

11. Formális nyelvek

Formális nyelvtanok és a Chomsky-féle nyelvosztályok



Alapfogalmak

Ábécé, szó

- **Ábécének** nevezzük egy tetszőleges véges szimbólumhalmazt.
- Az ábécé elemeit **betűknek** hívjuk.
- Egy V ábécé elemeiből képzett *véges sorozatokat* V feletti **szavaknak**.
- Adott $v \in V$ **szó hossza** a benne szereplő betűk száma. Jelölése: $l(v)$ vagy $|v|$.
- A V ábécé feletti ε szót **üres szónak** nevezzük, ha $l(\varepsilon) = 0$.
- $V^0 := \{\varepsilon\}$
- $V^n := V \times V^{n-1}$, $n \geq 1$
- A V ábécé feletti szavak halmazát (beleértve az üres szót is) V^* -gal jelöljük.
 $V^* := \bigcup_{i=0}^{\infty} V^i = V^0 \cup V^1 \cup \dots$ a V **ábécé lezártja**.
- A nemüres szavak halmazát V^+ -szal jelöljük ($V^+ := V^* \setminus \{\varepsilon\}$).
 $V^+ := \bigcup_{i=1}^{\infty} V^i = V^1 \cup V^2 \cup \dots$ a V **ábécé pozitív lezártja**.

Példa: $V := \{a, b\}$

$V^* := \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$

$V^+ := \{a, b, aa, ab, ba, bb, aaa, aab, \dots\}$

Szavak konkatenációja

Legyenek u és v szavak egy V ábécé felett. Ekkor a két szó **konkatenációjának** nevezzük azt a szimbólumsorozatot, amelyet a két szó szimbólumainak egymás után fűzésével kapunk. Jelölése: uv .

Példa: $V = \{a, b, c\}$, $u = abb$, $v = cbb$. Ekkor $uv = abbcbb$.

A konkatenáció asszociatív, de nem kommutatív művelet, melynek egységeleme ε .

Így V^* a konkatenációval mint művelettel és ε -nal egységelemes félcsoportot alkot.

- V^* zárt a konkatenációra.

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

- ε egységelem V^* -gal és a konkatenációval:

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

Legyen $v \in V^*$

- $v^0 = \varepsilon$ (bármely szó nulladik hatványa az üres szó)
- $v^n = vv^{n-1}$ ($n \geq 1$) (bármely szó n . hatványa a szó n -szeres konkatenációja)

Egyéb definíciók

- $u, v \in V$. Az u szót a v **részzavának** nevezzük, ha $v = xuy$, ($x, y \in V$) teljesül. Ha még $xy \neq \varepsilon$, akkor u **valódi részzó**.

Példa: $v = aabbbcc \in V$ Az $u = abbbc$ valódi részzava v -nek.

- $v = xuy$ ($x, y, u, v \in V$). Ekkor:
 - Ha $x = \varepsilon$, akkor u -t a v szó **prefix**ének nevezzük
Példa: $v = aabbbcc$. Az $u = aabbb$ szó prefixe v -nek.
 - Ha $y = \varepsilon$, akkor u -t a v szó **szuffix**ének nevezzük
Példa: $v = aabbbcc$. Az $u = bbbcc$ szó szuffixe v -nek.
- Egy $u \in V$ szó tükörképe alatt a szimbólumai fordított sorrendben való felírását értjük. (Jelölése: u^{-1}) Legyen $u = a_1 \dots a_n$, $a_i \in V$, $1 \leq i \leq n$, ekkor

$$u^{-1} = a_n \dots a_1$$

◁ ♡

Formális nyelvtanok és a Chomsky-féle nyelvosztályok

Nyelv

A V^* valamely részhalmazát (2^{V^*} valamely elemét) a V ábécé feletti **nyelv**nek nevezzük. Jelölése: L . ($L \subset V^*$).

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

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

$$L_1 = \{a, b, \varepsilon\} \text{ véges nyelv, } L_2 = \{a^i b^i \mid 0 \leq i\} \text{ végtelen nyelv.}$$

Megjegyzés: A formális nyelv nem más, mint egy adott ABC jeleiből alkotott tetszőleges hosszú szavak halmazának részhalmaza, vagyis a formális nyelv egy adott ABC jeleiből alkotható, meghatározott szavak halmaza. A formális nyelv állhat véges sok szóból, állhat végtelen sok szóból és tartalmazhatja az üres szót is.

Nyelvek valamely összességét nyelvosztálynak, nyelvcsaládnak hívjuk. Jelölése: \mathcal{L}_i .

Nyelvekre vonatkozó műveletek

- $L_1 \cup L_2 = \left\{ u \mid u \in L_1 \vee u \in L_2 \right\}$: az L_1 és L_2 nyelv uniója
- $L_1 \cap L_2 = \left\{ u \mid u \in L_1 \wedge u \in L_2 \right\}$: az L_1 és L_2 nyelv metszete
- $L_1 - L_2 = \left\{ u \mid u \in L_1 \wedge u \notin L_2 \right\}$: az L_1 és L_2 nyelv különbsége
- $\bar{L} = V^* - L$: az $L \subseteq V^*$ komplementere
- $L_1 L_2 = \left\{ u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2 \right\}$: az L_1 és L_2 nyelv konkatenációja

Minden nyelvre fennáll: $\emptyset L = L \emptyset = \emptyset$ illetve $\{\varepsilon\} L = L \{\varepsilon\} = L$

- L^i : az L nyelv i -edik hatványa, és
 - $L^0 = \{\varepsilon\}$
 - $L^i = L^{i-1} L \quad (i \geq 1)$
- $L^* = \bigcup_{i \geq 0} L^i$: az L nyelv iteratív lezártja
- $L^+ = \bigcup_{i \geq 1} L^i$ nyelvet értjük

Az *unió*, *konkatenáció* és *iteráció lezárása* műveleteket *reguláris műveletek*nek nevezzük.

Nyelvekre vonatkozó leképezések

Legyen V_1, V_2 ábécé.

A $h : V_1 \rightarrow V_2$ leképezést **homomorfizmus**nak nevezzük, ha

$$h(uv) = h(u)h(v) \quad \forall u, v \in V_1^*$$

Valamint minden $u = a_1 a_2 \dots a_n$ szóra, ahol $a_i \in V$, $1 \leq i \leq n$ fennáll, hogy

$$h(u) = h(a_1)h(a_2) \dots h(a_n)$$

Legyen $h : V_1 \rightarrow V_2$ homomorfizmus. A h **homomorfizmus** ε -mentes, ha

$$h(u) \neq \varepsilon, \quad \forall u \in V_1^* \text{ szóra, ahol } u \neq \varepsilon$$

Egy h homomorfizmust **izomorfizmus**nak nevezzük, ha bármely $\forall u, v \in V_1^*$ szóra teljesül, hogy

$$h(u) = h(v) \Rightarrow u = v$$

Példa: izomorfizmus (decimális számok bináris reprezentációja):

$$V_1 = \{0, 1, 2, \dots, 9\}, \quad V_2 = \{0, 1\} \\ h(0) = 0000, \quad h(1) = 0001, \quad \dots, \quad h(9) = 1001$$

Nyelvek megadása

A formális nyelvek megadási módszereitől elvárjuk, hogy a leírás véges legyen.

A formális nyelv megadható:

1. felsorolással (csak véges nyelvek esetén).
2. a szavakat alkotó szabály szöveges leírásával.
Például: *Legyen L_2 a páratlan számok nyelve.*
3. a szavakat alkotó szabály matematikai leírásával.
Például: $V := \{a, b\}$ és legyen $L := \{u \mid u = a^n b^n \wedge n \in \mathbb{N}\}$
4. generatív grammatika segítségével.

Grammatika

A nyelveket megadhatjuk nyelvtanukkal is, vagyis egy szabályrendszerrel, aminek a felhasználásával a nyelv mondatai levezethetők. Egy generatív nyelvtan a jelsorozatok transzformációs szabályait leíró szabályok halmazából áll. A nyelvet alkotó jelsorozatok létrehozásához szükséges, hogy legyen egy egyedi „kezdő” szimbólum, ezután csak a szabályokat kell egymás után alkalmazni (bárhányszor, tetszés szerinti sorrendben) a kezdő szimbólum átalakítására. A nyelv azokból a jelsorozatokból áll, amelyeket az említett módon elő lehet állítani.

Tegyük fel például, hogy egy ábécéhez a 'a' és a 'b' szimbólumok tartoznak, a kezdő szimbólum pedig legyen az 'S' és adottak a következő szabályok:

- $S \rightarrow aSb$
- $S \rightarrow ba$

Kezdő szimbólumunk az 'S', de ezután kiválaszthatjuk, hogy melyik szabályt alkalmazzuk a jelsorozat következő elemének előállításához.

Példa:

- Ha az 1-es szabályt választjuk, akkor annak alapján az 'S' szimbólumot a 'aSb'-al helyettesítjük, eredményül tehát a "aSb" jelsorozatot kapjuk.
- Ha most ismételten az 1-es szabály alkalmazását választjuk, akkor helyettesítjük az 'S' szimbólumot a 'aSb'-vel, és akkor a "aaSbb" jelsorozatot kapjuk.
- Ezt az eljárást addig ismételhetjük, amíg az ábécé szimbólumai megengedik.
- Befejezve a példát, most válasszuk a 2-es szabályt, helyettesítsük az 'S' szimbólumot a 'ba' jelsorozattal, eredményül pedig a "aababb" jelsorozatot kapjuk, és ezzel be is fejeztük.

A nyelvtan által meghatározott nyelv nem lesz más, mint az összes olyan jelsorozat halmaza, amelyeket ezzel az eljárással elő tudunk állítani:

$$\{ba, abab, aababb, aaababbb, \dots\}$$

A G generatív nyelvtan egy $\langle N, T, \mathcal{P}, S \rangle$ négyest értünk, ahol:

- N és T diszjunkt ábécék, a nemterminális (N) és terminális (T) szimbólumok ábécéi.
- $S \in N$ a kezdőszimbólum.
- \mathcal{P} az (x, y) rendezett párok halmaza, ahol $x, y \in (N \cup T)^*$ és x legalább egy nemterminális szimbólumot tartalmaz.
A \mathcal{P} halmaz elemeit átírási szabályoknak nevezzük.

Megjegyzés: $N \cap T = \emptyset$ és az (x, y) jelölés helyett az $x \rightarrow y$ jelölést alkalmazzuk.

Példa: $N = \{S, B\}$, $T = \{a, b, c\}$
 \mathcal{P} :

$$(1) S \rightarrow aBS c$$

$$(2) S \rightarrow abc$$

$$(3) Ba \rightarrow aB$$

$$(4) Bb \rightarrow bb$$

És $S \in N$ nemterminális kezdőszimbólum.

Ekkor $L(G)$, a G grammatika által generált nyelvben generált néhány szimbólumsorozat.

$$S \rightarrow (2) abc$$

$$S \rightarrow (1) aBS c \rightarrow (2) a\mathbf{B}abcc \rightarrow (3) aa\mathbf{B}bcc \rightarrow (4) aabbcc$$

$$S \rightarrow (1) aBS c \rightarrow (1) aBaBScc \rightarrow (2) a\mathbf{B}aBabccc \rightarrow (3) aaB\mathbf{B}abccc \rightarrow (3) aa\mathbf{B}aBbccc \rightarrow (3) aaaB\mathbf{B}bcc \rightarrow (4) aaa\mathbf{B}bbccc \rightarrow (4) aaabbbccc$$

(zárójelben az adott lépésre alkalmazott szabály azonosítója, a helyettesíthető rész pedig kiemelt terminális, nem terminális).

A G grammatika által leírt nyelv: $L(G) := \{a^n b^n c^n \mid n > 0\}$.

Levezetés

Mondatforma: Mondatformának nevezzük terminális és nemterminális szimbólumok véges sorozatát: $(N \cup T)^*$.

Közvetlen levezetés

$G = \langle N, T, \mathcal{P}, S \rangle$, és $u, v \in (N \cup T)^*$. A v mondatforma közvetlenül (egy lépésben) levezethető u mondatformából G -ben, ha

$$\mathbf{u} = w_1 \mathbf{x} w_2, \quad v = w_1 \mathbf{y} w_2, \quad \text{ahol } \mathbf{x} \rightarrow \mathbf{y} \in \mathcal{P} \quad (w_1, w_2 \in (N \cup T)^*)$$

Jelölése: $u \xrightarrow{G} v$

Közvetett levezetés

Azt mondjuk, hogy a v mondatforma közvetetten levezethető az u mondatformából G -ben, ha létezik olyan $(\mathbb{N} \ni k \geq 0)$ szám és $x_0, x_1, \dots, x_k \in (N \cup T)^*$ mondatformák úgy, hogy

$$\begin{array}{l} u = x_0 \\ v = x_k \end{array} \quad \text{és} \quad x_i \xrightarrow{G} x_{i+1} \quad (1 \leq i \leq k-1)$$


Másképp: A v mondatforma levezethető az u mondatformából G -ben (jele: $u \xrightarrow{G}^* v$), ha $u = v$ vagy $\exists z \in (N \cup T)^*$ mondatforma, hogy $u \xrightarrow{G}^* z$ és $z \xrightarrow{G} v$

Generált nyelv

A G formális nyelvtan által generált nyelv szavai a kezdőszimbólumból levezethető szavak.

$L(G)$ a $G = \langle N, T, \mathcal{P}, S \rangle$ grammatika által generált nyelv, ha:

$$L(G) = \left\{ w \in T^* \mid S \xrightarrow{G}^* w \right\}$$

 **Ekvivalens nyelvtanok:** A G_1 és G_2 nyelvtanok ekvivalensek, ha $L(G_1) = L(G_2)$, azaz ugyanazt a nyelvet generálják. Jelölése: $G_1 \sim G_2$.

Kvázi ekvivalens nyelvtanok: A G_1 és G_2 nyelvtanok kvázi ekvivalensek, ha $L(G_1) \setminus \{\varepsilon\} = L(G_2) \setminus \{\varepsilon\}$, azaz legfeljebb az üres szó tartalmazásában különböznek. Jelölése: $G_1 \overset{\sim}{\underset{kv}{\sim}} G_2 \triangleleft \mathfrak{P}$

Generatív nyelvtanok osztályozása

A generatív nyelvtanok osztályozását a szabályok alakja alapján tehetjük meg.

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

- **Mondatszerű grammatika** ($i=0$)

Tetszőleges, azaz $p\mathbf{A}q \rightarrow \mathbf{B}$ alakú, ahol, $A \in N$, $p, q, B \in (N \cup T)^*$

- **Környezetfüggő grammatika** ($i=1$)

$p\mathbf{A}q \rightarrow p\mathbf{B}q$ alakú, ahol, $A \in N$, $p, q, B \in (N \cup T)^*$

és $B \neq \varepsilon$, kivéve az $S \rightarrow \varepsilon$ szabály (ha létezik), ekkor S nem fordul elő egyetlen szabály jobb oldalán sem.

Itt az A nemterminális szimbólum adott p, q szavak esetén helyettesíthető egy $p\mathbf{A}q$ mondatformában egy $B \in (N \cup T)^+$ szóval. Azaz, az $p = q = \varepsilon$ kivételével A helyettesítése B -vel, függ A környezetétől.

Továbbá minden $\omega_1 \rightarrow \omega_2 \in \mathcal{P}$ szabályra (kivéve $S \rightarrow \varepsilon$): $l(\omega_1) \leq l(\omega_2)$

- **Környezetfüggetlen grammatika** ($i=2$)

$A \rightarrow q$, ahol $A \in N$, $q \in (N \cup T)^*$.

- **Reguláris grammatika** (i=3)

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

A 3. típusú grammatikákat ill. nyelveket *jobb lineáris*aknak is nevezzük, mivel minden szabály jobb oldalán legfeljebb egy nemterminális állhat és az is csak a jobb oldal végén.

A különböző típusú nyelvtanok összességét **nyelvtani osztályoknak** hívjuk és \mathcal{G}_i jelöljük.

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

A definíciók alapján: $\mathcal{G}_3 \subseteq \mathcal{G}_2 \subseteq \mathcal{G}_1 \subseteq \mathcal{G}_0$.

A nyelvtani osztályozást felhasználva a nyelveket is osztályozhatjuk:

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

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

Igaz a hierarchia erősebb változata is: $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$.

Tétel. Nem minden nyelv írható le nyelvtannal.

💡 ▶ Miért érdekes a Chomsky-féle osztályozás?

Az algoritmusok - mint ismeretes - problémátípusok (problémaosztályok) megoldására szolgáló lépéssorozatok. A generatív grammatikákkal kapcsolatban több, számítógéppel (esetleg) megoldható probléma merül fel, melyekre algoritmusok készíthetők. Nagyon fontos annak eldöntése, hogy ha egy konkrét generatív grammatika egy konkrét problémájának megoldására kifejlesztett algoritmus alkalmazható-e más grammatika ugyanazon problémájára, illetve mennyi módosítással alakítható át.

Amennyiben a két grammatika ugyanazon Chomsky-osztályba tartozik, úgy egy jól megfogalmazott algoritmus váza a paramétereiktől eltekintve elvileg alkalmazható lesz.

Másik fontos indok, ami miatt fontos ismerni egy grammatika Chomsky-osztályát, hogy igazolható bizonyos problémakörökről, hogy nem készíthető hozzá általános érvényű megoldó algoritmus. ◁💡

Az \mathcal{L}_i ($i = 0, 1, 2, 3$) *nyelvosztályok mindegyike zárt a nyelveken végezhető műveletekre nézve*. Tehát valahányszor \mathcal{L} -beli nyelveken végezzük el a műveletet, mindannyiszor \mathcal{L} -beli nyelvet kapunk eredményül.

Tétel. (*Kis Bar-Hillel lemma*): Tetszőleges $L \in \mathcal{L}_3$ nyelvhez $\exists n(L) > 0 \in \mathbb{Z}$, hogy $\forall u \in L$, $l(u) \geq n : u = xyz$ a következő tulajdonságokkal bír:

- $y \neq \varepsilon$
- $l(xy) \geq n$
- $\forall i = 0, 1, \dots : xy^iz \in L$

Azaz L nyelv minden szavának elég hosszú részsavában létezik elég rövid, nem üres, beiteralható részszó. pl. $K-B HL \Rightarrow HE \notin \mathcal{L}_3$. HE: A $\{(,)\}$ ábécé feletti helyes zárójelezések halmaza.

Tétel. (*Zárttság*): \mathcal{L}_3 zárt az unió, konkatenáció, lezáras, komplementerképzés, metszet, különbség és szimmetrikus differencia műveletekre nézve.

💡 **Zsákutca:** Olyan nyelvtani jel, melyből az adott 2. típusú nyelvtanban nem vezethető le terminális szó.

Nem elérhető nyelvtani jel: Olyan nyelvtani jel, mely az adott 2. típusú nyelvtanban semmilyen, a kezdőszimbólumból történő levezetésben nem szerepel.

Láncszabály: Egy G grammatika egy $x \rightarrow y \in \mathcal{P}$ szabálya láncszabály, ha $x, y \in N$, azaz nemterminálisok.

Láncszabálymentes nyelvtan: Egy nyelvtan láncszabálymentes, ha nincs láncszabálya.

Epsilon-szabály: Egy G grammatika egy $x \rightarrow y \in \mathcal{P}$ epsilon-szabály, ha $y = \varepsilon$.

Korlátozott epsilon-szabály (KeS): Egy $G = \langle N, T, \mathcal{P}, S \rangle$ nyelvtanra teljesül a korlátozott epszilonszabály, ha nincsenek epszilonszabályai az $S \rightarrow \varepsilon$ szabály esetleges kivétel, de ez esetben minden $x \rightarrow y \in \mathcal{P}$ szabályra $y \in (N \cup T \setminus \{S\})^*$ (azaz semelyik szabály jobboldala sem tartalmazza a kezdőszimbólumot). ◁ 💡

Epsilonmentes nyelvtan: Egy nyelvtan epsilonmentes, ha teljesül rá az epsilon-szabály.

Automaták

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

Véges automaták

Definíció. A véges automata egy rendezett ötös,

$$\mathcal{A} = \langle Q, T, \delta, q_0, F \rangle$$

- Q - állapotok véges, nemüres halmaza
- T - bementi szimbólumok ábécéje (véges)
- $\delta : Q \times T \rightarrow Q$ - állapot-átmeneti függvény
- $q_0 \in Q$ - kezdőállapot
- $F \subseteq Q$ - végállapotok halmaza

Működése:

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

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

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

Automata által felismert nyelv: Az automata által felismert nyelv azon szavak halmaza, amelyekre a kezdő állapotból valamelyik végállapotba jut, azaz

$$L(\mathcal{A}) = \left\{ u \in T^* \mid \delta(q_0, u) \in F \right\}.$$

Az automata megadási módszerei

- *Átmeneti gráf:* Olyan irányított gráf, melynek pontja az állapotok, és A állapotból t -vel címkézett él fut B állapotba, ha $\delta(A, t) = B$.
- *Táblázat:* egyik tengelyén az állapotok, a másikon a terminálisok találhatók
- lényegében az átmeneti gráf mátrixreprezentációja.
- *Képlet:* a δ függvény megadása valamilyen zárt számítási képlettel, formulával.

VDA - Véges determinisztikus automata

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

VNDA - Véges nemdeterminisztikus automata

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

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

Tulajdonságok

Az állapot-átmeneteket

$$qa \rightarrow p$$

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

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

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

- Redukció:

Az $A = (Q, T, \delta, q_0, F)$ véges automata az $u \in QT^*$ szót a $v \in QT^*$ szóra redukálja ($u \Longrightarrow_A^* v$), ha $u = v$, vagy $\exists z \in QT^*$, amelyre $u \Longrightarrow_A^* z$ és $z \Longrightarrow_A v$ teljesül.

Az automata által elfogadott nyelv:

Az $A = \langle Q, T, \delta, q_0, F \rangle$ véges automata által elfogadott/felismerett nyelv alatt az

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

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

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

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

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

Válasz: Nincs.

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

Automata konstrukciója 3. típusú nyelvből. A fentihez hasonló, fordított konstrukcióval készíthető 3. típusú nyelvtan alapján automata, ez azonban nem mindig lesz determinisztikus - ha nemdeterminisztikus állapotátmeneti függvényt kapunk, az automatát még determinisztikussá kell tennünk.

Veremautomaták

Definíció

Az egy veremmel rendelkező (1-verem) automatát a következő hetessel azonosítjuk

$$A = \langle Q, T, \Sigma, \delta, q_0, \sigma_0, F \rangle$$

ahol

- Q - az állapotok véges halmaza
- T - input ábécé
- Σ - veremábécé
- $\delta : Q \times (T \times \cup\{\varepsilon\}) \times \Sigma \rightarrow 2^{Q \times \Sigma^*}$, $|\delta(q, t, \sigma)| < \infty$ - átmeneti függvény
- $q_0 \in Q$ - kezdőállapot
- $\sigma_0 \in \Sigma$ - a verem kezdőszimbóluma
- $F \subseteq Q$ - az elfogadó állapotok halmaza

🔗▷ A veremautomata egy ütemben kiolvassa a központi egység állapotát, az input szó aktuális szimbólumát és a verem tetőelemét, ennek függvényében új állapotba kerül, a verem tetőelemét felülírja egy vagy több jellel (azaz egy szóval), az input szó következő betűjére áll az olvasófej (kivéve ε -mozgás) és a tetőmutató az új tetőelemre áll.

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

A kezdeti konfiguráció $\sigma_0 q_0$.

Működés:

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

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

Ha az input szalag a $w \in T^*$ szót tartalmazza és a $z_0 q_0$ kezdeti konfigurációból kiindulva a lépések sorozatát végrehajtva az A veremautomata egy up konfigurációba ér, ahol p elfogadó állapot, akkor azt mondjuk, hogy A elfogadta a w szó. \triangleleft 🔗

A veremautomata elfogadja a szót, ha üres a verem, és elfogadó állapotban van.

Tulajdonságok

- Közvetlen redukció:

$$\alpha, \beta \in Z^* Q T^*$$

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

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

hogy:

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

- Redukció:

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

- A veremautomata által elfogadott nyelv:

Az A veremautomata által (elfogadó állapottal) elfogadott nyelv:

$$L(A) = \left\{ w \in T^* \mid z_0 q_0 w \Rightarrow_A^* up, \text{ ahol } u \in Z^*, p \in F \right\}$$

- Determinizmus:

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

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

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

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

- Üres veremmel elfogadott nyelv:

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

$$N(A) = \left\{ w \in T^* \mid z_0 q_0 w \xRightarrow{*}_A p, \text{ ahol } p \in Q \right\}$$

Tétel. Tétel. Az 1-verem automatákkal felismerhető nyelvek osztálya megegyezik a 2. típusú grammatikák által generált nyelvek osztályával, azaz $\mathcal{L}_V = \mathcal{L}_2$.

Reguláris nyelvek tulajdonságai és alkalmazásai

Reguláris nyelv: Reguláris nyelveknek nevezzük az alábbi három tulajdonsággal definiált (\mathcal{L}_{REG}) nyelvosztály elemeit:

- (1) \mathcal{L}_{REG} tartalmazza az elemi nyelveket: $\emptyset, \{\varepsilon\}, \{a\}$ ($a \in U$) (U : univerzális ábécé)
 - (2) \mathcal{L}_{REG} zárt az *unió*, *konkatenáció* és a *lezárás* műveletekre.
- \mathcal{L}_{REG} a legszűkebb olyan nyelvosztály, mely az (1), (2) feltételeknek megfelel.



Reguláris kifejezés: A reguláris kifejezések a reguláris nyelvek egyszerűsített leírását jelentik, ahol:

- az elemi reguláris kifejezések: $\emptyset, \varepsilon, a$ ($a \in U$)
- ha R_1 és R_2 reguláris kifejezések, akkor $(R_1 \cup R_2), (R_1 R_2), R_1^*$ is V ábécé feletti reguláris kifejezések
- a reguláris kifejezések halmaza a legszűkebb, melyre az előbbi két pont teljesül

V ábécé feletti reguláris kifejezések (rekurzív definíció)

- az elemi reguláris kifejezések: $\emptyset, \varepsilon, a$ ($a \in V$)
- ha R_1 és R_2 V ábécé feletti reguláris kifejezések, akkor $(R_1 \cup R_2), (R_1 R_2), R_1^*$ is reguláris kifejezések
- a V ábécé feletti reguláris kifejezések halmaza a legszűkebb, melyre az előbbi két pont teljesül

V ábécé feletti általánosított reguláris kifejezések (rekurzív definíció)

- az elemi reguláris kifejezések: $\emptyset, \varepsilon, a$ ($a \in V$)
- ha R_1 és R_2 V ábécé feletti általánosított kifejezések, akkor $(R_1 \cup R_2), (R_1 R_2), R_1^*, (R_1 \cap R_2), \overline{R_1}$ is V ábécé feletti általánosított reguláris kifejezések
- a V ábécé feletti általánosított reguláris kifejezések halmaza a legszűkebb, melyre az előbbi két pont teljesül

Reguláris kifejezések szemantikája (rekurzív definíció)

- az $\emptyset, \varepsilon, a$ reguláris kifejezések rendre az $\emptyset, \{\varepsilon\}, \{a\}$ nyelveket reprezentálják
- ha R_1 az L_1 és R_2 az L_2 nyelvet reprezentálja, akkor $(R_1 \cup R_2), (R_1 R_2), R_1^*$ rendre az $L_1 \cup L_2, L_1 L_2, L_1^*$ nyelveket reprezentálja

Reguláris kifejezések általánosított szemantikája (rekurzív definíció)

- az $\emptyset, \varepsilon, a$ reguláris kifejezések rendre az $\emptyset, \{\varepsilon\}, \{a\}$ nyelveket reprezentálják
- ha R_1 az L_1 és R_2 az L_2 nyelvet reprezentálja, akkor $(R_1 \cup R_2), (R_1 R_2), R_1^*, \overline{R_1}$ rendre az $L_1 \cup L_2, L_1 L_2, L_1^*, \overline{L_1}$ nyelveket reprezentálja

Axiómák

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

- Asszociativitás:

$$P + (Q + R) = (P + Q) + R$$

$$P \cdot (Q \cdot R) = (P \cdot Q) \cdot R$$

- Kommutativitás:

$$P + Q = Q + P$$

- Disztributivitás:

$$P \cdot (Q + R) = P \cdot Q + P \cdot R$$

$$(P + Q) \cdot R = P \cdot R + Q \cdot R$$

- Egységelem:

$$\varepsilon \cdot P = P \cdot \varepsilon = P$$

$$P^* = \varepsilon + P \cdot P^*$$

$$P^* = (\varepsilon + P)^*$$

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

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

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

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

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

Reguláris kifejezések és reguláris nyelvek

Minden reguláris kifejezés egy reguláris (3-típusú) nyelvet jelöl, és megfordítva, minden reguláris nyelvhez megadható egy, ezen nyelvet jelölő reguláris kifejezés.

◁ ♡

Tétel. (*Kleene-tétel*): $\mathcal{L}_{REG} = \mathcal{L}_3$, azaz a reguláris nyelvek osztálya megegyezik a véges determinisztikus automatával előállítható nyelvek osztályával.

Tétel. Az \mathcal{L}_3 nyelvosztály zárt a komplementer, a metszet, a különbség és aszimmetrikus differencia műveletekre, a zártsági tétel miatt pedig zárt az unió, konkatenáció és a lezáras műveletekre is.

Reguláris nyelvek felhasználási területei

Bár a reguláris nyelvtanok az összes nyelvek viszonylag szűk osztályát jelölik ki, könnyű kezelhetőségük miatt gyakorlati alkalmazásuk rendszeres.

Lexikális elemző (scanner): A fordítóprogramok legalsó szintjén, a lexikális elemek felderítésében nagy szerepet játszanak a 3. típusú nyelvtanok: általában reguláris kifejezésekkel azonosítják a nyelv lexikális elemeit.

Mintaillesztés: Mintaillesztési feladatokban (pl. adatok között/szövegben történő keresés) reguláris kifejezéssel írható le az illesztendő minta. Ebből általában véges determinisztikus automatát (például Knuth-Morris-Pratt automatát) generálnak.



Lineáris grammatikák és nyelvek

Definíció

Egy $\langle N, T, \mathcal{P}, S \rangle$ környezetfüggetlen grammatikát lineárisnak nevezünk, ha minden szabálya:

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

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

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

Lineáris és reguláris grammatikák, nyelvek

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

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



Környezetfüggetlen nyelvek tulajdonságai és elemzésük

Környezetfüggetlen grammatikák normálformái

Környezetfüggetlen grammatikák normálformái olyan grammatikai transzformációval előállított grammatikák, melyek:

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

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

ε -mentes grammatika

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

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

Chomsky normálforma

A $G = \langle N, T, \mathcal{P}, S \rangle$ környezetfüggetlen grammatikát Chomsky-normálformájúnak mondjuk, ha minden egyes szabálya:

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

Tétel. Minden ε -mentes $\langle N, T, \mathcal{P}, S \rangle$ környezetfüggetlen grammatikához meg tudunk konstruálni egy vele ekvivalens $G' = \langle N', T, \mathcal{P}', S' \rangle$ Chomsky-normálformájú környezetfüggetlen grammatikát.

Tétel. (az előző következménye): Minden G környezetfüggetlen grammatika esetében eldönthető, hogy egy u szó benne van-e G grammatika által generált nyelvben.

Redukált grammatika

- A környezetfüggetlen grammatika egy nemterminálisát *inaktív*nak/*nem aktív*nek nevezzük, ha nem vezethető le belőle terminális szó.
- A környezetfüggetlen grammatika egy nemterminálisát *nem elérhetőnek* nevezzük, ha nem fordul elő egyetlen olyan szóban sem, amely a kezdőszimbólumból levezethető.
- Az, hogy egy A nemterminális elérhető-e vagy aktív-e, az eldönthető.
- Egy környezetfüggetlen grammatika redukált, ha minden nemterminális *aktív* és *elérhető*.

Elemzés

A környezetfüggetlen nyelvtanok segítségével szintaxisfát építhetünk. Sikeres szintaxisfafelépítés esetén az elemzés sikerült, különben szintaktikai hibát találunk. Módszerek: legbal levezetés, legjobb levezetés.

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

Levezetési fa

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

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

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

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

legbaloldalibb nemterminálist kell helyettesítenünk.

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

Bar-Hillel Lemma

Minden $L \in \mathcal{L}_2$ (környezetfüggetlen nyelvhez) meg tudunk adni két nyelvfüggő p és q egész konstans ($p = p(L)$, $q = q(L)$), amelyekre ha $u \in L$ és $|u| > p$, akkor u -nak létezik a következő felbontása

$$u = xyzvw \quad (u, x, w, y, v \in T^*), \text{ melyre}$$

- $|yzv| \geq q$,
- $|yv| > 0$,
- $xy^izv^iw \in L \quad (\forall i \geq 0)$

A Bar-Hillel lemma azt mondja ki, hogy egy végtelen környezetfüggetlen nyelvben minden elég hosszú szóhoz végtelen sok "hasonló szerkezetű szó" van.

G generált L nyelv esetén, ha $|N| = l$, akkor L -beli szavak hossza legfeljebb 2^l .

Kiegészítés

Minden környezet független nyelv felismerhető veremautomatával	✓
Minden környezet független nyelv felismerhető 1 veremmel	✓
Minden 3. típusú nyelv felismerhető VNDA-val	✓
Minden 0-veremautomata véges determinisztikus automata	✓
Minden rekurzív nyelv rekurzívan felsorolható	✓
Minden (kiterjesztett)3. típusú nyelvtan 1. típusú nyelvtan	✓
Minden nyelvtannal leírt nyelv parciálisan rekurzív	✓
Minden rekurzívan felsorolható nyelv parciálisan rekurzív	✓
Minden Chomsky 1. típusú nyelv rekurzív nyelv	✓
Minden Chomsky 2. típusú nyelv rekurzívan felsorolható nyelv	✓
Minden Chomsky 2. típusú nyelv rekurzív nyelv	✓
Minden parciálisan rekurzív nyelv környezetfüggő	⊗
Minden parciálisan rekurzív nyelv reguláris	⊗
Minden nyelvtannal leírt nyelv rekurzívan felsorolható	⊗
Minden nyelvtannal leírt nyelv reguláris	⊗
Minden nyelvtannal leírt nyelv rekurzív	⊗
Minden nyelvtannal leírt nyelv felismertethető VDA-val	⊗
Minden Chomsky 1. típusú nyelv felismerhető 1 veremmel	⊗
Minden Chomsky 1. típusú nyelv felismerhető VDA-val	⊗
Minden rekurzívan felsorolható nyelv reguláris	⊗