

Horváth Géza, Mecsei Zoltán, Nagy Benedek

Gyakorlati összefoglaló

mobiDIÁK könyvtár

Horváth Géza, Mecsei Zoltán, Nagy Benedek

Gyakorlati összefoglaló

mobiDIÁK könyvtár

SOROZATSZERKESZTŐ

Fazekas István

Horváth Géza, Mecsei Zoltán, Nagy Benedek

Gyakorlati összefoglaló

mobiDIÁK könyvtár

Lektor

Fazekas Attila

Copyright © Horváth Géza, Mecsei Zoltán, Nagy Benedek

Copyright © elektronikus közlés mobiDIÁK könyvtár

mobiDIÁK könyvtár

Debreceni Egyetem

Informatikai Intézet

4010 Debrecen, Pf. 12

<http://mobidiak.unideb.hu>

A mű egyéni tanulmányozás céljára szabadon letölthető. Minden egyéb felhasználás csak a szerző előzetes írásbeli engedélyével történhet.

A mű a *A mobiDIÁK önszervező mobil portál* (IKTA, OMFB-00373/2003) és a *GNU Iterátor, a legújabb generációs portál szoftver* (ITEM, 50/2003) projektek keretében készült.

Tartalomjegyzék

I. Formális nyelvek	9
1. Markov-féle normál algoritmus	9
2. Grammatikák főbb típusai	14
3. Az üresszó-lemma	21
4. Chomsky-féle normálforma	24
5. Kuroda-féle normálforma	30
6. Bar-Hillel lemma	34
7. Cook-Younger-Kasumi algoritmus	37
8. Early-algoritmus	46
II. Automaták	57
1. Mealy automaták minimalizálása	57
2. Moore automaták minimalizálása	60
3. Kimenő jel nélküli, iniciális, végállapotokkal bővített automaták minimalizálása	63
4. Nemdeterminisztikus és determinisztikus véges automaták ekvivalenciája	66
5. Ekvivalens reguláris nyelvtanok megadása reguláris kifejezésekhez	69
6. Ekvivalens véges automaták megadása reguláris nyelvtanokhoz	72
7. Ekvivalens reguláris kifejezések megadása véges automatákhoz	74
8. Veremautomaták	76
9. Turing-gépek	83

I. fejezet

Formális nyelvek

1. Markov-féle normál algoritmus

Legyen $W(V, H, H_1)$ rendszer, ahol V egy ábécé,

$H = \{P_1 \rightarrow Q_1, \dots, P_m \rightarrow Q_m\}$ helyettesítések rendezett halmaza,

$H_1 \subseteq H$ záróhelyettesítések (esetleg üres) halmaza!

Alkalmazható a $P \in V^*$ szóra a Markov-féle normál algoritmus, ha van olyan $P' \in V^*$, $P'' \in V^*$ és $i \in \{1, \dots, m\}$, hogy $P = P'P_iP''$.

Az algoritmust alkalmazzuk a P szóra a legkisebb i indexre, de amennyiben P -ben részszoikként többször is szerepel P_i , akkor a legbaloldalibb helyen.

(Vagyis $P'P_i$ -ben csak egyszer szerepeljen részszoikként P_i .)

Ekkor kapjuk a $Q = P'Q_iP''$ szót.

Ha $P_i \rightarrow Q_i$ záróhelyettesítés, akkor a P -n végzett számítás eredménye Q , ha nem, akkor Q -ra alkalmazzuk tovább a számítást.

Az algoritmus akkor áll meg, ha a helyettesítésünk záróhelyettesítés, vagy ha már nem alkalmazható rá egyetlen szabály sem.

1.1. példa.

$1^4 + 1^3$ szóra alkalmazzuk az alábbi algoritmust!

$W(V, H, H_1)$

$V = \{1, +\}$,

$H = \{+1 \rightarrow 1+, 1+ \rightarrow 1\}$,

$H_1 = \{1+ \rightarrow 1\}$.

A lépésenkénti levezetés a következő: $1111\underline{+}111 \Rightarrow 11111\underline{+}11 \Rightarrow 111111\underline{+}1 \Rightarrow 1111111\underline{+} \Rightarrow 1111111$.

1.2. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szó esetén annyi 1-est, majd annyi 0-ást ír, amennyi a bemenő szóban van!

Például: 001011011 bemenetre 111110000 szót adja eredményül.

Megoldás:

Egyik lehetséges megoldás a következő:

$W(\{0, 1\}, \{01 \rightarrow 10\}, \emptyset)$

Az algoritmus a buborékrendezést valósítja meg.

Ugyanez a példa három számjegy esetén:

$W(\{0, 1, 2\}, \{01 \rightarrow 10, 02 \rightarrow 20, 12 \rightarrow 21\}, \emptyset)$.

1.3. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szó tükörképét adja eredményül!

Például: 0110011 bemenetre 1100110 végeredményt ad.

Megoldás:

Egy lehetséges algoritmus a következő:

$W(\{0, 1, X, Y, U, V, Z\}, H, H_1)$, ahol H elemei:

1. $Y0 \rightarrow 0Y$, 2. $Y1 \rightarrow 1Y$, 3. $Y \rightarrow Z$, 4. $X0 \rightarrow XU$,
5. $X1 \rightarrow XV$, 6. $U0 \rightarrow 0U$, 7. $U1 \rightarrow 1U$, 8. $V0 \rightarrow 0V$,
9. $V1 \rightarrow 1V$, 10. $UZ \rightarrow Z0$, 11. $VZ \rightarrow Z1$, 12. $XZ \rightarrow \lambda$,
13. $\lambda 0 \rightarrow XY0$, 14. $\lambda 1 \rightarrow XY1$.

$H_1 = \{XZ \rightarrow \lambda\}$.

Az algoritmus a következőképpen működik:

1. A bemenő szó elejére írja XY -t.
2. Y -t elvisszük a szó végére és átírjuk Z -re.
3. Ha az X utáni első számjegy 1, akkor V -t írunk helyette.
4. Ha az X utáni első számjegy 0, akkor U -t írunk helyette.
5. U -t vagy V -t elvisszük a szó végére.
6. Ha a Z elé U kerül, akkor 0-t írunk Z után, és ha van még szám X és Z közt, akkor vissza a 3-as pontra.
7. Ha a Z elé V kerül, akkor 1-t írunk Z után, és ha van még szám X és Z közt, akkor vissza a 3-as pontra.
8. XY törlése.

1.4. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szó első és utolsó karakterét addig törli, ameddig azok megegyeznek!

Például: 10110001 esetén a kimenet 1100.

Megoldás:

Egy algoritmus lehet ez: $W(\{0, 1, X, Y, U, V, W\}, H, H_1)$, ahol H elemei:

1. $0Y \rightarrow Y0$, 2. $1Y \rightarrow Y1$, 3. $0W \rightarrow W0$, 4. $1W \rightarrow W1$,
5. $XY0 \rightarrow X$, 6. $XY1 \rightarrow X$, 7. $XW0 \rightarrow 0$, 8. $XW1 \rightarrow 1$,
9. $U0 \rightarrow 0U$, 10. $U1 \rightarrow 1U$, 11. $V0 \rightarrow 0V$, 12. $V1 \rightarrow 1V$,

13. $0U \rightarrow Y$, 14. $1U \rightarrow W1$, 15. $1V \rightarrow Y$, 16. $0V \rightarrow W0$,
 17. $X1 \rightarrow X1W$, 18. $X0 \rightarrow X0U$, 19. $\lambda 0 \rightarrow X0$, 20. $\lambda 1 \rightarrow X1$.
 $H_1 = \{XW0 \rightarrow 0, XW1 \rightarrow 1\}$.

Az algoritmus a következőképpen működik:

1. A bemenő szó elejére ír egy X -et.
2. Ha az első számjegy 0, akkor U -t írunk utána.
3. Ha az első számjegy 1, akkor V -t írunk utána.
4. U -t vagy V -t elvisszük a szó végére.
5. Ha az utolsó jegy 0 és U került mellé, vagy ha 1 és V , akkor ezek helyére Y -t írunk.
6. Ha az utolsó jegy 1 és U került mellé, vagy ha 0 és V , akkor ezek helyére W -t írunk és visszaírjuk az utolsó jegyet.
7. Y -t vagy W -t az elejére visszük.
8. Ha az X után Y kerül, akkor az Y -t és az első számjegyet töröljük, és vissza a 2. lépésre.
9. Ha az X után W kerül akkor töröljük XW -t és megállunk.

1.5. feladat.

Adjon meg olyan Markov algoritmust, amely $1^m * 1^n$ bemenetre 1^{mn} kimenettel áll meg!

Megoldás:

$W(\{1, *, a, b\}, H, H_1)$, ahol H a következő:

1. $*11 \rightarrow a*1$, 2. $*1 \rightarrow a$, 3. $1a \rightarrow a1b$, 4. $ba \rightarrow ab$,
5. $b1 \rightarrow 1b$, 6. $a1 \rightarrow 1a$, 7. $ab \rightarrow b$, 8. $b \rightarrow 1$.

$H_1 = \emptyset$.

1.6. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szám esetén eggyel nagyobb bináris számot szolgáltat eredményül!

Megoldás:

$W(\{0, 1, X, Y\}, H, H_1)$, ahol H elemei:

1. $X1 \rightarrow 1X$, 2. $X0 \rightarrow 0X$, 3. $X \rightarrow Y$,
4. $0Y \rightarrow 1$, 5. $1Y \rightarrow Y0$, 6. $Y \rightarrow 1$, 7. $\lambda 1 \rightarrow X1$.

$H_1 = \{0Y \rightarrow 1, Y \rightarrow 1\}$.

Az algoritmus működése:

1. A szó elejére X -et írunk.
2. X -et a szó végére visszük és Y -ra cseréljük.
3. Y előtt 0 van, 1-esre írjuk és kész vagyunk.
4. Y előtt 1 van, 0-ra írjuk és Y -t írunk elé.
5. Ha Y előtt nincs semmi, akkor 1-esre írjuk át.

1.7. feladat.

Adjon meg olyan Markov algoritmust, amely 1^n bemenetre az n szám bináris alakját adja eredményül!

Megoldás:

$W(\{0, 1, X, Y\}, H, H_1)$, ahol H elemei: 1. $0Y \rightarrow 1$, 2. $1Y \rightarrow Y0$, 3. $Y \rightarrow 1$, 4. $X1 \rightarrow YX$, 5. $X \rightarrow \lambda$, 6. $\lambda 1 \rightarrow 0X1$.

$H_1 = \{X \rightarrow \lambda\}$

Az algoritmus lépései:

1. Az 1-esek elé $0X$ -et ír.
2. $X1$ -et YX -re cseréli.
3. Y előtti bináris számot növeli eggyel és Y -t törli.
4. Ha van X után 1-es vissza 2-es pontra.
5. Törli X -et.

1.8. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szám esetén eggyel kisebb bináris számot ad eredményül!

Megoldás:

$W(\{0, 1, X, Y\}, H, H_1)$, ahol H elemei:

1. $X1 \rightarrow 1X$, 2. $X0 \rightarrow 0X$, 3. $X \rightarrow Y$, 4. $1Y \rightarrow 0$, 5. $0Y \rightarrow Y1$, 6. $Y \rightarrow \lambda$, 7. $\lambda 1 \rightarrow X1$, még a záróhelyettesítés: $H_1 = \{Y \rightarrow \lambda\}$.

Az algoritmus működése:

1. A szó elejére X -et írunk.
2. X -et a szó végére visszük és Y -ra cseréljük.
3. Y előtt 1 van, 0-ra írjuk és kész vagyunk.
4. Y előtt 0 van, 1-re írjuk, és Y -t írunk elé.
5. Ha Y előtt nincs semmi, akkor töröljük Y -t, és kész vagyunk.

1.9. feladat.

Adjon meg olyan Markov algoritmust, amely egy bemenő bináris szám unáris alakját adja vissza!

Megoldás:

$W(\{0, 1, X, Y, Z, V\}, H, H_1)$, ahol H elemei:

1. $X1 \rightarrow 1X$, 2. $X0 \rightarrow 0X$, 3. $X \rightarrow Y$, 4. $1Y \rightarrow 0Y1$, 5. $0Y \rightarrow Z1V1$, 6. $0Z \rightarrow Z1$, 7. $1Z \rightarrow \lambda$, 8. $Z \rightarrow \lambda$, 9. $V \rightarrow Y$, 10. $Y \rightarrow \lambda$, 11. $\lambda 1 \rightarrow X1$, még $H_1 = \{Y \rightarrow \lambda\}$ záróhelyettesítés.

Az algoritmus lépései:

1. A szó elé X -et ír.
2. X -et a végére viszi és Y -ra cseréli.
3. Y előtti bináris számot csökkenti, utáni unárist növeli eggyel.
4. Ha van Y előtt számjegy, akkor vissza 2-ra.
5. Törli Y -t.

(A feladatot úgy is meg lehetett volna oldani, hogy az unáris szám kettővel való szorzását, illetve az egyet hozzáadunk eljárásokat alkalmazva számjegyenként haladunk végig a bináris szón.)

2. Grammatikák főbb típusai

Vegyünk egy V véges nemüres hamazt, melynek az elemeit nevezzük betűknek!

Legyen $V_N \subset V$ és $V_T \subset V$ a nemterminálisok és a terminálisok halmaza úgy, hogy $V = V_N \cup V_T$ és $V_N \cap V_T = \emptyset$!

$-A \subseteq V$ elemeiből álló véges jelsorozatot jelöljük A^* -gal, amelybe bele vesszük a λ -val jelölt üres szót is.

$-A \subseteq V$ elemeiből álló véges jelsorozatot jelöljük A^+ -szal, amelybe nem vesszük bele a λ -val jelölt üres szót.

$S \in V_N$ egy kitüntetett elem, a kezdőszimbólum.

Legyen $H \subseteq \{V^*V_NV^* \times V^*\}$, jelölve $H = \{P_1 \rightarrow Q_1, \dots, P_n \rightarrow Q_n | n \in \mathbb{N}\}$!
 $G = (V_N, V_T, S, H)$ rendszert generatív grammatikának hívjuk.

0-ás típusú mondunk egy grammatikát, ha a szabályokra nincs semmilyen megkötésünk.

1-es típusú egy grammatika, ha szabályai:

$P_1PP_2 \rightarrow P_1QP_2$ alakúak, ahol $P_1, P_2 \in V^*, P \in V_N, Q \in V^+$ vagy

$S \rightarrow \lambda \in H$, de ekkor S nem szerepel egyik szabály jobb oldalán sem.

2-es típusú egy grammatika, ha szabályai:

$X \rightarrow P$ alakúak, ahol $X \in V_N$ és $P \in V^*$.

3-as típusú egy grammatika, ha szabályai:

$X \rightarrow x$ vagy $X \rightarrow xY$ alakúak, ahol $X, Y \in V_N, x \in V_T^*$.

A G generatív grammatikában az **X szóból egy lépésben levezethető az Y szó**, ha van olyan $i \in \{1, 2, \dots, n\}$ és $P', P'' \in V^*$, hogy $X = P'P_iP''$ és $Y = P'Q_iP''$. Jelölésben: $X \Rightarrow Y$

A G generatív grammatikában az **X szóból levezethető az Y szó**, ha van olyan $k \in \mathbb{N}$ és $X_0, X_1, \dots, X_k \in V^*$, hogy $X = X_0, Y = X_k$ és $X_i \Rightarrow X_{i+1}$, ahol $i = 0, 1, \dots, k-1$.

Jelölésben: $X \Rightarrow^* Y$.

Egy G **grammatika által generált nyelv** alatt azoknak a szavaknak a halmazát értjük, melyek a kezdőszimbólumból levezethetők, és nem tartalmaznak nemterminálist.

$L(G) = \{P \mid S \Rightarrow^* P, P \in V_T^*\}$

Két grammatikát ekvivalensnek mondunk, ha az általuk generált nyelv megegyezik.

Egy nyelv i=0,1,2,3 típusú, ha van olyan i-típusú grammatika, amely azt generálja.

Az i-típusú nyelvek osztályát L_i -vel jelöljük.

$L_3 \subseteq L_2$ nyilvánvaló.

$L_1 \subseteq L_0$ szintén egyértelmű.

$L_2 \subseteq L_1$ a következő fejezetben kerül bizonyításra.

Ezek alapján: $L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$.

2. 1. példa.

Az alábbi szabályok melyik grammatika követelményeinek tesznek eleget, ha a kisbetűk terminálist, a nagybetűk nemterminálist jelölnek?

$A \rightarrow B$

3-as típusú: $A, B \in V_N$ és $\lambda \in V_T^*$, így ez a szabály $X \rightarrow xY$ alakú, ahol $X, Y \in V_N$, $x \in V_T^*$

$A \rightarrow BC$

2-es típusú $A \in V_N$, és $BC \in V^*$, így ez a szabály $X \rightarrow P$ alakú, ahol $X \in V_N$ és $P \in V^*$.

$XY \rightarrow XaB$

1-es típusú, mert $X \in V^*$, $Y \in V_N$, $\lambda \in V^*$, és $aB \in V^+$, vagyis ez a szabály $P_1PP_2 \rightarrow P_1QP_2$ alakú, ahol $P_1, P_2 \in V^*$, $P \in V_N$, $Q \in V^+$.

$XYZ \rightarrow XZY$

2-es vagy 3-as típusú nem lehet, mert a baloldalon nem csak egy nemterminális van.

1-es típusú sem lehet, mert a közös jobboldali rész λ , azaz P a P_1PP_2 szóból csak Z lehet, még a P_1 így XY , de a másik oldal nem ezzel kezdődik, vagyis ez a szabály nem lehet $P_1PP_2 \rightarrow P_1QP_2$ alakú, ahol $P_1, P_2 \in V^*$, $P \in V_N$, $Q \in V^+$, következésképpen a szabály 0-ás típusú.

2. 2. feladat.

Milyen típusú a következő grammatika?

$G = (\{S, A, B\}, \{x, y\}, S, H)$, ahol H a következő szabályokból áll:

$S \rightarrow AB$, $A \rightarrow BSB$,

$A \rightarrow BB$, $B \rightarrow xAy$,

$B \rightarrow \lambda$, $B \rightarrow x$,

$B \rightarrow y$.

Megoldás:

Minden szabálynál azt kell nézni, hogy a 0, 1, 2, 3 indexek közül melyik az a legnagyobb index, amelyhez tartozó követelményeinek eleget tesz.

$S \rightarrow AB$; $A \rightarrow BSB$; $A \rightarrow BB$; $B \rightarrow xAy$ megfelel a kettes típusú grammatika követelményeinek, de a hármasénak nem.

$B \rightarrow \lambda$; $B \rightarrow x$; $B \rightarrow y$ hármas típusú, ezért a grammatikánk mivel tartalmaz kettes és hármas típusú grammatikához tartozó szabályokat is, kettes típusú lesz.

2. 3. feladat.

Milyen típusú a következő grammatika?

$G = \langle \{S, A, B\}, \{a\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow ABa$, $AB \rightarrow AaBB$,

$B \rightarrow aaa$, $S \rightarrow AS$,

$AAS \rightarrow ABS$.

Megoldás:

$S \rightarrow ABa$ kettes típusú szabály,

$AB \rightarrow AaBB$ egyes típusú szabály,

$B \rightarrow aaa$ hármas típusú szabály,

$S \rightarrow AS$ kettes típusú szabály,

$AAS \rightarrow ABS$ egyes típusú szabály, ezért a grammatikánk 1-es típusú lesz.

2. 4. feladat.

Hányas típusú a következő grammatika?

$G = \langle \{S, A, B\}, \{a, b, c\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow ABc$, $A \rightarrow aB$,

$A \rightarrow Bc$, $B \rightarrow aAc$,

$B \rightarrow bc$.

Adjuk meg az összes hatbetűs szót a grammatika által generált nyelvben!

Megoldás:

Az összes szabály kettes típusú, ezért a grammatika is az.

Most határozzuk meg a hatbetűs szavakat!

Először is azt kell megállapítsuk, hogy a grammatika szabályai közül a baloldal mindenhol rövidebb, mint a jobboldal, vagyis egy szóból sehol sem kapunk rövidebbet, tehát a levezetéseket elég hatbetűs szavakig nézni, és azok közül azok lesznek a grammatika által generált nyelvben, melyek csak terminálist tartalmaznak. A levezetés első lépése mindenképpen az $S \rightarrow ABc$ lesz. Az A helyére vagy Bc vagy aB kerülhet csak. Mindkét esetben lesz egy négybetűs szavunk, melyben lesz két terminális és két B nemterminális. B helyére csak bc kerülhet, mert különben úgy kapunk hatbetűs szót, hogy még marad benne nemterminális. Ezért a levezetéseink a következők lesznek:

$S \Rightarrow ABc \Rightarrow aBBc \Rightarrow abcBc \Rightarrow abcbcc$

$S \Rightarrow ABc \Rightarrow BcBc \Rightarrow bccBc \Rightarrow bccbcc$

Természetesen van még ezen kívül több levezetés is, de az csak a nemterminálisok helyettesítésének sorrendjében tér el.

2. 5. feladat.

Hányas típusú a következő grammatika?

$G = \langle \{S\}, \{a, b, +, *, \cdot, \wedge, \vee, \neg, \rightarrow, \leftrightarrow\}, \{S, H\} \rangle$, ahol H szabályai a következők:

$S \rightarrow S + S, S \rightarrow S * S, S \rightarrow (S), S \rightarrow a, S \rightarrow b.$

Adjunk meg az $y * (x + y)$ szó egy levezetését!

Megoldás:

$S \rightarrow S + S ; S \rightarrow S * S ; S \rightarrow (S)$ kettes típusú szabályok.

$S \rightarrow x ; S \rightarrow y$ pedig hármas típusú szabályok, vagyis a grammatika kettes típusú lesz. Az $y * (x + y)$ szó egyik levezetése a következő:

$S \Rightarrow S * S \Rightarrow y * S \Rightarrow y * (S) \Rightarrow y * (S + S) \Rightarrow y * (x + S) \Rightarrow y * (x + y)$

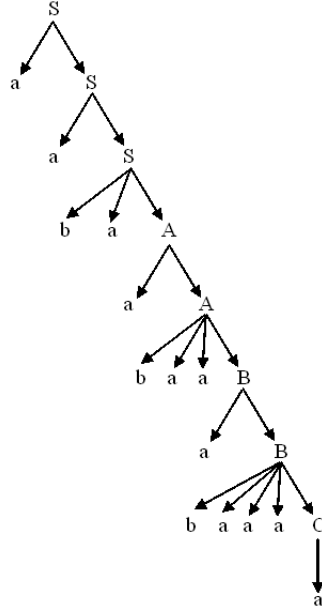
2. 6. feladat.

Adott a $G_1 = \langle \{S, A, B, C\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow baA, A \rightarrow aA, A \rightarrow baaB, B \rightarrow aB, B \rightarrow b, B \rightarrow baaaC, C \rightarrow aC, C \rightarrow a\} \rangle$ nyelvtan. Milyen típusú? Mely nyelvet generálja? Adjuk meg a $aabaabaaabaaaa$ szó levezetési fáját!

Megoldás. A nyelvtan szabályait megnézve, láthatjuk, hogy a $B \rightarrow b$ és $C \rightarrow a$ szabályok nem-terminális $\rightarrow$ terminális alakúak. Az összes többi szabály pedig nem-terminális $\rightarrow$ terminális(ok) – nem terminális alakú. Tehát a nyelvtan 3. típusú.

A generálás az első két szabály valamelyikével kezdődhet. Ahhoz, hogy az S szimbólum eltűnjön a mondatformából a második szabály használatára van szükség. Tehát az első szabály n -szeri (n tetszőleges nem-negatív egész) alkalmazásával $a^n S$ -et kapunk, majd a második szabállyal: $a^n baA$ -t. Ezután a harmadik szabály m -szeri alkalmazásával $a^n baam$ -et kapunk, majd a negyedik szabállyal: $a^n baam baab$ -t. Ezután a három alkalmazható szabály van. Az ötödik k -szori alkalmazása az $a^n baam baak$ alakhoz vezet. Ekkor a levezetés befejezhető a $B \rightarrow b$ szabállyal: $a^n baam baakb$. Alternatívaként a $B \rightarrow baaaC$ szabállyal folytatható a levezetés: $a^n baam baakbaaaC$. Ekkor a $C \rightarrow aC$ j -szeri alkalmazása a $a^n baam baakbaaa^jC$ formához vezet, a levezetés ekkor az utolsó szabállyal fejezhető be: $a^n baam baakbaaa^ja$ eredménnyel. Összefoglalva és felhasználva a Kleene-csillag jelölést a nyelv a következő reguláris kifejezéssel adható meg: $a^*baa^*ba^2a^*b(\lambda + a^4a^*)$.

Az $aabaabaaabaaaa$ szó levezetési fáját az ábra mutatja.



2. 7. feladat.

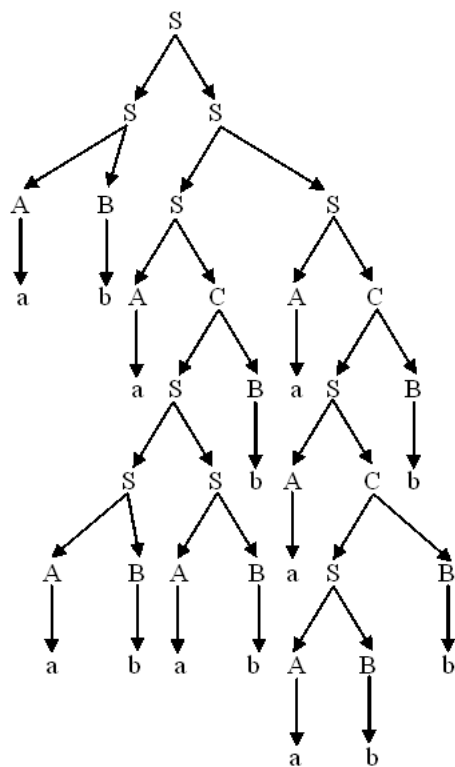
Adott a $G_2 = \langle \{S, A, B\}, \{a, b\}, S, H_2 \rangle$ nyelvtan, ahol $H_2 = \{S \rightarrow aA, A \rightarrow Sa, A \rightarrow a, S \rightarrow bB, B \rightarrow Sb, B \rightarrow b, S \rightarrow a, S \rightarrow b\}$. Milyen típusú? Mely nyelvet generálja? Adjuk meg a *abbabba* szó levezetési fáját!

Megoldás. Amint látjuk a szabályok közt van jobb-, illetve bal-lineáris alakú is, ezért a megadott grammatika nem reguláris. Az viszont minden szabályra teljesül, hogy a nyíl után legfeljebb egy nem-terminális betű szerepel. Tehát a nyelvtan lineáris.

A nyelvtan szabályait csoportokban vizsgálhatjuk. Az első szabály alkalmazása után a második vagy harmadik szabályt is fel kell használnunk a levezetésben. Az első után a harmadik szabállyal a levezetés befejeződik, az S szimbólum helyére aa kerül így a levezetésben. Az első és a második szabályok az eredeti S -t aSa -val helyettesítik. Hasonlóan a következő három szabály az S szimbólumot a bSb , illetve a bb szavakkal helyettesítheti. Ezek alapján könnyen belátható, hogy a generált szó az elejétől olvasva éppen ugyanaz, mint a végétől visszafelé olvasva. Az is látszik, hogy az $\{a, b\}$ ábécé felett minden ilyen szót előállít a nyelvtanunk. Az utolsó két szabály a páratlan hosszúságú ilyen tulajdonságú szavak levezetésének befejezésében játszik szerepet. Ezt a nyelvet egyébként a kétbetűs ábécé feletti *palindrom* nyelvnek nevezzük.

Az ábra a *abbabba* szó levezetési fáját mutatja.

Az ábra mutatja a *abaababbbaabbb* szó levezetési fáját.



3. Az üresszó-lemma

Minden környezetfüggetlen G nyelvtanhoz megadható vele ekvivalens G' nyelvtan, amelyre teljesülnek a következők:

$\lambda \notin L(G)$ esetén a G' -beli szabályok jobb oldalán λ nem fordul elő,

$\lambda \in L(G)$ esetén az egyetlen G' -beli szabály $S' \rightarrow \lambda$, ahol λ jobb oldalon megtalálható, és ekkor S' egyetlen szabály jobb oldalán sem fordul elő, ahol S' a G' kezdőszimbóluma.

3. 1. példa.

$G = (\{S, A, B\}, \{x, y\}, S, H)$, ahol H a következő szabályokból áll:

$S \rightarrow AB, A \rightarrow BSB, A \rightarrow BB, B \rightarrow xAy,$

$B \rightarrow \lambda, B \rightarrow x, B \rightarrow y.$

Először nézzük meg, hogy az üresszó benne van-e a G által generált nyelvben!

$U_1 := \{B\}$ - azok a nemterminálisok, amelyekből λ közvetlenül megkapható.

$U_2 := \{A, B\}$ - azok a nemterminálisok, amelyekből U_1^* -beli szavak közvetlenül megkaphatóak!

$U_3 := \{S, A, B\}$ - azok a nemterminálisok, amelyekből U_2^* -beli szavak közvetlenül megkaphatóak.

Mivel U_3 a nyelv összes nemterminálisát tartalmazza, ezért $U := U_3 = U_4 = U_5 = \dots$

Tehát U -beli nemterminálisokból kapható meg az üresszó.

Mivel $S \in U$, ezért az üresszó benne van a G által generált nyelvben.

Készítsük el a $G' = (\{S', S, A, B\}, \{x, y\}, S', H')$ grammatikát, ahol H' a következő szabályokból áll:

- $S' \rightarrow S, S' \rightarrow \lambda$. A második szabály azért került be a G' szabályai közé, mert $\lambda \in L(G)$, ezért $\lambda \in L(G')$ -nek is teljesülnie kell!

- Most vegyük azokat a H -beli szabályokat, amelyekben nem szerepel λ ! Ezek a következők: $S \rightarrow AB, A \rightarrow BSB, A \rightarrow BB, B \rightarrow xAy, B \rightarrow x, B \rightarrow y.$

- És vegyük még azokat a szabályokat, amelyek H -ban nem szerepeltek, és amelyeket úgy kapunk, hogy H -beli szabályok jobb oldaláról az összes lehetséges módon elhagyjuk U halmaz nemterminálisait, figyelve azonban arra, hogy a jobb oldalon ne λ maradjon:

$S \rightarrow A, S \rightarrow B,$

$A \rightarrow BS, A \rightarrow SB, A \rightarrow B, A \rightarrow S,$

$B \rightarrow xy.$

3. 2. feladat.

Készítsen a következő grammatikával ekvivalens 1-es típusú grammatikát!

$G = (\{S, A, B, C, D\}, \{a, b\}, S, H)$, ahol H szabályai:

$S \rightarrow AB, A \rightarrow CB, A \rightarrow SS, A \rightarrow a, B \rightarrow \lambda, C \rightarrow D, D \rightarrow \lambda, D \rightarrow b.$

Megoldás:

$U = \{S, A, B, C, D\}$

$S \in U$, ezért $\lambda \in L(G)$, vagyis új kezdőszimbólumot kell bevezetni.

$G' = (\{S', S, A, B, C, D\}, \{a, b\}, S', H')$

H' szabályai:

I. $S' \rightarrow S$ és $S' \rightarrow \lambda$, mert $\lambda \in L(G)$

II. $S \rightarrow AB, A \rightarrow CB, C \rightarrow D, A \rightarrow SS, A \rightarrow a, D \rightarrow b$ lesznek azok a szabályok H -ből melyekben nincs λ .

III. $S \rightarrow A, S \rightarrow B, A \rightarrow C, A \rightarrow B, A \rightarrow S$ lesznek az új szabályok, amelyeket U -beli nemterminálisok elhagyásával kapunk.

Ez a grammatika ekvivalens a G grammatikával, de már 1-es típusú.

3. 3. feladat.

Hozza a $G = (\{S, A, B\}, \{a, b\}, S, H)$ grammatikát 1-es típusúra!

$H = \{S \rightarrow \lambda, S \rightarrow AB, A \rightarrow SAB, A \rightarrow a, B \rightarrow S, B \rightarrow b\}.$

Megoldás:

$U = \{S, B\}$

$S \in U$, ezért $\lambda \in L(G)$, vagyis új kezdőszimbólumot kell bevezetni.

$G' = (\{S', S, A, B\}, \{a, b\}, S', H')$, ahol H' szabályai:

I. $S' \rightarrow S$ és $S' \rightarrow \lambda$, mert $\lambda \in L(G)$.

II. $S \rightarrow AB, A \rightarrow SAB, A \rightarrow a, B \rightarrow S, B \rightarrow b$ lesznek azok a szabályok H -ből, amelyekben nincs λ .

III. $S \rightarrow A, A \rightarrow SA, A \rightarrow AB$ lesznek az új szabályok, amelyeket U -beli nemterminálisok elhagyásával kapunk. Az $A \rightarrow A$ alakú szabályt nyugodtan elhagyhatjuk, mivel levezetés szempontjából nincs hatása.

3. 4. feladat.

Hozza a $G = (\{S, A, B, C, D, E\}, \{a, d\}, S, H)$ grammatikát 1-es típusúra!

$H = \{S \rightarrow SA, S \rightarrow BS, A \rightarrow a, B \rightarrow CDE, C \rightarrow DddE,$

$C \rightarrow d, C \rightarrow \lambda, D \rightarrow E, D \rightarrow d, E \rightarrow \lambda\}.$

Megoldás:

$U = \{B, C, D, E\}$

$S \notin U$, ezért $\lambda \notin L(G)$, vagyis nem kell új kezdőszimbólumot bevezetni.

$G' = (\{S, A, B, C, D, E\}, \{a, d\}, S, H')$

H' szabályai:

I. $S \rightarrow SA, S \rightarrow BS, A \rightarrow a, B \rightarrow CDE, C \rightarrow DddE, C \rightarrow d, D \rightarrow E, D \rightarrow d$ lesznek azok a szabályok H -ből melyekben nincs λ .

II. $B \rightarrow CD, B \rightarrow CE, B \rightarrow DE, B \rightarrow C, B \rightarrow D, B \rightarrow E, C \rightarrow ddE, C \rightarrow Ddd, C \rightarrow dd$ lesznek az új szabályok, melyeket U -beli nemterminálisok elhagyásával kapunk. ($S \rightarrow S$ alakú szabályt elhagytuk.)

3. 5. feladat.

Hozza a $G = (\{S, A, B, C\}, \{a, b\}, S, H)$ grammatikát 1-es típusúra!

$H = \{S \rightarrow ASC, S \rightarrow BA, B \rightarrow ACB, B \rightarrow AA, C \rightarrow bB, A \rightarrow \lambda, A \rightarrow a, B \rightarrow ba\}$

Megoldás:

$U = \{S, A, B\}$

$S \in U$, ezért $\lambda \in L(G)$, vagyis új kezdőszimbólumot kell bevezetni.

$G' = (\{S', S, A, B, C\}, \{a, b\}, S', H')$, ahol H' szabályai:

I. $S' \rightarrow S$ és $S' \rightarrow \lambda$, mert $\lambda \in L(G)$.

II. $S \rightarrow ASC, S \rightarrow BA, B \rightarrow ACB, B \rightarrow AA, C \rightarrow bB, A \rightarrow a, B \rightarrow ba$ lesznek azok a szabályok H -ből amelyekben nincs λ .

III. $S \rightarrow AC, S \rightarrow SC, S \rightarrow A, S \rightarrow B, B \rightarrow CB, B \rightarrow AC, B \rightarrow C, B \rightarrow A, C \rightarrow b$ lesznek az új szabályok, amelyeket U -beli nemterminálisok elhagyásával kapunk.

4. Chomsky-féle normálforma

Egy $G = (V_N, V_T, S, H)$ grammatikát **normális alakúnak** mondunk, ha minden terminálist tartalmazó szabálya $X \rightarrow x$ alakú, ahol $X \in V_N, x \in V_T$. (Minden 0, 1, 2 típusú grammatika normális alakra hozható.)

Ehhez tekintsük azt a G' grammatikát, amit a következőképpen konstruálhatunk meg!

Vezessünk be az összes terminális helyére -amely nem megfelelő alakú szabályban fordul elő- új nemterminálisokat, amelyek V_N -ben még nem szerepeltek! Jelöljük ezek halmazát A -val!

A $G' = (V'_N, V_T, S, H')$ lesz a megfelelő alakú új grammatika, ahol $V'_N := V_N \cup A$ és a H' -t pedig úgy hoztuk létre, hogy a H -beli szabályokban kicseréljük azokat a terminálisokat, amelyek nem megfelelő szabályokban fordulnak elő A -beli nemterminálisokra. Majd kiegészítjük $X \rightarrow x$ típusú szabályokkal, ahol $X \in A$, és x azon terminális, amely helyére X nemterminális került.

(Kettes típusú grammatika esetén elég csak azokban a szabályokban elvégezni a helyettesítést, amelyek nem normális alakúak.)

Minden G λ -mentes 2-es típusú grammatikához megadható vele ekvivalens **Chomsky-féle normálformájú** grammatika, azaz ami csak $X \rightarrow x$ és $X \rightarrow YZ$ alakú szabályokat tartalmaz.

(Ezeknek a grammatikáknak az az előnye, hogy a levezetésfa bináris fa lesz. pl. 2.9. feladat.)

A kettes típusú λ -mentes grammatikák esetén hasonló módszerrel, mint az üresszó-lemma esetén, könnyen elérhető, hogy ne legyen a grammatikánkban $X \rightarrow \lambda$ alakú szabály, ezért tegyük fel, hogy grammatikánk már olyan alakú, hogy csak $X \rightarrow p$ alakú szabályokat tartalmaz, ahol $X \in V_N$ és $P \in V^* \setminus \{\lambda\}$.

Ezután grammatikánkat három lépésben hozzuk Chomsky-féle normálalakra:

1. lépés: Hozzuk normális alakúra a grammatikánkat!

Ezután csak $X \rightarrow x$

$X \rightarrow X_1 X_2 \dots X_m$

$X \rightarrow YZ$

$X \rightarrow Y$ típusú szabályokat tartalmaz, ahol $X, Y, Z, X_1, \dots, X_m \in V_N$, $x \in V_T$ és $2 < m$, $m \in \mathbb{N}$.

2. lépés: Kiszöböljük ki az $X \rightarrow X_1 X_2 \dots X_m$ típusú szabályokat.

Vezessük be Y_1 -et $X_2 \dots X_m$ helyett, Y_2 -t $X_3 \dots X_m$ helyett, $\dots$ Y_{m-2} -t $X_{m-1} X_m$ helyett!

Majd vegyük fel az $X \rightarrow X_1 X_2 \dots X_m$ típusú szabály helyett az $X \rightarrow X_1 Y_1$, $Y_1 \rightarrow X_2 Y_2$, $\dots$; $Y_{m-2} \rightarrow X_{m-1} X_m$ szabályokat.

Például: $A \rightarrow BCDE$ helyett $A \rightarrow BX$, $X \rightarrow CY$, $Y \rightarrow DE$.

Ezzel már csak $X \rightarrow YZ$ és $X \rightarrow x$ és $X \rightarrow Y$ alakú szabályaink lesznek.

3. lépés: Az $X \rightarrow Y$ alakú szabályok kiküszöbölése.

Készítsük el az összes $X \rightarrow Y$ alakú szabály alapján az összes A_X és B_X halmazt, úgy, hogy A_X tartalmazza azokat a nemterminálisokat, melyekből X -hez juthatunk, még B_X tartalmazza azokat a nemterminálisokat, melyeket X -ből kaphatunk.

Ekkor az összes $X \rightarrow x$ típusú szabályban X -et helyettesítsük az összes A_X -beli elemmel, az $X \rightarrow YZ$ típusú szabályban X helyére A_X -beli, Y helyére B_Y -beli, Z helyére B_Z -beli nemterminálisokat írunk az összes lehetséges módon, majd az $X \rightarrow Y$ alakú szabályokat elhagyjuk.

Így már minden szabály Chomsky-féle normálformájú, és az így kapott grammatika ekvivalens az előzővel.

4. 1. példa.

Normalizáljuk a következő grammatikát!

$G = (\{A, B, S\}, \{a, b\}, S, H)$

$H = \{ S \rightarrow A, S \rightarrow Aa, A \rightarrow b, A \rightarrow B, B \rightarrow Bb, B \rightarrow a \}.$

A nem megfelelő szabályok: $S \rightarrow Aa$ és $B \rightarrow Bb$, vagyis a és b helyett vezessük be X és Y nemterminálisokat.

Cseréljük ki a -t és b -t az összes szabályban, ahol szerepelnek!

$S \rightarrow A, S \rightarrow AX, A \rightarrow Y, A \rightarrow B, B \rightarrow BY, B \rightarrow X.$

Majd vegyük hozzá az $X \rightarrow a$ és $Y \rightarrow b$ szabályokat!

Ezzel készen vagyunk H' -vel.

Tehát a G' grammatikánk, amely G -vel ekvivalens:

$G' = (\{S, A, B, X, Y\}, \{a, b\}, S, H').$

Mivel a G grammatika 2-es típusú, ezért $A \rightarrow b$ és $B \rightarrow a$ maradhatott volna változatlanul is, ekkor a H' szabályai: $S \rightarrow A, S \rightarrow AX, A \rightarrow b, A \rightarrow B, B \rightarrow BY, B \rightarrow a, X \rightarrow a$ és $Y \rightarrow b$ lennének.

4. 2.példa.

Készítsünk a következő grammatikával ekvivalens Chomsky-féle normálformájú grammatikát!

$G = (\{S, X, Y\}, \{a, b\}, S, H)$

$H = \{S \rightarrow X, S \rightarrow XY, X \rightarrow Y, X \rightarrow ababY, Y \rightarrow ab\}.$

1. lépés: Normalizáljuk a szabályokat.

Két olyan szabály van, melyben szerepel terminális, és nem $X \rightarrow x$ alakú. Ezekben a és b a terminális elemek, melyek helyett vegyük A és B nemterminálisokat.

Így a szabályaink olyanok lesznek, hogy:

$S \rightarrow X ; S \rightarrow XY$

$X \rightarrow Y ; X \rightarrow ABABY$

$Y \rightarrow AB ; A \rightarrow a$

$B \rightarrow b$

2. lépés: Küszöböljük ki az $X \rightarrow X_1X_2 \dots X_m$ alakú szabályokat!

Ebből csak egy van, az $X \rightarrow ABABY$.

Ekkor a szabályok ilyenek lesznek:

$S \rightarrow X, S \rightarrow XY,$

$X \rightarrow Y,$

$X \rightarrow AC, C \rightarrow BD, D \rightarrow AE, E \rightarrow BY,$

$Y \rightarrow AB, A \rightarrow a, B \rightarrow b.$

3. lépés: Küszöböljük ki az $X \rightarrow Y$ alakú szabályokat!

Ebből csak az $S \rightarrow X$ és $X \rightarrow Y$ van.

$A_S = \{S\}, B_S = \{S, X, Y\}$

$A_X = \{S, X\}, B_X = \{X, Y\}$

$A_Y = \{S, X, Y\}, B_Y = \{Y\}$

(Jelölés: $A_X \rightarrow B_Y B_Z$ azt jelöli, hogy a baloldalon az A_X halmaz egy eleme állhat, míg a jobboldal első betűje B_Y -ből, a második pedig B_Z halmazból

kerül ki az összes lehetséges módon.)

Az új szabályok: $S \rightarrow XY$ helyett $A_S \rightarrow B_X B_Y$ alakú szabályok:

$S \rightarrow XY$ és $S \rightarrow YY$.

$X \rightarrow AC$ helyett $A_X \rightarrow AC$ alakú szabályok:

$S \rightarrow AC$ és $X \rightarrow AC$.

$C \rightarrow BD$ marad,

$D \rightarrow AE$ marad.

$E \rightarrow BY$ helyett $E \rightarrow B B_Y$ alakú szabály: $E \rightarrow BY$.

$Y \rightarrow AB$ helyett $A_Y \rightarrow AB$ alakú szabályok:

$S \rightarrow AB$, $X \rightarrow AB$, $Y \rightarrow AB$.

$A \rightarrow a$ és $B \rightarrow b$ marad.

Ezzel a Chomsky-féle normálformára hozást befejeztük.

4. 3. feladat.

Készítsünk el a következő grammatikával ekvivalens Chomsky-féle normálformájú grammatikát!

$G = \langle \{S, A, B\}, \{a, b, c\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow ABc$, $B \rightarrow aAc$, $A \rightarrow aB$, $B \rightarrow bc$, $A \rightarrow Bc$

Megoldás:

1. lépés: Normalizáljuk a grammatikánkat.

A terminálisokat, amelyek nem $X \rightarrow x$ alakú szabályokban szerepelnek, helyettesítsük (a, b, c -t X, Y, Z -vel)!

$S \rightarrow ABZ$, $B \rightarrow XAZ$, $A \rightarrow XB$, $B \rightarrow YZ$, $A \rightarrow BZ$,

$X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$.

2. lépés: A két betűnél hosszabb jobboldalú szabályok kiküszöbölése.

Ilyen szabály a $S \rightarrow ABZ$ és a $B \rightarrow XAZ$, amelyeket kiválthatunk az $S \rightarrow AC$ és $C \rightarrow BZ$ illetve a $B \rightarrow XD$ és a $D \rightarrow AZ$ szabályokkal.

A szabályaink:

$S \rightarrow AC$, $C \rightarrow BZ$, $B \rightarrow XD$, $D \rightarrow AZ$,

$A \rightarrow XB$, $B \rightarrow YZ$, $A \rightarrow BZ$,

$X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$.

Ezek a szabályok már Chomsky-féle normálformában vannak.

4. 4. feladat.

Készítsünk el a következő grammatikával ekvivalens Chomsky-féle normálformájú grammatikát.

$G = \langle \{S, A, B, C\}, \{x, y\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow A$, $A \rightarrow BC$, $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$,

$B \rightarrow CC$, $C \rightarrow xBy$, $C \rightarrow xy$, $C \rightarrow x$, $B \rightarrow y$.

Megoldás:

1. lépés: Normalizáljuk a grammatikánkat:

A terminálisokat, amelyek nem $X \rightarrow x$ alakú szabályokban szerepelnek, helyettesítsük $(x, y$ -t X, Y -nal)!

$S \rightarrow A, A \rightarrow BC, A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow CC,$

$C \rightarrow XBY, C \rightarrow XY, C \rightarrow x, B \rightarrow y,$

$X \rightarrow x, Y \rightarrow y.$

2. lépés: A két betűnél hosszabb jobboldalú szabályok kiküszöbölése.

Ilyen szabály a $C \rightarrow XBY$, amelyet kiválthatunk az $C \rightarrow XD$ és $D \rightarrow BY$ szabályokkal.

$S \rightarrow A, A \rightarrow BC, A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow CC,$

$C \rightarrow XD, D \rightarrow BY,$

$C \rightarrow XY, C \rightarrow x, B \rightarrow y, X \rightarrow x, Y \rightarrow y.$

3. lépés: Az $X \rightarrow Y$ típusú szabályok kiiktatása.

Ezek: $S \rightarrow A, A \rightarrow B, A \rightarrow C, B \rightarrow C,$

$A_S = \{S\}, B_S = \{S, A, B, C\},$

$A_A = \{S, A\}, B_A = \{A, B, C\},$

$A_B = \{S, A, B\}, B_B = \{B, C\},$

$A_C = \{S, A, B, C\}, B_C = \{C\}.$

$G' = (\{S, A, B, C, D, X, Y\}, \{x, y\}, S, H'),$ ahol H' szabályai:

$A \rightarrow BC, S \rightarrow BC, A \rightarrow CC, S \rightarrow CC,$

$B \rightarrow CC, A \rightarrow CC,$

$C \rightarrow XD, A \rightarrow XD, B \rightarrow XD, S \rightarrow XD,$

$D \rightarrow BY, D \rightarrow CY,$

$C \rightarrow XY, S \rightarrow XY, A \rightarrow XY, B \rightarrow XY,$

$C \rightarrow x, S \rightarrow x, A \rightarrow x, B \rightarrow x,$

$B \rightarrow y, S \rightarrow y, A \rightarrow y,$

$X \rightarrow x, Y \rightarrow y.$

4. 5. feladat.

Készítsünk el a következő grammatikával ekvivalens Chomsky-féle normálformájú grammatikát!

$G = \langle \{S, A, B, C\}, \{0, 1\}, S, H \rangle,$ ahol H szabályai a következők:

$S \rightarrow 010, S \rightarrow C, S \rightarrow 0B1, A \rightarrow B, A \rightarrow 0, B \rightarrow A0,$

$B \rightarrow SAB, C \rightarrow B, S \rightarrow 0C1, C \rightarrow 1, C \rightarrow CC, A \rightarrow S$

Megoldás:

1. lépés: Normalizáljuk a grammatikánkat:

$S \rightarrow XYX, S \rightarrow C, S \rightarrow XBY, A \rightarrow B, A \rightarrow 0, B \rightarrow AX,$

$B \rightarrow SAB, C \rightarrow B, S \rightarrow XCY, C \rightarrow 1, C \rightarrow CC, A \rightarrow S,$

$X \rightarrow 0, Y \rightarrow 1.$

2. lépés: A két betűnél hosszabb jobboldalú szabályok kiküszöbölése:

Ilyen szabályok az $S \rightarrow XYX$, $S \rightarrow XBY$, $B \rightarrow SAB$, $S \rightarrow XCY$, amelyeket kiválthatunk az $S \rightarrow XD$, $D \rightarrow YX$, az $S \rightarrow XE$, $E \rightarrow BY$, a $B \rightarrow SF$, $F \rightarrow AB$ és az $S \rightarrow XG$, $G \rightarrow CY$ szabályokkal.

$S \rightarrow XD$, $D \rightarrow YX$, $S \rightarrow C$, $S \rightarrow XE$, $E \rightarrow BY$, $A \rightarrow B$,

$A \rightarrow 0$, $B \rightarrow AX$,

$B \rightarrow SF$, $F \rightarrow AB$, $C \rightarrow B$,

$S \rightarrow XG$, $G \rightarrow CY$, $C \rightarrow 1$,

$C \rightarrow CC$, $A \rightarrow S$, $X \rightarrow 0$, $Y \rightarrow 1$,

3. lépés: Az $X \rightarrow Y$ típusú szabályok kiiktatása.

Ezek: $S \rightarrow C$, $A \rightarrow B$, $C \rightarrow B$, $A \rightarrow S$.

$A_S = \{S, A\}$, $B_S = \{S, B, C\}$,

$A_A = \{A\}$, $B_A = \{S, A, B, C\}$,

$A_B = \{S, A, B, C\}$, $B_B = \{B\}$,

$A_C = \{S, A, C\}$, $B_C = \{B, C\}$.

$G' = (\{S, A, B, C, D, E, F, G, X, Y\}, \{0, 1\}, S, H')$, ahol H' szabályai:

$S \rightarrow XD$, $A \rightarrow XD$, $D \rightarrow YX$,

$S \rightarrow XE$, $A \rightarrow XE$, $E \rightarrow BY$,

$A \rightarrow 0$, $B \rightarrow AX$, $A \rightarrow AX$, $S \rightarrow AX$, $C \rightarrow AX$,

$B \rightarrow SX$, $A \rightarrow SX$, $S \rightarrow SX$, $C \rightarrow SX$,

$B \rightarrow BX$, $A \rightarrow BX$, $S \rightarrow BX$, $C \rightarrow BX$,

$B \rightarrow CX$, $A \rightarrow CX$, $S \rightarrow CX$, $C \rightarrow CX$,

$B \rightarrow SF$, $A \rightarrow SF$, $C \rightarrow SF$, $S \rightarrow SF$,

$F \rightarrow AB$, $F \rightarrow SB$, $F \rightarrow BB$, $F \rightarrow CB$,

$S \rightarrow XG$, $A \rightarrow XG$, $G \rightarrow CY$, $G \rightarrow BY$,

$C \rightarrow 1$, $A \rightarrow 1$, $S \rightarrow 1$,

$C \rightarrow CC$, $C \rightarrow BC$, $C \rightarrow CB$, $C \rightarrow BB$,

$A \rightarrow CC$, $A \rightarrow BC$, $A \rightarrow CB$, $A \rightarrow BB$,

$S \rightarrow CC$, $S \rightarrow BC$, $S \rightarrow CB$, $S \rightarrow BB$,

$X \rightarrow 0$, $Y \rightarrow 1$.

4. 6. feladat.

Készítsünk el a következő grammatikával ekvivalens Chomsky-féle normálformájú grammatikát!

$G = \langle \{S, A, B\}, \{a, b, c, +, (,)\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow S + A$, $S \rightarrow A$, $A \rightarrow AB$, $A \rightarrow B$,

$B \rightarrow (S)$, $B \rightarrow a$, $B \rightarrow b$, $B \rightarrow c$.

Megoldás:

1. lépés: Normalizáljuk a grammatikánkat.

A terminálisokat, amelyek nem $X \rightarrow x$ alakú szabályokban szerepelnek, helyettesítsük $(+, (,)$ -et C, D, E -vel)!

$S \rightarrow SCA, S \rightarrow A, A \rightarrow AB, A \rightarrow B,$

$B \rightarrow DSE, B \rightarrow a, B \rightarrow b, B \rightarrow c,$

$C \rightarrow +, D \rightarrow (, E \rightarrow)$.

2. lépés: A két betűnél hosszabb jobboldalú szabályok kiküszöbölése.

Ilyen szabályok az $S \rightarrow SCA, B \rightarrow DSE$, melyeket kiválthatunk az $S \rightarrow SX, X \rightarrow CA$ és a $B \rightarrow DY, Y \rightarrow SE$ szabályokkal.

$S \rightarrow SX, X \rightarrow CA, S \rightarrow A, A \rightarrow AB, A \rightarrow B,$

$B \rightarrow DY, Y \rightarrow SE, B \rightarrow a, B \rightarrow b, B \rightarrow c$

$C \rightarrow +; D \rightarrow (; E \rightarrow)$.

3. lépés: Az $X \rightarrow Y$ típusú szabályok kiiktatása.

Ezek: $S \rightarrow A, A \rightarrow B$.

$A_S = \{S\}, B_S = \{S, A, B\},$

$A_A = \{S, A\}, B_A = \{A, B\},$

$A_B = \{S, A, B\}, B_B = \{B\}.$

$G' = (\{S, A, B, C, D, E\}, \{a, b, c, +, (,)\}, S, H')$, ahol H' szabályai:

$S \rightarrow SX, X \rightarrow CA, X \rightarrow CB,$

$A \rightarrow AB, S \rightarrow AB, A \rightarrow BB, S \rightarrow BB,$

$B \rightarrow DY, S \rightarrow DY, A \rightarrow DY,$

$Y \rightarrow SE, Y \rightarrow AE, Y \rightarrow BE,$

$B \rightarrow a, B \rightarrow b, B \rightarrow c,$

$S \rightarrow a, S \rightarrow b, S \rightarrow c,$

$A \rightarrow a, A \rightarrow b, A \rightarrow c,$

$C \rightarrow +, D \rightarrow (, E \rightarrow)$.

5. Kuroda-féle normálforma

Egy G grammatika Kuroda-féle normálformában van, ha szabályai:

$X \rightarrow x, X \rightarrow Y, X \rightarrow YZ, XY \rightarrow ZV$ alakúak, ahol $X, Y, Z, V \in V_N,$
 $x \in V_T.$

Minden G λ -mentes környezetfüggő grammatikához megadható egy vele ekvivalens Kuroda-féle normálformában levő grammatika.

1. lépés: Normalizáljuk a grammatikát!

A szabályok ezután a következő formájúak lehetnek:

$X \rightarrow x$ alakú, ahol $X \in V_N, x \in V_T,$

$X \rightarrow Y$ alakú, ahol $X, Y \in V_N,$

$X \rightarrow YZ$ alakú, ahol $X, Y, Z \in V_N$,
 $X \rightarrow X_1 \dots X_m$ alakú, ahol $X, X_1, \dots, X_m$ és $m \geq 3$
 $XY \rightarrow ZV$ alakú, ahol $X, Y, Z, V \in V_N$,
 $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$, ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ ($m \geq 2, n \geq 3, n \geq m$).

2. lépés: Váltuk ki az $X \rightarrow X_1 X_2 \dots X_m$ típusú szabályokat!

Ez a lépés ugyanúgy történik, mint a Chomsky-féle normálalak esetében.

3. lépés: Küszöböljük ki az $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$ szabályokat,
ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ ($m \geq 2, n \geq 3, n \geq m$)!

Vezessük be az új $Z_1 \dots Z_n$ változókat és vegyük a következő szabályokat:

$$X_1 X_2 \rightarrow Y_1 Z_2,$$

$$Z_2 X_3 \rightarrow Y_2 Z_3,$$

$\vdots$

$$Z_{m-1} X_m \rightarrow Y_{m-1} Z_m,$$

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

$$Z_{m+1} \rightarrow Y_{m+1} Z_{m+2},$$

$\vdots$

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

Az összes ilyen szabályra végrehajtva ezt a helyettesítést, a grammatikánk már Kuroda-féle normálalakban lesz.

A harmadik lépés szemléltetése: $X_1 X_2 X_3 X_4 \rightarrow Y_1 Y_2 Y_3 Y_4 Y_5$ kiváltásával.

Nézzük, mi történik, ha egy szóban szerepel az $X_1 X_2 X_3 X_4$ szelet!

$\dots X_1 X_2 X_3 X_4 \dots$, és $X_1 X_2 \rightarrow Y_1 Z_2$

$\dots Y_1 Z_2 X_3 X_4 \dots$, és $Z_2 X_3 \rightarrow Y_2 Z_3$

$\dots Y_1 Y_2 Z_3 X_4 \dots$, és $Z_3 X_4 \rightarrow Y_3 Z_4$

$\dots Y_1 Y_2 Y_3 Z_4 \dots$, és $Z_4 \rightarrow Y_4 Y_5$

$\dots Y_1 Y_2 Y_3 Y_4 Y_5 \dots$

Látható, hogy az új szabályokat írva az $X_1 X_2 X_3 X_4 \rightarrow Y_1 Y_2 Y_3 Y_4 Y_5$ helyére, ugyanarra az eredményre jutunk.

5. 1. példa.

Alakítsuk Kuroda-féle normálformájúvá a G grammatikát!

$G = \langle \{S, A, B\}, \{a\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow ABa$, $AB \rightarrow AaBB$, $B \rightarrow aaa$, $S \rightarrow AS$, $AAS \rightarrow ABS$.

1. lépés: Normalizáljuk a grammatikánkat!

H_1 szabályai: $S \rightarrow ABC$, $AB \rightarrow ACBB$, $B \rightarrow CCC$, $S \rightarrow AS$,

$AAS \rightarrow ABS$, $C \rightarrow a$.

$G_1 = \langle \{S, A, B, C\}, \{a\}, S, H_1 \rangle$

2. lépés: Váltuk ki az $X \rightarrow X_1 X_2 \dots X_m$ típusú szabályokat!

H_2 szabályai: $S \rightarrow AD$ és $D \rightarrow BC$ kerül az $S \rightarrow ABC$ helyére, illetve $B \rightarrow CE$ és $E \rightarrow CC$ a $B \rightarrow CCC$ helyére, valamint a korábbi $AB \rightarrow ACBB$, $S \rightarrow AS$, $AAS \rightarrow ABS$ és $C \rightarrow a$ szabályok.

$G_2 = \langle \{S, A, B, C, D, E\}, \{a\}, S, H_2 \rangle$.

3. lépés: Küszöböljük ki az $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$ szabályokat, ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ $m \geq 2$, $n \geq 3$, $n \geq m$!

H_3 szabályai: $S \rightarrow AD$, $D \rightarrow BC$, $B \rightarrow CE$, $E \rightarrow CC$, $C \rightarrow a$ valamint $AB \rightarrow AF$, $F \rightarrow CG$, $G \rightarrow BB$ az $AB \rightarrow ACBB$ helyett és $AA \rightarrow AH$, $HS \rightarrow BS$ az $AAS \rightarrow ABS$ helyett, $G_3 = \langle \{S, A, B, C, D, E, F, G, H\}, \{a\}, S, H_3 \rangle$

5. 2. feladat.

Készítsük el a következő grammatikával ekvivalens Kuroda-féle normálformájú grammatikát!

$G = \langle \{S, A, B\}, \{a, b, d\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow AB$, $AB \rightarrow ABa$, $ABa \rightarrow Abbba$, $A \rightarrow ddaa$, $Abbba \rightarrow Bbbba$, $B \rightarrow a$.

Megoldás:

1. lépés: Normalizáljuk a grammatikát!

A terminálisokat, amelyek nem $X \rightarrow x$ alakú szabályokban szerepelnek, helyettesítsük (a, b, d -t X, Y, Z -vel)!

Legyen $H_1 = \{S \rightarrow AB, AB \rightarrow ABX, ABX \rightarrow AYYYYX, A \rightarrow ZZXX, AYYYYX \rightarrow BYYYYX, B \rightarrow a, X \rightarrow a, Y \rightarrow b, Z \rightarrow d\}$!

2. lépés: Váltssuk ki az $X \rightarrow X_1 X_2 \dots X_m$ típusú szabályokat, ahol

$X, X_1, X_2, \dots, X_m \in V_N$ és $m > 2$!

Ilyen az $A \rightarrow ZZXX$, amit cseréljünk ki $A \rightarrow ZC$, $C \rightarrow ZD$ és $D \rightarrow XX$ szabályokra!

$H_2 = \{S \rightarrow AB, B \rightarrow a, X \rightarrow a, Y \rightarrow b, Z \rightarrow d,$

$A \rightarrow ZC, C \rightarrow ZD, D \rightarrow XX,$

$AB \rightarrow ABX, ABX \rightarrow AYYYYX, AYYYYX \rightarrow BYYYYX\}$.

3. lépés: Küszöböljük ki az $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$ szabályokat,

ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ $m \geq 2$, $n \geq 3$, $n \geq m$!

Cseréljük ki az $AB \rightarrow ABX$ szabályt $AB \rightarrow AE$ és $E \rightarrow BX$ szabályokra, az $ABX \rightarrow AYYYYX$ -et $AB \rightarrow AF$, $FX \rightarrow YG$, $G \rightarrow YH$ és $H \rightarrow YX$ -re, $AYYYYYX \rightarrow BYYYYX$ -et pedig $AY \rightarrow BI$, $IY \rightarrow YJ$, $JY \rightarrow YK$ és $KX \rightarrow YX$ -re!

Így kapjuk a következő grammatikát, amely ekvivalens G -vel:

$G' = (\{S, A, B, C, D, E, F, G, H, I, J, K, X, Y, Z\}, \{a, b, d\}, S, H_3,$ ahol

$H_3 = \{S \rightarrow AB, B \rightarrow a, X \rightarrow a, Y \rightarrow b, Z \rightarrow d, A \rightarrow ZC, C \rightarrow ZD,$

$D \rightarrow XX, AB \rightarrow AE, E \rightarrow BX, AB \rightarrow AF, FX \rightarrow YG, G \rightarrow YH,$

$H \rightarrow YX, AY \rightarrow BI, IY \rightarrow YJ, JY \rightarrow YK, KX \rightarrow YX\}$.

5. 3. feladat.

Készítsük el a következő grammatikával ekvivalens Kuroda-féle normálformájú grammatikát!

$G = \langle \{S, A, B\}, \{a, b\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow SASA$, $SAS \rightarrow SBBS$, $SBBSA \rightarrow SBabaSA$, $S \rightarrow aa$, $B \rightarrow bab$,
 $A \rightarrow SB$,

Megoldás:

1. lépés: Normalizáljuk a grammatikát!

$H_1 = \{S \rightarrow SASA, SAS \rightarrow SBBS, SBBSA \rightarrow SBXYXSA, S \rightarrow XX, B \rightarrow YXY, A \rightarrow SB, X \rightarrow a, Y \rightarrow b\}$.

2. lépés: Váltuk ki az $X \rightarrow X_1X_2 \dots X_m$ típusú szabályokat, ahol $X, X_1, X_2, \dots, X_m \in V_N$ és $m > 2$!

$H_2 = \{S \rightarrow SC, C \rightarrow AD, D \rightarrow SA$ -az $S \rightarrow SASA$ helyett,
 $B \rightarrow YE, E \rightarrow XY$ - a $B \rightarrow YXY$ helyett,
 $S \rightarrow XX, A \rightarrow SB, X \rightarrow a, Y \rightarrow b, SAS \rightarrow SBBS, SBBSA \rightarrow SBXYXSA\}$.

3. lépés: Szüntessük meg az $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$ szabályokat, ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ $m \geq 2, n \geq 3, n \geq m$!

$H_3 = \{S \rightarrow SC, C \rightarrow AD, D \rightarrow SA, B \rightarrow YE, E \rightarrow XY, S \rightarrow XX, A \rightarrow SB, X \rightarrow a, Y \rightarrow b, SA \rightarrow SF, FS \rightarrow BG, G \rightarrow BS$ -az $SAS \rightarrow SBBS$ helyett,
 $SB \rightarrow SH, HB \rightarrow BI, IS \rightarrow XJ, JA \rightarrow YK, K \rightarrow XL, L \rightarrow SA$ szabályok az $SBBSA \rightarrow SBXYXSA$ helyett}.

A G -vel ekvivalens G' grammatika tehát:

$G' = (\{S, A, B, C, D, E, F, G, H, I, J, K, L, X, Y\}, \{a, b\}, S, H_3)$.

5. 4. feladat.

Adjunk a G grammatikával ekvivalens Kuroda-normálformájú grammatikát!

$G = \langle \{S, A, B, C\}, \{a, b\}, S, H \rangle$, ahol H szabályai a következők:

$S \rightarrow AB$, $A \rightarrow ABCA$, $SBCAb \rightarrow BCaab$, $B \rightarrow b$, $A \rightarrow b$, $C \rightarrow aaba$.

Megoldás:

1. lépés: Normalizáljuk a grammatikát!

$H_1 = \{S \rightarrow AB, A \rightarrow ABCA, SBCAY \rightarrow BCXXY, B \rightarrow Y, A \rightarrow Y, C \rightarrow XXYX, X \rightarrow a, Y \rightarrow b\}$.

2. lépés: Váltuk ki az $X \rightarrow X_1X_2 \dots X_m$ típusú szabályokat, ahol $X, X_1, X_2, \dots, X_m \in V_N$ és $m > 2$!

$H_2 = \{A \rightarrow AD, D \rightarrow BE, E \rightarrow CA, C \rightarrow XF, F \rightarrow XG, G \rightarrow YX, S \rightarrow AB, B \rightarrow Y, A \rightarrow Y, X \rightarrow a, Y \rightarrow b, SBCAY \rightarrow BCXXY\}$.

3. lépés: Szüntessük meg az $X_1 \dots X_m \rightarrow Y_1 \dots Y_n$ szabályokat, ahol $X_1, \dots, X_m, Y_1, \dots, Y_n \in V_N$ $m \geq 2, n \geq 3, n \geq m$!

$H_3 = \{A \rightarrow AD, D \rightarrow BE, E \rightarrow CA, C \rightarrow XF, F \rightarrow XG, G \rightarrow YX, S \rightarrow AB, B \rightarrow Y, A \rightarrow Y, X \rightarrow a, Y \rightarrow b, SB \rightarrow BH, HC \rightarrow CI, IA \rightarrow XJ, JY \rightarrow XY\}$.
 A G -vel ekvivalens $G' = (\{S, A, B, C, D, E, F, G, H, I, J, X, Y\}, \{a, b\}, S, H_3)$.

5. 5. feladat.

Adjunk a G grammatikával ekvivalens Kuroda-normálformájú grammatikát!
 $G = \langle \{S, A, B, C\}, \{a, b, c\}, S, H \rangle$, ahol H szabályai a következők:
 $S \rightarrow aBC, S \rightarrow aSBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc$.

Megoldás:

1. lépés: Normalizáljuk a grammatikát!

$H_1 = \{S \rightarrow XBC, S \rightarrow XSBC, CB \rightarrow BC, XB \rightarrow XY, YB \rightarrow YY, YC \rightarrow YZ, ZC \rightarrow ZZ, X \rightarrow a, Y \rightarrow b, Z \rightarrow c\}$.

2. lépés: Válsuk ki az $X \rightarrow X_1X_2 \dots X_m$ típusú szabályokat, ahol

$X, X_1, X_2, \dots, X_m \in V_N$ és $m > 2$!

$H_2 = \{S \rightarrow XD, D \rightarrow BC, S \rightarrow XE, E \rightarrow SD, CB \rightarrow BC, XB \rightarrow XY, YB \rightarrow YY, YC \rightarrow YZ, ZC \rightarrow ZZ, X \rightarrow a, Y \rightarrow b, Z \rightarrow c\}$.

Ezzel a grammatikánk már Kuroda-féle normálformában van, nincs szükség a korábbi feladatok megoldásában szereplő harmadik lépésre, tehát a keresett grammatika: $G' = (\{S, A, B, C, D, E, X, Y, Z\}, \{a, b, c\}, S, H_2)$.

6. Bar-Hillel lemma

Bármely L környezetfüggetlen nyelvhez megadható két természetes szám, p és q úgy, hogy minden $P \in L$ szó esetén, amelyre $|P| > p$ teljesül, felírható $P = UXWYZ$ alakban oly módon, hogy $|XWY| \leq q$, $XY \neq \lambda$, és minden $i \geq 0$ egész számra $UX^iWY^iZ \in L$ teljesül.

Ez a lemma röviden azt mondja ki, hogy egy környezetfüggetlen nyelvben minden elég hosszú szóhoz végtelen sok rokonszerkezetű további szó található.

6. 1. példa.

$L = \{a^i b^i c^i | i = 1, 2, \dots\}$ nyelv nem környezetfüggetlen.

Tegyük fel, hogy megadható olyan p és q , amely a Bar-Hillel lemma szerint minden környezetfüggetlen nyelv esetében létezik!

Legyen m olyan, hogy $a^m b^m c^m$ szó hosszabb p -nél!

Ekkor tekintsük azt, hogy $P_1 = a^m b^m c^m = UXWYZ \in L$, ahol $XY \neq \lambda$, és azt, hogy $P_2 = UX^2WY^2Z \in L$.

X és Y közül tartalmaznia kell valamelyiknek a -t, hiszen az a -k száma P_2 -ben nagyobb, mint P_1 -ben.

Hasonlóan X és Y valamelyikének tartalmaznia kell b és c -t is, vagyis X és Y valamelyike a három féle betű közül legalább kettőt tartalmaz.

Ha pedig X és Y valamelyike kétféle betűt is tartalmaz, akkor a második hatványa már nem lehet egy $a^i b^j c^i$ alakú szó részszoja, ami ellentmond a Bar-Hillel lemmának, vagyis L nem környezetfüggetlen.

($X = a...ab..b$ akkor $X^2 = a...ab...ba...ab...b$, ami nyilvánvalóan nem lehet egy az L -be tartozó szó részszoja.)

6. 2. feladat.

$$L = \{a^{n^2} | n = 1, 2, 3, \dots\}$$

Bizonyítsa be, hogy az L nyelv nem környezetfüggetlen!

Megoldás:

Tegyük fel, hogy L környezetfüggetlen, ekkor létezik ilyen p szám, és $UXWYZ = a^{m^2}$ hosszabb p -nél.

Legyen $|XY| = x!$

A lemma miatt $UXXWYYZ$, $UXXXWYYYYZ...$ is L -beli, ezért $m^2 + x, m^2 + 2x, m^2 + 3x...$ is négyzetszám. Kérdés tehát, hogy van-e olyan szigorúan monoton növekvő számtani sorozat, melynek minden eleme négyzetszám?

Mivel két szomszédos négyzetszám különbsége minden határon túl nő, ezért egyszer meghaladja x -et is, tehát nincs. Vagyis ellentmondásra jutottunk.

6. 3. feladat.

Környezetfüggetlen-e az $L = \{0^n 1^{2n} | n = 1, 2, \dots\}$ nyelv?

Ha igen, adjon meg hozzá környezetfüggetlen grammatikát, ha nem, bizonyítsa be!

Megoldás:

Mivel itt csak egyszerre két helyen "pumpáljuk" a szót, ezért ez nem mond ellent a Bar-Hillel lemmának:

$U := \lambda$

$X := 0$

$W := \lambda$

$Y := 11$

$Z := \lambda$

Ekkor $p=2, q=3$.

A grammatika pedig: $G = \langle \{S\}, \{0, 1\}, S, \{S \rightarrow 0S11, S \rightarrow 011\} \rangle$.

6. 4. feladat.

Környezetfüggetlen-e az $L = \{0^n 1^{2n} 0^m | n = 1, 2, \dots, m = 1, 2, \dots\}$ nyelv?

Ha igen, adjon meg hozzá környezetfüggetlen grammatikát, ha nem, bizonyítsa be!

Megoldás:

Mivel egyszerre(!!!) itt is csak két helyen “pumpáljuk” a szót, ezért ez nem mond ellent a Bar-Hillel lemmának:

$U := \lambda$

$X := 0$

$W := \lambda$

$Y := 11$

$Z := 0^m$.

Ekkor $p=m+2$, $q=3$.

$G = \langle \{S, A, B\}, \{0, 1\}, S, \{S \rightarrow AB, A \rightarrow 0A11, A \rightarrow 011, B \rightarrow 0, B \rightarrow 0B\} \rangle$ pedig egy grammatika, amely a keresett nyelvet állítja elő.

6. 5. feladat.

Környezetfüggetlen-e az $L = \{0^n 1^m 0^{2n} | n = 1, 2, \dots, m = 1, 2, \dots\}$ nyelv?

Megoldás:

Mivel egyszerre itt is csak két helyen “pumpáljuk” a szót, ezért ez nem mond ellent a Bar-Hillel lemmának.

A nehézséget az jelentheti, hogy a két rész, amely ugyanattól az értéktől függ, nincs egymás mellett.

Ekkor a W -nek nem kell üres szónak lennie, hanem egy függetlenül pumpálható résznek.

$U := \lambda$

$X := 0$

$W := 1^m$

$Y := 00$

$Z := \lambda$

$p=m+2$, $q=m+3$.

Vagyis a Bar-Hillel lemmával nem kerültünk ellentmondásba.

A $G = \langle \{S\}, \{0, 1\}, S, \{S \rightarrow 0S00, S \rightarrow 0A00, A \rightarrow 1A, A \rightarrow 1\} \rangle$

Tudtunk 2-es típusú grammatikát konstruálni, mely pontosan ezt a nyelvet generálja.

Először a két szélét “pumpáltuk”, majd a közepét töltöttük fel 1-esekkel.

6. 6. feladat.

$L = \{a^{p_i} | p_i \text{ az } i\text{-edik prim } i = 1, 2, \dots\}$

Bizonyítsa be, hogy az L nyelv nem környezetfüggetlen!

Megoldás:

Tegyük fel, hogy L környezetfüggetlen! Ekkor alkalmazható rá a Bar-Hillel lemma, vagyis, ha x olyan prím, hogy nagyobb a lemma által létező p -nél, és az XY hossza m , akkor $x + m, x + 2m, \dots, x + xm$ is prím. Márpedig $x + xm = x(1 + m)$ nem lehet prím. Vagyis ellentmondásra jutottunk.

6. 7. feladat.

$L = \{pp^{-1} | p^{-1} \text{ a } p \text{ tükörképe}\}$

Környezetfüggetlen-e az L nyelv?

Megoldás:

A grammatikának csak $S \rightarrow aSa$ és $S \rightarrow a$ típusú szabályokat kell tartalmaznia, tehát az.

6. 8. feladat.

$L = \{a^n | x = n^2 + 3n + 1, n = 1, 2, 3, \dots\}$

Bizonyítsa be, hogy az L nyelv nem környezetfüggetlen!

Megoldás:

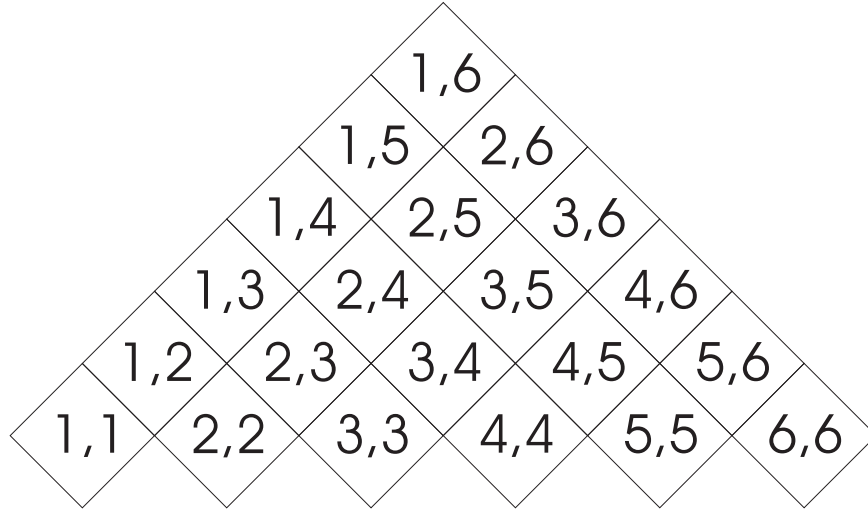
A bizonyítás azon múlik, hogy két szomszédos ilyen alakú szám távolsága is minden határon túl nőhet. Az $UXWYZ, UXXWYYZ, UXXXWYYYYZ, \dots$ hosszai pedig számtani sorozatot alkotnak.

7. Cook-Younger-Kasumi algoritmus

Ez az algoritmus annak eldöntésére szolgál, hogy egy Chomsky-féle normálformában lévő G grammatika által generált nyelvben benne van-e egy adott $a_1a_2 \dots a_n$ szó.

Az n betűs $a_1a_2 \dots a_n$ szó esetén egy $n \times n$ -es mátrix főátló feletti részére lesz szükségünk az algoritmushoz.

Például ez a mátrix egy hatbetűs szó esetén így néz ki:



A mátrix kitöltése azzal kezdődik, hogy a főátló i -edik cellájába írjuk be azokat a nemterminálisokat, amelyekből a_i ($i = 1, \dots, n$) közvetlenül levezethető!

Ezek után először azokat a cellákat töltjük ki, amelyeknél az első indexnél eggyel nagyobb a második, balról jobbra haladva.

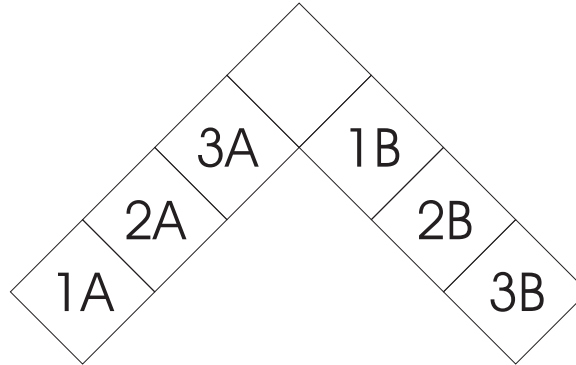
Majd melyeknél az első indexnél kétszer nagyobb a második stb.

Az (i, j) cella kitöltése:

1. $k := i$!
2. (i, j) cellába bekerül az X nemterminális, ha Y szerepel az (i, k) cellában, Z szerepel a $(k + 1, j)$ cellában, és $X \rightarrow YZ \in H$.
3. $k := k + 1$!
4. Ha $k < j$, akkor vissza a 2. pontra!

Ha a mátrix kitöltése után az $(1, n)$ cellában szerepel S , akkor az $a_1 a_2 \dots a_n$ szó benne van a G grammatika által generált nyelvben, különben nincs.

A könnyebb érthetőség kedvéért nézzük, hogy töltünk ki egy olyan cellát, ahol az első index 3-mal kisebb a másodiknál!



X bekerül az üres cellába, ha $X \rightarrow YZ$ szerepel a grammatika szabályai közt, és Y szerepel az 1A cellában és Z 1B-ben, vagy Y szerepel a 2A cellában és Z a 2B-ben, vagy Y szerepel a 3A cellában és Z a 3B-ben.

7. 1. példa.

Tekintsük a következő grammatikát:

$G = (\{S, A, B, X, Y, Z\}, \{a, b\}, S, H)$, ahol H szabályai:

$S \rightarrow AY$, $Y \rightarrow XB$, $X \rightarrow BA$, $X \rightarrow ZA$, $Z \rightarrow BX$, $A \rightarrow a$, $B \rightarrow b$.

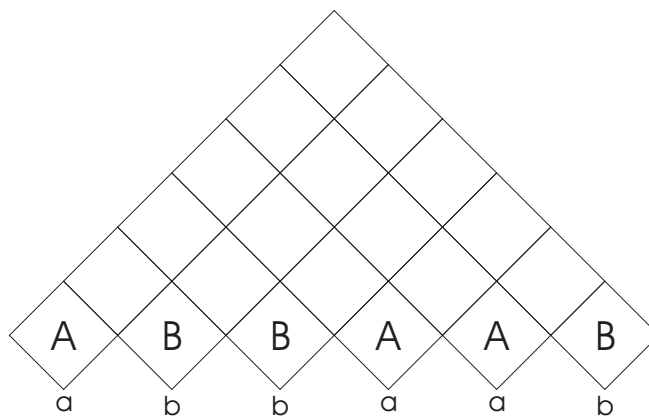
Benne van-e a G által generált nyelvben az *abbaab* szó?

Megoldás:

Készítsünk egy hatbetűs szóhoz mátrixot!

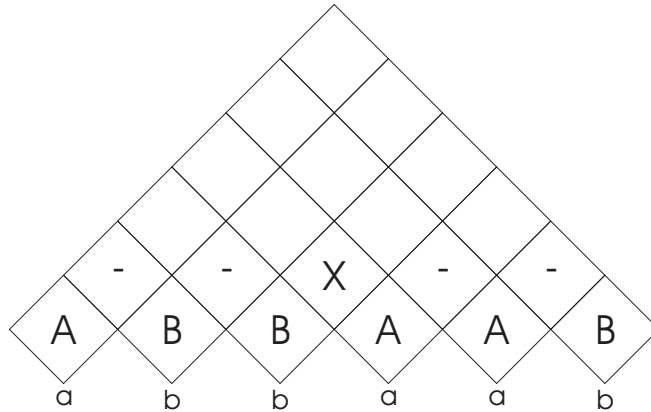
Ezt egy 6×6 -os mátrix főátló alatti részeinek elhagyásával kapjuk. Írjuk a nagyátló első cellájába azokat a nemterminálisokat, amelyből a szó első betűje közvetlenül levezethető!

Írjuk a nagyátló második cellájába azokat a nemterminálisokat, melyből a szó második betűje közvetlenül levezethető! stb.



Ezután azokat a cellákat töltjük ki, melyeknek második koordinátája eggyel nagyobb az első koordinátájától. (A továbbiakban a mátrix második sorát.)

- Az első cellába azokat a nemterminálisokat írjuk, melyekből AB közvetlenül levezethető. Ez üresen marad.
- A másodikba azokat, melyekből BB közvetlenül levezethető. Ez is üresen marad.
- A harmadikba azokat, melyekből BA közvetlenül levezethető. Ide $X \rightarrow BA$ miatt X kerül.
- A negyedikbe azokat, melyekből AA közvetlenül levezethető. Ez is üresen marad.
- Az ötödikbe azokat, melyekből AB közvetlenül levezethető. Ez is üresen marad.



Most a harmadik sor kitöltése következik:

- Az első cellába olyan nemterminálisok kerülhetnek, melyek olyan szabályok baloldalán állnak, ahol a jobboldal első betűje az $(1, 1)$, második betűje a $(2, 3)$ cellából, vagy első betűje az $(1, 2)$, második betűje a $(3, 3)$ cellából kerül ki.

Mivel a 2,3 cella és az 1,2 cella üres, ezért ez a cella is üresen marad.

- A második cellába olyan nemterminálisok kerülhetnek, melyek első betűje a $(2, 2)$, második betűje a $(3, 4)$ cellából, vagy első betűje a $(2, 3)$, második betűje a $(4, 4)$ cellából kerül ki.

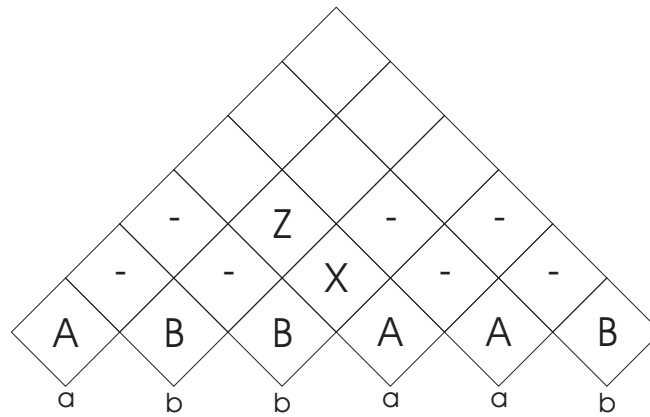
Ebbe a cellába a $Z \rightarrow BX$ szabály miatt Z kerül.

- A harmadik cellába olyan nemterminálisok kerülhetnek, melyek olyan szabályok bal oldalán állnak, ahol a jobb oldal első betűje a $(3, 3)$, második betűje a $(4, 5)$ cellából, vagy első betűje a $(3, 4)$, második betűje az $(5, 5)$ cellából kerül ki.

Mivel a $(4, 5)$ cella üres, és nincs olyan szabályunk, amelynek jobb oldalán XA áll, ezért ez a cella is üresen marad.

– A negyedik cellába olyan nemterminálisok kerülhetnek, amelyek olyan szabályok bal oldalán állnak, ahol a jobb oldal első betűje a $(4, 4)$, második betűje az $(5, 6)$ cellából, vagy első betűje a $(4, 5)$, második betűje a $(6, 6)$ cellából kerül ki.

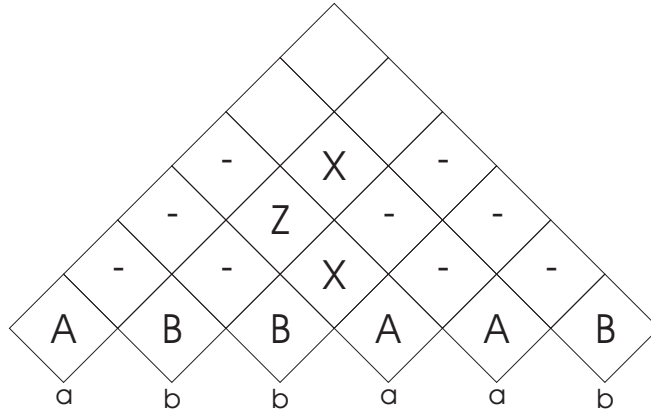
Mivel a $(4, 5)$ cella és az $(5, 6)$ cella üres, ezért ez a cella is üresen marad.



Most a negyedik sor kitöltése következik:

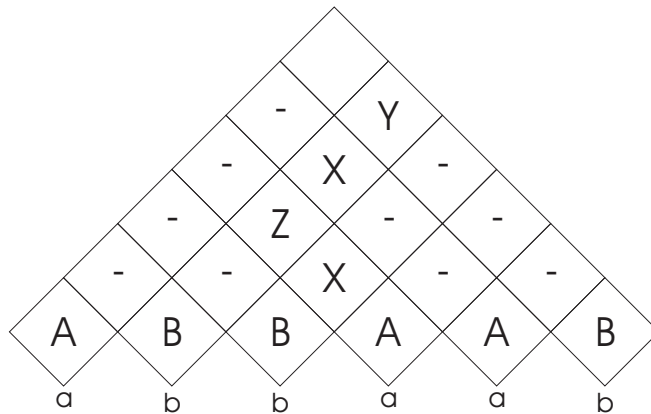
– Az első cella esetében az $(1, 1)$ cella a $(2, 4)$ cellával van párban, de olyan szabály, melynek jobb oldalán AZ van nincs. Az $(1, 2)$ és $(3, 4)$ párban az $(1, 2)$, az $(1, 3)$ és $(4, 4)$ párban az $(1, 3)$ cella üres cella, ezért ez a cellánk üresen marad.

– A második cella esetében a $(2, 2)$ cella párja a $(3, 5)$ cella üres cella, a $(2, 3)$ cella és párja a $(4, 5)$ cella is üres. A $(2, 4)$ cella tartalmazza a Z -t, az $(5, 5)$ cella pedig az A -t, és a grammatika $X \rightarrow ZA$ szabálya miatt cellánk tartalmazza X -et. – A harmadik cella esetén nincs olyan cella, melynek a párja ne lenne üres, ezért ez is üres lesz.



Az ötödik sor:

- Az első cellánál az (1, 1) cella párja a (2, 5) cella, és AX jobboldalú szabályt kell keresni, ami nincs, az (1, 2) az (1, 3) és az (1, 4) cellák pedig üresek, ezért ez a cella üresen marad.
- A második cella esetében a (2, 2) cella párja az üres (3, 6) cella, a (2, 3) cella üres, a (2, 4) cella párja az üres (5, 6) cella, még a (2, 5) cella X és a (6, 6) cella B betűje az $Y \rightarrow XB$ szabály jobb oldala, ezért a cellába Y kerül.

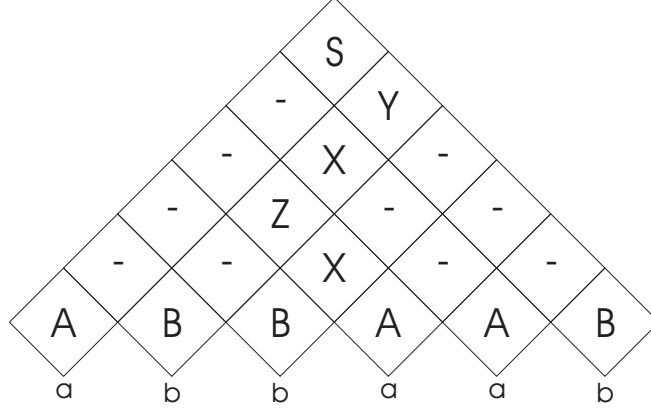


Az utolsó cella:

Az (1, 1) cellában A szerepel, a (2, 6)-ban Y , és van $S \rightarrow AY$ szabály, az (1, 2), az (1, 3), az (1, 5) és az (1, 5) cellák pedig üresek, ezért csak S kerül a cellába.

Mivel a legfelső (1, 6) cellában szerepel S , ezért az *abbbaab* szó benne van a grammatika által generált nyelvben.

A véglegesen kitöltött mátrixból leolvashatóak azok a levezetések, melyeken keresztül eljuthatunk az S kezdőszimbólumból az $abbaab$ szóig.



Az S kezdőszimbólum az $S \rightarrow AY$ miatt került be.

$S \Rightarrow AY$

A bal oldalon az A található, mely az $A \rightarrow a$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY$

Az Y az $Y \rightarrow XB$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB$

A bal oldalon az X található, mely az $X \rightarrow ZA$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB$

A bal oldalon a Z található, mely az $Z \rightarrow BX$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB$

A bal oldalon a B található, mely az $B \rightarrow b$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB$

A bal oldalon a X található, mely az $X \rightarrow BA$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB \Rightarrow abBAAB$

A bal oldalon a B található, mely a $B \rightarrow b$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB \Rightarrow abBAAB \Rightarrow abbAAB$

A bal oldalon az A található, mely a $A \rightarrow a$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB \Rightarrow abBAAB \Rightarrow abbAAB \Rightarrow abbaAB$

A bal oldalon az A található, mely a $A \rightarrow a$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB \Rightarrow abBAAB \Rightarrow abbAAB \Rightarrow abbaAB \Rightarrow abbaaB$

Majd a B maradt, mely a $B \rightarrow b$ szabály miatt került be.

$S \Rightarrow AY \Rightarrow aY \Rightarrow aXB \Rightarrow aZAB \Rightarrow aBXAB \Rightarrow abXAB \Rightarrow abBAAB \Rightarrow$

$abbAAB \Rightarrow abbaAB \Rightarrow abbaaB \Rightarrow abbaab$

Ha egy levezetésben mindig először a legbaloldalibb nemterminálist helyettesítjük, akkor egy legbaloldalibb levezetéshez jutunk, melynek a későbbiekben fontos szerepe lesz.

7. 2. feladat.

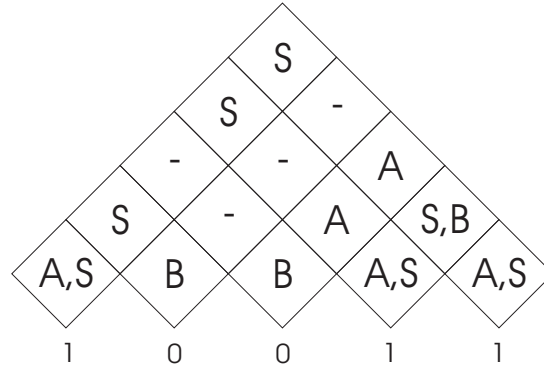
Tekintsük a következő grammatikát!

$G = (\{S, A, B\}, \{0, 1\}, S, H)$, ahol H szabályai:

$S \rightarrow SA, S \rightarrow AB, A \rightarrow BS, B \rightarrow SA, A \rightarrow 1, S \rightarrow 1, B \rightarrow 0$.

Bizonyítsuk be, hogy az 10011 szó benne van a grammatika által generált nyelvben, majd adjuk meg az összes legbaloldalibb levezetést!

Megoldás:



1. levezetésből: $S \Rightarrow SA \Rightarrow SAA \Rightarrow ABAA \Rightarrow 1BAA \Rightarrow 10AA \Rightarrow 10BSA \Rightarrow 100SA \Rightarrow 1001A \Rightarrow 10011$

Nézzük, hogyan kerülnek a levezetésbe az új nemterminálisok!

$S \rightarrow SA$: S (1, 4)-ből és A (5, 5)-ből;

$S \rightarrow SA$: S (1, 2)-ből és A (3, 4)-ből;

$S \rightarrow AB$: A (1, 1)-ből és B (2, 2)-ből;

$A \rightarrow BS$: B (3, 3)-ből és S (4, 4)-ből;

2. levezetésből: $S \Rightarrow SA \Rightarrow ABA \Rightarrow 1BA \Rightarrow 10A \Rightarrow 10BS \Rightarrow 100S \Rightarrow 100SA \Rightarrow 1001A \Rightarrow 10011$

$S \rightarrow SA$: S (1, 2)-ből és A (3, 5)-ből;

$S \rightarrow AB$: A (1, 1)-ből és B (2, 2)-ből;

$A \rightarrow BS$: B (3, 3)-ből és S (4, 5)-ből;

$S \rightarrow SA$: S (4, 4)-ből és A (5, 5)-ből.

7. 3. feladat.

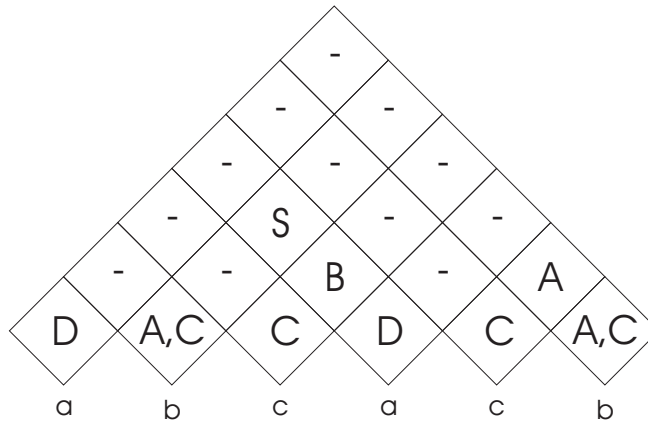
Tekintsük a következő grammatikát:

$G = (\{S, A, B, C, D\}, \{a, b, c\}, S, H)$, ahol H szabályai:

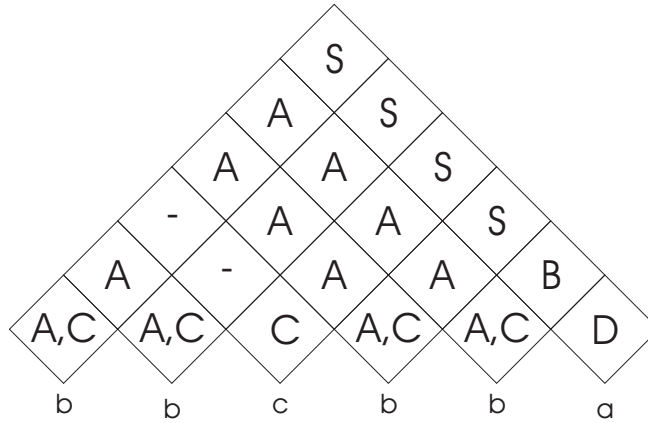
$S \rightarrow AB$, $A \rightarrow CA$, $A \rightarrow SS$, $B \rightarrow CD$, $A \rightarrow b$, $D \rightarrow a$, $C \rightarrow c$, $C \rightarrow b$.

Döntsük el, benne van-e a grammatika által generált nyelvben az $abcacb$ és a $bcbba$ szó, majd ha lehet adjuk meg az összes legbaloldalibb levezetését!

Megoldás:



Tehát mivel a legfelső cellában nem szerepel S , ezért az $abcacb$ szó nincs benne az adott grammatika által generált nyelvben.



Mivel a legfelső cellában szerepel S , ezért az $bcbba$ szó benne van az adott grammatika által generált nyelvben.

Az egyetlen legbaloldalibb levezetés pedig:

$S \Rightarrow AB \Rightarrow CAB \Rightarrow bAB \Rightarrow bCAB \Rightarrow bbAB \Rightarrow bbCAB \Rightarrow bbcAB \Rightarrow bcbB \Rightarrow bcbCD \Rightarrow bcbbbD \Rightarrow bcbba$

7. 4. feladat.

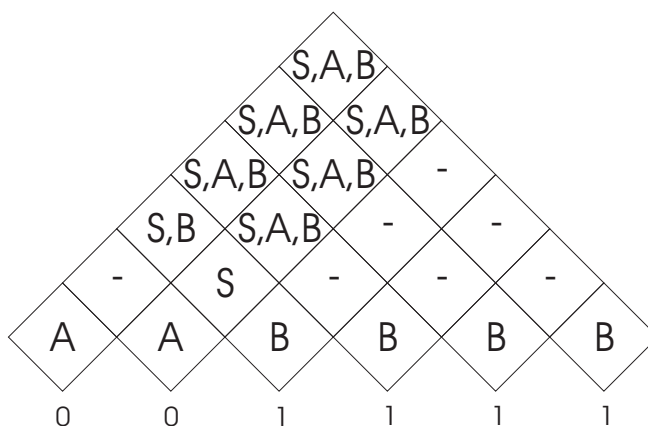
Tekintsük a következő grammatikát!

$G = (\{S, A, B\}, \{0, 1\}, S, H)$, ahol H szabályai:

$S \rightarrow AS, S \rightarrow SB, S \rightarrow AB, A \rightarrow BB, B \rightarrow SB; A \rightarrow SB, B \rightarrow AS, A \rightarrow 0, B \rightarrow 1.$

Döntsük el CYK-algoritmus segítségével, hogy levezethető-e a nyelvtanban a 001111 szó!

Megoldás:



Tehát mivel a legfelső cellában szerepel S , ezért a 001111 szó benne van az adott grammatika által generált nyelvben.

8. Early-algoritmus

Ez az algoritmus is annak az eldöntésére szolgál, hogy egy szó benne van-e egy 2-es típusú grammatika által generált nyelvben, vagy nem. Ennek az eldöntése is egy felismerési mátrix segítségével történik, de az algoritmus összetettebb, mint a CYK- algoritmus, viszont nem követeli meg, hogy a grammatikánk Chomsky- féle normálformára legyen hozva, sőt az algoritmus egy kis módosítással nem csak λ -mentes esetben használható. A mátrixunk celláiban úgynevezett "pontos" szabályok lesznek, és egy szó akkor lesz a grammatika által generált nyelv része, ha egy kitüntetett cellában lesz olyan szabály, aminek bal oldalán S áll, még a jobb oldala $\cdot$ -ra végződik.

A mátrixunk a következőképpen néz majd ki például egy 4-betűs szó esetén:

	a_1	a_2	a_3	a_4
0,0	0,1	0,2	0,3	0,4
	1,1	1,2	1,3	1,4
		2,2	2,3	2,4
			3,3	3,4
				4,4

A kitöltés után az $M_{0,4}$ cellát kell figyelni, hogy szerepel-e benne $S \rightarrow U$. alakú szabály, ahol $S \rightarrow U \in H$. A kitöltést a 0,0 cellából kell elkezdni, és haladunk balról jobbra, illetve alulról felfelé, de a főátlót akkor töltjük csak ki, ha az oszlop összes többi celláját már kitöltöttük. A sorrend tehát $M_{0,0}, M_{0,1}, M_{1,1}, M_{1,2}, M_{0,2}, M_{2,2}, M_{2,3}, M_{1,3}, M_{0,3}, M_{3,3}, M_{3,4}, M_{2,4}, M_{1,4}, M_{0,4}, M_{4,4}$. Látható, hogy az $M_{4,4}$ cella kitöltése már szükségtelen, mert előtte már tudjuk, hogy mi szerepel a $M_{0,4}$ cellában, vagyis el tudjuk dönteni, hogy a szó benne van-e a grammatika által generált nyelvben, vagy nincs. A kitöltéshez a következő algoritmust használjuk:

1. Minden $S \rightarrow W \in H$ szabályra legyen $S \rightarrow .W \in M_{0,0}$ és $j := 0$.
(Vagyis az S -ből kiinduló szabályokat írjuk a $M_{0,0}$ cellába, úgy, hogy a nyíl után egy pontot teszünk.)
2. Ha valamely $j \geq k$ -ra $A \rightarrow X.BY \in M_{k,j}$ valamely $X, Y \in V^*$ -ra, $A, B \in V_N$ -re, és $B \rightarrow U \in H$, valamely $U \in V^+$ -ra, akkor legyen $B \rightarrow .U \in M_{j,j}$!
(Ha az adott oszlopban van olyan nemterminális, amit pont előz meg, és szerepel H -beli szabály bal oldalán, akkor ezen H -beli szabályokat vegyük fel a diagonálisban lévő cellánkba, úgy, hogy a nyíl után pontot rakunk.)
3. Ha $j < n$, akkor $j := j + 1$, és $i := j - 1$. Ha $j = n$, akkor készen vagyunk, és az $a_1 a_2 \dots a_n$ szó benne van a grammatika által generált nyelvben, ha van a jobbfelső cellában S -ből induló pontra végződő szabály.
(Lépünk egy cellával jobbra, ha tudunk!)
4. Ha $A \rightarrow X.a_j Y \in M_{i,j-1}$ ahol $X, Y \in V^*$, $A \in V_N$ és a_j a szó j . betűje, akkor legyen $A \rightarrow XA_j.Y \in M_{i,j}$!

(Ha van a cellánk bal szomszédjában olyan szabály, ahol a pont az oszlopunk felé írt betűt előzi meg, "ugorjuk át" a ponttal az adott betűnket!)

5. Ha van olyan $i \leq k < j$, hogy $A \rightarrow X.BY \in M_{i,k}$ és $B \rightarrow U. \in M_{k,j}$, ahol $U \in V^+$, akkor legyen $A \rightarrow XBY \in M_{i,j}$!

(Ennek a lépésnek a szemléltetéséhez tekintsük a következő ábrát!

3B	2B	1B	3A
			2A
			1A

Ez az ábra a 3A cella kitöltéséhez készült. Vegyük azokat a cellákat, melyek a kitöltendőől balra, illetve lefele vannak! Megnézzük, hogy az 1A cellában van-e pontra végződő szabály, és annak milyen nemterminális szerepel a bal oldalán. Ha az 1B-ben ezt a nemterminálist pont előzi meg egy szabályban, akkor ezt a szabályt írjuk be a kitöltendő cellánkba, de úgy, hogy a .-al "átugorjuk" a nemterminálist. A kitöltés utána a 2A és 2B, majd a 3A és 3B cella összevetésével folytatódik.)

6. Ha $i > 0$, akkor $i := i - 1$ és térjünk a 4. lépésre.

(Ha nem vagyunk az oszlop tetején, akkor lépünk eggyel feljebb, és térjünk vissza az algoritmus 4. lépéséhez!)

7. Ha $i=0$, akkor térjünk a 2. lépésre!

8. 1. példa.

Tekintsük a következő grammatikát:

$G = (\{S, A, B\}, \{0, 1\}, S, \{S \rightarrow 1A0, S \rightarrow AS, A \rightarrow B0, B \rightarrow S0, S \rightarrow 0B1, A \rightarrow 1, B \rightarrow 0\})!$

Döntsük el Early-algoritmus segítségével, hogy levezethető-e a nyelvtanban a 011001 szó!

$M_{0,0}$ -ba az $S \rightarrow .1A0$, $S \rightarrow .AS$ és $S \rightarrow 0B1$ szabályokat írjuk az az 1. lépés alapján. $A \rightarrow .B0$ és $A \rightarrow .1$ szabályokat is írjuk az $M_{0,0}$ -ba a 2. lépés alapján, mivel az oszlopban van olyan szabály, ahol az A -betűt közvetlenül a pont előzi meg. Végül a $B \rightarrow .S0$ és $B \rightarrow .0$ szabályokat is írjuk az $M_{0,0}$ -ba a 2. lépés alapján, mivel az oszlopban van olyan szabály, ahol a B -betűt közvetlenül a pont előzi meg.

Lépünk egyet jobbra az algoritmus 3. lépése szerint!

$M_{0,1}$ -be $S \rightarrow 0.B1$ és $B \rightarrow 0.$ szabályok kerülnek a 4. lépés miatt, mert $M_{0,0}$ -ban a keresett szó első betűjét a pont előzi meg az $S \rightarrow .0B1$ és $B \rightarrow .0$ szabályokban. $A \rightarrow B.0$ -t szintén írjuk az $M_{0,1}$ -be az 5. lépés alapján, mivel $M_{0,1}$ -ben van pontra végződő szabály, melynek bal oldalán B áll, még az $M_{0,0}$ -ban az $A \rightarrow .B0$ szabályban a B -t a pont előzi meg.

Mivel az oszlop legfelső celláját kitöltöttük, ezért újra a 2. lépés következik.

$M_{1,1}$ cellába beírjuk a $B \rightarrow .S0$ és $B \rightarrow .0$ szabályokat, mert az oszlopban van olyan szabály, ahol a pont előzi meg a B -t. Ide kerülnek az S -ből induló szabályok is, mert már olyan szabály is van, ahol az S -t előzi meg a pont. Ezek lesznek az $S \rightarrow .1A0$, $S \rightarrow .AS$, $S \rightarrow .0B1$ szabályok. Illetve ugyanígy kerül be az $M_{1,1}$ cellába az $A \rightarrow .B0$ és az $A \rightarrow .1$ szabály is, mivel az oszlopban van olyan szabály, ahol az A -t előzi meg a pont.

Az első két oszlop kitöltése utáni ábrát mutatja a következő ábra.

0

S	→	.1A0	S	→	0.B1
S	→	.AS	B	→	0.
S	→	.0B1	A	→	B.0
A	→	.B0			
A	→	.1			
B	→	.S0			
B	→	.0			
			B	→	.S0
			B	→	.0
			S	→	.1A0
			S	→	.AS
			S	→	.0B1
			A	→	.B0
			A	→	.1

Ezután következik az algoritmus 3. lépése, vagyis egyet jobbra lépünk.

$M_{1,2}$ cellába kerülnek a 4. lépés alapján az $S \rightarrow 1.A0$ és $A \rightarrow 1.$ szabályok, mert az $M_{1,1}$ -ben az $A \rightarrow .1$ és $S \rightarrow .1A0$ szabályokban előzi meg pont a keresett szó második betűjét, az 1-et. Az 5. lépés alapján kerül be az $S \rightarrow A.S$ szabály, mert $M_{1,2}$ -ben pontra végződő szabály bal oldalán A betű áll, és az $M_{1,1}$ -ben az $S \rightarrow .AS$ szabályban követi a pontot A betű. Az algoritmus 6. lépése következik: mivel nem vagyunk az oszlop tetején, lépünk eggyel feljebb, és következzen újra a 4. lépés!

$M_{0,2}$ üres marad, mert az $M_{0,1}$ -ben nincs olyan szabály, ahol az 1-est pont előzné meg, és az 5. lépés alapján sem kerül be semmi ide, mert az $M_{1,2}$ -ben pontra végződő szabály bal oldalán csak A betű áll, de az $M_{0,1}$ -ben nincs olyan szabály, ahol pont előzi meg az A betűt. Mivel az oszlop tetején vagyunk, ezért következzen az oszlop diagonálisba eső cellájának kitöltése, az algoritmus 2. lépése után!

$M_{2,2}$ -be a 2. lépés alapján kerülnek az $A \rightarrow .B0$, $A \rightarrow .1$, $S \rightarrow .1A0$, $S \rightarrow .AS$, $S \rightarrow .0B1$ szabályok, mivel az oszlopban A -t és S -et előz meg pont, majd ezután lesz olyan szabály is, hogy B -t előz meg pont, ezért ide írjuk a $B \rightarrow .S0$ és $B \rightarrow .0$ szabályokat is.

Az első három oszlop után ilyen lesz a mátrixunk.

	0	1
$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	-
	$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$S \rightarrow 1.A0$ $A \rightarrow 1.$ $S \rightarrow A.S$
		$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$

Ezután következik az algoritmus 3. lépése, vagyis egyet jobbra lépünk.

$M_{2,3}$ cellába kerülnek a 4. lépés alapján az $A \rightarrow 1.$ és $S \rightarrow 1.A0$ szabályok, mert az $M_{2,2}$ -ben az $A \rightarrow .1$ és $S \rightarrow .1A0$ szabályokban előzi meg pont a keresett szó harmadik betűjét, az 1-et. Az 5. lépés alapján kerül be az $S \rightarrow A.S$ szabály, mert $M_{2,3}$ -ben pontra végződő szabály bal oldalán A betű áll, és az $M_{2,2}$ -ben az $S \rightarrow .AS$ szabályban. követi a pontot A betű. A 6. lépés alapján lépünk eggyel feljebb, majd az algoritmus 4. lépésére!

$M_{1,3}$ -ba a 4. lépés miatt nem kerül be szabály, hiszen a mellette levő cellában nincs olyan szabály, ahol az 1-est pont előzné meg, viszont bekerül az $S \rightarrow 1A.0$ az 5. lépés miatt, mivel az $M_{2,3}$ cellában van az $A \rightarrow 1.$ szabály, még $M_{1,2}$ -ben az $S \rightarrow 1.A0$ szabályban pont előzi meg az A -betűt. A 6. lépés szerint lépünk feljebb, és térjünk a 4. lépésre!

$M_{0,3}$ -ba sem a 4. lépés miatt nem kerül be semmi, hiszen a szomszédos cellában nem előzi meg pont az 1-est, sem a 5. lépés miatt nem kerül be semmilyen szabály.

$M_{3,3}$ -ba a 2. lépés miatt bekerülnek a H -ból az S -sel és A -val, majd miután ezeket beírtuk, a B -ből induló szabályok is. Az első négy oszlop kitöltése utáni ábra így néz ki.

	0	1	1
$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	-	-
	$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$S \rightarrow 1.A0$ $A \rightarrow 1.$ $S \rightarrow A.S$	$S \rightarrow 1A.0$
		$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$	$A \rightarrow 1.$ $S \rightarrow 1.A0$ $S \rightarrow A.S$
			$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$

Ezután következik az algoritmus 3. lépése, vagyis egyet jobbra lépünk.

$M_{3,4}$ -be ugyanaz kerül mint az $M_{0,1}$ -be, hiszen a helyzetük egyforma.

A 6. lépés alapján lépünk feljebb, és következzen a 4. lépés!

$M_{2,4}$ cella üres, mert $M_{2,3}$ -ban nincs pontot követő 0, még bár $M_{3,4}$ -ben van pontra végződő, B -ből induló szabály, de $M_{2,4}$ -ben nincs pont után B .

A 6. lépés alapján lépünk feljebb, és következzen a 4. lépés!

$M_{1,4}$ cellába kerül a 4. lépés miatt az $S \rightarrow 1A0$. szabály, mert $S \rightarrow 1A.0 \in M_{1,3}$. Az 5. lépés szerint $B \rightarrow S.0 \in M_{1,4}$, mert $S \rightarrow 1A0. \in M_{1,4}$ és $B \rightarrow .S0 \in M_{1,0}$. A 6. lépés alapján lépünk feljebb, és téjünk a 4. lépésre!

$M_{0,4}$ cellába a 4. lépés szerint nem kerül semmi, mert $M_{0,3}$ üres. És az 5. lépés szerint is üres marad, mert $B \rightarrow 0. \in M_{3,4}$, de $M_{0,3}$ üres, és bár $S \rightarrow 1A0. \in M_{1,4}$, de $M_{0,1}$ -ben nincs S -sel kezdődő szabály.

$M_{4,4}$ -be írjuk a B -vel, majd az S -sel, végül az A -val kezdődő szabályokat!

		0	1	1	0
S → .1A0 S → .AS S → .0B1 A → .B0 A → .1 B → .S0 B → .0	S → 0.B1 B → 0. A → B.0	-	-	-	
	B → .S0 B → .0 S → .1A0 S → .AS S → .0B1 A → .B0 A → .1	S → 1.A0 A → 1. S → A.S	S → 1A.0	S → 1A0. B → S.0	
		A → .B0 A → .1 S → .1A0 S → .AS S → .0B1 B → .S0 B → .0	A → 1. S → 1.A0 S → A.S	-	
			A → .B0 A → .1 S → .1A0 S → .AS S → .0B1 B → .S0 B → .0	S → 0.B1 B → 0. A → B.0	
				B → .S0 B → .0 S → .1A0 S → .AS S → .0B1 A → .B0 A → .1	

Ezután következik az algoritmus 3. lépése, vagyis egyet jobbra lépünk.

$M_{4,5}$ -be a 4. lépés miatt $B \rightarrow 0.$ és $S \rightarrow 0.B1$, még az 5. lépés szerint $B \rightarrow 0. \in M_{4,5}$ és $A \rightarrow .B0 \in M_{4,4}$ miatt $A \rightarrow B.0$ kerül .

A 6. lépés alapján lépünk feljebb, és térjünk a 4. lépésre!

$M_{3,5}$ -be a 4. lépés miatt $A \rightarrow B0.-t$, még az 5. alapján $B \rightarrow 0. \in M_{4,5}$ és $S \rightarrow 0.B1 \in M_{3,4}$ miatt $S \rightarrow 0B.1$ -et illetve $A \rightarrow B0. \in M_{3,5}$ valamint $S \rightarrow .AS \in M_{3,3}$ miatt $S \rightarrow A.S$ -t írunk.

A 6. lépés alapján lépünk feljebb, és térjünk a 4. lépésre!

$M_{2,5}$ -be $A \rightarrow B \in M_{3,5}$ és $S \rightarrow 1.A0 \in M_{2,3}$ miatt $S \rightarrow 1A.0$ -t írunk.

	0	1	1	0	0
$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	-	-	-	$S \rightarrow 0B.1$
	$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$S \rightarrow 1.A0$ $A \rightarrow 1.$ $S \rightarrow A.S$	$S \rightarrow 1A.0$	$S \rightarrow 1A0.$ $B \rightarrow S.0$	$B \rightarrow S0.$ $A \rightarrow B.0$
		$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$	$A \rightarrow 1.$ $S \rightarrow 1.A0$ $S \rightarrow A.S$	-	$S \rightarrow 1A.0$
			$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	$A \rightarrow B0.$ $S \rightarrow 0B.1$ $S \rightarrow A.S$
				$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$B \rightarrow 0.$ $S \rightarrow 0.B1$ $A \rightarrow B.0$
					$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$ $A \rightarrow .B0$ $A \rightarrow .1$

A 6. lépés alapján lépünk feljebb, és térjünk a 4. lépésre!

$M_{1,5}$ -be a $B \rightarrow S.0 \in M_{1,4}$ miatt $B \rightarrow S0.$ -t írjuk. Az 5. lépésben $B \rightarrow S0. \in M_{1,5}$ és $A \rightarrow .B0 \in M_{1,1}$ miatt $A \rightarrow B.0$ kerül.

Újra a 6. lépés alapján lépünk feljebb, és térjünk a 4. lépésre!

$M_{0,5}$ -be $B \rightarrow S0. \in M_{1,5}$ és $S \rightarrow 0.B1 \in M_{0,1}$ miatt az 5. lépés alapján $S \rightarrow 0B.1$ -et írjuk. Ezzel az oszlop alsó cellájára, és a 2. lépésre térhetünk.

$M_{5,5}$ -be írjuk az S -sel és a B -vel, majd az A -val kezdődő szabályokat. Ezzel az oszlop végére értünk, és a 3. lépés szerint lépünk egyet jobbra.

	0	1	1	0	0	1
$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	-	-	-	$S \rightarrow 0B.1$	$S \rightarrow 0B1.$ $B \rightarrow S.0$
$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$S \rightarrow 1.A0$ $A \rightarrow 1.$ $S \rightarrow A.S$	$S \rightarrow 1A.0$	$S \rightarrow 1A0.$ $B \rightarrow S.0$	$B \rightarrow S0.$ $A \rightarrow B.0$	$S \rightarrow AS.$ $B \rightarrow S.0$	
	$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$	$A \rightarrow 1.$ $S \rightarrow 1.A0$ $S \rightarrow A.S$	-		$S \rightarrow 1A.0$	$S \rightarrow AS.$
		$A \rightarrow .B0$ $A \rightarrow .1$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$	$S \rightarrow 0.B1$ $B \rightarrow 0.$ $A \rightarrow B.0$	$A \rightarrow B0.$ $S \rightarrow 0B.1$ $S \rightarrow A.S$	$S \rightarrow 0B1.$ $B \rightarrow S.0$	
			$B \rightarrow .S0$ $B \rightarrow .0$ $S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $A \rightarrow .B0$ $A \rightarrow .1$	$B \rightarrow 0.$ $S \rightarrow 0B.1$ $A \rightarrow B.0$	-	
				$S \rightarrow .1A0$ $S \rightarrow .AS$ $S \rightarrow .0B1$ $B \rightarrow .S0$ $B \rightarrow .0$ $A \rightarrow .B0$ $A \rightarrow .1$	$S \rightarrow 1.A0$ $A \rightarrow 1.$ $S \rightarrow A.S$	

$M_{5,6}$ -ba és $M_{1,2}$ -be a 4. és 5. lépés alapján ugyanazt írjuk, mivel helyzetük azonos, majd a 6. lépés alapján feljebb lépünk, és jön a 4. lépés.

$M_{4,6}$ -ba a 4. lépéssel nem kerül semmi, még az 5. miatt azért nem, mert $M_{5,6}$ -ban van egy $A \rightarrow 1$. szabály, de $M_{4,5}$ -ben nincs A előtt pont.

$M_{3,6}$ -ba a 4. lépéssel beírjuk a $S \rightarrow 0B1$ -et, még az 5. lépéssel $S \rightarrow 0B1. \in M_{3,6}$ és $B \rightarrow .S0 \in M_{3,3}$ miatt $B \rightarrow S.0$ -t.

$M_{2,6}$ -ba az 5. lépéssel az $S \rightarrow 0B1. \in M_{3,6}$ és $S \rightarrow A.S \in M_{2,3}$ miatt kerül $S \rightarrow AS$. szabály.

$M_{1,6}$ -ba két szabály: $S \rightarrow AS. \in M_{2,6}$ és $S \rightarrow A.S \in M_{1,2}$ miatt $S \rightarrow AS$. még $S \rightarrow AS. \in M_{1,6}$ és $B \rightarrow .S0 \in M_{1,1}$ miatt $B \rightarrow S.0$ kerül be az 5. lépés után, majd a 6. után feljebb lépünk, és jön a 4. lépés.

$M_{0,6}$ -ba a 4. lépés után kerül $S \rightarrow 0B1$. még az 5. alapján $B \rightarrow S.0$ az $S \rightarrow 0B1. \in M_{0,6}$ és $B \rightarrow .S0 \in M_{0,0}$ miatt.

Ezzel eljutottunk az algoritmus befejezéséhez, és mivel az $M_{0,6}$ -ban van S -sel kezdődő pontra végződő szabály, ezért azt mondhatjuk, hogy a 011001 szó benne van a fenti grammatika által generált nyelvben.

8. 2. feladat.

Tekintsük a $G = \langle \{S, A, B\}, \{a, b, c,), \rangle\}, S, H \rangle$ grammatikát, ahol H szabályai: $S \rightarrow S + A$, $S \rightarrow A$, $A \rightarrow AB$, $A \rightarrow B$, $B \rightarrow (S)$, $B \rightarrow a$, $B \rightarrow b$, $B \rightarrow c$!

Döntsük el Early-féle elemző algoritmus segítségével, benne van-e grammatika által generált nyelvben az $a(b + c)$ szó?

Megoldás:

A következő ábra mutatja, hogy az adott grammatika esetén az $a(b + c)$ szóhoz tartozó mátrix $M_{0,6}$ cellájában van olyan szabály, melynek bal oldalán S áll és pontra végződik, ezért a szó benne van a G grammatika által generált nyelvben.

a (b + c)						
$S \rightarrow .S+A$ $S \rightarrow .A$ $A \rightarrow .B$ $A \rightarrow .AB$ $B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	$B \rightarrow a.$ $A \rightarrow B.$ $A \rightarrow A.B$ $S \rightarrow A.$ $S \rightarrow S.+A$	-	-	-	-	$A \rightarrow AB.$ $S \rightarrow A.$ $A \rightarrow A.B$ $S \rightarrow S.+A$
	$B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	$B \rightarrow (.S)$	$B \rightarrow (S.)$	-	$B \rightarrow (S.)$	$B \rightarrow (S).$
		$S \rightarrow .S+A$ $S \rightarrow .A$ $A \rightarrow .B$ $A \rightarrow .AB$ $B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	$B \rightarrow b.$ $A \rightarrow B.$ $A \rightarrow A.B$ $S \rightarrow A.$ $S \rightarrow S.+A$	$S \rightarrow S+.A$	$S \rightarrow S+A.$ $S \rightarrow S.+A$	-
			$B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	-	-	-
				$A \rightarrow .B$ $A \rightarrow .AB$ $B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	$B \rightarrow .c$ $A \rightarrow B.$ $A \rightarrow A.B$	-
					$B \rightarrow .(S)$ $B \rightarrow .a$ $B \rightarrow .b$ $B \rightarrow .c$	-

II. fejezet

Automaták

1. Mealy automaták minimalizálása

1.1. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5, a_6\}, \{x, y\}, \{u, v, w\}, \delta, \lambda)$$

$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5	a_6
x	(a_2, u)	(a_1, w)	(a_1, u)	(a_5, w)	(a_4, u)	(a_3, u)
y	(a_3, v)	(a_2, v)	(a_4, v)	(a_4, v)	(a_3, v)	(a_2, v)

Mealy automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásához első lépésben osztályozzuk az állapotokat. Két állapot akkor és csak akkor esik egy osztályba, ha ugyanarra a bemenő jelre ugyanazzal a kimenő jellel reagálnak:

$$C_1 = \{\{a_1, a_3, a_5, a_6\}, \{a_2, a_4\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1, a_5\}, \{a_3, a_6\}, \{a_2, a_4\}\}.$$

$$C_3 = \{\{a_1, a_5\}, \{a_3\}, \{a_6\}, \{a_2, a_4\}\}.$$

$$C_4 = \{\{a_1, a_5\}, \{a_3\}, \{a_6\}, \{a_2, a_4\}\}.$$

Mivel $C_3 = C_4$, ezért az osztályozás véget ért. A C_3 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_1 = \{a_1, a_5\}, \quad b_2 = \{a_3\}, \quad b_3 = \{a_6\}, \quad b_4 = \{a_2, a_4\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_1, b_2, b_3, b_4\}$, a bemenő- és kimenő jeleinek a halmazai megegyeznek

az $\mathcal{A}$ automata megfelelő halmazai, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén és végül az aktuális b_i osztályhoz tartozó kimenőjelet megkapjuk, ha megnézzük, hogy az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely kimenőjelet szolgáltatott az $\mathcal{A}$ automata esetén. Jelen esetben:

$$\mathcal{A}_0 = (\{b_1, b_2, b_3, b_4\}, \{x, y\}, \{u, v, w\}, \delta', \lambda'),$$

$\mathcal{A}_0$	b_1	b_2	b_3	b_4
x	(b_4, u)	(b_1, u)	(b_2, u)	(b_1, w)
y	(b_2, v)	(b_4, v)	(b_4, v)	(b_4, v)

1.2. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5\}, \{x, y, z\}, \{u, v\}, \delta, \lambda)$$

$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5
x	(a_2, u)	(a_2, v)	(a_2, u)	(a_5, v)	(a_4, v)
y	(a_4, u)	(a_3, u)	(a_1, u)	(a_3, u)	(a_3, u)
z	(a_5, v)	(a_5, u)	(a_1, v)	(a_5, u)	(a_2, u)

Mealy automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásához első lépésben osztályozzuk az állapotokat. Két állapot akkor és csak akkor esik egy osztályba, ha ugyanarra a bemenő jelre ugyanazzal a kimenő jellel reagálnak:

$$C_1 = \{\{a_1, a_3\}, \{a_2, a_4, a_5\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1\}, \{a_3\}, \{a_2, a_4, a_5\}\}.$$

$$C_3 = \{\{a_1\}, \{a_3\}, \{a_2, a_4, a_5\}\}.$$

Mivel $C_2 = C_3$, ezért az osztályozás véget ért. A C_2 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_1 = \{a_1\}, \quad b_2 = \{a_3\}, \quad b_3 = \{a_2, a_4, a_5\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_1, b_2, b_3\}$, a bemenő- és kimenő jeleinek a halmazai megegyeznek az $\mathcal{A}$

automata megfelelő halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén és végül az aktuális b_i osztályhoz tartozó kimenőjelet megkapjuk, ha megnézzük, hogy az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely kimenőjelet szolgáltatottak az $\mathcal{A}$ automata esetén. Jelen esetben:

$$\mathcal{A}_0 = (\{b_1, b_2, b_3\}, \{x, y, z\}, \{u, v\}, \delta', \lambda'),$$

$\mathcal{A}_0$	b_1	b_2	b_3
x	(b_3, u)	(b_3, u)	(b_3, v)
y	(b_3, u)	(b_1, u)	(b_2, u)
z	(b_3, v)	(b_1, v)	(b_3, u)

1.3. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}, \{x, y\}, \{u, v\}, \delta, \lambda)$$

$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x	(a_8, u)	(a_3, v)	(a_8, u)	(a_1, v)	(a_8, u)	(a_4, u)	(a_3, v)	(a_5, v)
y	(a_2, v)	(a_1, u)	(a_7, u)	(a_2, v)	(a_7, v)	(a_2, v)	(a_6, u)	(a_3, v)

Mealy automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásához első lépésben osztályozzuk az állapotokat. Két állapot akkor és csak akkor esik egy osztályba, ha ugyanarra a bemenő jelre ugyanazzal a kimenő jellel reagálnak:

$$C_1 = \{\{a_1, a_5, a_6\}, \{a_2, a_7\}, \{a_3\}, \{a_4, a_8\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1, a_5, a_6\}, \{a_2, a_7\}, \{a_3\}, \{a_4\}, \{a_8\}\}.$$

$$C_3 = \{\{a_1, a_5\}, \{a_6\}, \{a_2, a_7\}, \{a_3\}, \{a_4\}, \{a_8\}\}.$$

$$C_4 = \{\{a_1, a_5\}, \{a_6\}, \{a_2\}, \{a_7\}, \{a_3\}, \{a_4\}, \{a_8\}\}.$$

$$C_5 = \{\{a_1\}, \{a_5\}, \{a_6\}, \{a_2\}, \{a_7\}, \{a_3\}, \{a_4\}, \{a_8\}\}.$$

Ebben az esetben, mivel a C_5 osztályozás esetén minden a_i állapot külön osztályba esik, ezért az $\mathcal{A}$ automata már minimális, így tovább nem minimalizálható, azaz $\mathcal{A}_0 = \mathcal{A}$.

2. Moore automaták minimalizálása

2.1. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5, a_6\}, \{x, y\}, \{u, v, w\}, \delta, \mu)$$

	u	v	u	u	w	w
$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5	a_6
x	a_4	a_1	a_4	a_1	a_1	a_3
y	a_1	a_3	a_6	a_4	a_6	a_5

Moore automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásának első lépése eltér a Mealy automaták esetén tárgyalt első lépéstől. Jelen esetben a kezdeti osztályozáskor két állapot akkor és csak akkor esik egy osztályba, ha ugyanaz az állapotjelük:

$$C_1 = \{\{a_1, a_3, a_4\}, \{a_2\}, \{a_5, a_6\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Tehát ez a lépés megegyezik a Mealy automata esetén tárgyalt lépéssel. Az osztályozás itt is véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1, a_4\}, \{a_2\}, \{a_3\}, \{a_5, a_6\}\}.$$

$$C_3 = \{\{a_1, a_4\}, \{a_2\}, \{a_3\}, \{a_5\}, \{a_6\}\}.$$

$$C_4 = \{\{a_1, a_4\}, \{a_2\}, \{a_3\}, \{a_5\}, \{a_6\}\}.$$

Mivel $C_3 = C_4$, ezért az osztályozás véget ért. A C_3 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_1 = \{a_1, a_4\}, \quad b_2 = \{a_2\}, \quad b_3 = \{a_3\}, \quad b_4 = \{a_5\}, \quad b_5 = \{a_6\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_1, b_2, b_3, b_4, b_5\}$, a bemenő- és kimenő jeleinek a halmazai megegyeznek az $\mathcal{A}$ automata megfelelő halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén és végül az aktuális b_i osztályhoz tartozó állapotjelet megkapjuk, ha megnézzük, hogy az adott b_i osztálybeli állapotokhoz mely állapotjel tartozott az $\mathcal{A}$ automata esetén. Jelen esetben:

$$\mathcal{A}_0 = (\{b_1, b_2, b_3, b_4, b_5\}, \{x, y\}, \{u, v, w\}, \delta', \mu'),$$

$\mathcal{A}_0$	u	v	u	w	w
	b_0	b_2	b_3	b_4	b_5
x	b_1	b_1	b_1	b_1	b_3
y	b_1	b_3	b_5	b_5	b_4

2.2. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}, \{x, y\}, \{u, v\}, \delta, \mu)$$

$\mathcal{A}$	u	u	v	u	u	v	v
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x	a_3	a_4	a_3	a_2	a_1	a_3	a_6
y	a_1	a_2	a_2	a_2	a_5	a_4	a_5

Moore automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A kezdeti osztályozáskor két állapot akkor és csak akkor esik egy osztályba, ha ugyanaz az állapotjelük:

$$C_1 = \{\{a_1, a_2, a_4, a_5\}, \{a_3, a_6, a_7\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1\}, \{a_2, a_4, a_5\}, \{a_3, a_6, a_7\}\}.$$

$$C_3 = \{\{a_1\}, \{a_2, a_4\}, \{a_5\}, \{a_3, a_6, a_7\}\}.$$

$$C_4 = \{\{a_1\}, \{a_2, a_4\}, \{a_5\}, \{a_3, a_6\}, \{a_7\}\}.$$

$$C_5 = \{\{a_1\}, \{a_2, a_4\}, \{a_5\}, \{a_3, a_6\}, \{a_7\}\}.$$

Mivel $C_4 = C_5$, ezért az osztályozás véget ért. A C_4 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_1 = \{a_1\}, \quad b_2 = \{a_2, a_4\}, \quad b_3 = \{a_5\}, \quad b_4 = \{a_3, a_6\}, \quad b_5 = \{a_7\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_1, b_2, b_3, b_4, b_5\}$, a bemenő- és kimenő jeleinek a halmazai megegyeznek az $\mathcal{A}$ automata megfelelő halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén

és végül az aktuális b_i osztályhoz tartozó állapotjelet megkapjuk, ha megnézzük, hogy az adott b_i osztálybeli állapotokhoz mely állapotjel tartozott az $\mathcal{A}$ automata esetén. Jelen esetben:

$$\mathcal{A}_0 = (\{b_1, b_2, b_3, b_4, b_5\}, \{x, y\}, \{u, v\}, \delta', \mu'),$$

$\mathcal{A}_0$	u	u	u	v	v
	b_0	b_2	b_3	b_4	b_5
x	b_4	b_2	b_1	b_4	b_4
y	b_1	b_2	b_3	b_2	b_3

2.3. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_1, a_2, a_3, a_4, a_5\}, \{x, y, z\}, \{u, v, w\}, \delta, \mu)$$

$\mathcal{A}$	u	v	v	w	w
	a_1	a_2	a_3	a_4	a_5
x	a_2	a_4	a_4	a_2	a_3
y	a_5	a_4	a_5	a_3	a_3
z	a_1	a_3	a_2	a_5	a_4

Moore automatához tartozó $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A kezdeti osztályozáskor két állapot akkor és csak akkor esik egy osztályba, ha ugyanaz az állapotjelük:

$$C_1 = \{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_1\}, \{a_2, a_3\}, \{a_4, a_5\}\}.$$

Jelen esetben már $C_2 = C_1$, ezért az osztályozás véget is ért. Az osztályokat jelöljük valamely új betűvel, például b -vel:

$$b_1 = \{a_1\}, \quad b_2 = \{a_2, a_3\}, \quad b_3 = \{a_4, a_5\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_1, b_2, b_3\}$, a bemenő- és kimenő jeleinek a halmazai megegyeznek az $\mathcal{A}$ automata megfelelő halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén és végül az

aktuális b_i osztályhoz tartozó állapotjelet megkapjuk, ha megnézzük, hogy az adott b_i osztálybeli állapotokhoz mely állapotjel tartozott az $\mathcal{A}$ automata esetén. Jelen esetben:

$$\mathcal{A}_0 = (\{b_1, b_2, b_3\}, \{x, y, z\}, \{u, v, w\}, \delta', \mu'),$$

$\mathcal{A}_0$	u	v	w
	b_0	b_2	b_3
x	b_2	b_3	b_2
y	b_3	b_3	b_2
z	b_1	b_2	b_3

3. Kimenő jel nélküli, iniciális, végállapotokkal bővített automaták minimalizálása

3.1. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_0, a_1, a_2, a_3, a_4, a_5\}, \{x, y\}, \delta, a_0, \{a_2, a_4, a_5\}),$$

$\mathcal{A}$	a_0	a_1	a_2	a_3	a_4	a_5
x	a_2	a_5	a_1	a_1	a_2	a_1
y	a_1	a_0	a_3	a_4	a_5	a_3

kimenő jel nélküli, iniciális, végállapotokkal bővített automatához tartozó, $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásának első lépése eltér mind a Mealy, mind pedig a Moore automaták esetén tárgyalt első lépéstől. Itt a kezdeti osztályozáskor mindig két osztályba kerülnek az állapotok, attól függően, hogy végállapotok, vagy pedig nem végállapotok.

$$C_1 = \{\{a_0, a_1, a_3\}, \{a_2, a_4, a_5\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Tehát ez a lépés megegyezik a Mealy és a Moore automata esetén tárgyalt lépéssel. Az osztályozás itt is véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_0, a_1\}, \{a_3\}, \{a_2, a_5\}, \{a_4\}\}.$$

$$C_3 = \{\{a_0, a_1\}, \{a_3\}, \{a_2, a_5\}, \{a_4\}\}.$$

Mivel $C_2 = C_3$, ezért az osztályozás már véget is ért. A C_2 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_0 = \{a_0, a_1\}, \quad b_1 = \{a_3\}, \quad b_2 = \{a_2, a_5\}, \quad b_3 = \{a_4\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_0, b_1, b_2, b_3\}$, a bemenő jeleinek a halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén, az új kezdőállapotunk az a b_i lesz, melynek eleme a_0 , és végül az $\mathcal{A}_0$ automata végállapotait azon $b_{k_1}, \dots, b_{k_m}$ osztályok alkotják, mely osztályok elemei tartalmaznak legalább egy elemet az $\mathcal{A}$ automata végállapotai közül. Jelen esetben:

$$\mathcal{A}_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, \delta', b_0, \{b_2, b_3\}),$$

$\mathcal{A}_0$	b_0	b_1	b_2	b_3
x	b_2	b_0	b_0	b_2
y	b_0	b_3	b_1	b_2

3.2. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_0, a_1, a_2, a_3, a_4\}, \{x, y\}, \delta, a_0, \{a_0, a_1, a_2, a_3\}),$$

$\mathcal{A}$	a_0	a_1	a_2	a_3	a_4
x	a_4	a_4	a_4	a_4	a_2
y	a_4	a_3	a_0	a_3	a_1

kimenő jel nélküli, iniciális, végállapotokkal bővített automatához tartozó, $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásának első lépéseként, a kezdeti osztályozáskor két osztályba kerülnek az állapotok, attól függően, hogy végállapotok, vagy pedig nem végállapotok.

$$C_1 = \{\{a_0, a_1, a_2, a_3\}, \{a_4\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$ valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_0\}, \{a_1, a_2, a_3\}, \{a_4\}\}.$$

$$C_3 = \{\{a_0\}, \{a_1, a_3\}, \{a_2\}, \{a_4\}\}.$$

$$C_4 = \{\{a_0\}, \{a_1, a_3\}, \{a_2\}, \{a_4\}\}.$$

Mivel $C_3 = C_4$, ezért az osztályozás véget ért. A C_3 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_0 = \{a_0\}, \quad b_1 = \{a_1, a_3\}, \quad b_2 = \{a_2\}, \quad b_3 = \{a_4\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_0, b_1, b_2, b_3\}$, a bemenő jeleinek a halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén, az új kezdőállapotunk az a b_i lesz, melynek eleme a_0 , és végül az $\mathcal{A}_0$ automata végállapotait azon $b_{k_1}, \dots, b_{k_m}$ osztályok alkotják, mely osztályok elemei tartalmaznak legalább egy elemet az $\mathcal{A}$ automata végállapotai közül. Jelen esetben:

$$\mathcal{A}_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, \delta', b_0, \{b_0, b_1, b_2\}),$$

$\mathcal{A}_0$	b_0	b_1	b_2	b_3
x	b_3	b_3	b_3	b_2
y	b_3	b_1	b_0	b_1

3.3. feladat. Készítsük el az Aufenkamp-Hohn-féle minimalizációs algoritmus segítségével az

$$\mathcal{A} = (\{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\}, \{x, y, z\}, \delta, a_0, \{a_2, a_4, a_5, a_7\}),$$

$\mathcal{A}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
x	a_6	a_3	a_1	a_7	a_1	a_0	a_4	a_1
y	a_4	a_7	a_0	a_2	a_3	a_1	a_5	a_6
z	a_5	a_2	a_6	a_1	a_3	a_6	a_0	a_6

kimenő jel nélküli, iniciális, végállapotokkal bővített automatához tartozó, $\mathcal{A}_0$ minimális állapotszámú automatát!

Megoldás: A feladat megoldásának első lépéseként, a kezdeti osztályozáskor két osztályba kerülnek az állapotok, attól függően, hogy végállapotok, vagy pedig nem végállapotok.

$$C_1 = \{\{a_0, a_1, a_3, a_6\}, \{a_2, a_4, a_5, a_7\}\}.$$

Ezek után a C_{i+1} -edik osztályozás esetén két állapot akkor esik egy osztályba, ha egyrészt a C_i -edik osztályozás esetén is azonos osztályba tartoztak, másrészt pedig minden bemenő jel hatására azonos C_i -beli osztályban található állapotokba mennek át. Az osztályozás véget ér, amennyiben $C_i = C_{i+1}$

valamely $i \geq 1$ esetén.

$$C_2 = \{\{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_4, a_5, a_7\}\}.$$

$$C_3 = \{\{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_5\}, \{a_4, a_7\}\}.$$

$$C_4 = \{\{a_0, a_1\}, \{a_3, a_6\}, \{a_2, a_5\}, \{a_4, a_7\}\}.$$

Mivel $C_3 = C_4$, ezért az osztályozás véget ért. A C_3 osztályait jelöljük valamely új betűvel, például b -vel:

$$b_0 = \{a_0, a_1\}, \quad b_1 = \{a_3, a_6\}, \quad b_2 = \{a_2, a_5\}, \quad b_3 = \{a_4, a_7\}.$$

Végül készítsük el az $\mathcal{A}$ automatával ekvivalens, minimális állapotszámú $\mathcal{A}_0$ automatát, mely állapothalmaza az osztályozás és az új jelölés bevezetése után $\{b_0, b_1, b_2, b_3\}$, a bemenő jeleinek a halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával, az átmenetfüggvényt megkapjuk úgy, hogy megnézzük az adott b_i osztálybeli állapotok az adott bemenő jel hatására mely b_j osztálybeli állapotokba mentek át az $\mathcal{A}$ automata esetén, az új kezdőállapotunk az a b_i lesz, melynek eleme a_0 , és végül az $\mathcal{A}_0$ automata végállapotait azon $b_{k_1}, \dots, b_{k_m}$ osztályok alkotják, mely osztályok elemei tartalmaznak legalább egy elemet az $\mathcal{A}$ automata végállapotai közül. Jelen esetben:

$$\mathcal{A}_0 = (\{b_0, b_1, b_2, b_3\}, \{x, y, z\}, \delta', b_0, \{b_2, b_3\}),$$

$\mathcal{A}_0$	b_0	b_1	b_2	b_3
x	b_1	b_3	b_0	b_0
y	b_3	b_2	b_0	b_1
z	b_2	b_0	b_1	b_1

4. Nemdeterminisztikus és determinisztikus véges automaták ekvivalenciája

4.1. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_0, a_1\}, \{x, y\}, \delta, \{a_0\}, \{a_1\}),$$

$\mathcal{A}$	a_0	a_1
x	$\{a_1\}$	$\{a_0, a_1\}$
y	$\{a_0\}$	—

nemdeterminisztikus, parciálisan definiált, kimenő jel nélküli, iniciális, végállapotokkal bővített véges automatával ekvivalens $\mathcal{A}_d$ determinisztikus, teljesen definiált automatát!

Megoldás: Az $\mathcal{A}_d$ automata állapothalmaza az $\mathcal{A}$ állapothalmazának a részhalmazai által alkotott halmaz lesz. Jelöljük egy új betűvel, például b -vel az $\{a_0, a_1\}$ halmaz részhalmazait:

$$b_0 = \{\emptyset\}, b_1 = \{a_0\}, b_2 = \{a_1\}, b_3 = \{a_0, a_1\}.$$

Az $\mathcal{A}_d$ automata bemenő jeleinek halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával. Az $\mathcal{A}_d$ automata átmenetfüggvényét a következőképpen kapjuk meg:

- (1) Ha az automata $b_0 = \{\emptyset\}$ állapotban van, akkor bármely bemenő jel hatására b_0 állapotban marad.
- (2) Ha az automata egy $b_i \neq b_0$ állapotban van, akkor meg kell vizsgálni, hogy a b_i halmazban lévő $a_{l_1}, \dots, a_{l_n}$ állapotok az adott bemenő jel hatására mely $a_{k_1}, \dots, a_{k_m}$ állapotokba mennek át. Az $\mathcal{A}_d$ automata a b_i állapotból az adott bemenő jel hatására az $a_{k_1}, \dots, a_{k_m}$ halmazhoz tartozó b_j állapotba megy át.

Az $\mathcal{A}_d$ automata kezdőállapota az $\mathcal{A}$ automata kezdőállapotainak a halmazához tartozó b_i lesz. Az $\mathcal{A}_d$ automata végállapotainak a halmaza pedig tartalmazni fog minden olyan b_j -t, melynek mint halmaznak eleme az $\mathcal{A}$ automata bármely végállapota. Jelen esetben:

$$\mathcal{A}_d = (\{b_0, b_1, b_2, b_3\}, \{x, y\}, \delta', b_1, \{b_2, b_3\}),$$

$\mathcal{A}_d$	b_0	b_1	b_2	b_3
x	b_0	b_2	b_3	b_3
y	b_0	b_1	b_0	b_1

4.2. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_0, a_1\}, \{x, y, z\}, \delta, \{a_0, a_1\}, \{a_0\}),$$

$\mathcal{A}$	a_0	a_1
x	$\{a_1\}$	—
y	—	$\{a_0, a_1\}$
z	$\{a_0, a_1\}$	$\{a_0\}$

nemdeterminisztikus, parciálisan definiált, kimenő jel nélküli, iniciális, végállapotokkal bővített véges automatával ekvivalens $\mathcal{A}_d$ determinisztikus, teljesen definiált automatát!

Megoldás: Az $\mathcal{A}_d$ automata állapothalmaza az $\mathcal{A}$ állapothalmazának a részhalmazai által alkotott halmaz lesz. Jelöljük egy új betűvel, például b -vel az $\{a_0, a_1\}$ halmaz részhalmazait:

$$b_0 = \{\emptyset\}, b_1 = \{a_0\}, b_2 = \{a_1\}, b_3 = \{a_0, a_1\}.$$

Az $\mathcal{A}_d$ automata bemenő jeleinek halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával. Az $\mathcal{A}_d$ automata átmenetfüggvényét a következőképpen kapjuk meg:

- (1) Ha az automata $b_0 = \{\emptyset\}$ állapotban van, akkor bármely bemenő jel hatására b_0 állapotban marad.
- (2) Ha az automata egy $b_i \neq b_0$ állapotban van, akkor meg kell vizsgálni, hogy a b_i halmazban lévő $a_{l_1}, \dots, a_{l_n}$ állapotok az adott bemenő jel hatására mely $a_{k_1}, \dots, a_{k_m}$ állapotokba mennek át. Az $\mathcal{A}_d$ automata a b_i állapotból az adott bemenő jel hatására az $a_{k_1}, \dots, a_{k_m}$ halmazhoz tartozó b_j állapotba megy át.

Az $\mathcal{A}_d$ automata kezdőállapota az $\mathcal{A}$ automata kezdőállapotainak a halmazához tartozó b_i lesz. Az $\mathcal{A}_d$ automata végállapotainak a halmaza pedig tartalmazni fog minden olyan b_j -t, melynek mint halmaznak eleme az $\mathcal{A}$ automata bármely végállapota. Jelen esetben:

$$\mathcal{A}_d = (\{b_0, b_1, b_2, b_3\}, \{x, y, z\}, \delta', b_3, \{b_1, b_3\}),$$

$\mathcal{A}_d$	b_0	b_1	b_2	b_3
x	b_0	b_2	b_0	b_2
y	b_0	b_0	b_3	b_3
z	b_0	b_3	b_1	b_3

4.3. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_0, a_1, a_2\}, \{x, y\}, \delta, \{a_0\}, \{a_1, a_2\}),$$

$\mathcal{A}$	a_0	a_1	a_2
x	$\{a_2\}$	$\{a_0\}$	$\{a_1, a_2\}$
y	$\{a_1\}$	$\{a_0, a_1, a_2\}$	—

nemdeterminisztikus, parciálisan definiált, kimenő jel nélküli, iniciális, végállapotokkal bővített véges automatával ekvivalens $\mathcal{A}_d$ determinisztikus, teljesen definiált automatát!

Megoldás: Az $\mathcal{A}_d$ automata állapothalmaza az $\mathcal{A}$ állapothalmazának a részhalmazai által alkotott halmaz lesz. Jelöljük egy új betűvel, például b -vel az $\{a_0, a_1\}$ halmaz részhalmazait:

$$b_0 = \{\emptyset\}, \quad b_1 = \{a_0\}, \quad b_2 = \{a_1\}, \quad b_3 = \{a_2\},$$

$$b_4 = \{a_0, a_1\}, \quad b_5 = \{a_1, a_2\}, \quad b_6 = \{a_0, a_2\}, \quad b_7 = \{a_0, a_1, a_2\}.$$

Az $\mathcal{A}_d$ automata bemenő jeleinek halmaza megegyezik az $\mathcal{A}$ automata bemenő jeleinek a halmazával. Az $\mathcal{A}_d$ automata átmenetfüggvényét a következőképpen kapjuk meg:

- (1) Ha az automata $b_0 = \{\emptyset\}$ állapotban van, akkor bármely bemenő jel hatására b_0 állapotban marad.
- (2) Ha az automata egy $b_i \neq b_0$ állapotban van, akkor meg kell vizsgálni, hogy a b_i halmazban lévő $a_{l_1}, \dots, a_{l_n}$ állapotok az adott bemenő jel hatására mely $a_{k_1}, \dots, a_{k_m}$ állapotokba mennek át. Az $\mathcal{A}_d$ automata a b_i állapotból az adott bemenő jel hatására az $a_{k_1}, \dots, a_{k_m}$ halmazhoz tartozó b_j állapotba megy át.

Az $\mathcal{A}_d$ automata kezdőállapota az $\mathcal{A}$ automata kezdőállapotainak a halmazához tartozó b_i lesz. Az $\mathcal{A}_d$ automata végállapotainak a halmaza pedig tartalmazni fog minden olyan b_j -t, melynek mint halmaznak eleme az $\mathcal{A}$ automata bármely végállapota. Jelen esetben:

$$\mathcal{A}_d = (\{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7\}, \{x, y\}, \delta', b_1, \{b_2, b_3, b_4, b_5, b_6, b_7\}),$$

$\mathcal{A}_d$	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
x	b_0	b_3	b_1	b_5	b_6	b_7	b_5	b_7
y	b_0	b_2	b_7	b_0	b_7	b_7	b_2	b_7

5. Ekvivalens reguláris nyelvtanok megadása reguláris kifejezésekhez

5.1. feladat. Adjunk meg az

$$L = \{a^*b \cup b^*a\}$$

reguláris kifejezéssel megadott L nyelvet generáló reguláris nyelvtant!

Megoldás: A feladat megoldásához első lépésben vezessünk be minden terminális betűhöz egy reguláris nyelvtant, mely kizárólag az adott terminálist generálja. Jelen esetben legyenek a kiinduló nyelvtanok a következők: $G_a = (\{A\}, \{a\}, A, \{A \rightarrow a\})$, $G_b = (\{B\}, \{b\}, B, \{B \rightarrow b\})$. Ezek után a kifejezésben belülről kifelé haladva építsük fel a nyelvtant a következőképpen:

- (1) Adott $G = (V_N, V_T, S, H)$ nyelvtan esetén az $L(G)^*$ nyelvet generáló nyelvtant megkaphatjuk, ha az összes $A \rightarrow p$ alakú szabály mellé – ahol $A \in V_N$, $p \in V_T^*$, – bevezetjük az $A \rightarrow pS$ alakú szabályt is, valamint bevezetjük az $S' \rightarrow \lambda$, $S' \rightarrow S$ szabályokat. Jelen esetben:

$$G_{a^*} = (\{A, S\}, \{a\}, S, \{A \rightarrow a, A \rightarrow aA, S \rightarrow \lambda, S \rightarrow A\}),$$

$$G_{b^*} = (\{B, S\}, \{b\}, S, \{B \rightarrow b, B \rightarrow bB, S \rightarrow \lambda, S \rightarrow B\}).$$
- (2) Adott $G_1 = (V_N, V_T, S, H)$ és $G_2 = (V'_N, V'_T, S', H')$ nyelvtanok esetén – ahol $V_N \cap V'_N = \emptyset$ – az $L(G) = L(G_1) \cdot L(G_2)$ konkatenált

nyelvet generáló nyelvtan előállításához első lépésben az összes H -beli $A \rightarrow p$ szabályt – ahol $A \in V_N$, $p \in V_T^*$ – az $A \rightarrow pS'$ szabályra cseréljük. Az így kapott szabályhalmazt jelöljük H'' -vel. Ezek után $G = (V_N \cup V'_N, V_T \cup V'_T, S, H'' \cup H')$. Jelen esetben:

$$G_{a*b} = (\{A, S, B\}, \{a, b\}, S, \{A \rightarrow aB, A \rightarrow aA, S \rightarrow B, S \rightarrow A, B \rightarrow b\}),$$

$$G_{b*a} = (\{B, S, A\}, \{b, a\}, S, \{B \rightarrow bA, B \rightarrow bB, S \rightarrow A, S \rightarrow B, A \rightarrow a\}).$$

- (3) Adott $G_1 = (V_N, V_T, S, H)$ és $G_2 = (V'_N, V'_T, S', H')$ nyelvtanok esetén – ahol $V_N \cap V'_N = \emptyset$ – az $L(G) = L(G_1) \cup L(G_2)$ nyelvet generáló nyelvtan előállításához a $G = (V_N \cup V'_N \cup \{S''\}, V_T \cup V'_T, S'', H \cup H' \cup \{S'' \rightarrow S, S'' \rightarrow S'\})$ nyelvtan megfelelő lesz, ahol S'' nem eleme a V_N, V'_N, V_T, V'_T hamazok egyikének sem. Jelen esetben a 2. pontban megadott G_{a*b} és G_{b*a} nyelvtanok esetén nem teljesül az a feltétel, miszerint a nemterminálisok hamazai diszjunktak, ezért az egyik nyelvtanban először át kell jelölni a nemterminálisokat. Legyen például

$$G'_{a*b} = (\{C, S', D\}, \{a, b\}, S', \{C \rightarrow aD, C \rightarrow aC, S' \rightarrow D, S' \rightarrow C, D \rightarrow b\}).$$

Természetesen a nemterminálisok átjelölése nem befolyásolja a generált nyelvet, azaz $L(G'_{a*b}) = L(G_{a*b}) = \{a^*b\}$. Ezek után már meg tudjuk adni a

$$G_{a*b \cup b*a} = (\{C, S', D, B, S, A, S''\}, \{a, b\}, S'', \{C \rightarrow aD, C \rightarrow aC, S' \rightarrow D, S' \rightarrow C, D \rightarrow b, B \rightarrow bA, B \rightarrow bB, S \rightarrow A, S \rightarrow B, A \rightarrow a, S'' \rightarrow S', S'' \rightarrow S\})$$

reguláris nyelvtant, mely pontosan az L nyelvet generálja.

5.2. feladat. Adjunk meg az

$$L = \{(ab \cup c)^*(ba)^*\}$$

reguláris kifejezéssel megadott L nyelvet generáló reguláris nyelvtant!

Megoldás: A feladat megoldásához első lépésben vezessünk be minden terminális betűhöz egy reguláris nyelvtant, mely kizárólag az adott terminálist generálja. Jelen esetben legyenek a kiinduló nyelvtanok a következők: $G_a = (\{A\}, \{a\}, A, \{A \rightarrow a\})$, $G_b = (\{B\}, \{b\}, B, \{B \rightarrow b\})$, $G_c = (\{C\}, \{c\}, C, \{C \rightarrow c\})$. Ezek után a kifejezésben belülről kifelé haladva építsük fel a nyelvtant a következőképpen:

- (1) $G_{ab} = (\{A, B\}, \{a, b\}, A, \{A \rightarrow aB, B \rightarrow b\})$,
- (2) $G_{ba} = (\{B, A\}, \{b, a\}, B, \{B \rightarrow bA, A \rightarrow a\})$,

- (3) $G_{ab\cup c} = (\{A, B, C, S\}, \{a, b, c\}, S, \{A \rightarrow aB, B \rightarrow b, C \rightarrow c, S \rightarrow A, S \rightarrow C\})$,
- (4) $G_{(ab\cup c)^*} = (\{A, B, C, S, S'\}, \{a, b, c\}, S', \{A \rightarrow aB, B \rightarrow b, B \rightarrow bS, C \rightarrow c, C \rightarrow cS, S \rightarrow A, S \rightarrow C, S' \rightarrow \lambda, S' \rightarrow S\})$,
- (5) $G_{(ba)^*} = (\{B, A, S\}, \{b, a\}, S, \{B \rightarrow bA, A \rightarrow a, A \rightarrow aB, S \rightarrow \lambda, S \rightarrow B\})$,
- (6) $G'_{(ba)^*} = (\{D, C, Z\}, \{b, a\}, Z, \{D \rightarrow bC, C \rightarrow a, C \rightarrow aD, Z \rightarrow \lambda, Z \rightarrow D\})$,
- (7) $G_{(ab\cup c)^*(ba)^*} = (\{A, B, C, S, S', D, C, Z\}, \{a, b, c\}, S', \{A \rightarrow aB, B \rightarrow bZ, B \rightarrow bS, C \rightarrow cZ, C \rightarrow cS, S \rightarrow A, S \rightarrow C, S' \rightarrow Z, S' \rightarrow S, D \rightarrow bC, C \rightarrow a, C \rightarrow aD, Z \rightarrow \lambda, Z \rightarrow D\})$.

5.3. feladat. Adjunk meg az

$$L = \{(a^*ba^* \cup aa)^*\}$$

reguláris kifejezéssel megadott L nyelvet generáló reguláris nyelvtant!

Megoldás: A feladat megoldásához első lépésben vezessünk be minden terminális betűhöz egy reguláris nyelvtant, mely kizárólag az adott terminálist generálja. Jelen esetben legyenek a kiinduló nyelvtanok a következők: $G_a = (\{A\}, \{a\}, A, \{A \rightarrow a\})$, $G_b = (\{B\}, \{b\}, B, \{B \rightarrow b\})$. Ezek után a kifejezésben belülről kifelé haladva építsük fel a nyelvtant a következőképpen:

- (1) $G_{a^*} = (\{A, S\}, \{a\}, S, \{A \rightarrow a, A \rightarrow aA, S \rightarrow \lambda, S \rightarrow A\})$,
- (2) $G_{a^*b} = (\{A, S, B\}, \{a, b\}, S, \{A \rightarrow aB, A \rightarrow aA, S \rightarrow B, S \rightarrow A, B \rightarrow b\})$,
- (3) $G'_{a^*} = (\{C, Z\}, \{a\}, Z, \{C \rightarrow a, C \rightarrow aC, Z \rightarrow \lambda, Z \rightarrow C\})$,
- (4) $G_{a^*ba^*} = (\{A, S, B, C, Z\}, \{a, b\}, S, \{A \rightarrow aB, A \rightarrow aA, S \rightarrow B, S \rightarrow A, B \rightarrow bZ, C \rightarrow a, C \rightarrow aC, Z \rightarrow \lambda, Z \rightarrow C\})$,
- (5) $G'_a = (\{E\}, \{a\}, E, \{E \rightarrow a\})$,
- (6) $G_{aa} = (\{A, E\}, \{a\}, A, \{A \rightarrow aE, E \rightarrow a\})$,
- (7) $G'_{aa} = (\{F, E\}, \{a\}, F, \{F \rightarrow aE, E \rightarrow a\})$,
- (8) $G_{a^*ba^*\cup aa} = (\{A, S, B, C, Z, F, E, S'\}, \{a, b\}, S', \{A \rightarrow aB, A \rightarrow aA, S \rightarrow B, S \rightarrow A, B \rightarrow bZ, C \rightarrow a, C \rightarrow aC, Z \rightarrow \lambda, Z \rightarrow C, F \rightarrow aE, E \rightarrow a, S' \rightarrow S, S' \rightarrow F\})$,
- (9) $G_{(a^*ba^*\cup aa)^*} = (\{A, S, B, C, Z, F, E, S', S''\}, \{a, b\}, S'', \{A \rightarrow aB, A \rightarrow aA, S \rightarrow B, S \rightarrow A, B \rightarrow bZ, C \rightarrow aS', C \rightarrow aC, Z \rightarrow S', Z \rightarrow C, F \rightarrow aE, E \rightarrow aS', S' \rightarrow S, S' \rightarrow F, S'' \rightarrow \lambda, S'' \rightarrow S'\})$.

6. Ekvivalens véges automaták megadása reguláris nyelvtanokhoz

6.1. feladat. Adjunk meg a $G = (\{S, A\}, \{a, b\}, S, \{S \rightarrow abaS, S \rightarrow A, A \rightarrow bA, A \rightarrow aaa, A \rightarrow \lambda\})$ nyelvtannal ekvivalens, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatát!

Megoldás: A feladat megoldásához első lépésben megadunk az eredeti G nyelvtannal ekvivalens olyan G' nyelvtant, amelyben az összes szabály $A \rightarrow aA$, $A \rightarrow B$ vagy $A \rightarrow \lambda$ alakú, ahol $A, B \in V_N$ és $a \in V_T$. Ehhez először vezessünk be egy új V nemterminális, és a G' szabályai közé vegyük fel a $V \rightarrow \lambda$ szabályt, valamint vegyük át a G nyelvtan szabályai közül azokat, amelyek megfelelnek a megadott alakok valamelyikének. Ezek után:

- (1) A G nyelvtanbeli $A \rightarrow a_1a_2\dots a_n$, $A \in V_N$, $a_i \in V_T$, $2 \leq n$ alakú szabályok helyett új nemterminálisok bevezetésével használjuk az $A \rightarrow a_1Z_1$, $Z_1 \rightarrow a_2Z_2, \dots, Z_{n-2} \rightarrow a_{n-1}Z_{n-1}$, $Z_{n-1} \rightarrow a_nV$ alakú szabályokat a G' nyelvtanban, ahol a Z_i -k az új nemterminálisok.
- (2) A G nyelvtan $A \rightarrow a_1a_2\dots a_nB$, $A, B \in V_N$, $a_i \in V_T$, $2 \leq n$ alakú szabályai helyett új nemterminálisok bevezetésével használjuk az $A \rightarrow a_1Z_1$, $Z_1 \rightarrow a_2Z_2, \dots, Z_{n-2} \rightarrow a_{n-1}Z_{n-1}$, $Z_{n-1} \rightarrow a_nB$ alakú szabályokat G' -ben, ahol a Z_i -k az új nemterminálisok.

Jelen esetben $G' = (\{S, A, Z_1, Z_2, Z_3, Z_4, V\}, \{a, b\}, S, \{S \rightarrow aZ_1, Z_1 \rightarrow bZ_2, Z_2 \rightarrow aS, S \rightarrow A, A \rightarrow bA, A \rightarrow aZ_3, Z_3 \rightarrow aZ_4, Z_4 \rightarrow aV, A \rightarrow \lambda, V \rightarrow \lambda\})$.

Ezek után második lépésben már meg tudjuk konstruálni az automatát. Az $\mathcal{A}$ automata állapothalmaza megegyezik a G' nyelvtan nemterminálisainak halmazával. Az $\mathcal{A}$ automata bemenő jeleinek a halmaza megegyezik a G' nyelvtan terminálisainak a halmazával. Az új kezdőállapot a G' nyelvtan mondatshimbóluma lesz, az új végállapot pedig azon nemterminálisok halmaza, melyekből (akár több lépésben) levezethető az üres szó. Jelen esetben

$$\mathcal{A} = (\{S, A, Z_1, Z_2, Z_3, Z_4, V\}, \{a, b\}, \delta, S, \{S, A, V\}).$$

A δ átmenetfüggvényt a következőképpen kapjuk meg:

- (1) $\delta(A, a) = B$, ha létezik $A \rightarrow aB$ szabály G' -ben,
- (2) $\delta(A, a) = B$, ha létezik $C \in V_N$ úgy, hogy $A \xrightarrow{*}_G C$ és létezik $C \rightarrow aB$ szabály G' -ben.

Mindezek alapján

$\mathcal{A}$	S	A	Z_1	Z_2	Z_3	Z_4	V
a	$\{Z_1, Z_3\}$	$\{Z_3\}$	$\emptyset$	$\{S\}$	$\{Z_4\}$	$\{V\}$	$\emptyset$
b	$\{A\}$	$\{A\}$	$\{Z_2\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

6.2. feladat. Adjunk meg a $G = (\{S, A, B\}, \{a, b\}, S, \{S \rightarrow A, S \rightarrow B, S \rightarrow \lambda, A \rightarrow aB, A \rightarrow aabb, B \rightarrow bA, B \rightarrow S\})$ nyelvtannal ekvivalens, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatát!

Megoldás: A feladat megoldásához első lépésben megadunk az eredeti G nyelvtannal ekvivalens olyan G' nyelvtant, amelyben az összes szabály $A \rightarrow aA$, $A \rightarrow B$ vagy $A \rightarrow \lambda$ alakú, ahol $A, B \in V_N$ és $a \in V_T$. Jelen esetben $G' = (\{S, A, B, X, Y, Z, V\}, \{a, b\}, S, \{S \rightarrow A, S \rightarrow B, S \rightarrow \lambda, A \rightarrow aB, A \rightarrow aX, X \rightarrow aY, Y \rightarrow bZ, Z \rightarrow bV, B \rightarrow bA, B \rightarrow S, V \rightarrow \lambda\})$. Ezek után második lépésben már meg tudjuk konstruálni az automatát:

$$\mathcal{A} = (\{S, A, B, X, Y, Z, V\}, \{a, b\}, \delta, S, \{S, B, V\})$$

$\mathcal{A}$	S	A	B	X	Y	Z	V
a	$\{B, X\}$	$\{B, X\}$	$\{B, X\}$	$\{Y\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$
b	$\{A\}$	$\{\emptyset\}$	$\{A\}$	$\{\emptyset\}$	$\{Z\}$	$\{V\}$	$\{\emptyset\}$

6.3. feladat. Adjunk meg a $G = (\{S, A, B\}, \{a, b\}, S, \{S \rightarrow aaS, S \rightarrow bS, S \rightarrow bA, A \rightarrow B, A \rightarrow aaa, B \rightarrow abaB, B \rightarrow \lambda\})$ nyelvtannal ekvivalens, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatát!

Megoldás: A feladat megoldásához első lépésben megadunk az eredeti G nyelvtannal ekvivalens olyan G' nyelvtant, amelyben az összes szabály $A \rightarrow aA$, $A \rightarrow B$ vagy $A \rightarrow \lambda$ alakú, ahol $A, B \in V_N$ és $a \in V_T$. Jelen esetben $G' = (\{S, A, B, X, Y, Z, P, Q, V\}, \{a, b\}, S, \{S \rightarrow aX, X \rightarrow aS, S \rightarrow bS, S \rightarrow bA, A \rightarrow B, A \rightarrow aY, Y \rightarrow aZ, Z \rightarrow aV, B \rightarrow aP, P \rightarrow bQ, Q \rightarrow aB, B \rightarrow \lambda, V \rightarrow \lambda\})$.

Ezek után második lépésben már meg tudjuk konstruálni az automatát:

$$\mathcal{A} = (\{S, A, B, X, Y, Z, P, Q, V\}, \{a, b\}, \delta, S, \{A, B, V\})$$

$\mathcal{A}$	S	A	B	P	Q	X	Y	Z	V
a	$\{X\}$	$\{Y, P\}$	$\{P\}$	$\{\emptyset\}$	$\{B\}$	$\{S\}$	$\{Z\}$	$\{V\}$	$\{\emptyset\}$
b	$\{S, A\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{Q\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$	$\{\emptyset\}$

7. Ekvivalens reguláris kifejezések megadása véges automatákhoz

7.1. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_1, a_2\}, \{x, y\}, \delta, a_1, \{a_2\})$$

$\mathcal{A}$	a_1	a_2
x	a_2	a_2
y	a_1	$\emptyset$

determinisztikus, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatával ekvivalens reguláris kifejezést!

Megoldás: Jelöljük n -el az automata állapotainak számát. Jelen esetben $n = 2$. A feladat megoldásához előállítjuk az r_{ij}^k reguláris kifejezéseket, a következő rekurzív definíció segítségével:

- (1) $r_{ij}^0 = \{x \mid \delta(a_i, x) = a_j\}$, ha $i \neq j$,
- (2) $r_{ij}^0 = \{x \mid \delta(a_i, x) = a_j\} \cup \{\lambda\}$, ha $i = j$,
- (3) $r_{ij}^k = r_{ik}^{k-1} (r_{kk}^{k-1})^* r_{kj}^{k-1} \cup r_{ij}^{k-1}$, ha $1 \leq k \leq n$.

Jelen esetben

	$k = 0$	$k = 1$	$k = 2$
r_{11}^k	$y \cup \lambda$	y^*	y^*
r_{12}^k	x	y^*x	y^*xx^*
r_{21}^k	$\emptyset$	$\emptyset$	$\emptyset$
r_{22}^k	$x \cup \lambda$	$x \cup \lambda$	x^*

Például $r_{11}^1 = r_{11}^0 (r_{11}^0)^* r_{11}^0 \cup r_{11}^0 = (y \cup \lambda)(y \cup \lambda)^*(y \cup \lambda) \cup (y \cup \lambda) = y^*$,

$r_{12}^2 = r_{12}^1 (r_{22}^1)^* r_{22}^1 \cup r_{12}^1 = y^*x(x \cup \lambda)^*(x \cup \lambda) \cup y^*x = y^*xx^*$.

Ezek után $L(\mathcal{A}) = \bigcup r_{lm}^n$, ahol a_l a kezdőállapot, az a_m pedig befutja a végállapotok halmazát. Jelen esetben egyetlen végállapotunk van, így $L(\mathcal{A}) = r_{12}^2 = y^*xx^*$. Megfigyelhető, hogy $k = n$ esetén elegendő azokat az r_{ij}^k elemeket kiszámolni, ahol i kezdőállapot, és j végállapot. A továbbiakban ezek szerint fogunk eljárni.

7.2. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_1, a_2\}, \{x, y, z\}, \delta, a_1, \{a_1, a_2\})$$

$\mathcal{A}$	a_1	a_2
x	a_2	$\emptyset$
y	a_1	a_2
z	$\emptyset$	a_1

determinisztikus, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatával ekvivalens reguláris kifejezést!

Megoldás: A feladat megoldásához először előállítjuk az r_{ij}^k reguláris kifejezéseket:

	$k = 0$	$k = 1$
r_{11}^k	$y \cup \lambda$	y^*
r_{12}^k	x	y^*x
r_{21}^k	z	zy^*
r_{22}^k	$y \cup \lambda$	$zy^*x \cup y \cup \lambda$

Ezek után meghatározzuk az r_{11}^2 és r_{12}^2 reguláris kifejezéseket:

$$r_{11}^2 = r_{12}^1(r_{22}^1)^*r_{21}^1 \cup r_{11}^1 = y^*x(zy^*x \cup y \cup \lambda)^*zy^* \cup y^* = y^*x(zy^*x \cup y)^*zy^* \cup y^*,$$

$$r_{12}^2 = r_{12}^1(r_{22}^1)^*r_{21}^1 \cup r_{12}^1 = y^*x(zy^*x \cup y \cup \lambda)^*(zy^*x \cup y \cup \lambda) \cup y^*x = y^*x(zy^*x \cup y)^*.$$

$$\text{Végül } L(\mathcal{A}) = r_{11}^2 \cup r_{12}^2 = (y^*x(zy^*x \cup y)^*zy^* \cup y^*) \cup (y^*x(zy^*x \cup y)^*) = y^*x(zy^*x \cup y)^*(zy^* \cup \lambda) \cup y^*.$$

7.3. feladat. Adjunk meg az

$$\mathcal{A} = (\{a_1, a_2, a_3\}, \{x, y\}, \delta, a_1, \{a_2, a_3\})$$

$\mathcal{A}$	a_1	a_2	a_3
x	a_2	a_3	a_1
y	a_3	a_2	$\emptyset$

determinisztikus, kimenő jel nélküli, iniciális, végállapotokkal bővített, véges, felismerő automatával ekvivalens reguláris kifejezést!

Megoldás: A feladat megoldásához előállítjuk az r_{ij}^k reguláris kifejezéseket:

	$k = 0$	$k = 1$	$k = 2$
r_{11}^k	λ	λ	λ
r_{12}^k	x	x	xy^*
r_{13}^k	y	y	$y \cup xy^*x$
r_{21}^k	$\emptyset$	$\emptyset$	$\emptyset$
r_{22}^k	$y \cup \lambda$	$y \cup \lambda$	y^*
r_{23}^k	x	x	y^*x
r_{31}^k	x	x	x
r_{32}^k	$\emptyset$	xx	xy^*
r_{33}^k	λ	$\lambda \cup xy$	$\lambda \cup xy \cup xxy^*x$

$$\text{Ezek után } r_{12}^3 = (y \cup xy^*x)(\lambda \cup xy \cup xxy^*x)^*xxy^* \cup xy^* = (y \cup xy^*x)(xy \cup xxy^*x)^*xxy^* \cup xy^*,$$

$$r_{13}^3 = (y \cup xy^*x)(\lambda \cup xy \cup xxy^*x)^*(\lambda \cup xy \cup xxy^*x) \cup (y \cup xy^*x) = (y \cup xy^*x)(xy \cup xxy^*x)^*,$$

$$L(\mathcal{A}) = r_{12}^3 \cup r_{13}^3 = ((y \cup xy^*x)(xy \cup xxy^*x)^*xxy^* \cup xy^*) \cup ((y \cup xy^*x)(xy \cup xxy^*x)^*) = (y \cup xy^*x)(xy \cup xxy^*x)^*(xxy^* \cup \lambda) \cup xy^*.$$

8. Veremautomaták

A **verem** egy olyan adatszerkezet, amelyből az utoljára beírt elemet lehet először kivenni.

A **veremautomata** az egy végtelen veremmel bővített véges automata.

$\mathcal{A}(\Sigma, Z, Q, q_0, F, \phi, z_0)$ hetes veremautomata, ahol

Σ a szalag ábécé, (véges nemüres)

Z a verem ábécé, (véges nemüres)

Q a belsőállapotok halmaza, (véges nemüres)

$q_0 \subseteq Q$ a kezdőállapot,

$F \subseteq Q$ a záróhelyettesítések halmaza,

$\phi : \Sigma \cup \{\lambda\} \times Q \times Z \rightarrow 2^{Q \times Z^*}$ az átmenetfüggvény,

$z_0 \in Z$ a verem alja jel.

Az automata ha a **szalagról olvas** egy jelet, akkor a következő jelle lép az olvasó fejjel.

Az automata ha a **veremből olvas** egy jelet, akkor az törlődik a verem tetejéről.

Az automata működése:

A szalagról olvas egy jelet, vagy nem, a verem tetejéről olvas egy jelet, van valamilyen belső állapotban, majd ezektől függően átmehet új belső állapotokba, és írhat egy szót a verem tetejére.

Az automata akkor ismer fel a szalagról egy szót, ha vagy végállapotba kerül, vagy kiürül a verem a szó olvasása után.

Tetszőleges kettes típusú grammatikához megadható olyan veremautomata, mely ugyanazt a nyelvet ismeri fel.

8.1. feladat. Adjunk meg a G 2-es típusú grammatikához egy olyan veremautomatát, amely a G grammatika által generált nyelvet ismeri fel, majd mutassuk meg, hogy az 10011 szót felismeri az automata!

$G = (\{S, A, B\}, \{0, 1\}, S, H)$, ahol H szabályai:

$S \rightarrow SA, S \rightarrow AB,$

$A \rightarrow BS, B \rightarrow SA,$

$A \rightarrow 1, S \rightarrow 1, B \rightarrow 0.$

Megoldás:

A szalagábécé álljon a grammatika terminálisaiból, a veremábécé pedig tartalmazza a terminálisokat, a nemterminálisokat és a veremalja jelet!

Három belsőállapot elég lesz, melyből egy kezdőállapot, egy végállapot.

Az átmenetfüggvény definiálása:

Kell egy olyan szabály, hogy a veremalja jel fölé írja az S kezdőszimbólumot, és átviszi az automatát általános állapotba!

Kellenek olyan szabályok, hogy az automata a szalagról nem olvas, a verem tetejéről olvassa a H -beli szabályok bal oldalát, majd visszaírja fordított sorrendben a jobb oldalát!

Kellenek olyan szabályok, hogy a szalagról és a veremtetejéről ugyanazt olvassuk, majd nem írunk semmit a verem tetejére!

És kell egy olyan szabály, hogy ha általános állapotban olvassa az automata a veremalja jelet, akkor menjen át végállapotba!

$\mathcal{A}(\{0, 1\}, \{S, A, B, 0, 1, z_0\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \phi, z_0)$.

$\phi(\lambda, q_0, z_0) = (q_1, z_0 S)$,

$\phi(\lambda, q_1, S) = (q_1, AS)$,

$\phi(\lambda, q_1, S) = (q_1, BA)$,

$\phi(\lambda, q_1, A) = (q_1, SB)$,

$\phi(\lambda, q_1, B) = (q_1, AS)$,

$\phi(\lambda, q_1, A) = (q_1, 1)$,

$\phi(\lambda, q_1, S) = (q_1, 1)$,

$\phi(\lambda, q_1, B) = (q_1, 0)$,

$\phi(1, q_1, 1) = (q_1, \lambda)$,

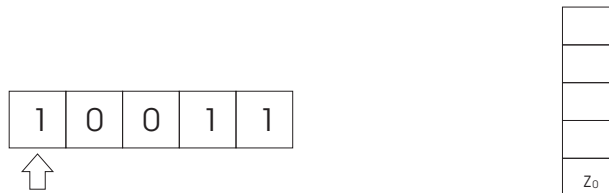
$\phi(0, q_1, 0) = (q_1, \lambda)$,

$\phi(\lambda, q_1, z_0) = (q_2, \lambda)$.

Az első fejezet 7.2. feladata alapján CYK algoritmus segítségével találtunk a következő legbaloldalibb levezetésére az 10011 szónak:

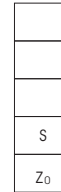
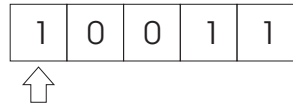
$S \Rightarrow SA \Rightarrow SAA \Rightarrow ABAA \Rightarrow 1BAA \Rightarrow 10AA \Rightarrow 10BSA \Rightarrow 100SA \Rightarrow 1001A \Rightarrow 10011$

Az automata működése előtt a szalag és a verem:

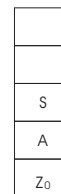
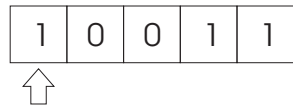


Egy legbaloldalibb levezetést használva az automata a következő lépéseken át ismeri fel az 10011 szót:

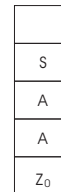
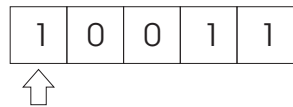
1. lépés: Írjuk a veremalja jel felé S -t! $\phi(\lambda, q_0, z_0) = (q_1, z_0 S)$.



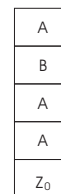
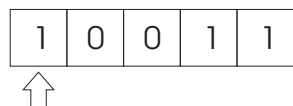
2. lépés: Alkalmazzuk az $S \rightarrow SA$ szabályt! $\phi(\lambda, q_1, S) = (q_1, AS)$.



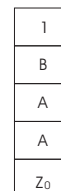
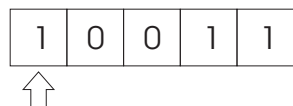
3. lépés: Alkalmazzuk az $S \rightarrow SA$ szabályt! $\phi(\lambda, q_1, S) = (q_1, AS)$.



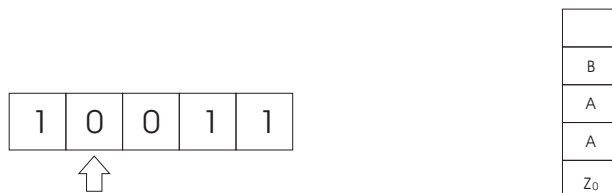
4. lépés: Alkalmazzuk az $S \rightarrow AB$ szabályt! $\phi(\lambda, q_1, S) = (q_1, BA)$.



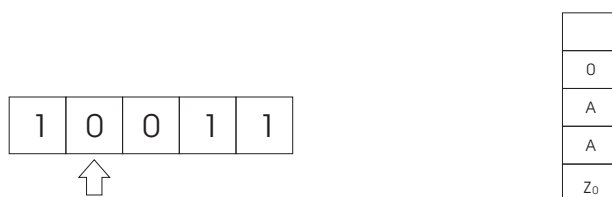
5. lépés: Alkalmazzuk az $A \rightarrow 1$ szabályt! $\phi(\lambda, q_1, A) = (q_1, 1)$.



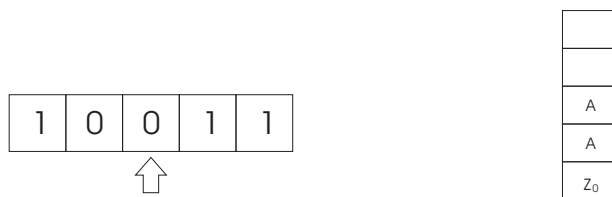
6. lépés: A szalagról olvasunk, a veremről törölünk. $\phi(1, q_1, 1) = (q_1, \lambda)$.



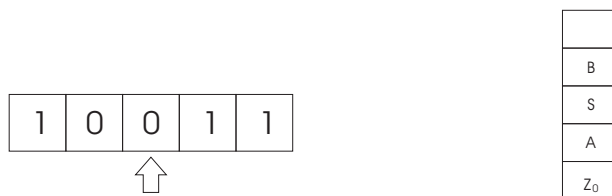
7. lépés: Alkalmazzuk a $B \rightarrow 0$ szabályt! $\phi(\lambda, q_1, B) = (q_1, 0)$.



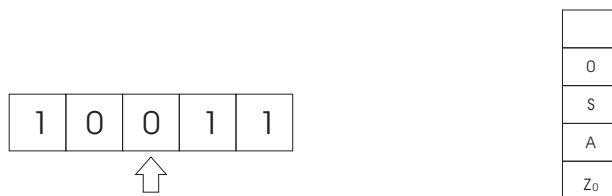
8. lépés: A szalagról olvasunk, a veremről törölünk. $\phi(0, q_1, 0) = (q_1, \lambda)$.



9. lépés: Alkalmazzuk az $A \rightarrow BS$ szabályt! $\phi(\lambda, q_1, A) = (q_1, SB)$.



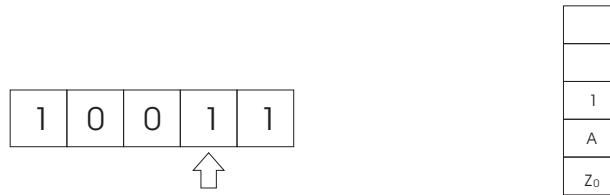
10. lépés: Alkalmazzuk a $B \rightarrow 0$ szabályt! $\phi(\lambda, q_1, B) = (q_1, 0)$.



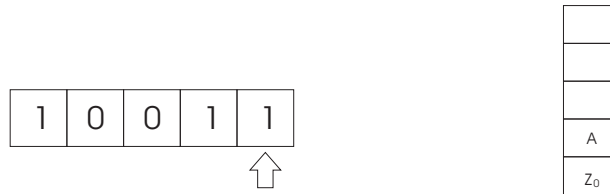
11. lépés: A szalagról olvasunk, a veremről törölünk. $\phi(0, q_1, 0) = (q_1, \lambda)$.



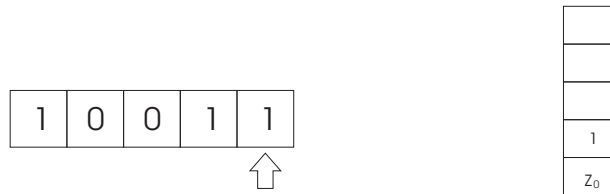
12. lépés: Alkalmazzuk az $S \rightarrow 1$ szabályt! $\phi(\lambda, q_1, S) = (q_1, 1)$.



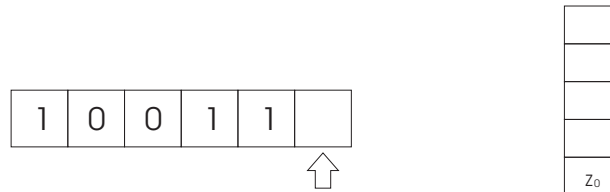
13. lépés: A szalagról olvasunk, a veremről törölünk. $\phi(1, q_1, 1) = (q_1, \lambda)$.



14. lépés: Alkalmazzuk az $A \rightarrow 1$ szabályt! $\phi(\lambda, q_1, A) = (q_1, 1)$.



15. lépés: A szalagról olvasunk, a veremről törölünk. $\phi(1, q_1, 1) = (q_1, \lambda)$.



16. lépés: $\phi(\lambda, q_1, z_0) = (q_2, \lambda)$, vagyis végállapotba került az automata, és így felismerte az 10011 szót.

8.2. feladat. Adjunk meg olyan veremautomatát, amely a következő grammatika által generált nyelvet ismeri fel, és mutassuk meg, hogy a bbcbba szót is elfogadja!

$G = (\{S, A, B, C, D\}, \{a, b, c\}, S, H)$, ahol H szabályai:

$S \rightarrow AB, A \rightarrow CA, A \rightarrow SS, B \rightarrow CD, A \rightarrow b, D \rightarrow a, C \rightarrow c, C \rightarrow b.$

Megoldás:

$\mathcal{A}(\{a, b, c\}, \{S, A, B, C, D, a, b, c, z_0\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \phi, z_0).$

$\phi(\lambda, q_0, z_0) = (q_1, z_0 S),$

$\phi(\lambda, q_1, S) = (q_1, BA),$

$\phi(\lambda, q_1, A) = (q_1, AC),$

$\phi(\lambda, q_1, A) = (q_1, SS),$

$\phi(\lambda, q_1, B) = (q_1, DC),$

$\phi(\lambda, q_1, A) = (q_1, b),$

$\phi(\lambda, q_1, D) = (q_1, a),$

$\phi(\lambda, q_1, C) = (q_1, c),$

$\phi(\lambda, q_1, C) = (q_1, b),$

$\phi(a, q_1, a) = (q_1, \lambda),$

$\phi(b, q_1, b) = (q_1, \lambda),$

$\phi(c, q_1, c) = (q_1, \lambda),$

$\phi(\lambda, q_1, z_0) = (q_2, \lambda).$

Az egyetlen legbaloldalibb levezetés az első fejezet 7.3. feladata alapján:

$S \Rightarrow AB \Rightarrow CAB \Rightarrow bAB \Rightarrow bCAB \Rightarrow bbAB \Rightarrow bbCAB \Rightarrow bbcAB \Rightarrow$
 $bbcbB \Rightarrow bbcbCD \Rightarrow bbcbD \Rightarrow bbcbba$

A szó felismerésének lépései:

$\phi(\lambda, q_0, z_0) = (q_1, z_0 S),$

$\phi(\lambda, q_1, S) = (q_1, BA),$

$\phi(\lambda, q_1, A) = (q_1, AC),$

$\phi(\lambda, q_1, C) = (q_1, b),$

$\phi(b, q_1, b) = (q_1, \lambda),$

$\phi(\lambda, q_1, A) = (q_1, AC),$

$\phi(\lambda, q_1, C) = (q_1, b),$

$\phi(b, q_1, b) = (q_1, \lambda),$

$\phi(\lambda, q_1, A) = (q_1, AC),$

$\phi(\lambda, q_1, C) = (q_1, c),$

$\phi(c, q_1, c) = (q_1, \lambda),$

$\phi(\lambda, q_1, A) = (q_1, b),$

$\phi(b, q_1, b) = (q_1, \lambda),$

$$\begin{aligned}
\phi(\lambda, q_1, B) &= (q_1, DC), \\
\phi(\lambda, q_1, C) &= (q_1, b), \\
\phi(b, q_1, b) &= (q_1, \lambda), \\
\phi(\lambda, q_1, D) &= (q_1, a), \\
\phi(a, q_1, a) &= (q_1, \lambda), \\
\phi(\lambda, q_1, z_0) &= (q_2, \lambda).
\end{aligned}$$

8.3. feladat. Adjunk meg olyan veremautomatát, mely a G grammatika által generált nyelvet ismeri fel!

$G = (\{S, A, B\}, \{x, y\}, S, H)$, ahol H a következő szabályokból áll:

$S \rightarrow AB, A \rightarrow BSB, A \rightarrow BB, B \rightarrow xAy, B \rightarrow \lambda, B \rightarrow x, B \rightarrow y$.

Megoldás:

$\mathcal{A}(\{x, y\}, \{S, A, B, x, y, z_0\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \phi, z_0)$.

$$\begin{aligned}
\phi(\lambda, q_0, z_0) &= (q_1, z_0 S), \\
\phi(\lambda, q_1, S) &= (q_1, BA), \\
\phi(\lambda, q_1, A) &= (q_1, BSB), \\
\phi(\lambda, q_1, A) &= (q_1, BB), \\
\phi(\lambda, q_1, B) &= (q_1, yAx), \\
\phi(\lambda, q_1, B) &= (q_1, \lambda), \\
\phi(\lambda, q_1, B) &= (q_1, x), \\
\phi(\lambda, q_1, B) &= (q_1, y), \\
\phi(x, q_1, x) &= (q_1, \lambda), \\
\phi(y, q_1, y) &= (q_1, \lambda), \\
\phi(\lambda, q_1, z_0) &= (q_2, \lambda),
\end{aligned}$$

(Ha egy szó levezetését szeretnénk megkeresni, ahhoz meg kell találni egy legbaloldalibb levezetését a szónak, melyben segíthet nekünk az üresszó lemma és a Chomsky-féle normálformára hozás alkalmazása után a CYK algoritmus.)

Előfordul olyan tárgyalása a veremautomatáknak, hogy a veremmel csak három műveletet lehet végezni:

- legfelső betű törlése,
- legfelső betű cseréje egy másikra,
- legfelső betű felé egyet rakunk,

ekkor például a $\phi(\lambda, q_1, A) = (q_1, BSB)$ helyett a $\phi(\lambda, q_1, A) = (q_3, B)$,

$\phi(\lambda, q_3, B) = (q_4, BS)$ és $\phi(\lambda, q_4, S) = (q_1, SB)$ szabályokat használjuk.

9. Turing-gépek

A Turing-gép az egy olyan automata, amely el van látva mindkét irányban végtelen szalagokkal, ahány szalagot használ annyi író-olvasó fejjel. Turing-gép megadása az $\mathcal{A}(K, \Sigma, Q, q_0, Q_v, M, \phi)$ hetessel történénik.

K a szalagok száma,

Σ a szalagábécé,

Q a belső állapotok halmaza,

$q_0 \in Q$ a kezdőállapot,

$Q_v \subseteq Q$ a végállapotok halmaza,

$M = L, R, H$ a szalag mozgásainak lehetséges iránya (Left, Right, Halt (balra, jobbra, helyben)),

$\phi : Q \times \Sigma^K \rightarrow 2^{Q \times \Sigma^K \times M^K}$ a leképezés függvénye.

Ha a $\phi : Q \times \Sigma^K \rightarrow Q \times \Sigma^K \times M^K$ alakú, akkor determinisztikus Turing-gépről beszélünk. A kiinduló helyzet az az lesz a feladatok során, hogy az író-olvasó fej a "#" karakteren van, de az eggyel mellette levő betű már "értékes", még a feldolgozandó jelek után szintén "#" jel áll, mely nem esik egybe az üres jellel.

9.1. feladat. Készítsünk egyszalagos Turing-gépet, amely a bemenő bináris szót negálja!

$\phi(q_0, \#) = (q_1, \#, L)$

$\phi(q_1, 1) = (q_1, 0, L)$

$\phi(q_1, 0) = (q_1, 1, L)$

$\phi(q_1, \#) = (q_v, \#, H)$

Először kezdőállapotban van a gép, és a szó előtt áll.

1. Menjen általános állapotba, majd álljon a szó első betűjére!

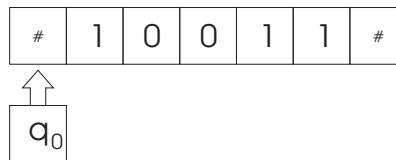
2. Ha az olvasott betű 1-es, akkor írja át 0-ra, majd lépjen a következőre!

3. Ha az olvasott betű 0, akkor írja át 1-re, majd lépjen a következőre!

4. Ha az olvasott betű #, akkor írja vissza, majd lépjen végállapotba, különben újra lépünk a 2. pontra!

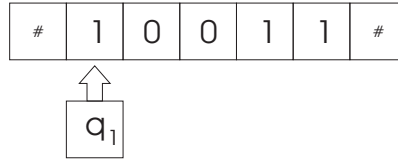
Nézzük, hogyan működik a gép az 10011 szóra!

Először q_0 állapotban van a gép és #-t olvas.

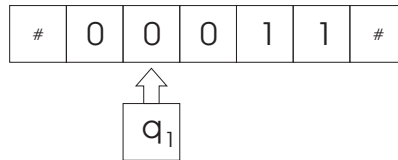


Determinisztikus a Turing-gép, tehát egy lehetséges előírás van erre a párra.

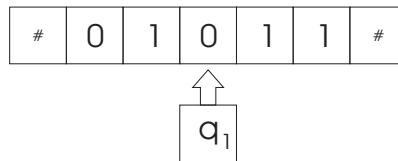
1. lépés $\phi(q_0, \#) = (q_1, \#, L)$.



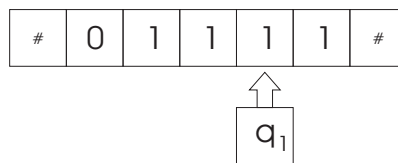
2. lépés $\phi(q_1, 1) = (q_1, 0, L)$.



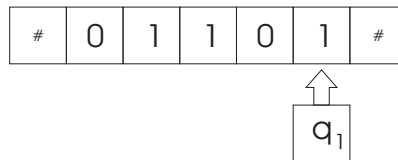
3. lépés $\phi(q_1, 0) = (q_1, 1, L)$.



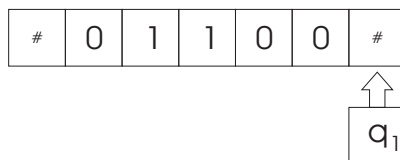
4. lépés $\phi(q_1, 0) = (q_1, 1, L)$.



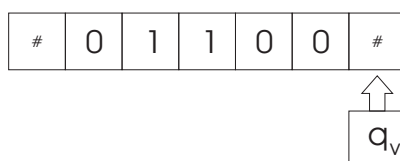
5. lépés $\phi(q_1, 1) = (q_1, 0, L)$.



6. lépés $\phi(q_1, 1) = (q_1, 0, L)$.



7. lépés $\phi(q_1, \#) = (q_v, \#, H)$.



És a Turing-gép végállapotba került.

A gép teljes megadása: $\mathcal{A}(1, \{1, 0, \#\}, \{q_0, q_1, q_v\}, q_0, \{q_v\}, M, \phi)$, ahol

$$\phi(q_0, \#) = (q_1, \#, L),$$

$$\phi(q_1, 1) = (q_1, 0, L),$$

$$\phi(q_1, 0) = (q_1, 1, L),$$

$$\phi(q_1, \#) = (q_v, \#, H).$$

9.2. feladat. Adjunk meg olyan Turing-gépet, amely egy bemenő bináris számhoz hozzáad egyet, majd a fejet a szám utánra viszi!

Megoldás:

$$\phi(q_0, \#) = (q_1, \#, L),$$

$$\phi(q_1, 1) = (q_1, 1, L),$$

$$\phi(q_1, 0) = (q_1, 0, L),$$

$$\phi(q_1, \#) = (q_2, \#, R),$$

$$\phi(q_2, 0) = (q_3, 1, L),$$

$$\phi(q_2, \#) = (q_3, 1, L),$$

$$\phi(q_2, 1) = (q_2, 0, R),$$

$$\phi(q_3, 1) = (q_3, 1, L),$$

$$\phi(q_3, 0) = (q_3, 0, L),$$

$$\phi(q_3, \#) = (q_v, \#, H).$$

1. lépés: Egyet lép, és átmegy q_1 állapotba.

2. lépés: Ellép az utolsó számjegyig, és átmegy q_2 állapotba.

3. lépés: Az utolsó számjegytől visszafele az 1-eseket átírja 0-ra.

4. lépés: Ha 0-ra, vagy #-ra talál átírja 1-esre, majd felveszi a q_3 állapotot.

5. lépés: Végigmegy a szón, majd átmegy végállapotba.

$\mathcal{A}(1, \{1, 0, \#\}, \{q_0, q_1, q_2, q_3, q_v\}, q_0, \{q_v\}, M, \phi)$, ahol ϕ a fenti módon adott.

9.3. feladat. Adjunk meg olyan kétszalagos Turing-gépet, amely az első szalagon levő szóról eldönti, hogy tükörszó-e!

Megoldás:

$$\begin{aligned}
 \phi(q_0, \begin{pmatrix} \# \\ \# \end{pmatrix}) &= q_1, \begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} L \\ R \end{pmatrix}, \\
 \phi(q_1, \begin{pmatrix} 1 \\ \# \end{pmatrix}) &= q_1, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} L \\ R \end{pmatrix}, \\
 \phi(q_1, \begin{pmatrix} 0 \\ \# \end{pmatrix}) &= q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} L \\ R \end{pmatrix}, \\
 \phi(q_1, \begin{pmatrix} \# \\ \# \end{pmatrix}) &= q_2, \begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} R \\ H \end{pmatrix}, \\
 \phi(q_2, \begin{pmatrix} 1 \\ \# \end{pmatrix}) &= q_2, \begin{pmatrix} 1 \\ \# \end{pmatrix}, \begin{pmatrix} R \\ H \end{pmatrix}, \\
 \phi(q_2, \begin{pmatrix} 0 \\ \# \end{pmatrix}) &= q_2, \begin{pmatrix} 0 \\ \# \end{pmatrix}, \begin{pmatrix} R \\ H \end{pmatrix}, \\
 \phi(q_2, \begin{pmatrix} \# \\ \# \end{pmatrix}) &= q_3, \begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}, \\
 \phi(q_3, \begin{pmatrix} 1 \\ 1 \end{pmatrix}) &= q_3, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}, \\
 \phi(q_3, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) &= q_3, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}, \\
 \phi(q_3, \begin{pmatrix} 1 \\ 0 \end{pmatrix}) &= q_n, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}, \\
 \phi(q_3, \begin{pmatrix} 0 \\ 1 \end{pmatrix}) &= q_n, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} L \\ L \end{pmatrix}, \\
 \phi(q_3, \begin{pmatrix} \# \\ \# \end{pmatrix}) &= q_i, \begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} H \\ H \end{pmatrix}.
 \end{aligned}$$

1. lépés Az első szalag tartalmát fordított irányban haladva átmásolja a másikkra.

2. lépés Az első szalag elejére lépked a gép.

3. lépés Összehasonlítja megegyezik-e a két szalag tartalma, majd megáll q_i elfogadó vagy q_n elutasító állapottal.

9.4. feladat. Adott a következő grammatika:

$G = \langle \{S, A, B, C\}, \{a, b, c\}, S, \{S \rightarrow ABC, S \rightarrow ABCS, AB \rightarrow BA, BA \rightarrow AB, AC \rightarrow CA, CA \rightarrow AC, BC \rightarrow CB, CB \rightarrow BC, A \rightarrow a, B \rightarrow b, C \rightarrow c\} \rangle$.

Mely nyelvet generálja ez a nyelvtan?

A nyelvtan alapján adjunk meg egy Turing gépet amely az adott nyelvet ismeri fel.

Megoldás:

A generált nyelv:

$$L = \{w \in \{a, b, c\}^+ \mid W\text{-ben az } a, a \text{ } b \text{ és a } c \text{ betűk száma azonos}\}$$

Az L nyelv nem környezetfüggetlen, környezetfüggő, tehát nincs sem olyan véges automata, sem olyan veremautomata amely képes azt felismerni.

A nyelvtan normálformában van, vagyis terminális betű csak $N \rightarrow t$ alakú szabályban fordul elő.

Tehát vegyük a következő $A(1, \Sigma, Q, q_0, \{q_v\}, \{L, R, N\}, \phi)$ Turing-gépet, ahol

$\Sigma = \{\#, a, b, c, A, B, C, S\}$ (a nyelvtan terminális és nem-terminális jelei kiegészítve a $\#$ szimbólummal)

A ϕ átmenetfüggvény pedig legyen a szabályok alapján a következő:

$$\phi(q_0, \#) = (q_1, \#, L) \text{ (iniciális lépés)}$$

$\phi(q_1, a) = (q_1, A, L)$ (a terminálisokat bevezető szabályoknak megfelelően)

$$\phi(q_1, b) = (q_1, B, L)$$

$$\phi(q_1, c) = (q_1, C, L)$$

$$\phi(q_1, \#) = (q_2, \#, R)$$

Eddig tehát a normálformában levő grammatika alapján a szalagon levő terminális szót visszaírtuk nem-terminális formára.

Most következik a többi szabály feldolgozása:

$$\phi(q_2, C) = (q_3, C, R)$$

$$\phi(q_3, B) = (q_4, B, R)$$

$$\phi(q_4, A) = (q_5, A, R) \text{ (a szó végén az } S \rightarrow ABC \text{ szabály alapján)}$$

$$\phi(q_5, A) = (q_5, A, L)$$

$$\phi(q_5, B) = (q_5, B, L)$$

$$\phi(q_5, C) = (q_6, \#, R)$$

$$\phi(q_6, B) = (q_6, \#, R)$$

$$\phi(q_6, A) = (q_{30}, S, R)$$

$$\phi(q_{30}, C) = (q_{31}, C, R)$$

$$\phi(q_{31}, B) = (q_{32}, B, R)$$

$$\phi(q_{32}, A) = (q_{33}, A, R) \text{ (a szó végén az } S \rightarrow ABCS \text{ szabály alapján)}$$

$$\phi(q_{33}, A) = (q_{33}, A, L)$$

$$\phi(q_{33}, B) = (q_{33}, B, L)$$

$$\phi(q_{33}, C) = (q_{33}, C, L)$$

$$\phi(q_{33}, S) = (q_{34}, \#, R)$$

$$\begin{aligned}\phi(q_{34}, C) &= (q_{34}, \#, R) \\ \phi(q_{34}, B) &= (q_{34}, \#, R) \\ \phi(q_{34}, A) &= (q_{30}, S, R)\end{aligned}$$

$\phi(q_{30}, \#) = (q_v, \#, H)$ (a szalagon egyetlen S van, elfogadjuk az eredeti szót)

ahhoz, hogy ne csak a szó végén alkalmazhassunk átíró lépéseket a szalag mozgatását a következő lépések teszik lehetővé:

$$\begin{aligned}\phi(q_2, C) &= (q_2, C, R) \\ \phi(q_2, B) &= (q_2, B, R) \\ \phi(q_2, A) &= (q_2, A, R) \\ \phi(q_2, \#) &= (q_{50}, \#, L) \\ \phi(q_{50}, A) &= (q_{50}, A, L) \\ \phi(q_{50}, B) &= (q_{50}, B, L) \\ \phi(q_{50}, C) &= (q_{50}, C, L) \\ \phi(q_{50}, \#) &= (q_2, \#, R)\end{aligned}$$

és, ha már S van a szalag végén:

$$\begin{aligned}\phi(q_{30}, C) &= (q_{35}, C, R) \\ \phi(q_{30}, B) &= (q_{35}, B, R) \\ \phi(q_{30}, A) &= (q_{35}, A, R) \\ \phi(q_{35}, C) &= (q_{35}, C, R) \\ \phi(q_{35}, B) &= (q_{35}, B, R) \\ \phi(q_{35}, A) &= (q_{35}, A, R) \\ \phi(q_{35}, \#) &= (q_{36}, \#, L) \\ \phi(q_{36}, A) &= (q_{36}, A, L) \\ \phi(q_{36}, B) &= (q_{36}, B, L) \\ \phi(q_{36}, C) &= (q_{36}, C, L) \\ \phi(q_{36}, S) &= (q_{30}, S, R)\end{aligned}$$

könnyen belátható, hogy a további hat szabálynak megfelelő lépések lehetnek a következők:

$$\begin{aligned}\phi(q_2, B) &= (q_{10}, A, R) \\ \phi(q_{10}, A) &= (q_2, B, R) \\ \phi(q_{35}, B) &= (q_{40}, A, R) \\ \phi(q_{40}, A) &= (q_{35}, B, R) \\ \phi(q_2, A) &= (q_{11}, B, R) \\ \phi(q_{11}, B) &= (q_2, A, R) \\ \phi(q_{35}, A) &= (q_{41}, B, R)\end{aligned}$$

$\phi(q_{41}, B) = (q_{35}, A, R)$ (ezek voltak az $AB \rightarrow BA$ és a $BA \rightarrow AB$ szabályoknak megfelelő lépések)

$$\begin{aligned}
\phi(q_2, B) &= (q_{12}, C, R) \\
\phi(q_{12}, C) &= (q_2, B, R) \\
\phi(q_{35}, B) &= (q_{42}, C, R) \\
\phi(q_{42}, C) &= (q_{35}, B, R) \\
\phi(q_2, C) &= (q_{13}, B, R) \\
\phi(q_{13}, B) &= (q_2, C, R) \\
\phi(q_{35}, C) &= (q_{43}, B, R) \\
\phi(q_{43}, B) &= (q_{35}, C, R) \\
\text{végül:} \\
\phi(q_2, A) &= (q_{14}, C, R) \\
\phi(q_{14}, C) &= (q_2, A, R) \\
\phi(q_{35}, A) &= (q_{44}, C, R) \\
\phi(q_{44}, C) &= (q_{35}, A, R) \\
\phi(q_2, C) &= (q_{15}, A, R) \\
\phi(q_{15}, A) &= (q_2, C, R) \\
\phi(q_{35}, C) &= (q_{45}, A, R) \\
\phi(q_{45}, A) &= (q_{35}, C, R)
\end{aligned}$$

Megjegyzés: általános esetben szükség lehet arra, hogy a Turing-gép "hely-csináló", illetve "hely-törlő" lépéseket is végezzen (amikor is a fej egyesével jobbra, illetve balra másolja a szalagon jevő jeleket).