Formális nyelvek

Kidolgozott gyakorlati példák
(Szerkesztés alatt álló példatár)

1 Mely nyelveket generálják az alábbi grammatikák? Adjuk meg a grammatikák típusait is!

1.1 Feladat:

$G =< \{0,1\}, \{S,A,B\}, S, \{S \to 01S|1A,\ A \to B0,\ B \to 1\} >$

1.2 Megoldás

A $S \to 01S$ szabállyal a '01' stringek számát tudjuk növelni. Majd ha ezt meguntuk, az $S \to (01)^n 1A \vdash (01)^n 1B0 \vdash (01)^n 110$ levezetés az egyetlen amivel folytatni tudjuk a generálást. Lezárni a generálást csak az $B \to 0$ szabály alkalmazásával tudjuk.

Tehát a keresett nyelv a következő:

$$L = \{\ (01)^n 110,\ ahol\ n \in \mathbf{N}\ \}$$

A G(L) pedig 2-es típusú, azaz környezetfüggetlen, mivel alakja:
$A \to \omega\ , ahol\ \omega \in (V \cup W)^*,\ valamint\ A \in W^*$.

1.3 Feladat:

$G =< \{0,1\}, \{S,A,B\}, S, \{S \to 0A,\ A \to 0A|1A|0B|1B,\ B \to 1B0|10\} >$

1.4 Megoldás

Kezdetben az $S \to 0A$ szabályt tudjuk csak alkalmazni, ezután viszont választhatunk A és B generálási irány közül. Először válasszuk A irányt: $0A \vdash 00A|01A$ ... használatával az első '0' string után '0'-at vagy '1'-eseket generálhatunk tetszőleges sorrendben és mennyiségben, ez formalizálva így néz ki: $0\{0,1\}^*$. Ezek után a B irány maradt, ami először egyetlen darab '0'-t vagy '1'-et ad az eddigi stringhez. A $B \to 1B0$ szabály a bal oldalon az '1'-esek, jobb oldalon a '0'-ák számát tudja növelni azonos mértékben. Viszont zárásképp '10'-val zárhatunk, ami jó, hiszen ez nem borítja fel az imént leírt szabályszerűséget. Így most tovább írhatjuk az eddig feltárt $0\{0,1\}^*$ összefüggést: $0\{0,1\}^*1^m0^m$. Ámde mert A-ból B-be el kell jutnunk valahogy amit csak egy '0' vagy '1' automatikus hozzáadásával tehetünk meg, így a képlet $\{0,1\}^*$ részénél ki kell vennünk az üres szót: $\{0,1\}^* \setminus \epsilon$.

Így a keresett nyelv végül a következőképp alakul:

$$0\alpha 1^m 0^m \mid \alpha \in \{0,1\}^* \setminus \epsilon\ ,\ m \in \mathbf{N},\ m \geq 1$$

A G(L) 2-es típusú grammatika. Indoklás a rész legelső feladatánál.

2 Készítsünk reguláris grammatikákat az alábbi nyelvekhez!

2.1 Feladat:

$L = \{\alpha \in \{0,1\}^* \mid \alpha - ra\ igaz\ X\}$

Ahol X az, hogy: nem kezdődik egyessel vagy nem végződik nullára.

2.2 Megoldás

Először is nézzük végig a lehetséges eseteket: 0x0; 0x1; 1x0; 1x1 (itt az x bármilyen és bármennyi véges V*-beli string-et jelöl). A 'vagy'-al öszekötött állítások igazságértéke csak akkor hamisak ha mindkét állítás hamis. Ebből az adódik, hogy 0x1 kivételével az összes eset megengedett.

A grammatika tehát így néz ki:

$$G =< \{\{0,1\}, \{S,A,B\}, S, \{S \to 0|1|1A|0B,\ A \to 0|1|0A|1A,\ B \to 0B|1B|1\}\} >$$

A működésének magyarázata esetszétválasztással egyszerűen elintézhető. A kezdőszimbólumból való távozások közül az B jelöli, hogy egyessel csak végezhetünk ($B \to 1$), A pedig hogy bármivel ($A \to 0|1$). Az $A \to 0A|1A$ pedig azt biztosítja, hogyha a kezdő -és zárószimbólum közt bármilyen string lehessen, ugyanígy a $B \to 0B|1B$ esetnél is.

2.3 Feladat:

$L = \{\alpha \in \{a,b\}^* \mid \alpha - ra\ igaz\ X\}$

Ahol X az, hogy: első és utolsó betűje különbözik.

2.4 Megoldás

Először is nézzük végig a lehetséges eseteket: axa; axb; bxa; bxb. Ebből rögtön ki tudjuk szűrni mi kell nekünk: axb és bxa (itt az x bármilyen és bármennyi véges V*-beli string-et jelöl). Ezek alapján az előző feladat mintájára tudjuk megkonstruálni a grammatikánkat:

$$G =< \{\{a,b\}, \{S,A,B\}, S, \{S \to aB|bA|\epsilon,\ B \to aB|bB|b,\ A \to aA|bA|a.\}\} >$$

A működésének magyarázata igen egyszerű és hasonló az előző feladatéhoz: esetszétválasztást használunk itt is. A B irány it a a-val való kezdést és b-vel való zárást takarja. Az A irány pedig a fordítottját, azaz a b-vel való kezdést és a-val történő lezárást. Az $A \to aA|bA$ pedig azt biztosítja, hogyha a kezdő -és zárószimbólum közt bármilyen string lehessen, ugyanígy a $B \to aB|bB$ esetnél is.

3 Készítsünk (bármilyen) grammatikákat az alábbi nyelvekhez!

3.1 Feladat:

$L = \{a^{2n}b^{3n}ab,\ n \in \mathbf{N}\}$

3.2 Megoldás

Ezt a feladatot sajnos nem tudjuk reguláris grammatikával megoldani, amit készíteni fogunk az környezetfüggetlen lesz. Az ötletünk a következő: $S \to aaAbbbab,\ A \to aaAbbb|aabbb$. Ezzel először S-ből átmegyünk A-ba, ami azért van mert nem szeretnénk a szó végi 'ab' generálását a szó belsejében ismételgetni. A-ból viszont véges sokszor tudjuk a szó belsejében az 'aabbb'-k generálását ismételgetni a $A \to aaAbbb$ szabály alkalmazásával. Ha mindezt meguntuk, akkor a $A \to aabbb$ szabállyal zárunk. De ne feledkezzünk meg arról sem, hogy az 'aabbbab' és 'ab' szavakat is tudnia kell generálnia a grammatikánknak, ezt a $S \to aabbbab|ab$ szabályok biztosítják. Így végül a következő grammatika lesz a megoldás:

$$G =< \{\{a,b\}, \{S,A\}, S, \{S \to aaAbbbab|aabbbab|ab,\ A \to aaAbbb|aabbb\}\} >$$

3.3 Feladat:

$L = \{10^{3k}1^{2k}0,\ k \in \mathbf{N}\}$

3.4 Megoldás

A megoldásnál itt is hasonlóan járunk el, mint az előző esetben, szintén környezetfüggetlen grammatikát gyártunk. Az ötletünk a következő: $S \to 000A11,\ A \to 000A11|00011$. Ezzel először S-ből átmegyünk A-ba, ami azért van mert nem szeretnénk a szó eleji '1' és szóvégi '0' generálását a szó belsejében ismételgetni. A-ból viszont véges sokszor tudjuk a szó belsejében az '00011'-k generálását ismételgetni a $A \to 00A11$ szabály alkalmazásával. Ha mindezt meguntuk, akkor a $A \to 00011$ szabállyal zárunk. De ne feledkezzünk meg arról sem, hogy az '1000110' és '10' szavakat is tudnia kell generálnia a grammatikánknak, ezt a $S \to 1000110|10$ szabályok biztosítják. Így végül a következő grammatika lesz a megoldás:

$$G =< \{\{0,1\}, \{S,A\}, S, \{S \to 1000A110|1000110|10,\ A \to 000A11|00011\}\} >$$

4 Formális nyelvek metszete, tükörképe és iteráltja

4.1 Feladat:

Legyen $L_1 = \{ab^n\ :\ n \in \mathbf{N}\},\ L_2 = \{a^nb\ :\ n \in \mathbf{N}\}$.
a) $L_1 \cup L_2 =?$, *valamint* $L_1 \cap L_2 =?$
b) *Az* $L_1^* = \{\alpha^{-1}\ :\ \alpha \in L_2\}$ *jelolessel,* $L_1 * L_2^{-1} =?$
c) $Igaz - e,\ hogy\ L_1^* = L_2^*\ ?\ Ha\ nem,\ miert?\ Ha\ igen\ mi\ ez\ a\ nyelv?$

4.2 Megoldás

Az a) feladatrészre a megoldás a következő: $L = \{ab^na^mb,\ n,m \in \mathbf{N}\}$, mivel a formális nyelvek metszete nem más mint az összeszorzásuk, ez utóbbi pedig konkatenálást (egymás után illesztést) jelent. A metszet az $L = \{ab\}$ lesz, mivel az az az egy szó, ami mindkét nyelvben előfordul.

A b) feladatrészre a válaszunk: $L = \{ab^nba^m\ n,m \in \mathbf{N}\}$. Mivel itt az α^{-1}-en nem jelent mást, mint a α szó tükörképét, ami valójában az őt alkotó string-ek fordított sorrendben való egymás utáni leírását takarja.

Végezetül a c) feladarészhez érkeztünk. Az $L^* = \bigcup_{n\geq 0} L^n$ összefüggés jelentését kell itt tulajdonképp tárgyalnunk. Tehát L nyelv iteráltja nem más, mint a szavainak tetszőleges véges sok tényezőből álló szorzatainak halmaza. Visszakanyarodva a kérdésünkhöz: az egyenlőség nem teljesül.
Indoklás: mivel $L_1 = \{a, ab, abb, abbb, ...\}$ és $L_2 = \{b, ab, aab, aaab, ...\}$ nyelvek szavainak szorzata között L_2 esetén előállhat pl. 'bab', de az L_1 nyelvnél ez nem lehetséges, tehát találtunk egy ellenpéldát.

5 Két nyelv metszete és uniója

5.1 Feladat:

Készítsünk két grammatikát, aminek az uniója az alábbi nyelvet generálja: $L = \{ba^nab^mba\ :\ n,m \in \mathbf{N}\}$

5.2 Megoldás

A nyelvet két részre szedjük és mindkét "részéhez" készítünk egy reguláris grammatikát, majd vesszük a két grammatika unióját. Megjegyezzük, hogy két grammatika uniója is reguláris kell legyen. A felosztás nézzen ki a következőképp: $L_1 = \{ba^na\ ...\}$ és $L_2 = \{b^nba\ ...\}$. Itt fontos megemlíteni, hogy az n hatvány az nyelvenként van érvényben. Tehát az únióban majd végül n és m fog szerepelni.

Az $G(L_1) = \{..., \ S \to bA, A \to aA|a\}$ generálja, a $G(L_2) = \{..., \ S \to bS|bA, A \to a\}$ generálja. Ennek a kettőnek a működése triviális, azokat most terjedelmi okok miatt nem részletezzük. Végül vesszük a két nyelv unióját, ami valójában a szorzatukat (azaz magyarán egymás után helyezésüket, konkatenálásukat) jelenti. Ezt úgy érjük el, hogy először a két nyelv terminális jeleinek halmazait diszjunktá tesszük, ez valójában a "nagybetűk bevesszőzése". Majd az L_1 nyelvben a zárószimbólumok végét kiegészítjük a L_2 grammatika kezdőszimbólumával, az egyik nyelvből a másikba való átmenet végett.

Itt megemlítjük azt, hogy ebben a feladatban az ϵ sehol sem szerepelt a kezdőszimbólumokból kiindulva, így most ezzel nem kell vacakolnunk. De egyébként ha lenne ilyen arra is kitérünk:
Tegyük fel, hogy az L_1-ben az $S \to bA$ helyett az $S \to bA|\epsilon$ szabály szerepel, ekkor ϵ-t csak egyszerűen kivesszük az unióból és helyére az "átvivőszabályt" írjuk be, ami jelen esetben az $A \to aS^{'}$. Ez biztosítja, hogy "semmi" generálás után rögtön a második nyelv elejére jutunk. Ha az L_1 és L_2 nyelvben egyszerre szerepel az ϵ, akkor egyszerűen az új kezdőszimbólum jobb oldalára be kell írni az ϵ-t is. Ezek után viszont figyeljünk rá, hogy az új kezdőszimbólum nem fordul elő-e a szabályok jobb oldalán.

Visszatérve az eredeti feladatunkhoz a megoldás a következőképp alakul:

$$G =< \{\{a,b\}, \{S, S^{'}, A, A^{'}, Q\}, Q, \{Q \to bA, \ A \to aA|aS^{'}, \ S^{'} \to bS^{'}|bA^{'}, \ A^{'} \to a\}\} > .$$

5.3 Feladat:
Készítsünk két grammatikát, aminek a metszete az alábbi nyelvet generálja: $L = \{\alpha ab^n b \mid \alpha \in \{\epsilon, a, b\} \ , \ n \in \mathbf{N}\}$

5.4 Megoldás

A nyelvet két részre szedjük és mindkét "részéhez" készítünk egy reguláris grammatikát, majd vesszük a két nyelv metszetét, ami az összeadásukat jelenti. Az összeadás itt valami olyasmit takar, hogy egyik vagy másik nyelv generálható, hozzátéve hogy mindkét nyelv nem generálható egyszerre, de legalább az egyik mindeképp generálható.
Az $L_1 = \{..., \ aab^n b\}$ legyen, az $L_2 = \{..., \ bab^n b\}$. Ezek után az L_1 és L_2-höz késztünk grammatikákat, úgy hogy a kezdő szabályokhoz még egy ϵ-t is beveszünk.
A következőképpen néznek ki: $G(L_1) = \{..., \ S \to aA|\epsilon, \ A \to aB, \ B \to bB|b\}$, valamint $G(L_2) = \{..., \ S^{'} \to bA^{'}|\epsilon, \ A^{'} \to aB^{'}, \ B^{'} \to bB^{'}|b\}$. Már a terminális jelek diszjunktá tétele is megtörtént. Ezek után nincs más dolgunk, minthogy bevezetünk egy új kezdőállapotot, amiből elérhetővé tesszük az L_1 és L_2 béli szavakat is. Tehát $G(L) = \{..., \ Q \to aA|bA^{'}|\epsilon\}$, majd ezek után az összes szabályt változatlanul bemásoljuk: $G(L) = \{..., \ Q \to aA|bA^{'}|\epsilon, A \to aB, B \to bB|b, A^{'} \to aB^{'}, B^{'} \to bB^{'}|b\}$. Mivel $Q \to \epsilon$ szerepel, így át kell néznünk, hogy Q nem-e fordul elő a szabályok jobb oldalán, de nem. Jók vagyunk! A teljes megoldás így írható fel:

$$G(L) =< \{\{a,b\}, \{S, S^{'}, A, A^{'}, B, B^{'}, Q\}, \ Q, \{Q \to aA|bA^{'}|\epsilon, A \to aB, B \to bB|b, A^{'} \to aB^{'}, B^{'} \to bB^{'}|b\}\} >$$

6 A formális nyelvek mint halmazok

6.1 Feladatok és megoldásai:

$V = \{a, b, c\}, \ W = \{c, d, e\}$.

$V * (W \setminus V) * W^* \ = \ \{a, b, c\} * \{d, e\} * \{\epsilon, c, d, e, cd, ec, ed, cde, ecd, ...\}$
$= \ \{ad, ae, bd, be, cd, ce\} * \{\epsilon, c, d, e, cd, ec, ed, cde, ecd, ...\} \ = \ \{ad\alpha, ae\alpha, bd\alpha, be\alpha, cd\alpha, ce\alpha\}$, ahol $\alpha \in \{c, d, e\}^*$

$(V \setminus W)(W \setminus V) = \{a, b\} * \{d, e\} = \{ad, ae, bd, be\}$
$V^* \setminus W^* \implies$ Minden lehet vélgülis, csak az ϵ-okat és a 'c'-t vettük ki. Tehát minden $\{a, b, c, d, e\}$-t tartalmazó szó kijöhet, ami nem tartalmaz ϵ-t és nem csak 'c'-ből áll (c lehet benne).