jelentése lehet. Ugyanis ha több, egymástól különböző szintaxisfát lehet konstruálni egy adott szóhoz, akkor már több különböző levezetése létezik. Tehát lényeges probléma az egyértelműség vizsgálata.

Definíció:

Egy adott G nyelvtan egyértelmű, ha minden $u \in L(G)$ szónak pontosan egy szintaxisfája létezik.

Definíció:

Az $L \in \mathcal{L}_2$ nyelv egyértelmű, ha van olyan $G \in \mathcal{G}_{\text{kit2}}$ nyelvtan, mely egyértelmű.

Definíció:

Az $L \in \mathcal{L}_2$ nyelv lényegesen nem egyértelmű, ha az adott nyelvhez nem létezik egyértelmű nyelvtan.

Probléma: Megtalálni az egyértelmű nyelvtanokat.

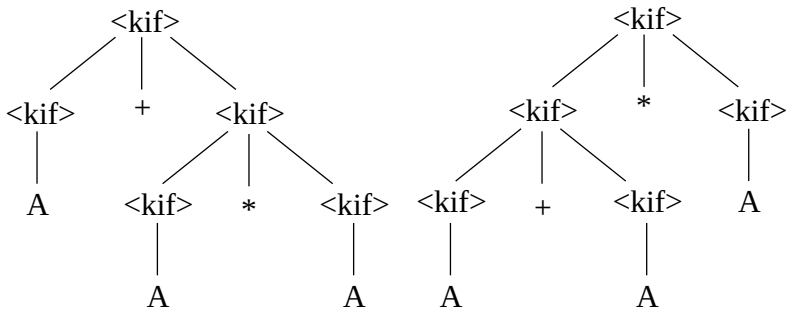
1.Példa: Nem egyértelmű nyelvtan: Legyen A egy azonosító (terminális jel), és G_1 pedig olyan nyelvtan, ahol:

$$T := \{A, +, *\},$$

$$N := \{< kif >\},$$

$$\mathcal{P} := \{< kif > \longrightarrow < kif > + < kif > | < kif > * < kif > | A\}.$$

Nézzük meg az $u = A + A * A$ szót. Erre két különböző levezetés is adható:



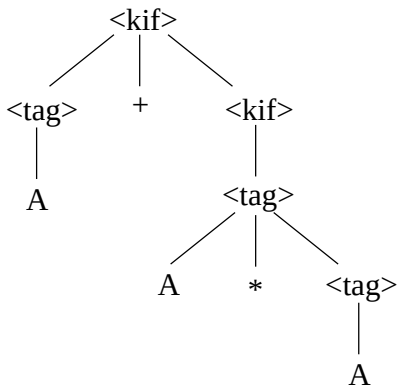
13.2. Ábra: Példa nem egyértelmű nyelvtanra

Ez a nyelvtan nem egyértelmű, de a generált nyelv mégis egyértelmű, mert tudunk adni hozzá egyértelmű nyelvtant. Legyen G_2 nyelvtan szabályrendszere a következő:

$$< kif > \longrightarrow < tag > | < tag > + < kif >$$

$$< tag > \longrightarrow A \mid A* < tag >$$

Ugyanezen $u = A + A * A$ példában már csak egy szintaxisfa létezik:



13.3. Ábra: Levezetés egyértelmű nyelvtanban

2.Példa: Lényegesen nem egyértelmű nyelv: Legyen a nyelv a következő:

$$\{a^n b^n c^m \mid n, m \geq 0\} \cup \{a^m b^n c^n \mid n, m \geq 0\}.$$

Itt nyilván lehet adni 2-es típusú nyelvtant az unió mindkét tagjára külön-külön, de magára az unióra már nem. Ugyanis az $a^n b^n c^n$ szavaknak mindig lesz két különböző szintaxisfája. (Ennek bizonyítását az olvasóra bízzuk.)

14. Nagy Bar-Hillel-lemma

Lemma:

Minden $L \in \mathcal{L}_2$ esetén léteznek $p, q > 0$ nyelvfüggő egész konstansok $(p = p(L), q = q(L))$, amelyekre ha $u \in L$, és $l(u) > p$, akkor u-nak létezik $u = xyzvw$ felbontása, ahol $l(yv) > 0$, $l(yzv) \leq q$ és minden $i \geq 0$ egészre $xy^i zv^i w \in L$.

Bizonyítás:

Mivel $L \in \mathcal{L}_2$, ezért létezik hozzá G Chomsky-normálformájú nyelvtan, melyre $L(G) = L$.

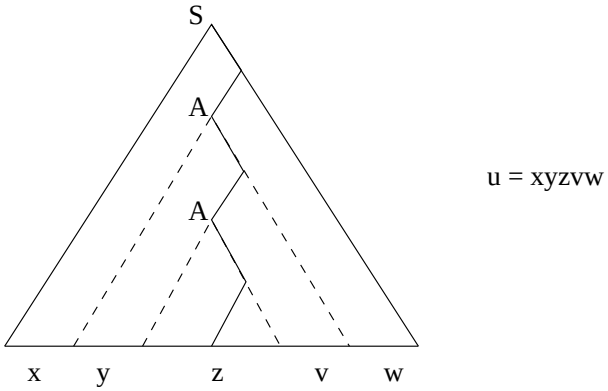
Legyen $G = \langle T, N, \mathcal{P}, S \rangle$ Chomsky-nyelvtan. (Szabályai $A \longrightarrow BC, A \longrightarrow a, S \longrightarrow \varepsilon$ alakúak.) Adjuk meg a konstansokat: $p := 2^{|N|-1}$ és $q := 2^{|N|}$. Ezek a konstansok függnek a nyelvtől, de az u szótól már nem.

Bizonyítani kell, hogy ekkor létezik egy $u = xyzvw$ felbontás. Legyen $u \in L$ olyan, hogy $l(u) > p$. Tudjuk, hogy u-hoz létezik olyan G feletti t szintaxisfa, melyre $\mathrm{gy}(t) = S$ és $\mathrm{front}(t) = u$. Tudjuk továbbá, hogy t bináris fa, mert G Chomsky-normálformájú. Így ha elhagyjuk a leveleit, egy pontos bináris fa keletkezik, de ugyanannyi levele lesz akkor is. (Az $A \longrightarrow a$ alakú szabályok miatt.) Jelöljük ezt a fát $\mathrm{red}(t)$-vel.

Becsüljük meg t magasságát $\mathrm{front}(t)$ hossza segítségével. Először $\mathrm{red}(t)$ magasságát becsüljük:

$$h(t) - 1 = h(\mathrm{red}(t)) \geq \log_2 l\big(\mathrm{front}(\mathrm{red}(t))\big) = \log_2 l(\mathrm{front}(t)) = \log_2 l(u).$$

Valamint $\log_2 l(u) > \log_2 2^{|N|-1} = |N| - 1$. Ebből már következik, hogy $h(t) > |N|$.



14.4. Ábra: A szintaxisfa egyik leghosszabb útja

A leghosszabb úton ezért legalább $|N|+2$ darab pont van. Ebből egy terminális jel, a maradék $|N|+1$ pont nyelvtani jellel van címkézve. Tehát biztosan van legalább egy ismétlődő nyelvtani jel. Válasszuk ki a leghosszabb úton alulról a legelőször ismétlődő párt. Jelölje ezt az ismétlődő nyelvtani jelet A. Vegyük azt a két t_1, t_2 részfát, ahol ez az ismétlődő nyelvtani jel a gyökér. Ezáltal felosztottuk a szót öt részre. Ezek alapján a következő levezetéseket adhatjuk meg:

a) $S \xrightarrow[G]{*} xAw$, a $t - t_1$ szintaxisfából,

b) $A \xrightarrow[G]{*} yAv$, a $t_1 - t_2$ szintaxisfából,

c) $A \xrightarrow[G]{*} z$, a t_2 szintaxisfából.

Alkalmazzuk egyszer az a) pontot, majd i-szer a b) pontot, végül ismét egyszer c)-t. Az eredmény:

$$xy^i zv^i w \in L(G).$$

Be kell még látnunk, hogy $l(yv) > 0$. Nézzük meg a b) pontban szereplő levezetést. Ha $l(yv) = 0$ lenne, az azt jelentené, hogy a G nyelvtanban $A \xrightarrow[G]{(k)} A$ teljesülne valamely $k > 0$-ra. Viszont ez a Chomsky-tulajdonságnak ellentmond, tehát $l(yv) > 0$.

Bizonyítani kell még, hogy $l(yzv) \leq q$. Tudjuk, hogy a t fában a leghosszabb út t_1-re eső szakasza t_1-ben is a leghosszabb, erre viszont a következő becslést adhatjuk:

$$\mathrm{sz}(t_1) \leq |N| - 1 + 1,$$

ahol $\mathrm{sz}(t_1)$ t_1 fa szintjeinek számát, $|N|$ a különböző nyelvtani jelek számát, -1 az ismétlést, $+1$ pedig a terminális jelet jelenti. Tehát

$$h(t_1) \leq |N| + 1.$$

Ebből következik, hogy

$$|N| \geq h(t_1) - 1 = h(\mathrm{red}(t_1)) \geq \log_2 l\big(\mathrm{front}(\mathrm{red}(t_1))\big) = \log_2 l(\mathrm{front}(t_1)) = \log_2 l(yzv).$$

Tehát összegezve megkaptuk, hogy

$$q = 2^{|N|} \geq l(yzv).$$

△

14.1. A nagy Bar-Hillel-lemma alkalmazása

Mivel ez a lemma egy szükséges feltételt ad meg, ezért arra használhatjuk, hogy kimutassuk, egy adott nyelv nem 2-es típusú.

Példa:

Legyen T olyan ábécé, amelyre $|T| \geq 2$, és két különböző jele legyen t_1, t_2. Legyen L az úgynevezett dadogós szavak nyelve, tehát $L = \{uu \mid u \in T^*\}$.

Állítás:

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

Bizonyítás:

Indirekt módon. Tegyük fel, hogy $L \in \mathcal{L}_2$. Ekkor a nagy Bar-Hillel lemma alapján léteznek a p, q nyelvfüggő konstansok. Legyen $M := \max\{p, q\}$. Vizsgáljuk az $\alpha = t_1^M t_2^M t_1^M t_2^M \in L$ szót. Nyilván $l(\alpha) > p$, tehát léteznie kell két, párhuzamosan beiterálható, együtt nemüres részszónak, amire a kapott új szavak ismét benne vannak a nyelvben.

Végezzük el az $i = 0$-ra az iterációt, azaz hagyjuk el a megfelelő részszavakat. Legyen az iteráció eredménye $t_1^{m_1} t_2^{m_2} t_1^{m_3} t_2^{m_4} \in L$, ekkor valamely $1 \leq j \leq 4$-re $m_j < M$, mert nemüres részszót hagytunk el. Tegyük fel, hogy t_1-ből hagytunk el (a bizonyítás ugyanilyen módon történik t_2-re is). Ekkor viszont $m_1 = m_3 < M$. Ebből következik, hogy az yzv részszó mindenképpen lefedi a teljes t_2-es blokkot, és belemetsz mindkét t_1-be, legalább egy-egy jellel. Ez viszont ellentmondás, ugyanis ellentétben a nagy Bar-Hillel-lemmával az $l(yzv) \not\leq M$.

△

Következmény:

A programozási nyelvek szintaxisa nem írható le tisztán 2-es típusú nyelvtanok segítségével. Ugyanis pl. a deklarációk az úgynevezett dadogós szavak legegyszerűbb modellje, amire beláttuk, hogy nem írhatók le 2-es típusú nyelvtannal. (Az első u részszó ugyanis a deklaráció, a második pedig a hivatkozás.)

14.2. Algoritmikusan eldönthető problémák 2-es típusú nyelveknél

Tétel:

Legyen L egy 2-es típusú nyelv. Az alábbi problémák algoritmikusan eldönthetők:

I) $u \in L$,

II) $L \neq \emptyset$,

III) $|L| = \infty$.

Bizonyítás:

I) Már beláttuk 1-es típusú nyelvekre is. (A 2-es típusú nyelvtanok szabályai hosszt nem csökkentő szabályok.)

II) Ez visszavezethető $L \neq \emptyset \iff L \cap T^{\leq p} \neq \emptyset$ alapján az I) problémára, ahol p például a nagy Bar-Hillel-lemma konstansa.

III) Ez is visszavezethető az $|L| = \infty \iff L \cap T^{p<i\leq p+q} \neq \emptyset$ állítás alapján, ahol a p, q konstansok ismét a nagy Bar-Hillel-lemma konstansai.

△

Kérdés: Miért nem vihető át az $L_1 \subseteq L_2$, illetve az $L_1 = L_2$ kérdés algoritmikus eldönthetősége $\mathcal{L}_2$-re? Erre ad választ a következő tétel.

Tétel:

$\mathcal{L}_2$ nem zárt a komplementer, a metszet, a különbség és a szimmetrikus differencia műveletekre.

Bizonyítás:

Metszet:

Direkt módon. Vizsgáljuk meg a következő nyelveket:

$$L_1 = \{a^n b^n c^m \mid n, m > 0\},$$

$$L_2 = \{a^m b^n c^n \mid n, m > 0\}.$$

Könnyen látszik, hogy $L_1, L_2 \in \mathcal{L}_2$ (olyan szabályok adhatók, ahol a két részt külön-külön vezetjük le), de ugyanakkor

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{L}_2.$$

Ezt a nagy Bar-Hillel-lemma segítségével láthatjuk be, indirekt módon. (Hasonlóan, mint ahogy azt a dadogós szavak esetében tettük.)

Komplementer:

Indirekt módon. Tegyük fel, hogy a komplementer is 2-es típusú lesz. Ekkor legyen $\overline{L} = T(L)^* \setminus L$, és a DeMorgan-azonosság alapján $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. Viszont ha a komplementerre zárt lenne, a metszetre is zártnak kéne lennie, ami nyilván ellentmondás.

Különbség:

Indirekt módon. Tegyük fel, hogy $T^* \setminus L \in \mathcal{L}_2$, ha $L \in \mathcal{L}_2$, és $\mathcal{L}_2$zárt a különbségre. De tudjuk, hogy $T^* \setminus L = \overline{L}$ (nyilván $T^* \in \mathcal{L}_2$ bármely T esetén), és ekkor a komplementer képzésére is zárt lenne, ami nyilvánvaló ellentmondás.

Szimmetrikus differencia:

Indirekt módon. Ekkor $T^* \triangle L = T^* \setminus L = \overline{L}$ miatt a komplementer képzésére is zárt lenne, de ez ellentmondás.

△

14.3. A veremautomaták és a 2-es típusú nyelvek kapcsolata

A továbbiakban 1-vermekkel fogunk foglalkozni. Legyen $\mathcal{V} = \langle A, T, \Sigma, \delta, a_0, \sigma_0, F \rangle$.

Definíció:

A $\mathcal{V}$ veremautomata által végállapottal elfogadott nyelv:

$$L^F(\mathcal{V}) = \{u \in T^* \mid [a_0, u, \sigma_0] \overset{*}{\underset{\mathcal{V}}{\dashv}} [f, \varepsilon, \alpha] \text{ valamely } f \in F\text{-re}\}.$$

Definíció:

A $\mathcal{V}$ veremautomata által üres veremmel elfogadott nyelv:

$$L^\varepsilon(\mathcal{V}) = \{u \in T^* \mid [a_0, u, \sigma_0] \overset{*}{\underset{\mathcal{V}}{\dashv}} [b, \varepsilon, \varepsilon] \text{ valamely } b \in A\text{-re}\}.$$

Ekkor az automata definíciójában a végállapotok halmazát nem szokták jelölni.

Jelölések:

$\mathcal{L}^F_{1\mathcal{V}}$ a végállapottal elfogadott nyelvek osztálya.

$\mathcal{L}^\varepsilon_{1\mathcal{V}}$ az üres veremmel elfogadott nyelvek osztálya.

Lemma:

Tetszőleges $\mathcal{V}$ 1-veremhez létezik olyan $\mathcal{V}'$ 1-verem, melyre $L^\varepsilon(\mathcal{V}') = L^F(\mathcal{V})$.

Bizonyítás:

Nyilván $\mathcal{V}'$-nek hasonlóan kell működnie, mint $\mathcal{V}$-nek, csak védekeznie kell az ellen, hogy idő előtt ürüljön ki a verem, illetve ha $\mathcal{V}$ végállapotba kerül, akkor ki kell ürítse a vermet. Tehát lényegében ugyanazt csinálja, mint $\mathcal{V}$, azaz szimulálja.

Legyen

$\mathcal{V} = \langle A, T, \Sigma, \delta, a_0, \sigma_0, F \rangle$, és

$\mathcal{V}' = \langle A \cup \{a'_0, d\}, T, \Sigma \cup \{\sigma'_0\}, \delta', a'_0, \sigma'_0, \ \rangle$, ahol

a'_0 az új kezdőállapot, σ'_0 az új kezdőszimbóluma a veremnek, ami mindig a veremben marad, d az ürítőállapot, δ' megegyezik δ-val, csak még néhány új értelmezése van:

— $\delta'(a'_0, \varepsilon, \sigma'_0) = (a_0, \sigma_0\sigma'_0)$,

— minden $f \in F, \sigma \in \Sigma$ esetén $\delta'(f, \varepsilon, \sigma) = (d, \varepsilon)$, és $\delta'(d, \varepsilon, \sigma) = (d, \varepsilon)$.

Ekkor nyilván $L^\varepsilon(\mathcal{V}') = L^F(\mathcal{V})$.

△

Lemma:

Tetszőleges $\mathcal{V}$ 1-veremhez létezik olyan $\mathcal{V}'$ 1-verem, melyre $L^F(\mathcal{V}') = L^\varepsilon(\mathcal{V})$.

Bizonyítás:

Mivel az előző lemma megfordítása, hasonlóan bizonyítjuk itt is, azaz konstruálunk egy $\mathcal{V}'$ 1-vermet. Fel kell venni egy új veremtetőt, hogy ki ne ürüljön a verem, és kell egy új állapot, mely az egyetlen végállapot lesz. Amikor $\mathcal{V}$ elfogad egy szót (kiürül a verem), akkor $\mathcal{V}'$-nek az új végállapotba kell mennie. Tehát itt is szimuláljuk az eredeti $\mathcal{V}$-t. δ' új értelmezései:

$\delta'(a'_0, \varepsilon, \sigma'_0) = (a_0, \sigma_0\sigma'_0)$,

$\delta'(b, \varepsilon, \sigma'_0) = (a_{\text{vég}}, \sigma'_0)$, minden $b \in A$ esetén.

Ebben az esetben $F' = \{a_{\text{vég}}\}$ és σ'_0 az új jel, ami a veremtető funkcióját tölti be.

△

Tétel:

$\mathcal{L}^F_{1\mathcal{V}} = \mathcal{L}^\varepsilon_{1\mathcal{V}}$.

Bizonyítás:

Az előző két lemma alapján nyilvánvaló, hogy $\mathcal{L}_{1\mathcal{V}} = \mathcal{L}^F_{1\mathcal{V}} = \mathcal{L}^\varepsilon_{1\mathcal{V}}$.

$\triangle$

Definíció:

Legbal, kezdőszelet-egyeztetéses elemzésnek hívjuk azt, amikor egy $u \in L(G)$ szó egy legbal levezetését konstruáljuk meg levezetés kezdőszeletének az eredeti u szó kezdőszeletével való egyeztetésével.

Példa:

$$\mathrm{S} \longrightarrow \mathrm{SS} \mid a\mathrm{S}b \mid b\mathrm{S}a \mid ab \mid ba.$$

Legyen $u = abba$. Ekkor egy levezetés például $\mathrm{S} \longrightarrow \mathrm{SS} \longrightarrow ab\mathrm{S}$. Itt az ab szó megegyezik u szó kezdőszeletével, tehát ez egy legbal, kezdőszelet-egyeztetéses elemzés kezdete lehet. Ha a továbbiakban nincs egyezés, más alternatívát kell keresni.

Tétel:

$\mathcal{L}_{1\mathcal{V}} = \mathcal{L}_2$.

Bizonyítás:

„$\supseteq$”:

Ehhez elég belátni, hogy tetszőleges $G = \langle T, N, \mathcal{P}, S \rangle \in \mathcal{G}_{\mathrm{kit2}}$ esetén létezik olyan $\mathcal{V}$ 1-verem, melyre $\mathrm{L}^\varepsilon(\mathcal{V}) = L(G)$.

Olyan $\mathcal{V}$ 1-vermet készítünk, mely a veremben egy legbal, kezdőszelet-egyeztetéses elemzést végez. Legyen $\mathcal{V} = \langle \{a_0\}, T, T \cup N, \delta, a_0, S \rangle$. Kétféle szabály lesz. Az első az ε-olvasásos szabály. Ez a legbal levezetésnek felel meg a következőképpen:

$$(a_0, p) \in \delta(a_0, \varepsilon, B) \iff B \longrightarrow p \in \mathcal{P} \quad \big(p \in (T \cup N)^*\big).$$

(A verem teteje a legbal levezetés bal oldali jele, így ezt helyettesítjük a szabály jobb oldalával.)

A másik szabály a kezdőszelet-egyeztetéses:

$\delta(a_0, t, t) = (a_0, \varepsilon)$, minden $t \in T$-re.

Tulajdonképpen ez az összehasonlítás, hogy a levezetett terminális jel benne van-e a szóban.

Erre a $\mathcal{V}$-re nyilván $L^\varepsilon(\mathcal{V}) = L(G)$.

Példánkban:

$$[a_0, abba, S] \underset{\mathcal{A}}{\vdash} [a_0, abba, SS] \underset{\mathcal{A}}{\vdash} [a_0, abba, abS] \underset{\mathcal{A}}{\vdash} [a_0, bba, bS] \underset{\mathcal{A}}{\vdash} [a_0, ba, S] \underset{\mathcal{A}}{\vdash}$$

$$\underset{\mathcal{A}}{\text{—}\!\dashv}[a_0, ba, ba] \underset{\mathcal{A}}{\text{—}\!\dashv}[a_0, a, a] \underset{\mathcal{A}}{\text{—}\!\dashv}[a_0, \varepsilon, \varepsilon].$$

„$\subseteq$”:

Azt kell belátni, hogy tetszőleges $\mathcal{V} = \langle A, T, \Sigma, \delta, a_0, \sigma_0 \rangle$ 1-veremhez létezik olyan $G \in \mathcal{G}_{\text{kit2}}$, melyre $L(G) = L^{\varepsilon}(\mathcal{V})$.

Tudjuk, hogy $\mathcal{V}$ akkor és csak akkor fogadja el az u szót, ha $[a_0, u, \sigma_0] \overset{*}{\underset{\mathcal{V}}{\text{—}\!\dashv}}[b, \varepsilon, \varepsilon]$. Tehát miközben $\mathcal{V}$ elolvasta az inputot, és átment az a_0 állapotból a b állapotba, a verem feldolgozta a σ_0 jelet.

Definíció:

Ezt a folyamatot nevezzük σ_0 feldolgozásának. Egy feldolgozás adatai:

u input, a tulajdonképpeni vezérlés,

a_0 kiindulóállapot,

b végállapot.

G nyelvtani jelei ilyen feldolgozások lesznek. Egy nyelvtani jel alapvetőleg három információt fog tartalmazni: honnan indult, hová érkezett, és mit dolgozott fel. Tehát az $[a, u, \sigma] \overset{*}{\underset{\mathcal{V}}{\text{—}\!\dashv}}[b, \varepsilon, \varepsilon]$ feldolgozásnak az (a, σ, b) nyelvtani jelet feleltetjük meg úgy, hogy a következő feltétel teljesüljön:

$$(a, \sigma, b) \overset{*}{\underset{G}{\longrightarrow}} u \iff [a, u, \sigma] \overset{*}{\underset{\mathcal{V}}{\text{—}\!\dashv}}[b, \varepsilon, \varepsilon].$$

Tehát a feldolgozások lesznek a nyelvtani jelek, azaz

$$N := \{(a, \sigma, b) \mid a, b \in A, \sigma \in \Sigma\} \cup \{S\}.$$

Mik legyenek a nyelvtan szabályai?

1) $S \longrightarrow (a_0, \sigma_0, b)$ minden $b \in A$ esetén, azaz el kell tudni jutni az S jelből feldolgozásokba.
2) Egylépéses feldolgozás esetén $u = t, t \in T \cup \{\varepsilon\}$, tehát t hatására σ-t a veremből kivesszük:
   — $(a, \sigma, b) \longrightarrow t \in \mathcal{P} \iff \delta(a, t, \sigma) \ni (b, \varepsilon), t \in T \cup \{\varepsilon\}$.
3) Nem egylépéses feldolgozás esetén a következő szabályokat vesszük fel:
   — $(a, \sigma, b) \longrightarrow t(c_0, \sigma_1, c_1)(c_1, \sigma_2, c_2) \dots (c_{k-1}, \sigma_k, c_k)$, ahol $c_k = b$,
   — $(c_0, \sigma_1 \dots \sigma_k) \in \delta(a, t, \sigma)$, és $c_1, \dots, c_{k-1} \in A$ tetszőlegesek, $|A|^{k-1}$ új szabály definiáltunk.

Ezek után bizonyítanunk kell, hogy ez tényleg jó nyelvtan lesz, azaz $L(G) = L^{\varepsilon}(\mathcal{V})$, ahol $G = \langle \{S\} \cup \{(a, \sigma, b) \mid a, b \in A, \sigma \in \Sigma\}, T, \mathcal{P}, S \rangle$ és ahol $\mathcal{P}$ a most definiált szabályokból áll. (Az így definiált levezetés a feldolgozásnak felel meg.)

Állítás:

Tetszőleges $u \in T^*$ esetén $(a, \sigma, b) \overset{*}{\underset{G}{\longrightarrow}} u$ akkor és csak akkor, ha $[a, u, \sigma] \overset{*}{\underset{\mathcal{V}}{\text{—}\!\dashv}}[b, \varepsilon, \varepsilon]$.

Bizonyítás:

Teljes indukcióval látjuk be.

„$\Longrightarrow$”:

A levezetés hossza szerint mutatjuk meg, hogy minden levezetés egy feldolgozásnak felel meg. Mivel $u \in T^*$, ezért a legrövidebb levezetés 1 hosszúságú lehet. Ekkor $u \in T \cup \{\varepsilon\}$ esetén

$$(a, \sigma, b) \underset{G}{\overset{(1)}{\longrightarrow}} u \in T \cup \{\varepsilon\} \iff (b, \varepsilon) \in \delta(a, u, \sigma) \iff [a, u, \sigma] \underset{\nu}{\overset{(1)}{\dashv\!\!-\!\!-}} [b, \varepsilon, \varepsilon].$$

Most tegyük fel, hogy a h hosszú levezetésekre az állítás igaz, és vizsgáljuk meg az $(a, \sigma, b) \underset{G}{\overset{(h+1)}{\longrightarrow}} u \in T^*$ levezetést $u \in T^*$ esetén. Az nyilvánvaló, hogy ekkor biztos lesz legelső lépés. De ez a lépés csak a 3) pontban definiált szabállyal lehetséges, tehát az első lépés a következő volt:

$$(a, \sigma, b) \underset{G}{\overset{(1)}{\longrightarrow}} t(c_0, \sigma_1, c_1) \dots (c_{k-1}, \sigma_k, c_k), \qquad (*)$$

ahol $c_k = b$, továbbá $(c_0, \sigma_1 \sigma_2 \dots \sigma_k) \in \delta(a, t, \sigma)$.

Mivel G 2-es típusú nyelvtan, ezért az u szó felbontható $u = t_1 v_1 v_2 \dots v_k$ alakra, ahol

$$(c_0, \sigma_1, c_1) \underset{G}{\overset{\leq h}{\longrightarrow}} v_1, \ \dots \ , (c_{k-1}, \sigma_k, c_k) \underset{G}{\overset{\leq h}{\longrightarrow}} v_k.$$

Ezekre alkalmazzuk az indukciós feltételt, azaz

$$[c_0, v_1, \sigma_1] \underset{\nu}{\overset{*}{\dashv\!\!-\!\!-}} [c_1, \varepsilon, \varepsilon], \dots, [c_{k-1}, v_k, \sigma_k] \underset{\nu}{\overset{*}{\dashv\!\!-\!\!-}} [c_k, \varepsilon, \varepsilon].$$

Most „rakjuk össze” ezeket a működéseket, így kapjuk, hogy

$$[c_0, v_1 \dots v_k, \sigma_1 \sigma_2 \dots \sigma_k] \underset{\nu}{\overset{*}{\dashv\!\!-\!\!-}} [c_k, \varepsilon, \varepsilon].$$

De nyilván az első lépés $(*)$ és a levezetés definíciója miatt

$$[a, t v_1 \dots v_k] \underset{\nu}{\overset{(1)}{\dashv\!\!-\!\!-}} [c_0, v_1 \dots v_k, \sigma_1 \dots \sigma_k] \underset{\nu}{\overset{*}{\dashv\!\!-\!\!-}} [c_k, \varepsilon, \varepsilon],$$

ahol természetesen $c_k = b$, amivel az állítást beláttuk.

„$\Longleftarrow$”:

Most feltesszük, hogy $[a, u, \sigma] \underset{\nu}{\overset{*}{\dashv\!\!-\!\!-}} [b, \varepsilon, \varepsilon]$, és a feldolgozás hossza szerinti indukcióval bizonyítjuk, hogy az (a, σ, b) nyelvtani jelből levezethető az $u \in T^*$ szó.

$h = 1$ hossz:

Egy jel feldolgozásra került, ezért világos, hogy a feldolgozás legalább 1 hosszú. Tehát feltehetjük fel, hogy $[a, u, \sigma] \underset{\nu}{\overset{(1)}{\dashv\!\!-\!\!-}} [b, \varepsilon, \varepsilon]$, és $u \in T \cup \{\varepsilon\}$, amiből nyilván következik, hogy $(b, \varepsilon) \in \delta(a, \sigma, b)$. Viszont a 2)-es szabálydefiníció alapján $(a, \sigma, b) \longrightarrow u \in \mathcal{P}$ és így levezethető az u szó.

$h+1$ hossz:

Most tegyük fel, hogy h hosszú feldolgozásra van levezetés, és lássuk be $h+1$-re. Legyen a feldolgozás $[a,u,\sigma] \overset{(h+1)}{\underset{\mathcal{V}}{\dashv}} [b,\varepsilon,\varepsilon]$. Mivel h legalább 1 volt, ezért ez nem egy közvetlen konfigurációátmenet, tehát létezik egy legelső lépése. Ezek alapján tehát az u szó felbontható $u=tv$ alakban, ahol $t \in T \cup \{\varepsilon\}$, és $v \in T^*$. Ekkor a legelső lépés külön felírva:

$$(**) \qquad [a,tv,\sigma] \overset{(1)}{\underset{\mathcal{V}}{\dashv}} [c_0,v,\sigma_1 \ldots \sigma_k] \overset{\leq h}{\underset{\mathcal{V}}{\dashv}} [b,\varepsilon,\varepsilon] \quad .$$

Nyilván a $\sigma_1 \ldots \sigma_k$ jelek teljesen feldolgozásra kerülnek, így létezik egy legelső időpont, amikor σ_1 eltűnik. Legyen az automata ebben az időpontban a c_1 állapotban. (Ekkor nyilván σ_2 van a verem tetején.) Folytassuk ezt a jelölést, és legyen az automata c_{k-1} állapotban abban az időpontban, amikor már csak σ_k van a verem tetején. Végül vezessük be a $c_k = b$ jelölést is. Jelölje $v_1, \ldots, v_k$ azokat a részszavakat, amiket a $c_0, c_1, \ldots, c_k$ állapotok között olvastunk. Ezen jelölések alapján nyilván igaz, hogy

$$[c_0,v_1,\sigma_1] \overset{\leq h}{\underset{\mathcal{V}}{\dashv}} [c_1,\varepsilon,\varepsilon], \ \ldots \ , [c_{k-1},v_k,\sigma_k] \overset{\leq h}{\underset{\mathcal{V}}{\dashv}} [c_k,\varepsilon,\varepsilon].$$

Ezekre már alkalmazhatjuk az indukciós feltételt, miszerint

$$\forall i = 1,\ldots,k : (c_{i-1},\sigma_i,c_i) \overset{*}{\underset{G}{\longrightarrow}} v_i.$$

Viszont a legelső konfigurációátmeneti lépés $(**)$ miatt igaz, hogy

$$(a,\sigma,b) \longrightarrow t(c_0,\sigma_1,c_1)\ldots(c_{k-1},\sigma_k,c_k) \in \mathcal{P}.$$

Innen nyilván már következik az állítás, mivel $u = tv_1v_2\ldots v_k$.

$\triangle$

Most térjünk vissza a tétel bizonyításához. Az előbbi állítás és az 1)-es szabálydefiníció alapján könnyen látszik, hogy

$$S \overset{*}{\underset{G}{\longrightarrow}} u \iff \exists b \in A : [a_0,u,\sigma_0] \overset{*}{\underset{\mathcal{V}}{\dashv}} [b,\varepsilon,\varepsilon] \iff u \in L^{\varepsilon}(\mathcal{V}).$$

Tehát ez alapján a $\mathcal{V}$ automatához tudunk adni $G \in \mathcal{G}_{\mathbf{kit2}}$ nyelvtant, melyre $L(G) = L^{\varepsilon}(\mathcal{V})$.

$\triangle$

15. Determinisztikus 1-vermek

Láttuk, hogy a determinisztikus 0-vermek ugyanazt tudják, mint a nemdeterminisztikus 0-vermek. Kérdés az, hogy mit mondhatunk 1-vermek esetén.

Emlékeztető:

Az 1-verem determinisztikus, ha minden konfigurációjának legfeljebb 1 átmenete (r´akövetkezője) van. Ez formálisa leírva:

a) $\forall a \in A, \sigma \in \Sigma, t \in T \cup \{\varepsilon\} : |\delta(a, t, \sigma)| \leq 1.$

b) $\forall a \in A, \sigma \in \Sigma : |\delta(a, \varepsilon, \sigma)| \neq 0 \implies \forall t \in T : |\delta(a, t, \sigma)| = 0.$

15.1. Veremautomaták normálformái

Definíció:

A $\mathcal{V}$ 1-verem **1-es normálformájú**, ha $(b, \gamma) \in \delta(a, t, \sigma)$ esetén $\gamma \in \{\varepsilon\} \cup \{\sigma\} \cup (\Sigma\{\sigma\})$.

Megjegyzés: Ez a normálforma tulajdonképpen a klasszikus verem megvalósítója, ugyanis

— $\gamma = \varepsilon$ a *pop* művelet,

— $\gamma = \sigma$ a *top* művelet,

— $\gamma \in (\Sigma\{\sigma\})$ a *push* művelet (itt a $\Sigma\{\sigma\}$ a komplexusszorzatot jelöli).

Definíció:

A $\mathcal{V}$ 1-verem **2-es normálformájú**, ha 1-es normálformájú, és ha bármely $[a, u, \alpha]$ termináló konfiguráció esetén $u = \varepsilon$. (Azaz ha az automata megáll, akkor biztosan végigolvasta az inputot.)

Definíció:

A $\mathcal{V}$ 1-verem **3-as normálformájú**, ha 2-es normálformájú, és biztosan mindig terminál.

Megjegyzés: most csak végállapottal történő elfogadással foglalkozunk.

Lemma:

Tetszőleges $\mathcal{V}$ 1-veremautomatához létezik 1-es normálformájú $\mathcal{V}'$ 1-veremautomata, amire $L(\mathcal{V}') = L(\mathcal{V})$.

Bizonyítás:

$\mathcal{V}'$ csak kevésben fog eltérni $\mathcal{V}$-től. Mik lesznek ezek az eltérések? Világos, hogy a normálforma szempontjából rossz átmeneteket kell megkeresni, és ezeket megfelelőkkel szimulálni. Hogy néznek

ki ezek a rossz átmenetek? Nyilván így:

$$(b, \sigma_k\sigma_{k-1}\ldots\sigma_1) \in \delta(a, t, \sigma).$$

Ennek szimulációja 1-es normálformájú működésekkel, ahol $c_0, c_1, \ldots, c_{k-1}$ új állapotok:

$$\delta'(a, t, \sigma) = (c_0, \varepsilon),$$

$$\delta'(c_0, \varepsilon, \sigma') = (c_1, \sigma_1\sigma') \text{ minden } \sigma' \in \Sigma \text{ esetén},$$

$$\delta'(c_1, \varepsilon, \sigma_1) = (c_2, \sigma_2\sigma_1),$$

$$\vdots$$

$$\delta'(c_{k-1}, \varepsilon, \sigma_{k-1}) = (b, \sigma_k\sigma_{k-1}).$$

Ezek nyilván már 1-es normálformájúak.

Végül még az elindulást kell biztosítanunk a $\delta'(a'_0, \varepsilon, \sigma'_0) = (a_0, \sigma_0\sigma'_0)$ átmenettel, hogy az eredeti működéssorozat elvégezhető legyen, ahol a'_0 az új kezdőállapot, σ'_0 pedig a verem új kezdőszimbóluma.

△

Lemma:

Tetszőleges $\mathcal{V}$ 1-veremautomatához létezik $\mathcal{V}'$ 2-es normálformájú 1-veremautomata, melyre $L(\mathcal{V}') = L(\mathcal{V})$.

Bizonyítás:

Az előző lemma értelmében feltehető, hogy $\mathcal{V}$ már 1-es normálformában adott.

Ekkor gond a termináló konfigurációval lehet, ugyanis lehet ugy termináló, hogy $[a, u, \alpha]$ esetén $u \neq \varepsilon$. De ekkor nyilván — mivel egy jel biztos van u-ban — létezik $u = tv$ felbontása, ahol $t \in T$, és $v \in T^*$.

A veremtartalom szerint két eset lehetséges. Vagy $\alpha = \varepsilon$, de ekkor új veremkezdő-szimbólum bevezetése a megoldás (feltehető, hogy $\mathcal{V}$ már eleve ilyen), vagy $\alpha \neq \varepsilon$. Ekkor legyen $\alpha = \sigma\alpha'$, ahol $\sigma \in \Sigma$. A problémát az jelenti, ha $\delta(a, t, \sigma) = \emptyset$ minden $t \in T$-re, és $\delta(a, \varepsilon, \sigma) = \emptyset$. (Ez jelenti a terminálást ebben az állapotban.) Megoldásként bevezetünk egy új d állapotot, és minden ilyen $a \in A$ és $\sigma \in \Sigma$ esetén hozzávesszük a következő átmeneteket:

$$\delta'(a, t, \sigma) := (d, \sigma),$$

$$\delta'(d, t, \sigma') := (d, \sigma') \quad \text{minden } t \in T\text{-re és } \sigma' \in \Sigma\text{-ra}.$$

Ekkor a d állapotban végig lehet olvasni az inputot. Persze nyilván $d \notin F'$.

Tehát kaptunk egy $\mathcal{V}'$ 2-es normálformájú veremautomatát $\mathcal{V}$-ből, melyre $L(\mathcal{V}') = L(\mathcal{V})$.

△

Megjegyzés: Mindkét lemma esetében nyilvánvaló, hogy ha a $\mathcal{V}$ veremautomata determinisztikus volt, akkor $\mathcal{V}'$ is az lesz.

Lemma:

Tetszőleges $\mathcal{V}$ determinisztikus 1-veremautomatához létezik $\mathcal{V}'$ 3-as normálformájú determinisztikus 1-veremautomata, melyre $L(\mathcal{V}') = L(\mathcal{V})$.

Bizonyítás:

Az előző lemma értelmében feltehető, hogy a $\mathcal{V}$ 1-verem 2-es normálformában adott. Ekkor az egyetlen probléma, ha létezik végtelen konfigurációátmenet-lánc.

Jelölés: $[a, u, \alpha] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty$.

Tegyük fel, hogy $[a, u, \alpha] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty$. Ekkor tudunk találni a működés során olyan állapotot, ahol már vagy elolvasta az input szót, vagy már nem olvas többet. Legyen ez a b állapot: $[b, \varepsilon, \beta] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty$. Tegyük fel, hogy ekkor $\beta := \sigma_1 \dots \sigma_k$. Mivel $\mathcal{V}$ 2-es normálformájú, ezért 1-es is, tehát σ_k sose kerül ki a veremből. (Ugyanis a *pop* ki tudja venni a veremből, de az automata üres veremmel nem működik.)

Legyen j_0 a legkisebb indexű olyan σ_j, mely a működés során végig a veremben van, azaz $j_0 := \min\{j \mid \sigma_j \text{ mindig a veremben van}\}$. Nyilván ekkor létezik egy olyan időpillanat, amikor ez a σ_{j_0} lesz a verem tetején. Legyen ennek az időpillanatnak a konfigurációja $[c_0, \varepsilon, \sigma_{j_0}\sigma_{j_0+1} \dots \sigma_k]$. Tudjuk, hogy σ_{j_0} sose kerül ki a veremből, tehát az utána következő $\sigma_{j_0+1}, \dots, \sigma_k$ jeleket el is hagyhatjuk, tehát

$$[c_0, \varepsilon, \sigma_{j_0}] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty.$$

Tehát ezeket az átmeneteket kell átdefiniálni.

$$\delta'(a, \varepsilon, \sigma) := \begin{cases} \delta(a, \varepsilon, \sigma), & \text{ha } [a, \varepsilon, \sigma]\text{-ből indulva nem létezik végtelen konfigurációátmenet-lánc,} \\ (d, \sigma), & \text{ha } [a, \varepsilon, \sigma] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty, \text{ és közben nem érint végállapotot,} \\ (f, \sigma), & \text{ha } [a, \varepsilon, \sigma] \overset{*}{\underset{\mathcal{V}}{\dashv}} \infty, \text{ és közben érint végállapotot.} \end{cases}$$

Itt d, f új állapotok, és $F' := F \cup \{f\}$. Ekkor d ismét csak az input szó végigolvasását végzi el.

$$\delta'(d, t, \sigma') := (d, \sigma') \text{ minden } \sigma' \in \Sigma\text{-ra.}$$

Egyetlen probléma még az f végállapotnál maradt, ugyanis lehet, hogy itt még nem olvastuk végig az input szót. Ezért ha van még jel az inputon, azt a szót se fogadhatjuk el, azaz

$$\delta'(f, t, \sigma') := (d, \sigma') \text{ minden } \sigma \in \Sigma\text{-ra.}$$

Az így definiált $\mathcal{V}'$ veremautomára tényleg $L(\mathcal{V}') = L(\mathcal{V})$, és ez a $\mathcal{V}'$ is determinisztikus lesz.

$\triangle$

Definíció:

A $\mathcal{V}$ veremautomatát inputterminátoros veremautomatának nevezzük, ha az inputja végén van egy vége jel (EOF), amit # jelöl, és δ a # jelre is definiálva van. Ekkor

$$L(\mathcal{V}) = \{u \in T^* \mid [a_0, u\#, \sigma_0] \overset{*}{\underset{\mathcal{V}}{\vdash}} [f, \varepsilon, \alpha] \text{ valamely} f \in F\text{-re}\}.$$

Jelölése: $\mathcal{L}_{\mathrm{It1}\mathcal{V}} = \{L(\mathcal{V}) \mid \mathcal{V}$ inputterminátoros 1-verem $\}$.
Megjegyzés: ha $\mathcal{V}$ determinisztikus, akkor jelölésben $\mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$-et használunk.

Könnyen belátható, hogy $\mathcal{L}_{\mathrm{It1}\mathcal{V}}$=$\mathcal{L}_{1\mathcal{V}}$.

Tegyük fel, hogy $\mathcal{V}$ egy inputterminátoros veremautomata, $\mathcal{V}'$ pedig egy nem inputterminátoros veremautomata. Ekkor ha $(b, \alpha) \in \delta(a, \#, \sigma)$, akkor $(b, \alpha) \in \delta'(a, \varepsilon, \sigma)$, és így $\mathcal{V}'$ már nem biztos, hogy determinisztikus.

De ugyanakkor fordítva már igaz, hogy egy $\mathcal{V}$ veremautomatából tudunk csinálni egy $\mathcal{V}'$ inputterminátoros veremautomatát $\delta'(f, \#, \sigma) := (f^{(2)}, \sigma)$ átmenetek segítségével (minden $\sigma \in \Sigma$-ra), ahol $f^{(2)}$ új végállapot.

Tehát most kimutattuk, hogy $\mathcal{L}_{D1\mathcal{V}} \subseteq \mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$. Összefoglalva:

$$\mathcal{L}_{D1\mathcal{V}} \subseteq \mathcal{L}_{\mathrm{DIt1}\mathcal{V}} \subseteq \mathcal{L}_{\mathrm{It1}\mathcal{V}} = \mathcal{L}_{1\mathcal{V}} = \mathcal{L}_2.$$

Célunk az, hogy kimutassuk, $\mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$ zárt a komplementer képzésre. Ugyanis $\mathcal{L}_2$ nem zárt a komplementer képzésre, s így belátnánk a valódi tartalmazást ($\mathcal{L}_{\mathrm{DIt1}\mathcal{V}} \subsetneq \mathcal{L}_2$).

Lemma:

$\mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$ zárt a komplementer műveletre.

Bizonyítás:

Legyen $L \in \mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$ tetszőleges nyelv. Be kell látni, hogy ekkor $\overline{L} \in \mathcal{L}_{\mathrm{DIt1}\mathcal{V}}$. Legyen $\mathcal{V} = \langle A, T, \Sigma, \delta, a_0, \sigma_0, F, \# \rangle$ az L nyelvet elfogadó inputterminátoros 1-verem, melyre $L(\mathcal{V}) = L$.

A korábbi lemmák alapján feltehető, hogy $\mathcal{V}$ 3-as normálformában adott, azaz ha tekintünk egy adott $u\#$ inputot, akkor erre $\mathcal{V}$ mindig megáll, valamint az inputot mindig végigolvassa. Miután elolvasta a # jelet, tesz még néhány ε-lépést, majd megáll.

Ha tehát tekintünk egy tetszőleges $u \notin L(\mathcal{V})$ szót, akkor $\mathcal{V}$ az u szó elolvas sa ut n az ε-lépések során nem érint végállapotot (legyen ez a ' tulajdonság), ha pedig $u \in L(\mathcal{V})$, akkor nyilván u szó elolvas sa ut n az ε-lépések során érint végállapotot (legyen ez a " tulajdonság).

Most próbáljuk megkonstruálni a $\overline{\mathcal{V}}$ determinisztikus, inputterminátoros 1-vermet. Ez lényegét tekintve $\mathcal{V}$-vel azonos, eltérés csak a # jel elolvasása után van.

Legyen $\overline{\mathcal{V}} = \langle A \cup A' \cup A'' \cup \{a_{\text{vég}}\}, T, \Sigma, \delta', a_0, \sigma_0, \{a_{\text{vég}}\}, \# \rangle$, ahol $A' = \{b' \mid b \in A\}$ és $A'' = \{b'' \mid b \in A\}$. Ekkor # elolvasására δ' definíciója:

$$\delta'(a, \#, \sigma) := \begin{cases} (b', \alpha), & \text{ha } \delta(a, \#, \sigma) = (b, \alpha), \text{ és } b \notin F, \\ (b'', \alpha), & \text{ha } \delta(a, \#, \sigma) = (b, \alpha), \text{ és } b \in F \end{cases}.$$

Valamint

$$\delta'(a',\varepsilon,\sigma) := \begin{cases} (b',\alpha), & \text{ha } \delta(a,\varepsilon,\sigma) = (b,\alpha) \text{ és } b \notin F, \\ (b'',\alpha), & \text{ha } \delta(a,\varepsilon,\sigma) = (b,\alpha) \text{ és } b \in F \end{cases}.$$

Tehát ha megérkezünk $\mathcal{V}$-ben az első végállapotba $\overline{\mathcal{V}}$ a b'' llapotba megy át. Tehát legyen

$$\delta'(a'',\varepsilon,\sigma) := (b'',\alpha), \text{ ha } \delta(a,\varepsilon,\sigma) = (b,\alpha).$$

Ha $\overline{\mathcal{V}}$ egy b' llapotban áll meg, akkor nyilván $u \notin L(\mathcal{V})$, és ha egy b'' llapotban, akkor $u \in L(\mathcal{V})$. Amikor $\overline{\mathcal{V}}$ egy b' llapotba megy, nyilván végállapotba kell mennünk. De itt az eredeti δ nincs értelmezve, tehát δ'-t defini lhatjuk így:

$$\delta'(a',\varepsilon,\sigma) := (a_{\text{vég}},\sigma), \text{ ha } \delta(a,\varepsilon,\sigma) = \emptyset.$$

Nyilván erre a konstrukcióra igaz, hogy $L(\overline{\mathcal{V}}) = \overline{L(\mathcal{V})} = \overline{L}$. △

Foglalkozhatnánk még a 2-vermekkel, illetve az ezeknél nagyobb veremszámú automatákkal. Ezek már minden $\mathcal{L}_0$-beli nyelvet el tudnak fogadni, azaz az 1-vermekhez képest már két osztálynyit ugrottunk. További érdekesség, hogy az 1-vermekkel ellentétben a determinisztikus 2-vermek ugyanazt a nyelvosztályt adják, mint a nemdeterminisztikus 2-vermek.

16. Elemzési módszerek

A most következő fejezetben az eddigiekben tárgyaltak egyik gyakorlati alkalmazását vizsgáljuk meg, és itt kapcsolódunk be a fordítóprogramok elméletének vizsgálatába. Nem célunk, hogy teljes részletességgel tárgyaljuk az egyes módszereket, hiszen ez már az automaták és formális nyelvek keretein túlmutat.

16.1. Az elemzési módszerek célja

A fordítóprogramok az inputként kapott karaktersorozatot először az adott nyelvtan szabályai szerint analizálják. Ezt persze több lépésben teszik, melyre már volt néhány utalás az előző fejezetek során. Az analízis a lexikális elemzés során felismeri a programnyelv lexikális egységeit és adott szimbólumokkal helyettesíti. Ez a művelet 3-as típusú (reguláris) nyelvtan segítségével írható le, mely könnyen implementálható, ezért ezzel a továbbiakban nem foglalkozunk. A szintaktikus elemző a lexikálisan elemzett szimbólumsorozatot szerkezetileg ellenőrzi egy 2-es típusú nyelvtan felhasználásával. Ennek az elemzési fázisnak rendszerint egy szintaxisfa a kimenete. Az analízis utolsó lépésében pedig szemantikusan ellenőrzi a felépített szintaxisfát. Mi már nem foglalkozunk a szemantikus ellenőrzéssel e fejezetben.

Nézzük tehát, mi is az elemzők célja. Adott egy $G = \langle T, N, \mathcal{P}, S \rangle \in \mathcal{G}_{\text{kit2}}$ és egy $u \in T^*$ szó. Az elemzési algoritmusok feladata azt eldönteni, hogy u szó eleme-e $L(G)$-nek. Ha igen, akkor további feladata, hogy előállítsa a szintaxisfát, ellenkező esetben hibajelzéssel leálljon. A szintaxisfát azért célszerű felépíteni, mivel ebből könnyen megadható az inputként megadott program jellemzése, szerkezete.

16.2. Módszerek és osztályozásaik

Az elemzési módszerek célja tehát, hogy létezik-e $S \xrightarrow[G]{*} u$, ahol S az adott G nyelvtan kezdőszimbóluma, és u szó az input szimbólumsorozat. Ennek eldöntésére többféle módszer létezik, melyeket a szintaxisfa felépítésének iránya szerint osztályozhatunk. Az alábbiakban a két legnagyobb csoporttal fogunk foglalkozni:

a) felülről lefelé (Top-down), és

b) alulról fölfelé (Bottom-Up).

Az elemzők másik szempont szerinti osztályozása lehet az input karaktersorozat kezelésének módszere. Ez általában meghatározott, hiszen például egy programszöveg is csak szekvenciális fileban érhető el a legtöbb esetben, így u szót csak balról jobbra lehet elolvasni. Ezeket az elemzőket nevezzük *Left to right* elemzőknek.

Továbbá osztályozhatjuk az algoritmusokat aszerint is, hogy determinisztikus, illetve nemde-

terminisztikus algoritmusok-e. A nemdeterminisztikus algoritmusoknak viszont csak elvi jelentőségük van, hiszen a gyakorlatban használhatatlanok. Ellenben ezen algoritmusokból programtranszformációk segítségével már deterministikus programok hozhatók létre.

16.3. Felülről lefelé módszerek

Ezek a módszerek a szintaxisfát a gyökértől (S kezdőszimbólumtól) kezdve építik fel.

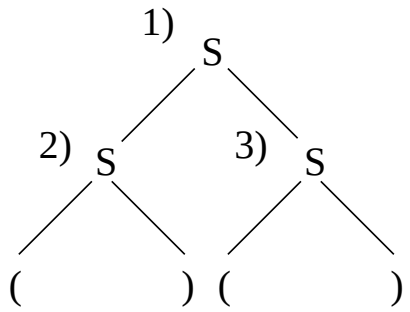
Először nézzük egy példát erre a módszerre. Legyen $T = \{(,)\}$, $N = \{\mathrm{S}\}$, $u = ()()$, és legyen

$G_1 = \langle T, N, \mathcal{P}, \mathrm{S} \rangle$, ahol

$\mathcal{P} = \{S \longrightarrow SS \mid (S) \mid ()\}$.

Ekkor a szintaxisfa felépítése S kezdőszimbólumból:

S$\longrightarrow$SS$\longrightarrow$ ()S$\longrightarrow$ ()(), azaz:



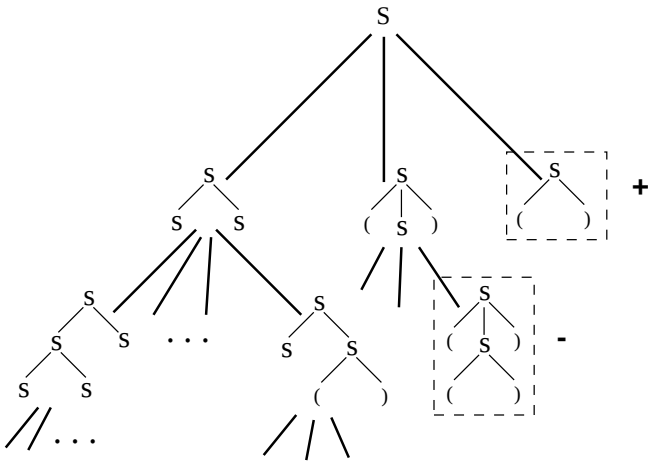
16.5. Ábra: Top-Down elemző működése

Most nézzünk meg egy nem deterministikus Top-Down elemzőt részletesebben. A struktogramban használt t változó most egy szintaxisfát jelent és felhasználjuk a köv(t) függvényt is, mely az összes t szintaxisfából egy szabály alkalmazásával származtatható új szintaxisfának a halmazát adja.

$t := S$	
köv$(t) \neq \emptyset$	
$t :\in$ köv(t)	
$front(t) = u$	Különben
Exit$(igen, t)$	Exit(nem)

Annak érdekében, hogy használható legyen ez az algoritmus, determinisztikussá kell tenni. Egy nemdeterminisztikus program mindig szemléltethető a saját lefutásfájával. Ez tulajdonképpen az

összes lehetőség felvételét jelenti a fába. A jelenlegi algoritmus lefutásfájában minden pont egy szintaxisfa lesz. Nézzük meg, mi lesz a szintaxisfája a konkrét példánknak:



16.6. Ábra: Példa a Top-Down elemző lefutási fájára

Ebben a lefutási fában egy levél „+" cimkét kap, ha megoldja a feladatot, ellenkező esetben „-" cimkét kap. Így tehát „+" cimkéjű leveleket keresünk a fában. Tehát a determinisztikussá transzformálás algoritmusa a lefutási fa valamilyen bejárása. Ebben az esetben ha érintünk „+" cimkét, akkor *igen*, különben pedig *nem* választ ad.

Fontos kérdés, hogy milyen bejárást alkalmazzunk. Tekintsük először a szintfolytonos bejárást. Mivel minden pontnak csak véges számú leágazása lehet, ezért egyszer biztos megtalálja a „+" cimkét, ha van ilyen. Különben vagy nem terminál (végtelen magasságú fa esetén), vagy *nem* választ ad.

Meg kell vizsgálni, hogy mennyi lehet így az algoritmus műveletigénye. Ezt $l(u)$ függvényében adhatjuk meg. Mivel minden nyelvtani jelre van legalább két alkalmazható szabály, ezért a lefutási fa legalább bináris. (Ugyanis ha csak egy szabály létezne, azt el is hagyhatnánk.) Legyen K az összes szabály jobboldalát tekintve a leghosszabb helyettesítés hossza. Ekkor biztosan kell legalább $\frac{l(u)}{K}$ levezetési lépés. Ettől csak nagyobb lehet a „+" cimkejű pont magassága: $h(t) \geq \frac{l(u)}{K}$. Ekkor a bejárt pontok száma viszont legalább $2^{\frac{l(u)}{K}}$, mert a lefutási fa legalább bináris. Így viszont exponenciális az időbonyolultsága a szintfolytonos bejárási algoritmusnak. A gyakorlatban ez így tulajdonképpen használhatatlan.

Most nézzük meg, **preorder** bejárás esetén mit mondhatunk. Ez már tulajonképpen a visszalépéses keresés algoritmusa lesz. Probléma viszont ezzel az algoritmussal, hogy nem minden esetben találná meg a „+" cimkét. (Az előbbi példa ilyen.) A „+" cimke megtalálása csak akkor garantált, ha a fa véges. Persze megoldásként megtehetjük, hogy a re-

kurzív szabályokat a fa „végére" helyezzük, ezáltal sokkal nagyobb eséllyel fogjuk megtalálni a „+" cimkéket. A lefutási idő *nem* válasz esetén itt is exponenciális, de *igen* válasz esetén már lehet polinomiális is az időbonyolultság.

Meg kell vizsgálni, milyen módszerekkel lehet az időbonyolultságot csökkenteni.

— Csökkentjük a lefutási fa szélességét bizonyos feltételek kikötésével a levezetésekkel szemben.

— Vágási kritériumokat határozunk meg, azaz ha egy tulajdonság a lefutási fa egy adott pontjára fennáll, akkor az abból származtatott pontokra is fenn fog állni.

Az előző problémák és még néhány, most nem említett probléma miatt az elemzők csak bizonyos feltételeknek megfelelő nyelvtanok esetén használhatók. Ezek a következő feltételek:

a) Egyértelműség
   — Ez nyilvánvaló, hiszen ellenkező esetben két szintaxisfát rendelne egy adott input szóhoz.

b) redukáltság

c) ciklusmentesség
   — A végtelen lefutási fa elkerülése érdekében követeljük meg.

d) ε-mentesség

— Ezt nem minden esetben fogjuk megkövetelni.

16.4. $LL(k)$ nyelvtanok

16.5. Alulról fölfelé módszerek

16.6. $LR(k)$ nyelvtanok

16.7. Kapcsolat $\mathcal{L}_{\mathrm{DIt1}\nu}$ és az $LR(k)$ nyelvek között

Jelöljül az $LR(k)$ nyelvtanok által generált nyelvek halmazát $\mathcal{L}_{LR(k)}$-val.

Tétel:

$\mathcal{L}_{\mathrm{DIt1}\nu} = \mathcal{L}_{LR(k)}$

Ezt a tételt nem bizonyítjuk.

Definíciójegyzék

Tartalomjegyzék