Formális nyelvek és automaták

Nagy Sára gyakorlatai alapján

Készítette: Nagy Krisztián

2. gyakorlat

Ismétlés:

Megjegyzés: Az ismétlés egy része nem szerepel a dokumentumban, mivel lényegében a teljes 1. gyakorlat elméletibb részét foglalja magában.

<u>Reguláris műveletek nyelveken:</u>

- unio
- konkatenáció
- lezárás

<u>Reguláris nyelvek családja:</u>

- elemi reguláris nyelvek $\emptyset, \{\varepsilon\}, \{a\} \quad a \in U, ahol\ U\ az\ univerzális\ jelkészletet\ jelenti$
- reguláris nyelv az elemi reguláris nyelvekből a reguláris műveletek véges sokszori alkalmazásával előállíthatók

Feladatok:

$L_1 = \{ab^{2k+1} \mid k \geq 0\} = \left\{ab, ab^3, ab^5, \dots\right\} \rightarrow \{a\}(\{b\}\{b\})^*\{b\}$

- a-val kezdődik ezért a reguláris kifejezésünknek is azzal kell kezdődnie, továbbá a $2k+1$ hatvány miatt fel tudjuk bontani a b^{2k+1} úgy, hogy $b^{2k}b$, ezt tovább lehet bontani úgy, hogy $(bb)^k b$ és mivel 0-nál nagyobb vagy egyenlő k ezért képzeljük el iterációként. Lényegében nekünk az a-val kezdődő és páratlan b-re végződő szavak halmazát kell meghatároznunk. A felbontásból adódóan ha k-t *-al helyettesítjük, akkor az iteráció szempontjából ez az alábbit jelenti: $\{bb\}^* = \left\{\varepsilon, b^2, b^4, \dots\right\}$, azaz páros b komponenseket „gyárt" számunkra. Mivel nekünk páratlan b komponensekre van szükségünk, így hozzá kell konkatenálnunk egy b -t az iterációnkhoz. Így a reguláris kifejezésünk esetén $\{a\}$ – hoz $(\{b\}\{b\})^*\{b\}$ -t kell konkatenálnunk.

$L_2 = \{u \in \{a,b\}^* \mid l_a(u) = 1\} = \{olyan\ szavak\ halmaza, ami\ 1\ db\ a\ betűt\ tartalmaz\ \} =$
$= \{a, ab, ba, abb, bab, bba, \dots\} \rightarrow \{b\}^*\{a\}\{b\}^*$

Reguláris kifejezés:

- elemei: $\emptyset, \varepsilon, a \quad a\ tetszőleges\ betű$
- ha $r_1, r_2\ reguláris\ kifejezés, akkor\ r_1|r_2\ [vagy \dots vagy \dots \quad (az\ \cup\ művelet)]$;
  $r_1 r_2\ (konkatenáció)$
  $r^*\quad (lezárt)$
  $(r)\ (helyes\ zárójel\ használat)$

  is reguláris kifejezés
- nincs más reguláris kifejezés

1. feladat:
$L_1 = \{ab^{2k+1} | \, k \geq 0\} = \left\{ab, ab^3, ab^5, \dots\right\} \rightarrow \{a\}(\{b\}\{b\})^*\{b\} \rightarrow a(bb)^*b$
$L_2 = \{u \in \{a,b\}^* | \, l_a(u) = 1\} = \{a, ab, ba, abb, bab, bba, \dots\} \rightarrow \{b\}^*\{a\}\{b\}^* \rightarrow b^*ab^*$

2. feladat:
Az 1. feladat-beli nyelveket használjuk fel és határzzuk meg $L_1 \cap L_2$ -t és $L_2 \backslash L_1$-t
Megoldás: $L_1 \cap L_2 = L_1$ $\qquad L_2 \backslash L_1 = b^*a(bb)^*|bb^*ab^*$

3. feladat:
Írjuk le reguláris kifejezésekkel a néggyel osztható bit sorozatokat (bináris számokat) leíró nyelvet!
További megkötés: Ne legyenek vezető 0-ák (Azaz ne legyen ilyesmi: 0000, 00010, 01001)
Megoldás:
(Megjegyzés: Kettes számrendszer! ☺)
Jelen esetünkben $X^* = \{0,1\}^*$
Néggyel osztható egy bináris szám, ha vagy csak 0 vagy 00-ra végződik.
Ezek alapján két esetre tudjuk bontani a feladatunkat. A vagy…. vagy… szófordulat alapján a | művelettel fogjuk a megoldásokat felírni.
Az első eset, ha vagy csak 0 a bináris számunk. Ez a megoldás egyszerű, mivel binárisan a 0 az 0.
A második esetben, mivel nem engedjük meg a vezető 0-ákat, ezért biztos, hogy 1-el fog kezdődni a bit sorozatunk (így a reguláris kifejezésünk is.) Ezek után tetszőlegesen sok 0 vagy 1-es állhat és mivel néggyel oszthatónak kell lennie, ezért 00-ra kell végződnie.
Egy reguláris kifejezést több féle képpen is fel lehet írni, így ennél a feladatnál két féle lehetséges megoldást is adunk, hogy szemléltessük ezt. Bármelyik választás helyes.
$L_3 = 0|1(0|1)^*00 = 0|1(0^*1^*)^*00$
A két megoldás végeredményeként ugyan azt a nyelvet kapjuk. A különbség annyi, hogy az első esetben a $(0|1)^*$ miatt, ha a nyelvet egy program konstruálná, egyesével tudna választani, hogy az iteráción belül milyen elemek jöjjenek. Hogy szemléltessem: $(0|1)^*$ esetén vagy 0-át vagy egyet választhat a programunk tetszőlegesen sokszor, míg a másik esetben (a $(0^*1^*)^*$ megoldásunkban) tetszőlegesen sok 0 és tetszőlegesen sok 1-esből választhat tetszőlegesen sokat.

Azonosítós feladatok:

A feladatok során legyen b betű, s számjegy és a aláhúzás (_)

1. feladat:
Írjuk fel reguláris kifejezésekkel a betűvel kezdődő és betűre vagy számjegyre végződő azonosítókat leíró nyelvet!

Megoldás: $L_4 = b(b|s)^*$

2. feladat:
Írjuk fel reguláris kifejezésekkel azt az aláhúzást tartalmazható azonosítókat leíró nyelvet, melyek betűvel kezdődnek, számokat is tartalmazhatnak, továbbá aláhúzás nem áll az azonosító végén és nem szerepel több aláhúzás egymás mellett!

Megoldás: $L_5 = b(b|s|ab|as)^* = b\big(b|s|a(b|s)\big)^*$

Grammatikák:

A generatív nyelvtanok speciális produkciós rendszerek az alábbi négy feltétellel:

(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.

Ezeket a feltételeket figyelembe véve az alábbi közvetlen definiciót adhatjuk a formális nyelvtan fogalmára.

1.16. definíció. *Generatív nyelvtannak (formális nyelvtannak) nevezzük az alábbi négyest:*

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

ahol T a terminális jelek ábécéje, N a nyelvtani jelek ábécéje, $\mathcal{P}$ véges szabályhalmaz, melynek bármely szabálya bal oldalán van legalább egy nyelvtani jel, és $S \in N$ a kezdő-szimbólum.

A levezetés fogalom a formális nyelvtanok esetében megegyezik a produkciós rendszereknél definiált levezetés fogalommal. Mivel most kitüntetett szerepük van a terminális jeleknek, ezért külön elnevezést használunk azokra a szavakre, melyek vegyesen tartalmazhatnak terminális jeleket és nyelvtani jeleket, és külön a csak terminális jelekből állókra. Az előbbieket *mondatformának* hívjuk és konvenció szerint a görög ábécé első betűinek kisbetűs formáival (α, β, γ) jelöljük, míg az utóbbiakat *terminális szavaknak* és a latin ábécé végéről vett kis betűkkel (u, v, w, z) jelöljük.

Az S kezdőszimbólumból G-ben levezethető mondatformákat, terminális szavakat *G-ben elérhető* mondatformáknak, terminális szavaknak nevezzük (Ha G rögzített vagy nem lényeges, akkor egyszerűen csak elérhetőségről beszélünk).

1.17. definíció. *A G nyelvtan által generált $L(G)$ nyelv a G-ben elérhető terminális szavak összessége, azaz:*

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

/Forrás: Hunyadvári László – Manhertz Tamás Automaták és Formális Nyelvek című könyve./

Próbáljuk a fentebbi két definíciót felfogni feladatokkal.

1. Feladat:

Legyen $L_6 = \{a^n b^n | \, n \geq 0\}$. Az első gyakorlaton láttuk, hogy ez a nyelv nem reguláris. A most megtanult módszerrel viszont fel tudjuk írni. Adjuk meg a grammatikáját!

Megoldás:
A fenti definíció alapján láthatjuk, hogy $T = \{a, b\}, N = \{S\}\ mivel\ S \in N$, amennyiben más nyelvtani jelre is szükségünk lenne, azt a nem terminálisokhoz írjuk.
Ezért: $G_6 = (\{a, b\}, \{S\}, S, P_6)$.
P_6:
$S \to aSb$
$S \to \varepsilon$

A gondolkodás mód: Vizsgáljuk meg a felírt szabályrendszer alapján, hogy meg kapjuk-e az L_6-ban található szavakat.

Nézzük először ε-t: $S \to \varepsilon$ szabályunk miatt megkapjuk.
Most nézzük meg ab-t: $ab: S \to aSb \to ab$
Nézzük meg a^8b^8-t: a^8b^8: $aSb \to aaSbb \to aaaSbbb \to \cdots \to a^7Sb^7 \xrightarrow{8.lépés} a^7aSbb^7 \to a^8b^8$
Az utóbbi lépést az $S \to \varepsilon$ szabályunk miatt kaptuk míg előtte folyamatosan az $S \to aSb$ szabályt alkalmaztuk. Ezek alapján látható, hogy a szabályrendszerünk helyes, így a feladatunk is készen van.

2. Feladat:

Legyen $L_7 = b(b|s)^*$. Adjuk meg a grammatikáját!

Megoldás:

Emlékezzünk vissza a régi feladatunkra amiben szerepelt ez a nyelv:
[Írjuk fel reguláris kifejezésekkel a betűvel kezdődő és betűre vagy számjegyre végződő azonosítókat leíró nyelvet! (b – betű, s-számjegy)]

Azt tudjuk, hogy jelen esetben a Terminális jelek ábécéje a $T = \{b, s\}$ és azt is, hogy $S \in N$.
Ki kell találnunk a szabályrendszert!
Tudjuk, hogy betűvel kell kezdődni-e az azonosítóknak, így definiáljuk ezt az első szabályunkba, a feladat többi részét (betűre vagy számjegyre végződjön az azonosító) egy másik szabályba írjuk le: $S \to bA$. Ebben az esetben létre hoztunk egy nyelvtani jelet (,amely a nem terminálisokhoz tartozik). Most, hogy definiáltuk azt, hogy betűvel kezdődjön az azonosítónk meg kell határoznunk, hogy A -t hogyan szeretnénk használni a szabályrendszerünkben. $(b|s)^*$-ból láthatjuk, hogy vagy betűre vagy számjegyre végződik az azonosítónk vagy pedig a lezárás művelet miatt ε-ra [azaz kimarad az azonosítonkból a $(b|s)^*$ rész], továbbá a lezárás miatt tetszőlegesen sokszor felhasználhatjuk a betűket és a számjegyeket. Ezeket a megállapításokat kell A-ból levezetni. Amennyiben egy adott szabály jobb és bal oldalán is szerepel ugyan az a nem terminálisunk, továbbá a jobboldalon terminális is szerepel (tetszőleges mennyiségben) úgy egy iterációt tudunk létrehozni, ami végtelen sokszor ismétli ön magát, viszont ilyen esetben szükségünk van egy ε-ra is, hogy az iteráció bármikor ki tudjon lépni. Ezek alapján az alábbi szabályokat tudjuk felírni A -ra:

$A \rightarrow bA$
$A \rightarrow sA$
$A \rightarrow \varepsilon$
Mivel az adott nem terminális többször szerepel a szabály bal oldalán, ezért | műveletünkkel fel tudjuk írni egyben ezeket a lehetőségeket:
$A \rightarrow bA|sA|\varepsilon$
Ezzel meg is oldottuk a feladatunkat. A grammatikánk így néz ki:

$G_7 = (\{b, s\}, \{S, A\}, S, P_7)$
P_7:
$S \rightarrow bA$
$A \rightarrow bA|sA|\varepsilon$

Feladat:
Írjuk fel az előző feladatban leírt grammatikánkat BNF-el.

BNF:
<valami> - nem tárminálisokat jelöl
Ha nincs <>-ben, akkor terminálisokat jelöl
továbbá ::= választja el a szabályrendszer bal és jobb oldalát.
Bal oldalon csak nem terminális szerepel.

Ezek alapján:
<azonosító>::=<betű><azonosító_vége>
<betű>::= a|b|c|…|z
<azonosító_vége>::=<betű><azonosító_vége>|<számjegy><azonosító_vége>| ε
<számjegy>::= 0|1|2|…|9

<u>Feladat:</u>
Vezessük le b12-t az előző két feladat alapján (grammatika,BNF)

Grammatika alapján:
$S \rightarrow bA \rightarrow bsA \rightarrow bssA \rightarrow bss$
- b betű és az 1 és a 2 számjegy, tehát készen vagyunk.

BNF:
<azonosító>→<betű><azonosító_vége>→
→ b< azonosító_vége>→b<számjegy><azonosító_vége>→
→b<számjegy><számjegy><azonosító_vége>$\overset{*}{\rightarrow}$b12

Feladat:
Milyen nyelvet generál az alábbi grammatika?
$G_8 = (\{a, b\}, \{S, B\}, S, P_8)$
P_8:
$S \to aSB|\varepsilon$
$aB \to Ba$
$B \to b$

Megoldás:
Induljunk el a kezdő szimbólumunkból és próbáljunk észrevenni valami logikát:
$S \to aSB \to aaSBB \to aaaSBBB \to \cdots \to \ldots aaaSBBB \ldots$
Nézzük meg a többi szabályt: $aB \to Ba$
Nézzük meg az $aaBB$ esetre: $aaBB \to aBaB \to BaBa \to BBaa$
Nézzük meg a $B \to b$ szabályt: $BBaa \to bbaa$

Mint láthatjuk a szabályrendszerünkben az $S \to aSB|\varepsilon$ szabály egy iteráció, ami ugyan annyi a-t iterál, mint B-t, továbbá az $aB \to Ba$ szabály egy felcserélés melynek segítségével a B nem terminálisokat a szavaink elejére „csúsztatja”, majd a $B \to b$ kicsréli a nem terminális B-ket terminális b-k re. Ezek alapján egy olyan nyelvet ír le, amelyek tetszőlegesen sok $\{a, b\}$ terminálisokból állnak és ugyan annyi $\{a\}$ terminális található a szavakban, mint $\{b\}$ terminális, ezért ez a grammatika az egyenlő hosszúságú a *és* b betűkből álló szavakat generáló nyelvet írja le. Formálisan:
$L(G_7) = \{u \in \{a, b\}^* | l_a(u) = l_b(u)\}$