

DÖMÖSI PÁL, FAZEKAS ATTILA, HORVÁTH GÉZA, MECSEI ZOLTÁN

Formális Nyelvek és Automaták

Egyetemi jegyzet

Lektorálta NAGY BENEDEK

2003
Debreceni Egyetem
Informatikai Intézet
Debrecen

Bevezetés

Az algoritmikus szemlélet kialakulására nagy hatással volt az 1900-ban Párizsban rendezett első matematikai világkongresszus, aholis többek között elhangzott David Hilbert híres előadása, melynek hatására tág teret kapott az absztrakt gondolkodás-mód széleskörű alkalmazása. Hilbert még úgy vélte, hogy létezik olyan univerzális módszer, melynek segítségével minden matematikai kérdés igaz vagy hamis volta eldönthető.

A modern elméleti számítástudomány egyik fontos fogalma a formális rendszer, melynek bevezetése Axel Thue (ejtsd: tyué) nevéhez fűződik, aki az 1910-es évek elején kezdte tanulmányozni az adatsorozatokkal megadott utasítások tulajdonságainak vizsgálatát. A fejlődésnek e téren is újabb lökést adtak a fiatal Kurt Gödel felfedezései, aki a 20-as, illetve 30-as években kimutatta, hogy létezik olyan precízen megfogalmazható matematikai állítás, melynek sem igaz, sem hamis volta nem bizonyítható. Ez a korszakalkotó felfedezés alapjaiban rengette meg a matematikát. Ezután már senki sem lehetett biztos abban, hogy egy tetszőlegesen kiválasztott matematikai állítás igaz vagy hamis voltának eldöntése lehetséges. Erre csattanós példa volt az ugyancsak ifjú Jurij V. Matijaszevics esete, aki a 70-es évek elején kimutatta Hilbert már említett híres előadásában szereplő egyik kérdésről, hogy eldönthetetlen. (Hilbert 10. problémája.) A 30-as évek második felében egy másik fiatal óriás, Alan Turing (ejtsd: tyuring) vizsgálataival párhuzamosan, Alonzo Church (ejtsd: csörcs) kapott hasonló módon meglepő eredményt. Nevezetesen, hogy nem létezik univerzális feladatmegoldó módszer.

A gépies bizonyítások vizsgálata elvezetett az automaták matematikai elméletének 50-es évekre kialakult megalapozásához, az idegen szövegek gépi fordításának vizsgálata pedig ugyancsak az 50-es években a formális rendszerek egy fontos típusa, a formális nyelvek matematikai elméletének kezdeteihez. Ezen a területen Noam Chomsky (ejtsd: csomszki) alapvető felfedezései jelentettek nagy lépéseket. Végül, de nem utolsósorban meg kell említenünk, hogy az automaták matematikai elméletének megalapozásában nagy szerepet játszott a pályafutásának nagy részét az Egyesült Államokban töltő magyar származású nagy tudós, Neumann János, aki John von Neumann néven szerte a világon mint a számítástudomány atyja, ismeretes. 1960-ban bekövetkezett korai halála sajnos megakadályozta abban, hogy az e téren (is) végzett úttörő kutatásait teljeskörűen megalapozza.

Már Neumann János is látta, hogy a bonyolultság és az azzal kapcsolatos kérdések nagy szerepet fognak játszani a számítástudományban. Valóban, a számítógépek 60-as évektől bekövetkezett széleskörű elterjedésével mind az elméleti, mind pedig a gyakorlati szakemberek azt tapasztalták, hogy bizonyos kérdéseket még a legmodernebb számítástechnikai eszközök felhasználásával sem képesek kezelni. Ezek a problémák nemcsak elméleti, hanem nagyon fontos gyakorlati feladatokkal kapcsolatban is felléptek. Hamarosan kiderült, hogy ez a gond nem fog megoldódni a számítógépek rohamos fejlődésével sem, mivel valószínűleg elvi akadályokról van szó. A kérdéses feladatok túl bonyolultnak látszanak ahhoz hogy kezelni tudjuk őket. A 70-es évek elején ezen kérdések kezelésére S. A. Cook (ejtsd: kuk), R. M. Karp és L. A.

Levin egy új elmélet megalapozását kezdeményezték. Megszületett az algoritmusok bonyolultságelmélete. Az eltelt több mint 30 év alatt azonban sajnos ez az elmélet nem adott választ a feltett legégetőbb kérdésekre. Ezek közül is a legfontosabb, az úgynevezett $P = NP?$ probléma, melynek megoldása választ adna arra a nagyon fontos kérdésre, hogy egyes, a gyakorlatban is lépten-nyomon felmerülő feladatokhoz létezik-e hatékony megoldó algoritmus. Ez a több mint 30 éves nagy kérdés valószínűleg még sokáig megoldatlan marad.

A bonyolultságelmélet által felvetett problémák a 80-as évek közepére-végére elvezettek egy új terület, az úgynevezett kooperatív rendszerek megszületéséhez. Kezdetben csak arról volt szó, hogy olyan számítástechnikai eszközök születtek, melyek egyes feladatok egyidejű, párhuzamos végrehajtására voltak képesek. Ma már a kooperatív rendszerek széleskörűen elterjedtek és mind elméleti, mind pedig gyakorlati szempontból egyre nagyobb teret kapnak olyan berendezések és módszerek, amik eddig nem, vagy csak csekély mértékben voltak használatosak. Talán ezen az úton haladva fogunk eredményt elérni a még mindig megközelíthetetlennek látszó problémák kezelésében.

A formális nyelvek és automaták elmélete kiváló eszköznek bizonyult ezen kérdések absztrakt szintű vizsgálatában. Jelen jegyzet célja ezen elmélet legalapvetőbb kérdéseinek megismertetése. Az ismertett módszerek ma már lépten-nyomon használatosak nemcsak elméleti problémák vizsgálatában, hanem egyes gyakorlati számítástechnikai feladatok megoldásában is. Helyenként nagy figyelmet fordítunk annak igazolására is, hogy bizonyos struktúrák és módszerek nem léteznek, illetve nem lehetségesek. Ez a fáradság a gyakorlati számítástechnikai szakember számára feleslegesnek tűnik. Jelen jegyzet szerzői, akik egyike több mint másfél évtizedig gyakorlati számítástechnikai szakemberként dolgozott, másképp gondolják. Ha ismerjük a leggyakoribb számítástechnikai zsákutcákat, akkor könnyebben ki tudjuk kerülni azokat. Ha nem, akkor úgy járhatunk mint ahogy a jegyzet nevezett szerzője is járt nem egy esetben: esetleg olyat próbálunk keresni, amiről már kiderült, hogy nincs.

Tartalomjegyzék

1. A formális rendszer fogalma és főbb típusai	1
1.1. Formális nyelv, szabad monoid, szabad félcsoporth	1
1.2. Formális rendszer és néhány főbb típusa	3
1.3. Algoritmus, nyelvtan	9
2. Az automaták elméletének alapjai	17
2.1. Az automata fogalma és főbb típusai	17
2.2. Az automaták megadása	21
2.3. Az automata mint algebrai struktúra: részautomata, homomorfizmus, izomorfizmus, kompatibilis osztályozás	25
2.4. Az automaták által indukált leképezések	31
2.5. Redukált automata. Véges automaták minimalizálása	37
2.6. Automaták Ekvivalenciája, Gill Tétele	45
2.7. Reguláris Kifejezések	47
2.8. Automaták Analízise és Szintézise. Nyelvek Előállítását Automatákban	49
2.9. Véges Automaták Analízise	52
2.10. Véges Automaták Szintézise	55
2.11. Véges Automaták Által Indukálható Leképezések	57
3. Formális Nyelvek: Definíciók és jelölések	61
3.1. Alapfogalmak	61
3.2. A Chomsky-féle nyelvosztályok	62
3.3. Nyelvekre értelmezett műveletek	63
4. Nyelvek és Automaták	67
4.1. Reguláris Nyelvtanok és Véges Automaták	67
4.2. Környezetfüggetlen Nyelvtanok és Veremutamaták	69
4.3. Mondatszerkezetű Nyelvek s a Turing-gépek	72
4.4. Környezetfüggő Nyelvek s a Lineárisan Korlátolt Automaták	74
4.5. A Turing-gépek megállási problémája és az algoritmikusan eldönthetetlen feladatosztályok kapcsolata	75
5. Környezetfüggetlen nyelvek	79
5.1. Chomsky-féle normál alak	79
5.2. Levezetési gráf	80
5.3. Lineáris grammatikák	81
6. Környezetfüggő nyelvek	83
7. Eldönthetőség	85
8. Szintaktikai elemzés	87
8.1. Az Early-féle algoritmus	90
Irodalomjegyzék	95

1. A formális rendszer fogalma és főbb típusai

1.1 Formális nyelv, szabad monoid, szabad félcsoporth

Legyen V nem üres és véges halmaz, amelyet gyakran *ábécének* fogunk hívni, elemeit pedig *betűknek* is fogjuk mondani. Jelölje V^+ (ejtsd: véplusz) az V -beli betűkből felírható $p = x_1x_2 \dots x_k$ ($x_1, \dots, x_k \in V$) alakú véges hosszúságú sorozatok, az úgynevezett V feletti nem üres szavak halmazát. Egy p szó hosszát $|p|$ -al (ejtsd p hossza) jelöljük és rajta a p -beli betűk számát értjük (esetleges többszöri előfordulással együtt). Így ha $p = x_1 \dots x_k$ ($x_1, \dots, x_k \in V$) akkor $|p| = k$.

Példa 1.1 Legyen az V ábécé két betűje a és b . Ekkor $bab \in V^+$ és $|bab| = 3$ (és nem 2, mert a b betű kétszer is előfordul, s a többszöri előfordulást figyelembe kell venni.)

Szokás beszélni az úgynevezett *üres szóról* is, ami egy matematikai absztrakció, ugyanis olyan szót jelent, melynek egyetlen betűje sincs. Speciálisan, jelölje λ a továbbiakban az üres szót. Tehát $|\lambda| = 0$. Az $V^+ \cup \{\lambda\}$ halmazt az V feletti (összes) szavak halmazának hívjuk, s V^* -al (ejtsd vécsillag) jelöljük. Speciálisan, ha valamely $x \in V$ -re $V = \{x\}$, azaz V egyelemű halmaz, s egyetlen eleme egy bizonyos x betű, akkor gyakran írunk $\{x\}^*$ helyett x^* -t, illetve $\{x\}^+$ helyett x^+ -t. Valamely V^* -beli $p = x_1 \dots x_m$ és $q = y_1 \dots y_n$ ($x_1, \dots, x_m, y_1, \dots, y_n \in V$) szavakat pontosan akkor tekintjük egyenlőknek, ha $m = n$ és minden $i (= 1, \dots, m)$ -re $x_i = y_i$. Ezt a tényt ki szokás úgy is fejezni, hogy V^* -ban csak *grafikus* (betűről - betűre megegyező) *egyenlőségek* léteznek. Az V^* halmaz részhalmazait V feletti *formális nyelveknek*, vagy röviden V feletti *nyelveknek*, vagy csak egyszerűen *nyelveknek* hívjuk. Valamely $L \subseteq V^*$ nyelvet *üresnek*, *végesnek* vagy *végtelennek* hívunk ha az L (mint halmaz) üres, véges, illetve végtelen.

A szóban forgó nyelvfogalom természetesen nagyon általános. Magában foglalja mind a mesterséges mind pedig a természetes (írott) nyelvek összességét. Mi a továbbiakban olyan nyelveket fogunk csak vizsgálni, melyek véges sok adat segítségével speciális módon, az úgynevezett generatív nyelvtannal (lásd 1.3. fejezet) megadhatók. Megjegyezzük azt is, hogy az általunk használt szófogalom nem esik egybe a természetes nyelvek szófogalmával, hisz egy természetes nyelvet úgy szokás tekinteni, mint az adott nyelv összes mondatainak halmazát. De tekinthetünk egy természetes nyelvet úgy is mint a nyelv összes véges hosszú szövegeinek halmazát. A szóközt és egyéb írásjeleket is felvéve az ábécébe, az általunk definiált szófogalom amellet, hogy magában foglalja a szokásos szófogalmat, magában foglalja mind a mondat fogalmát, mind pedig a tetszőleges végeshosszúságú szöveg fogalmát is.

Példa 1.2 Legyen $V = \{1, 2, +\}$. Ekkor (V^* - ban !) fennáll, hogy $1 + 1 \neq 2$, mivel az $1 + 1$ szó nem egyezik meg "betűről - betűre", azaz grafikusan a 2 szóval.

Példa 1.3 Legyen $V = \{0, \dots, 9\}$. A (magyar) történelmi dátumok $\{1514, 1526,$

$1606, 1711, 1849\}$ halmaza ekkor egy V feletti véges nyelv. A (tízes számrendszerbeli) páros számok halmaza egy V feletti végtelen nyelv. Természetesen az üres halmaz is (jele a továbbiakban: $\emptyset$) is egy V feletti (üres) nyelv. Ugyancsak V feletti (véges) nyelv az egy elemű $\{\lambda\}$ halmaz is, amit (helytelenül) gyakran össze szoktak tévesztetni $\emptyset$ -el (az üres nyelvvel.)

Az V^* halmazon (és az V^+ halmazon is) szokás bevezetni egy (nagyon egyszerű tulajdonságú) műveletet, melyet *szorzásnak* nevezzük. A $p = x_1 \dots x_m$ és $q = y_1 \dots y_n$ ($x_1, \dots, x_m, y_1, \dots, y_n \in V$) szavak szorzatán a $pq = x_1 \dots x_m y_1 \dots y_n$ szót értjük. Két V^* -beli (vagy két V^+ -beli) szót tehát úgy szorzunk össze, hogy a szavakat (megfelelő sorrendben) egymás mellé írjuk. E műveletet *konkatenációnak* vagy *összefűzésnek* is szokás hívni. Természetesen ez a szorzásfajta általában *nem kommutatív*, azaz általában nem teljesül minden $p, q \in V^*$ párra a $pq = qp$ egyenlőség. Amennyiben $p = p_1 \dots p_k$ ($\in V^*, k = 1, 2, \dots$), továbbá $p_1 = \dots = p_k = q$ ($\in V^*$), úgy alkalmazni fogjuk a $p = q^k$ jelölést, s ezesetben p -t a q szó k -adik hatványának is nevezzük. Továbbá megállapodunk abban, hogy minden szó *nulladik hatványa* az üres szó (jelekben: $\forall q \in V^* : q^0 = \lambda$.) Röviden úgy is mondhatjuk, hogy egy szó k -adik hatványa nem más mint k -szoros ismétlődése (beleértve a nullaszeres ismétlődést is).

Példa 1.4 Legyen ismét $V = \{a, b\}$. Ekkor az abba V^* -beli szó baba V^* -beli szóval való szorzata abba**ba** lesz (ami persze nem egyezik meg a baba**abba** szóval. A szorzás tehát valóban, általában *nem kommutatív*). Igaz továbbá (definíció szerint), hogy $baba = (ba)^2$.

Feladat 1.5 Legyen most $V = \{a\}$, azaz álljon ábécénk egyetlen betűből. Mutassuk meg hogy ebben a kivételes esetben a szorzás kommutatív.

Az V^* -ban ezen szorzás műveletre nézve a λ üres szó egységelem lesz, hisz minden $p \in V^*$ -ra $p\lambda = \lambda p = p$ (annak megfelelően, hogy ha egy szó - beleértve az üres szót is - elé vagy mögé nem írunk egyetlen betűt sem, azaz az üres szót "írjuk", akkor marad az eredeti szó). Nyilvánvaló továbbá, hogy minden $p, q \in V^*$ párra $|pq| = |p| + |q|$. Az előbbieken definiált szorzás műveletével ellátott V^* halmazt az V által generált egységelemes szabad félcsoporthnak, más néven V feletti egységelemes szabad félcsoporthnak, vagy röviden, V feletti monoidnak hívjuk, s rá ugyancsak az V^* jelölést használjuk. Hasonlóan, a szorzás műveletével ellátott V^+ halmazt az V által generált szabad félcsoporthnak, más néven V feletti szabad félcsoporthnak hívjuk, s rá ugyancsak az V^+ jelölést használjuk. (V^* tehát egyidejűleg jelöli az összes V feletti szavak halmazát és az V feletti szabad monoidot, míg V^+ az összes V feletti nem üres szavak halmazát és az V feletti szabad félcsoporthot.) Ha mást nem mondunk, a továbbiakban V^* , illetve V^+ a szóban forgó (V feletti) szabad monoidot, illetve szabad félcsoporthot jelöli. Ha tehát nem ezen műveletekkel ellátott halmazokról, azaz *struktúrákról*, hanem csak a hozzájuk tartozó halmazokról (vagyis az összes, avagy összes nem üres V feletti szavak halmazáról) lesz szó, azt a félreértések elkerülése végett külön hangsúlyozni fogjuk. (Nyilvánvaló, hogy V^+ zárt marad a szorzás, azaz az összefűzés műveletére, hisz két nem üres szót egymás után írva, azaz összefűzve, nem kaphatunk üres szót.)

Legyen p és q tetszőleges két szó V^* -ban. Azt mondjuk, hogy p *kezdő szelete* q -nak, ha van olyan $r \in V^*$, hogy $q = pr$. Pontosabban, ha $q = pr$ mellett $|p| = k$, akkor a p

szót a q szó k hosszúságú kezdő szeletének nevezzük. Hasonlóan, ha van olyan $s \in V^*$ szó, hogy $q = sp$, akkor azt mondjuk hogy p a q -nak *végződése* és ha $|p| = m$ akkor m hosszúságú végződésről beszélünk. Végül, a $p \in V^*$ szót a $q \in V^*$ szó *rész-szavának* mondjuk, ha van olyan $r, s \in V^*$, hogy $q = rps$. Amennyiben $p \neq q$, *valódi rész-szóról* beszélünk. Hasonlóan, egy szó önmagától különböző kezdőszeleteit, illetve végződéseit *valódi kezdő szeletnek*, illetve *valódi végződésnek* hívjuk.

Egy $p \in V^+$ szó utolsó betűjére használni fogjuk a $\overline{p}$ jelölést.

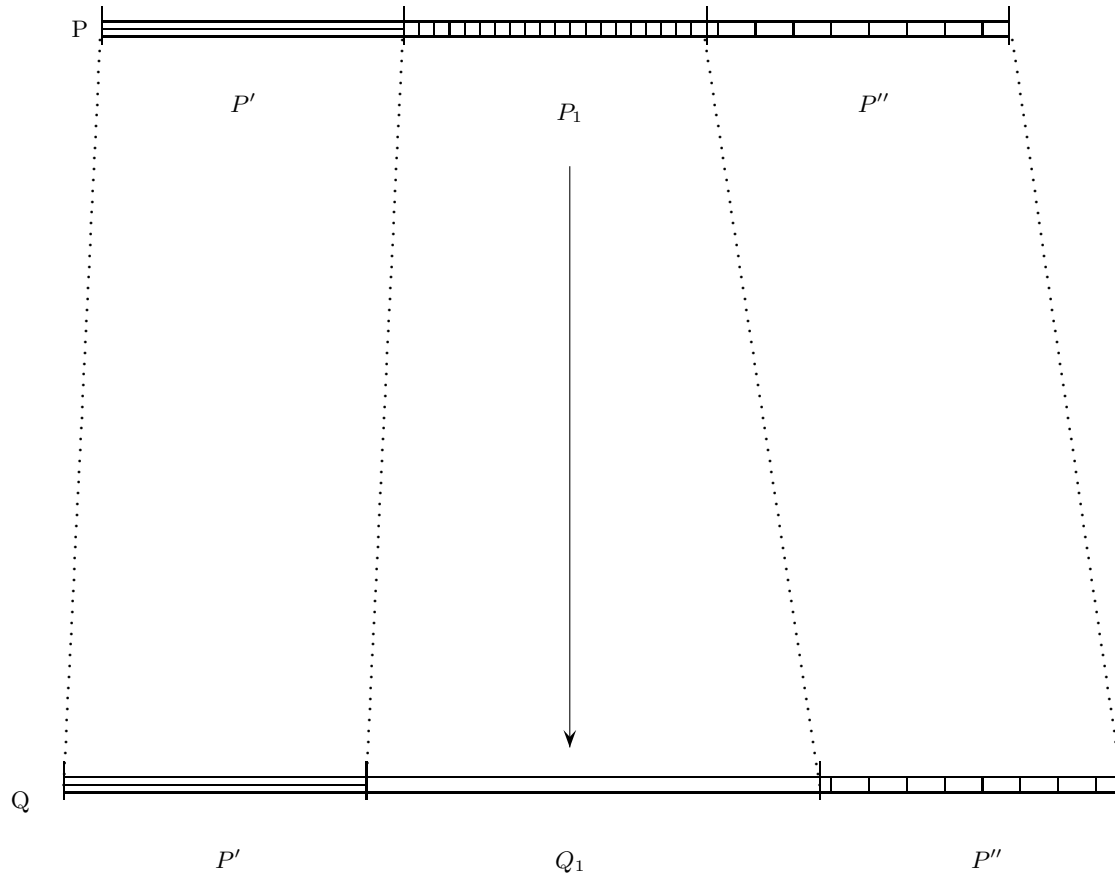
Példa 1.6 Legyen $V = \{a, b\}$, $p = abbababa$. Ekkor p -nek 4 hosszúságú kezdő szelete *abba*, 4 hosszúságú végződése pedig *baba* lesz. Ugyanekkor p -nek például a *bab* szó (valódi) rész-szava. Végül, $\overline{abbababa} = a$ (*abbababa* utolsó betűje).

1.2 Formális rendszer és néhány főbb típusa

Formális rendszernek (vagy átírási rendszernek) nevezzük minden olyan $W = (V, H)$ párt, amelyben V egy ábécé, H pedig a $V^* \times V^*$ direkt szorzat egy véges részhalmaza. A H elemeit *helyettesítési szabályoknak* hívjuk és ha $(P, Q) \in H$, akkor a $P \rightarrow Q$ (ejtsd: P nyíl Q) jelölést használjuk.

Példa 1.7 $W = (\{a, b, d, e, é, f, k, l, ő, s, v, z\}, \{aked \rightarrow le, leves \rightarrow f őzelék\})$.

Legyen $P, Q \in V^*$. Akkor mondjuk, hogy P -ből a Q *közvetlenül* (vagy *egy lépésben*) *levezethető* W -ben, jelekben: $P \xrightarrow{W} Q$, vagy ha nem vezet félreértéshez, W elhagyásával $P \Rightarrow Q$, ha léteznek olyan $P', P_1, P'', Q_1 \in V^*$ szavak, hogy $P = P'P_1P''$, $Q = P'Q_1P''$ és $P_1 \rightarrow Q_1 \in H$. (Természetesen megengedett, hogy akár a P' , akár a P'' , vagy akár mindkettő az üres szó legyen.) Szemléletesen, $P \xrightarrow{W} Q$ azt jelenti, hogy a Q szó megkapható a P szóból úgy, hogy P -ben valamely H -beli szabály baloldalán álló P_1 rész-szó helyébe az e szabály jobboldalán álló Q_1 szót írjuk (lásd ábra).



Azt mondjuk, hogy P -ből a Q *levezethető* W -ben, jelekben: $P \xrightarrow{W}^* Q$, vagy ha nem vezet félreértéshez, W elhagyásával: $P \Rightarrow^* Q$ ha léteznek olyan $P_0, P_1, \dots, P_k \in V^*$ szavak, hogy $P_0 = P, P_k = Q$ és $P_i \xrightarrow{W} P_{i+1}$ ($i = 0, \dots, k-1$). Emellett, amint szokásos, megállapodunk abban, hogy minden $P \in V^*$ szóra $P \xrightarrow{W}^* P$ fennáll. Használni fogjuk még a $P \xrightarrow{W}^+ Q$ jelölést is abban az esetben, ha azt akarjuk hangsúlyozni, hogy léteznek olyan $P_0, P_1, \dots, P_k \in V^*$ szavak, hogy $P_0 = P, P_k = Q$ és $P_i \xrightarrow{W} P_{i+1}$ ($i = 0, \dots, k-1$). Világos, hogy a két jelölés $P = Q$ esetén kap eltérő értelmet.

Példa 1.8 Tekintsük az 1.7. példában megadott formális rendszert és a *babakedves* szót. Ekkor a $\text{bab}\boxed{\text{aked}}\text{ves} \Rightarrow \text{bab}\boxed{\text{leves}} \Rightarrow \text{babfőzelék}$ levezetés alkalmazásával látjuk, hogy például a W -beli *babakedves* szóból a *bableves* közvetlenül levezethető, a *babfőzelék* pedig levezethető.

Amennyiben olyan formális rendszereket tekintünk, melyekben a helyettesítési szabályok szimmetrikusak, eljutunk az *asszociatív kalkulus* fogalmához. Ha tehát valamely asszociatív kalkulus esetén egy P szó helyettesíthető Q -val, akkor a Q szó is helyettesíthető P -vel. Emiatt nem is helyettesítési szabályokról, hanem *megengedett cserékről* szokás beszélni. Továbbá, ha egy P szóból egy Q szó levezethető az asszociatív kalkulusban, akkor azt mondjuk hogy P ekvivalens Q -val.

Példa 1.9 Legyen $W = (\{a, c, k, m, ó, u, s, t, y\}, \{acs - ó, ka - kus\})$. Asszociatív kalkulus esetén a szabályok bal és jobboldalát $\rightarrow$ helyett $-$ -al szokás elválasztani, ezzel is hangsúlyozva, hogy a szabályok szimmetrikusak. Példánknál maradva, az acs helyettesíthető az $ó$ -val és az $ó$ is helyettesíthető acs -sal. Ugyanígy, a ka helyettesíthető kus -sal és a kus a ka -val. Tekintsük a $m\boxed{acs}ka \Rightarrow m\boxed{ka} \Rightarrow m\acute{o}kus$ levezetést.

Ekkor a macska ekvivalens W -ben a mókus-al. Ehhez $\xrightarrow{*}_W$ helyett $\simeq_W$ -t, vagy ha nem okoz félreértést, akkor egyszerűen (az ekvivalencia egyik szokásos jelét,) $\simeq$ -t szokás használni, azaz $macska \simeq m\acute{o}kus$. Vizsgáljuk meg most ezt az $\simeq$ -t mint relációt. Ez az $\simeq$ reláció reflexív, azaz minden V^* -beli P szóra $P \simeq P$. Például, $macska \simeq macska$. Az $\simeq$ szimmetrikus, azaz minden V^* -beli P, Q szópárra $P \simeq Q$ akkor és csak akkor ha $Q \simeq P$. Például, $macska \simeq m\acute{o}kus$ következménye $m\acute{o}kus \simeq macska$ és viszont. Ugyanígy, példánkban $kutya \not\simeq macska$ következménye $macska \not\simeq kutya$ és viszont. Az $\simeq$ tranzitív, azaz minden V^* -beli P, Q, R szó-hármasra $P \simeq Q$ és $Q \simeq R$ esetén $P \simeq R$.

Példánkban $macska \simeq m\acute{o}ka$ és $m\acute{o}ka \simeq m\acute{o}kus$ miatt $macska \simeq m\acute{o}kus$. (De ugyanígy igaz, hogy $macska \simeq m\acute{o}kus$ és $m\acute{o}kus \simeq m\acute{o}ka$ miatt $macska \simeq m\acute{o}ka$.) (Meggjegyzés: a reflexív és tranzitív tulajdonság közvetlenül következik a levezethetőség definíciójából. A szimmetria a szabályok szimmetrikus volta miatt, továbbá a tranzitivitás miatt áll fenn, melynek igazolását az olvasóra bizzuk.)

Emlékeztető 1.10 Egy M halmaz önmagával való Descartes szorzatának (jelekben: $M \times M$ -nek) részhalmazait M feletti bináris relációnak, vagy röviden, relációnak szokás nevezni. Speciálisan, $M \times M$ véges részhalmazait véges relációknak is szokás hívni (M felett).

Felhasználva az előző példánkban tárgyaltakat, vegyük észre, hogy ha egy adott $W = (V, H)$ formális rendszerbeli közvetlen levezethetőséget mint a V^* halmaz feletti (véges) bináris relációt tekintjük, úgy a belőle származtatható levezethetőség nem más mint a közvetlen levezethetőség reflexív és tranzitív lezárása, vagyis az a legszűkebb reflexív és tranzitív reláció, melynek a tekintett levezethetőség rész-relációja. Asszociatív kalkulus esetén, mint példánkban már említettük, ez a levezethetőség (mint reláció) ekvivalencia reláció lesz, ugyanis a reflexív és tranzitív tulajdonság mellett a szimmetrikus tulajdonsággal is rendelkezni fog. Emiatt szokás egy $W = (V, H)$ asszociatív kalkulusban *ekvivalensnek* hívni a $P, Q \in V^*$ szavakat, ha $P \xrightarrow{*}_W Q$, azaz ha $P \simeq Q$.

Akkor mondjuk, hogy a $W = (V, H)$ asszociatív kalkulusban a szóprobléma *algoritmikusan megoldható*, ha létezik olyan algoritmus, melynek segítségével tetszőleges P, Q párra eldönthető, hogy P és Q ekvivalens-e (azaz $P \simeq Q$ fennáll-e) és ha igen, akkor az algoritmus egy $P \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \dots \Rightarrow P_{n-1} \Rightarrow Q$ levezetést is szolgáltat. Ezen fogalomnak azért van különös jelentősége, mert kimutatható, hogy nem minden asszociatív kalkulusban oldható meg a szóprobléma algoritmikusan (ami egyúttal azt is igazolja, hogy nincs olyan univerzális módszer, ami minden matematikai problémát képes megoldani). Később még vissza fogunk térni erre a dologra.

Egy $W = (V, H)$ formális rendszert *generatív rendszernek* nevezünk, ha ki van tüntetve V^* egy nem üres és véges részhalmaza, amelyet W *axiómarendszerének*

nevezünk (és rá többnyire az Ax jelölést használjuk). Egy ilyen W generatív rendszert $W = (V, Ax, H)$ alakban szokás megadni (aholis V az ábécé, Ax az axiómák, H pedig a szabályok nem üres és véges halmaza). W -hez hozzárendelünk egy L_{gW} halmazt az $L_{gW} = \{P \in V^* \mid \exists A \in Ax : A \xrightarrow{*}_W P\}$ definícióval, s ezt a halmazt a W generatív rendszer által generált nyelvnek hívjuk. L_{gW} tehát tartalmazza mindazon V^* -beli szavakat, melyek legalább az egyik axiómából levezethetők.

Egy $W = (V, Ax, H)$ generatív rendszert úgy is szokás interpretálni (azaz értelmezni), hogy V bizonyos fogalmak rendszere, Ax az axiómarendszer (vagyis az alapigazságok, vagy igaznak tartott alapvető állítások gyűjteménye), V^* elemei a V fogalomkörben megfogalmazható (értelmes vagy értelmetlen) formális mondatok halmaza, H a W -ben megengedett elemi bizonyítási lépések halmaza, $P \Rightarrow P_1 \Rightarrow \dots \Rightarrow P_n \Rightarrow Q$ a Q formális mondat P -ből való bizonyítása, L_{gW} pedig a V fogalomkörben megfogalmazható és az Ax axiómarendszerből W -ben bebizonyítható formális mondatok (kijelentések) halmaza.

Analóg módon, tekinthetjük az összes olyan szavak halmazát is, melyekből az axiómák közül legalább az egyik levezethető. Ebben az esetben W -t (a generatív rendszer elnevezés helyett) *analitikus rendszernek*, az $L_{aW} = \{P \in V^* \mid \exists A \in Ax : P \xrightarrow{*}_W A\}$ halmazt pedig a W analitikus rendszer által *acceptált*, vagy más szóval, a W analitikus rendszer által *elfogadott nyelvnek*, vagy még másképp a W analitikus rendszer által *felismert nyelvnek* hívjuk.

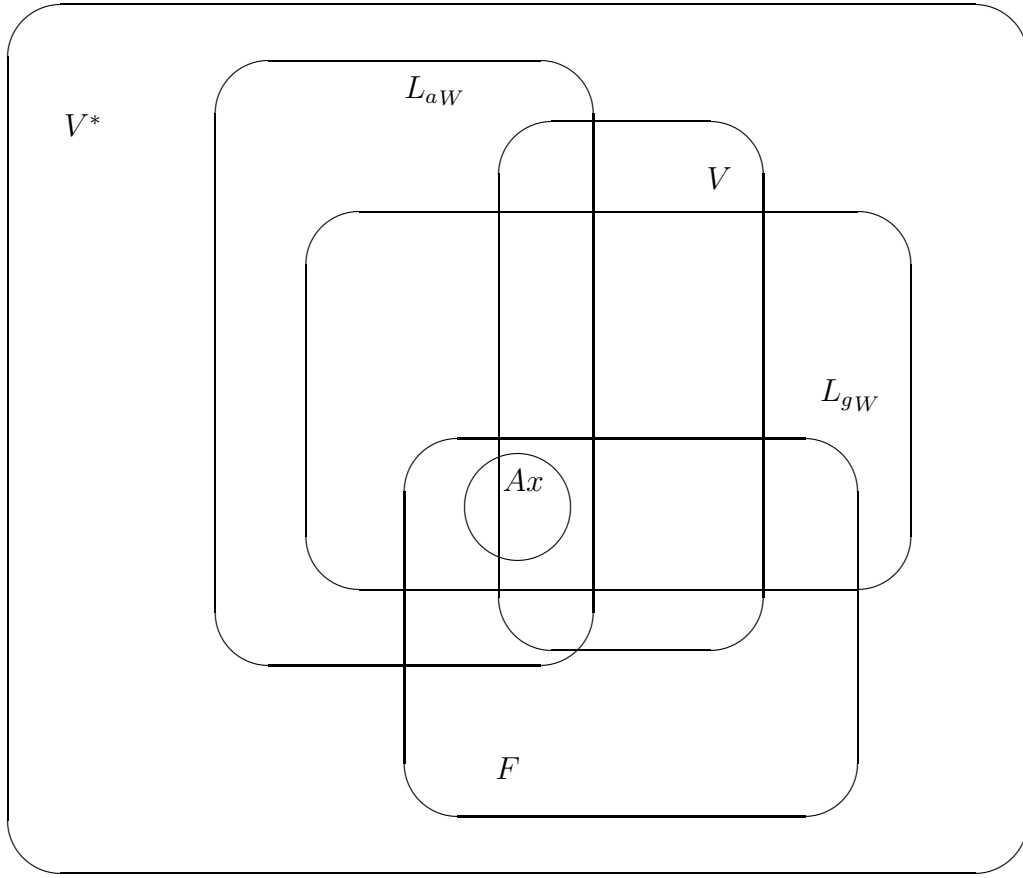
Az analitikus rendszer által elfogadott nyelvet a későbbiekben fogjuk interpretálni (azaz értelmezni).

Példa 1.11 Vegyük a $W = (\{1, 2, +, =\}, \{1 + 1 = 2\}, \{= \rightarrow +1 = 1+\})$ generatív rendszert. Foglmaink az 1, 2 természetes számok, továbbá az összeadás és az egyenlőség (jele). Egyetlen axiómánk van, $1 + 1 = 2$. Az $1 + 1 + 1 + 1 = 1 + 1 + 2$ kijelentés (vagy "tétel"-nek is lehetne mondani) bebizonyítható W -ben. Ime a "bizonyítás:" (Egyetlen) axiómánk $1 + 1 = 2$. Ebből indulunk ki :

$$1 + 1 \boxed{=} 2 \Rightarrow 1 + 1 + 1 \boxed{=} 1 + 2 \Rightarrow 1 + 1 + 1 + 1 = 1 + 1 + 2.$$

Példa 1.12 Tekintsük a $W = (\{<, >, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{< 0 >, < 5 >\}, \{00 \rightarrow 0, 10 \rightarrow 0, 20 \rightarrow 0, 30 \rightarrow 0, 40 \rightarrow 0, 50 \rightarrow 0, 60 \rightarrow 0, 70 \rightarrow 0, 80 \rightarrow 0, 90 \rightarrow 0, 05 \rightarrow 5, 15 \rightarrow 5, 25 \rightarrow 5, 35 \rightarrow 5, 45 \rightarrow 5, 55 \rightarrow 5, 65 \rightarrow 5, 75 \rightarrow 5, 85 \rightarrow 5, 95 \rightarrow 5\})$ analitikus rendszert. Az olvasóra bízunk annak igazolását, hogy L_{aW} épp az öttel osztható, $< a_1 \dots a_n >$ ($a_1, \dots, a_n \in \{0, \dots, 9\}$) alakban megadott nemnegatív egészek halmaza. W tehát "felismeri" az 5-tel osztható nemnegatív egészeket.

Egy $W = (V, Ax, H)$ generatív rendszert *szemi-Thue rendszernek* (ejtsd: tyué) nevezünk (az angol semi[ejtsd:szemi] = félig-, fél szó alapján), ha ki van tüntetve V^* egy olyan F részhalmaza, melyre $Ax \subseteq F$. Egy ilyen szemi-Thue rendszert $W = (V, Ax, H, F)$ alakban adunk meg, ahol F a *formulák* halmaza, és a W -nek a generatív rendszerhez hasonló értelmezése esetén mondhatjuk még azt is, hogy F az igaz (vagy igaznak tartott) állítások (kijelentések, mondatok, tételek) halmaza (lásd ábra).



Példa 1.13 Képezzünk egy szemi-Thue rendszert oly módon, hogy az 1.11. példabeli generatív rendszert kiegészítsük formulákkal. Ezzel összhangban, legyen

$W = (\{1, 2, +, =\}, \{1 + 1 = 2\}, \{= \rightarrow +1 = 1+\}, \{1 + 1 = 2, 1 + 1 + 1 + 1 = 2 + 2, 1 + 1(+1)^n = (1+)^n + 2, \quad n = 1, 2, \dots\})$.

Amint szokásos, $(+1)^n$ itt azt jelenti, hogy $+1$ n -szer van írva. Például: $(+1)^3 = +1 + 1 + 1$. Hasonlóan, $(1+)^n$ jelentése az, hogy $1+$ van n -szer írva. Például: $(1+)^2 = 1 + 1+$. A nyilvánvalóan igaz $1 + 1 + 1 + 1 = 2 + 2$ összefüggés a példabeli szemi-Thue rendszerben nem bizonyítható. Mint már a bevezetésben elmondtuk, ezen a játékos példán bemutatott probléma a matematika egyes nehezebb fejezeteiben is fennáll. Nevezetesen, vannak precízen megfogalmazható, de nem bizonyítható és nem is cáfolható matematikai kijelentések.

Legyen $W = (V, Ax, H, F)$ tetszőleges szemi-Thue rendszer, s jelölje $Ax \xRightarrow{*} P$ röviden azt a tényt, hogy létezik olyan $P_0 \in Ax$, melyre $P_0 \xRightarrow{*} P$. Ezt a jelölést használva $L_W = \{P \in V^* \mid Ax \xRightarrow{*} P\}$ és W -ben absztrakt módon az F halmaz mint elmélet (= igaz vagy igaznak tartott állítások halmaza, rendszere, vagy gyűjteménye) axiomatizálásának tipikus kérdései a következőképp fogalmazhatók meg:

a.) teljes-e az elmélet, vagyis az Ax axiómarendszerből minden igaz állítás bebizonyítható-e. Jelöléssel: fennáll-e az $F \subseteq L_{gW}$ tartalmazás;

b.) *ellentmondásmentes* (, azaz *helyes*)-e az *elmélet*, vagyis igaz-e, hogy az axiómarendszerből nem vezethető le egyidejűleg igaz és hamis állítás is. Képletben: igaz-e, hogy akárhogy is adjuk meg a $P \in F$ és $Q \notin F$ párokat, $Ax \stackrel{*}{\Rightarrow} P$ mellett $Ax \stackrel{*}{\Rightarrow} Q$ nem állhat fenn egyidejűleg. (Vegyük észre, hogy ha $Ax \stackrel{*}{\Rightarrow} Q$ és $Q \notin F$, akkor felhasználva $Ax \stackrel{*}{\Rightarrow} Q$ jelentését, lenne olyan $P_0 \in Ax$, hogy $P_0 \stackrel{*}{\Rightarrow} Q$, ami $Ax \subseteq F$ és $P_0 \stackrel{*}{\Rightarrow} P_0$ miatt azt is jelentené, hogy $P_0 \in F$ és $Q \notin F$ mellett $Ax \stackrel{*}{\Rightarrow} P_0$ és $Ax \stackrel{*}{\Rightarrow} Q$. Az ellentmondásmentesség feltétele tehát tulajdonképp azt jelenti, hogy hamis kijelentés nem vezethető le az axiómarendszerből.)

c.) *kategórikus* (= *konzekvens*)-e az *elmélet*, vagyis az Ax axiómarendszerből minden igaz állítás, de csakis az igaz állítások bebizonyíthatók-e. Jelöléssel: $F = L_{gW}$ fennáll-e; (Vegyük észre, hogy egy elmélet pont akkor konzekvens, ha egyidejűleg teljes és helyes is.)

d.) *minimális*-e az *elmélet*, azaz igaz-e, hogy egyik axióma sem bizonyítható be a másiktól. Képletben: igaz-e, hogy ha $P, Q \in Ax$ és $P \stackrel{*}{\Rightarrow} Q$, akkor szükségképp $P = Q$.

Példa 1.14 Az 1.13. példabeli "elméletünket" vizsgálva megállapíthatjuk, hogy nem teljes, mert az $F = \{1+1=2, 1+1+1+1=2+2, 1+1(+1)^n = (1+)^n + 2, n=1, 2, \dots\}$ elmélet nem minden kijelentése bizonyítható W -ben (pl. az $1+1+1+1=2+2$ nem). Emiatt "elméletünk" nem is kategórikus. Mivel "hamisnak tartott", azaz F -en kívüli állítás $1+1=2$ -ből (egyetlen axiómánkból) nem vezethető le (ugyanis minden belőle levezethető "állítás" vagy maga az $1+1=2$ lesz, vagy $1+1(+1)^n = (1+)^n + 2$ alakú lesz, ahol n tetszőleges természetes szám), "elméletünk" ellentmondásmentes. Végül, lévén csak egyetlen axiómánk, "elméletünk" nyilvánvalóan minimális.

Végül megjegyezzük, hogy lehetett volna a formális rendszer és a belőle származtatott különféle speciális formális rendszer típusok fogalmának kialakításakor a közvetlen levezethetőséget és a levezethetőséget másképp is definiálni. Erre az egyik legnevezetesebb példa a Post-féle normál rendszer.

A *Post-féle normál rendszer* (hasonlóképp a generatív rendszerhez) egy olyan V ábécéből, Ax axiómarendszerből és H helyettesítési szabályokból felépülő $W = (V, Ax, H)$ hármas (Ax itt is nem üres és véges részhalmaza V^* -nak, továbbá H ezesetben is nem üres és véges részhalmaza $V^* \times V^*$ -nak), melyben H elemei $PX \rightarrow XQ$ alakúak, aholis $X \in V$ egy olyan betű, mely sem P -ben, sem Q -ban nem fordul elő. Akkor mondjuk, hogy a V^* -beli P_1 szóból a V^* -beli Q_1 szó *közvetlenül levezethető* (jelekben, mint korábban : $P_1 \xrightarrow{w} Q_1$, vagy ha nem vezet félreértéshez, egyszerűen csak $P_1 \Rightarrow Q_1$), ha található olyan V -beli X betű, továbbá találhatók olyan $(V \setminus \{X\})^*$ -beli P, Q, R szavak, hogy $P_1 = PR, Q_1 = RQ$, és $PX \rightarrow XQ \in H$. Ezenkívül, mint korábban, azt mondjuk, hogy P -ből a Q *levezethető* W -ben, jelekben: $P \xrightarrow{w}^* Q$, vagy ha nem vezet félreértéshez, W elhagyásával: $P \stackrel{*}{\Rightarrow} Q$ ha léteznek olyan $P_0, P_1, \dots, P_k \in V^*$ szavak, hogy $P_0 = P, P_k = Q$ és $P_i \xrightarrow{w} P_{i+1}$ ($i = 0, \dots, k-1$). Emellett, amint szokásos, itt is megállapodunk abban, hogy minden $P \in V^*$ szóra $P \xrightarrow{w}^* P$ fennáll. (Másként mondva, a levezethetőség mint V^* feletti reláció, ezesetben is reflexív és tranzitív lezárása a közvetlen levezethetőségnak.) Ugyanúgy mint a generatív rendszer esetén,

a Post-féle normál rendszerhez is szokás tekinteni az $L_{gW} = \{P \in V^* \mid \exists A \in Ax : A \xrightarrow{W}^* P\}$ halmazt, amit a W Post-féle normál rendszer *által generált nyelvnek* hívunk. Természetesen mint a generatív rendszerből "képzett" analitikus rendszer esetén, itt is lehet tekinteni az $L_{aW} = \{P \in V^* \mid \exists A \in Ax : P \xrightarrow{W}^* A\}$ halmazt mint a W *által elfogadott nyelvet*. (Emlékeztetőül: szerkezetileg a generatív rendszer ugyanaz mint az analitikus rendszer, csak a hozzájuk rendelt nyelvek szerkezete más.)

A Post-féle normál rendszer fogalmának kis módosításával jutunk el a Post-féle tag (tag [ejtsd: teg] = toldalék, vmit vmihez hozzáfűző eszköz vége) rendszer fogalmához. A *Post-féle tag rendszer* egy olyan $W = (V, S, H)$ hármas, ahol V egy ábécé, H helyettesítési szabályok egy halmaza (azaz $V^* \times V^*$ egy nem üres és véges részhalmaza), S pedig egy tetszőleges, rögzített eleme V^* -nak. Erre az S elemre a *startszó* elnevezés használatos. Akkor mondjuk, hogy a V^* -beli P szóból az ugyancsak V^* -beli Q szó *közvetlenül (vagy egy lépésben) levezethető* W -ben, jelekben: $P \xrightarrow{W} Q$, vagy ha nem vezet félreértéshez, W elhagyásával $P \Rightarrow Q$, ha léteznek olyan $P_1, Q_1, R \in V^*$ szavak, hogy $P = P_1 R$, $Q = R Q_1$ és $(P_1, Q_1) \in H$. (Természetesen megengedett, hogy az R az üres szó legyen.) A levezethetőségre (is) ugyanazt a jelölést használjuk mint korábban, továbbá a levezethetőség mint V^* feletti reláció, ez esetben is reflexív és tranzitív lezárása a közvetlen levezethetőségnak. Ugyancsak összhangban az előzőekkel, a W *által generált nyelv*: $L_{gW} = \{P \in V^* \mid S \xrightarrow{W}^* P\}$. Definiálhatjuk a W *által elfogadott nyelvet is*: $L_{aW} = \{P \in V^* \mid P \xrightarrow{W}^* S\}$.

1.3 Algoritmus, nyelvtan

Az algoritmus intiutív fogalmát körülbelül a következőképp szokás megfogalmazni: A fegyelmezett, egyértelműen előírt elemi lépésekből egyértelműen felépíthető olyan feladatmegoldás modellje, melynek eredménye végrehajtójától függetlenül ugyanazon feladatokra akárhányszor is megismételve ugyanazon eredményt szolgáltatja. Segítségével egy adott feladatosztály minden feladata véges számú lépésben megoldható. (A legáltalánosabb értelemben természetesen egy algoritmus bármilyen emberi vagy nem emberi tevékenységre vonatkozhat, nemcsak matematikai avagy számítástechnikai feladatok megoldására.)

Fontos tulajdonságai közé tartozik tehát a *függetlenség*, ami azt jelenti, hogy ugyanazon feladatra ugyanaz az eredménye függetlenül attól hogy ki (vagy mi) a végrehajtója, az *egyértelműség*, ami azt jelenti, hogy ugyanazon feladatra akárhányszor is alkalmazzuk az algoritmust, mindig ugyanazt az eredményt szolgáltatja, az *elemi lépésekre bonthatóság*, ahol persze ezen elemi lépések, továbbá ezen lépések sorrendje is egyértelműen meg van határozva, valamint a *végesség*, vagyis az, hogy a feladatosztály minden egyes feladatára mindig véges sok lépésben befejeződik. Amennyiben a véges lépésben történő befejeződés követelményétől eltekintünk, az *eljárás* fogalmához jutunk. Ezen és ehhez hasonló intiutív megfogalmazások után születtek meg a 30-as évek második felétől a különféle matematikailag is precíz algoritmusfogalmak, melyek közös érdekessége, hogy ekvivalensek. Ez azt jelenti, hogy ha egy algoritmust az egyik

algoritmusfogalom segítségével megadtuk, akkor ugyanez az algoritmus a másik algoritmusfogalom segítségével is megadható. Ezen fogalmak egyike a Markov-féle normál algoritmus.

Egy $W = (V, H)$ formális rendszert (*Markov-féle*) *normál algoritmusnak* nevezzük, ha H rendezett halmaz és ki van tüntetve benne egy H_1 részhalmaz (megengedve, hogy esetleg $H_1 = \emptyset$, azaz üres halmaz legyen). A H_1 elemeit *záróhelyettesítéseknek* hívjuk és $P \rightarrow Q \in H_1$ esetén használni fogjuk a $P \rightarrow .Q$ jelölést is. Egy ilyen normál algoritmust $W = (V, H, H_1)$ alakban adunk meg, ahol a $H = \{P_1 \rightarrow Q_1, \dots, P_m \rightarrow Q_m\}$ megadás, azaz H elemeinek felsorolása feltételezésünk szerint egyben a H -beli rendezést is mutatja. Legyen $P \in V^*$. Azt mondjuk, hogy a P szóra a $W = (V, H, H_1)$ algoritmus *alkalmazható*, ha van olyan 1 és m közötti $i (= 1, \dots, m)$ index és olyan $P', P'' \in V^*$ szópár, hogy $P = P'P_iP''$. Legyen i a legkisebb ilyen index és válasszuk meg a P szó P' kezdő szeletét úgy, hogy P_i a $P'P_i$ szóban rész-szóként csak egyszer forduljon elő. Rendeljük hozzá P -hez a $Q = P'Q_iP''$ szót és ezt a hozzárendelést jelöljük a következőképpen: legyen $P \Rightarrow_w Q$ ha $P_i \rightarrow Q_i \notin H_1$ és legyen $P \Rightarrow_w Q$ ha $P_i \rightarrow Q_i \in H_1$. E hozzárendelést, mely a formális rendszer fogalmánál bevezetett közvetlen levezethetőség fogalmának a H -beli rendezést is figyelembe vevő módosítása, *elemi helyettesítésnek* szokás nevezni. (Az elemi helyettesítés nem tévesztendő össze a helyettesítéssel, azaz a helyettesítési szabályok halmazának egy elemével.) $P \Rightarrow_w Q$ esetén elemi záróhelyettesítésről beszélünk.

Ha záróhelyettesítést hajtottunk végre az algoritmus megáll, egyébként az algoritmus helyettesítő lépése újra lefut az imént kapott szóra. Az algoritmus tehát mindaddig fut, amíg vagy záróhelyettesítés be nem következik, vagy nincs alkalmazható helyettesítési szabálya. Itt jegyezzük meg, hogy a Markov-féle normál algoritmus a szó eredeti értelmében nem algoritmus, hanem eljárás, mivel – ahogy majd példát is mutatunk rá – nem feltétlenül fejeződik be, tehát a végesség feltétele nem teljesül.

Példa 1.15 Vegyük a $W = (\{a, b, e, i, k, o, r, s, t, ú, z\}, \{szoba \rightarrow kerti, bab \rightarrow szob\}, \{szoba \rightarrow kerti\})$ normál algoritmust és a bababútor szót. Az $i = 2$, hiszen a második szabály alkalmazható rá. Vigyázni kell viszont, hisz a bab szó rész-szóként kétszer is előfordul. P' tehát nem ba lesz (mert ekkor $P'P_1 = babab$, aminek első három és utolsó három betűje is a bab szót alkotja). Így $P' = \lambda$, azaz az üres szó. Ennek megfelelően a $\boxed{bab}abútor \xRightarrow{w} \boxed{szoba}bútor \xRightarrow{w} .kertibútor$ levezetéshez jutunk. Itt a bababútor $\xRightarrow{w}$ szobabútor egy elemi helyettesítés, szobabútor $\xRightarrow{w} .kertibútor$ pedig egy elemi záróhelyettesítés. (Igaz, hogy a szobabútor szóban is benne van a bab rész-szó, de mindig az első olyan szabályt kell alkalmazni, amit lehet. Emiatt a szobabútor szó után a kertibútor szó következik, nem a szoszobútor.

Azt mondjuk, hogy a $P \in V^*$ szóból kiindulva egy $Q \in V^*$ szó eleme a W algoritmus *levezetési láncának*, ha teljesülnek az alábbi feltételek:

$Q = P$, vagy

- (i) W alkalmazható P -re és
- (ii) léteznek olyan $R_0, \dots, R_k \in V^*$ szavak, hogy $R_0 = P$, $R_k = Q$, $R_i \xRightarrow{w} R_{i+1}$ ($i = 0, \dots, k-1$), ahol az $R_0 \xRightarrow{w} \dots \xRightarrow{w} R_{k-1}$ lánc nem tartalmaz záróhelyettesítést.

Azt mondjuk, hogy a $P \in V^*$ szóra egy $Q \in V^*$ szó a W algoritmus *eredménye* (jelekben: $P \xrightarrow{W}^* Q$), ha Q eleme a W P -ből induló ($P = R_0, \dots, R_k = Q$) levezetési láncának és vagy $R_{k-1} \xrightarrow{W} R_k$, vagy pedig $R_{k-1} \xrightarrow{W} R_k$, és W az $R_k (= Q)$ szóra már nem alkalmazható.

Emellett megállapodás, hogy ha W a P szóra nem alkalmazható, akkor $P \xrightarrow{W}^* P$.

Ha $P \xrightarrow{W}^* Q$, akkor szokás mondani, hogy Q a P szóval megadott adatsorozaton a W normál algoritmussal végrehajtott számítás eredménye. Eszerint, egy $W = (V, H, H_1)$ normál algoritmus esetén egy $P \in V^*$ szóra a következő esetek állhatnak fenn:

(I) W a P -re nem alkalmazható. Ekkor a W -vel P -n végzett számítás eredményének magát a P adatot tekintik.

(II) P -ből kiindulva létezik záróhelyettesítéssel végződő (azaz $P \xrightarrow{W} R_1, R_1 \xrightarrow{W} R_2, \dots, R_{k-2} \xrightarrow{W} R_{k-1}, R_{k-1} \xrightarrow{W} R_k$ alakú), vagy tovább nem folytatható helyettesítési lánc. (Ez utóbbi azt jelenti tehát, hogy a $P \xrightarrow{W} R_1, R_1 \xrightarrow{W} R_2, \dots, R_{k-1} \xrightarrow{W} R_k$ láncban R_k -ra W már nem alkalmazható.) Ekkor a P adaton a W -vel végzett számítás eredménye az $R_k = Q$ szó.

(III) P -ből olyan helyettesítési lánc indul ki, mely soha nem fejeződik be. Ilyenkor azt mondjuk, hogy W a P -re nem áll meg, W a P adatból eredményt nem szolgáltat.

Példa 1.16 Tekintsük a $W = (\{1, +\}, \{1+ \rightarrow +1, ++ \rightarrow +, + \rightarrow \lambda\}, \{+ \rightarrow \lambda\})$ normál algoritmust (ahol λ az üres szót jelöli). Ekkor $1 + 1 + 1 + 1 \xrightarrow{W} 11 + 1 + 1$

$\xrightarrow{W} +1 + 11 + 1 \xrightarrow{W} ++111 + 1 \xrightarrow{W} ++11 + 11 \xrightarrow{W} ++1 + 111 \xrightarrow{W} ++1111 \xrightarrow{W} ++1111$
 $\xrightarrow{W} +1111 \xrightarrow{W} .1111$.

Vegyük észre, hogy általában is, $1^m + 1^n \xrightarrow{W}^* 1^{m+n}$, vagyis a példabeli algoritmus az "egyes számrendszerben" való "összeadás" algoritmus. Ahány 1-et "adunk össze", annyi 1-esből álló sorozatot kapunk a végén záróhelyettesítéssel. Nincs alkalmazható szabálya az algoritmusnak például a 11 szóra. Ekkor eredménynek a korábbi definícióval összhangban magát a 11-t tekintjük.

Példa 1.17 Egészítsük ki az előbbi példát az $1 \rightarrow 11$ szabállyal, méghozzá úgy, hogy ez legyen az első szabályunk. Az így kapott újabb $W = (\{1, +\}, \{1 \rightarrow 11, 1+ \rightarrow +1, ++ \rightarrow +, + \rightarrow \lambda\}, \{+ \rightarrow \lambda\})$ algoritmus már nem fog befejeződni az $1 + 1 + 1 + 1$ szóra (és sok más szóra sem): $1 + 1 + 1 + 1 \xrightarrow{W} 11 + 1 + 1 + 1 \xrightarrow{W} 111 + 1 + 1 + 1 \xrightarrow{W} 1111 + 1 + 1 + 1 \xrightarrow{W} 11111 + 1 + 1 + 1 \dots$. A módosított algoritmus tehát az $1 + 1 + 1 + 1$ szóra nem áll meg, eredményt nem szolgáltat.

Példa 1.18 Vizsgáljuk meg az 1.15. példabeli algoritmus viselkedését a babbútor szóra. babbútor $\xrightarrow{W}$ szobbútor és itt az algoritmus megáll, mert nincs ugyan záróhelyettesítés, de a legutolsó szóra W már nem alkalmazható. A számítás eredménye tehát a szobbútor szó lesz.

Példa 1.19 Tekintsük a $W = (\{1, \times, a, b\}, \{\times 11 \rightarrow a \times 1, \times 1 \rightarrow a, 1a \rightarrow a1b, ba \rightarrow ab, b1 \rightarrow 1b, a1 \rightarrow a, ab \rightarrow b, b \rightarrow 1\}, \emptyset)$. normál algoritmust. (Feladatunkban tehát H_1 üres halmaz.) Igazoljuk, hogy tetszőleges m, n pozitív egészekre $1^m \rightarrow 1^n \xrightarrow{W}^* 1^{mn}$. W tehát a szorzás egy algoritmus (”egyes” számrendszerben).

Példa 1.20 Legyen $W = (\{a, b, c, d, e, \&\}, \{ac \rightarrow ca, a\&a \rightarrow c\&, a\& \rightarrow \&d, d \rightarrow a, c \rightarrow e, e \rightarrow a, \& \rightarrow \lambda\}, \emptyset)$ (azaz H_1 ebben a feladatban is üres halmaz) normál algoritmus. Bizonyítsuk be, hogy tetszőleges i, j pozitív egész számpárra $a^i \& a^j \xrightarrow{*}_W a^k$, ahol k a legnagyobb közös osztója i -nek és j -nek.

Egy $W = (V, Ax, H)$ genetatív rendszert *generatív nyelvtannak* (vagy *generatív grammatikának*) hívunk, ha V fel van bontva közös elemet nem tartalmazó nem üres (és V végessége miatt véges) V_N és V_T részhalmazokra, az Ax egyetlen V_N -beli (általában S -el jelölt) elemből áll, s minden H -beli szabály baloldala legalább egy V_N -beli betűt tartalmaz. Jelekben:

$$V = V_N \cup V_T, V_N \cap V_T = \emptyset, V_N \neq \emptyset, V_T \neq \emptyset, Ax = \{S\}, S \in V_N, H \subseteq V^* V_N V^* \times V^*.$$

Egy ilyen generatív nyelvtant a továbbiakban $G = (V_N, V_T, S, H)$ által jelölünk. Benne V_N a *változók*, vagy más néven, a *nemterminális (szimbólum)ok* halmaza, V_T a *konstansok* vagy *terminálisok* halmaza, S a *mondatszimbólum*, H pedig (mint korábban) a (helyettesítési) szabályok halmaza. Amint szokásos, használni fogjuk a $V = V_N \cup V_T$ jelölést is. V^* elemeit *mondatformáknak*, V_N^* elemeit *nemterminális szavaknak*, V_T^* elemeit pedig *terminális szavaknak*, vagy a nyelvtani analógia miatt *mondatoknak* is hívjuk. Mivel minden generatív nyelvtan egyben formális rendszer is, használhatjuk a közvetlen levezethetőség és a levezethetőség fogalmait. A G *generatív nyelvtan által generált nyelven* a V_T^* -beli szavak következő halmazát értjük: $L(G) = \{p \in V_T^* \mid S \xrightarrow{*}_G p\}$. Úgy is mondhattuk volna, hogy $L(G) = L_{gG} \cap V_T^*$, ahol L_{gG} a G mint generatív rendszer, pontosabban a (V, S, H) generatív rendszer által generált nyelvet jelöli. (Tehát vigyázat: Nem tévesztendő össze a G generatív nyelvtan és a G mint generatív rendszer által generált nyelv, hiszen $L(G) \subseteq L_{gG}$, vagyis a két nyelv általában nem esik egybe!)

A fogalom megfelel annak, hogy a változók a nyelvi fogalmak (ige, főnév, stb.), a konstansok pedig a szótári szavak elemeinek absztrakciói.

Példa 1.21 Legyen $G = (V_N, V_T, S, H)$, ahol $V_N = \{S, \langle ige \rangle, \langle főnév \rangle, \langle melléknév \rangle\}$, $V_T = \{\text{magyar ábécé}\}$, $H = \{S \rightarrow \langle ige \rangle, \langle ige \rangle \rightarrow \langle főnév \rangle \langle ige \rangle, \langle főnév \rangle \rightarrow \langle melléknév \rangle \langle főnév \rangle, \langle melléknév \rangle \rightarrow \text{az} \langle melléknév \rangle, \langle ige \rangle \rightarrow \text{tanul}, \langle főnév \rangle \rightarrow \text{diák}, \langle melléknév \rangle \rightarrow \text{engedelmes}\}$.

Tekintsük a következő levezetést.

$S \Rightarrow \langle ige \rangle \Rightarrow \langle főnév \rangle \langle ige \rangle \Rightarrow \langle melléknév \rangle \langle főnév \rangle \langle ige \rangle \Rightarrow \text{az} \langle melléknév \rangle \langle főnév \rangle \langle ige \rangle \Rightarrow \text{az engedelmes} \langle főnév \rangle \langle ige \rangle \Rightarrow \text{az engedelmes diák} \langle ige \rangle \Rightarrow \text{az engedelmes diák tanul}.$

Ugyanígy levezethető "nyelvtanunkkal" például: az az engedelmes diák tanul, stb.

A generatív nyelvtan persze nemcsak nyelvtani elemzésre alkalmas eszköz. Mint a következő (M. Soittola -tól , illetve M. Penttonen-től származó) példák is mutatják, segítségével elő tudjuk állítani az összes négyzetszámot, illetve a prímszámokat.

Példa 1.22 $G = (\{S, X, Y, Z, X_1, X_2, Y_1, Y_2\}, \{a\}, S, \{S \rightarrow a, S \rightarrow aX X_2 Z, X_2 Z \rightarrow aa, Xa \rightarrow aa, Ya \rightarrow aa, X_2 Z \rightarrow Y_1 Y X Z, X X_1 \rightarrow X_1 Y X, Y X_1 \rightarrow Y_1 Y X, X Y_1 \rightarrow X_1 Y, Y Y_1 \rightarrow Y_1 Y, a X_1 \rightarrow a X X Y X_2, X_2 Y \rightarrow X Y_2, Y_2 Y \rightarrow Y Y_2, Y_2 X \rightarrow Y X_2\})$.

Bizonyítható (lásd kiegészítés a fejezet végén), hogy a generált terminális szavak hossza eleme lesz az $1, 1 + 3, 1 + 3 + 5, \dots, 1 + 3 + \dots + (2n - 1), \dots$ sorozatnak. Másrészt igazolható, hogy $n^2 = 1 + 3 + \dots + (2n - 1)$ ($n = 1, 2, \dots$). Így $L(G) = \{a^{n^2} \mid n = 1, 2, \dots\}$.

Példa 1.23 $G = (\{S, A, B, C, D, E, X_1, X_2, X_3, Y_A, Y_B, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8, Y_9\}, \{a\}, S, \{S \rightarrow a^2, S \rightarrow a^3, S \rightarrow a^5, S \rightarrow a^7, S \rightarrow Y_1 A^6 X_3, Y_1 \rightarrow Y_1 A^2, Y_1 A^3 \rightarrow X_1 A^3 X_2, i A X_2 A \rightarrow D Y_2 Y_i E, i D Y_2 \rightarrow D Y_2 i, X_1 D Y_2 \rightarrow X_1 B Y_3, Y_3 i \rightarrow i Y_3, Y_3 Y_i E \rightarrow Y_i Y_3 E, Y_3 E i \rightarrow i Y_3 E, Y_3 E X_3 \rightarrow Y_4 C X_3, i Y_4 \rightarrow Y_4 i, Y_i Y_4 \rightarrow i X_2, (i \in \{A, B\}), X_1 B \rightarrow X_1 Y_5, Y_5 B \rightarrow B Y_5, Y_5 X_2 A \rightarrow Y_6 X_2 A, B Y_6 \rightarrow Y_6 A, X Y_6 \rightarrow X_1 A, C X_3 \rightarrow Y_7 X_3, C Y_7 \rightarrow Y_7 A, B A^j X_2 Y_7 \rightarrow Y_8 A^{j+1} X_2, (j \in \{0, 1, 2\}), A^4 X_2 Y_7 \rightarrow Y_8 A^4 X_2, A Y_8 \rightarrow Y_8 A, B Y_8 \rightarrow Y_8 A, X_1 Y_8 \rightarrow X_1 A, X_2 X_3 \rightarrow Y_9 a, A Y_9 \rightarrow Y_9 a, X_1 Y_9 \rightarrow a a\})$. *Igazolható (lásd kiegészítés a fejezet végén), hogy $L(G) = \{a^p \mid p \text{ prímszám}\}$.*

A generatív nyelvtan duális fogalma az *analitikus nyelvtan*, amit a generatív nyelvtanhoz hasonlóan, $G = (V_N, V_T, S, H)$ alakban adunk meg, benne V_N a *változó (szimbólum)ok*, vagy még más néven, a *nemterminálisok* halmaza, V_T a *konstansok* vagy *terminálisok* halmaza, S a *mondatszimbólum*, H pedig (mint korábban) a szabályok halmaza. E fogalomnál is használjuk a $V = V_N \cup V_T$ jelölést, aholis V^* elemeit *mondatformáknak*, V_N^* elemeit *nemterminális szavaknak*, V_T^* elemeit pedig *terminális szavaknak*, vagy a nyelvtani analógia miatt *mondatoknak* hívjuk (ugyanúgy mint a generatív nyelvtannál. Szemben a generatív nyelvtannal, itt azt tételezzük fel, hogy minden H -beli szabály jobb oldala legalább egy V_N -beli betűt tartalmaz. A G *analitikus nyelvtan által acceptált*, vagy *elfogadott*, vagy még más néven, *felismert nyelven* a V_T^* -beli szavak következő halmazát értjük: $L(G) = \{p \in V_T^* \mid p \xrightarrow{*}_G S\}$. Itt is mondhattuk volna, hogy $L(G) = L_{aG} \cap V_T^*$, ahol L_{aG} a G mint analitikus rendszer, pontosabban a (V, S, H) analitikus rendszer által felismert nyelvet jelöli. (Tehát vigyázat: Hasonlóan a generatív változatokhoz, nem tévesztendő össze a G analitikus nyelvtan és a G mint analitikus rendszer által felismert nyelv, hiszen $L(G) \subseteq L_{aG}$, vagyis a két nyelv általában nem esik egybe!)

Érvényes a következő tétel, melynek bizonyítását az olvasóra bízunk.

Tétel 1.1 *Tekintsünk egy tetszőleges $G = (V_N, V_T, S, H)$ tetszőleges generatív (analitikus) nyelvtant. Alkossuk meg hozzá az összes olyan $Q \rightarrow P$ szabályok H_1 halmazát, melyekre $P \rightarrow Q \in H$. Ekkor a $G_1 = (V_N, V_T, S, H_1)$ analitikus (generatív) nyelvtanra $L(G_1) = L(G)$.* $\square$

Feladat 1.24 *Az 1.1. tétel felhasználásával, továbbá az 1.22. és 1.23. példák segítségével készítsünk analitikus nyelvtanokat, melyek által felismert nyelvek $L(G) = \{a^{n^2} \mid n = 1, 2, \dots\}$, illetve $L(G) = \{a^p \mid p \text{ prímszám}\}$.*

Az 1.1. tétel alapján minden (generatív vagy analitikus) nyelvtan befoglalható egy vele ekvivalens duális nyelvtannal alkotott párba. (Nevezetesen, minden generatív nyelvtanhoz tartozik egy vele ekvivalens analitikus nyelvtan, s megfordítva, minden analitikus nyelvtanhoz tartozik egy vele ekvivalens generatív nyelvtan.) A továbbiakban a "nyelvtan" elnevezést fenntartjuk a "generatív nyelvtan" elnevezés

rövidítésére. Nyelvtan alatt tehát mindig generatív nyelvtant fogunk érteni (míg analitikus nyelvtan esetén mindig kitesszük az "analitikus" jelzöt).

Kiegészítés az 1.22. példához. Az összes négyzetszámot a következőképp állíthatjuk elő: Csoportosítsuk a szabályokat az alábbi módon.

- (1) $S \rightarrow a, S \rightarrow aX X_2 Z,$
- (2) $X_2 Z \rightarrow aa,$
- (3) $Xa \rightarrow aa, Ya \rightarrow aa,$
- (4) $X_2 Z \rightarrow Y_1 Y X Z,$
- (5) $XX_1 \rightarrow X_1 Y X, YX_1 \rightarrow Y_1 Y X,$
- (6) $XY_1 \rightarrow X_1 Y, YY_1 \rightarrow Y_1 Y,$
- (7) $aX_1 \rightarrow aX X Y X_2,$
- (8) $X_2 Y \rightarrow XY_2, Y_2 Y \rightarrow YY_2, Y_2 X \rightarrow YX_2\}.$

A továbbiakban (1) – (8) a megfelelő csoport valamelyik szabályát jelöli. (1) (mely a mondatszimbólumból kiinduló szabályokat mutatja) az a és $aX X_2 Z$ szavak valamelyikét eredményezi. Tekintsünk a $P_i = aX Q_i X_2 Z, Q_i \in \{X, Y\}^*$ szóból terminális szavakhoz vezető levezetések. Először csak (2) vagy (4) alkalmazható. Ha (2) került alkalmazásra, akkor a folytatás egyedüli módja (3) alkalmazása egész addig, míg egy $1 + 3 + |Q_i|$ hosszú terminális szót nem kapunk. (Mivel a terminális ábécé egy betűből áll, minden szó meg van határozva a hossza által.) Ha (4) kerül alkalmazásra, az $aX Q_i Y_1 Y X Z$ szót kapjuk. A folytatás egyedüli módja az "1" index balra léptetése (5)-el és (6) egész addig, amíg (7) alkalmazhatóvá válik. (7) alkalmazása után egy $aX X Y X_2 Q_i' Y Y X Z$ szóhoz jutunk, ahol Q_i' -t úgy kapjuk Q_i -ből, hogy X minden előfordulásakor Y -t írunk. A "2" index jobbra léptetésének egyedüli lehetősége a (8), mely a $P_{i+1} = aX Q_{i+1} X_2 Z, Q_{i+1} = XY Q_i' Y Y$ szavakhoz vezet. (Megjegyezzük, hogy (8) eleget tesz a "2" index jobbra léptetési követelményének, hisz $X_2 X \rightarrow X X_2$ nem szükséges, mivel nem fordul elő két egymást követő X .) Egy, a kiinduló P_i szavunkkal megegyező formájú szóhoz jutottunk, s kezdődhet egy új ciklus (2) vagy (4) alkalmazásával. Következésképp, $|Q_{i+1}| = |Q_i| + Q_i[X] + 5$, ahol $Q_i[X]$ jelöli az X betű előfordulásainak számát Q_i -ben. Mivel világos, hogy $Q_{i+1}[X] = Q_i[X] + 2$, kapjuk, hogy a generált terminális szavak hosszai elemei az $1, 1 + 3, 1 + 3 + 5, \dots, 1 + 3 + \dots + (2n - 1), \dots$ sorozatnak. $\square$

Kiegészítés az 1.23. példához. Az összes prím számot a következőképp állíthatjuk elő: Csoportosítsuk a szabályokat a következő módon.

- (1) $S \rightarrow a^2, S \rightarrow a^3, S \rightarrow a^5, S \rightarrow a^7,$
- (2) $S \rightarrow Y_1 A^6 X_3, Y_1 \rightarrow Y_1 A^2, Y_1 A^3 \rightarrow X_1 A^3 X_2,$
- (3) $iAX_2 A \rightarrow DY_2 Y_i E, iDY_2 \rightarrow DY_2 i, X_1 DY_2 \rightarrow X_1 BY_3, Y_3 i \rightarrow iY_3, Y_3 Y_i E \rightarrow Y_i Y_3 E, Y_3 E i \rightarrow iY_3 E, Y_3 E X_3 \rightarrow Y_4 C X_3, iY_4 \rightarrow Y_4 i, Y_i Y_4 \rightarrow iX_2, i \in \{A, B\},$
- (4) $X_1 B \rightarrow X_1 Y_5, Y_5 B \rightarrow BY_5, Y_5 X_2 A \rightarrow Y_6 X_2 A, BY_6 \rightarrow Y_6 A, XY_6 \rightarrow X_1 A,$
- (5) $CX_3 \rightarrow Y_7 X_3, CY_7 \rightarrow Y_7 A, BA^j X_2 Y_7 \rightarrow Y_8 A^{j+1} X_2, A^4 X_2 Y_7 \rightarrow Y_8 A^4 X_2, AY_8 \rightarrow Y_8 A, BY_8 \rightarrow Y_8 A, X_1 Y_8 \rightarrow X_1 A, j \in \{0, 1, 2\}$
- (6) $X_2 X_3 \rightarrow Y_9 A, AY_9 \rightarrow Y_9 A, X_1 Y_9 \rightarrow aa.$

Az (1)-beli szabályok generálják minden 10 alatti prím n -re az a^n alakú terminális szavakat. A (2)-beli szabályok generálják minden 9-nél nem kisebb n -re az

$X_1A^3X_2A^{n-6}X_3$ alakú szavakat. A (3) – (5) szabályok tesztelik, hogy n osztható-e 3-al, s ha nem, akkor eljutunk az $X_1A^4X_2A^{n-7}X_3$ szóhoz. Ezután következik a 4-el oszthatóság tesztje. Negatív eredmény esetén X_2 egyet lép jobbra. A pozitív válasz olyan levezetést eredményez, mely nem vezet terminális szóhoz. Ha az oszthatósági teszt eredménytelen, eljutunk az $X_1A^{n-3}X_2X_3$ szóhoz, s (6) válik alkalmazhatóvá, mely elvezet a^n -hez. A részletesebb bizonyítást mellőzzük. $\square$

2. Az automaták elméletének alapjai

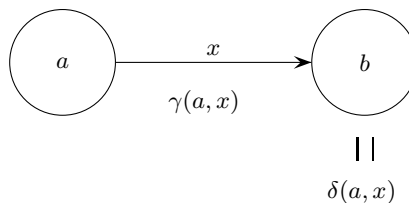
2.1 Az automata fogalma és főbb típusai

Automatán egy olyan absztrakt rendszert fogunk érteni, mely egy diszkrétnek képzelt időskála időpillanataiban érkezett ingerek hatására ezen időpillanatokban válasszal reagál, miközben belső állapotát megadott szabályok szerint változtatja a külső ingerek hatására. Az ingerekre adott válasz függ mind az ingerektől mind pedig a pillanatnyi belső állapottól. Ebben az értelemben tehát nemcsak a gépek, hanem bármiféle élő vagy élettelen objektumok tekinthetők automatának, ha ezen séma szerint vizsgáljuk őket, azaz ilyenszerű működést tulajdonítunk nekik.

Példa 2.1 *A legkülönbözőbb létező vagy nem létező dolgok tekinthetők automatának. Automatának tekinthető a lakásunk ajtaja, ablaka. Bizonyos értelemben automatának tekinthető a kedvenc macskánk, a számítógépünk, vagy a lakáscsengőnk is. Vizsgálhatjuk a főnökünket is mint automatát, hisz minden biztonnyal különféleképp reagál arra, hogy az elvárásainak megfelelően cselekszünk-e vagy sem. (És az is valószínű, hogy ezek a dolgok a főnök állapotát is befolyásolják.) De automatának tekinthető az inkák esőistene, aki imádságra esővel, káromkodásra szárazsággal reagál.*

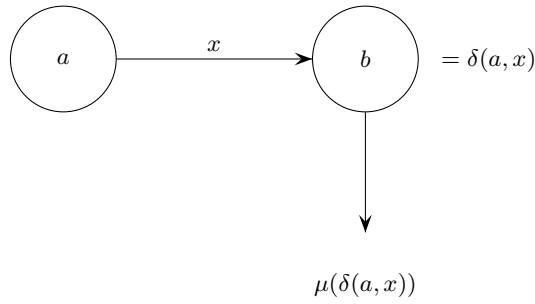
Az absztrakt automaták egyik nevezetes típusa a Mealy (ejtsd: mli) automata. *Mealy automatán* fogunk érteni egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ ötöst, aholis A a (belső) állapotok nem üres halmaza, X a bemenő jelek nem üres halmaza, Y a kimenő jelek nem üres halmaza, $\delta : A \times X \rightarrow A$ az átmeneti függvény és $\gamma : A \times X \rightarrow Y$ a kimeneti függvény.

Úgy képzeljük, hogy a Mealy automata diszkrét időskála mentén működik, s annak minden egyes időpillanatában egy-egy jól meghatározott állapotban van. Ha valamely időpillanatban egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy automata egy $a \in A$ állapotában az $x \in X$ bemenő jelet kapja, akkor ugyanezen időpillanatban a $\gamma(a, x)$ kimenő jellel reagál, majd a következő időpillanatra átmegy a $\delta(a, x)$ állapotba.



A MEALY-AUTOMATA MŰKÖDÉSI SÉMÁJA

Amennyiben az $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatához létezik olyan $\mu : A \rightarrow Y$ függvény, hogy tetszőleges $a \in A$ állapota és $x \in X$ bemenő jele esetén teljesül a $\gamma(a, x) = \mu(\delta(a, x))$ egyenlőség, akkor *Moore-automatáról* beszélünk. A Moore-automatát $\mathcal{A} = (A, X, Y, \delta, \mu)$ alakban szokás megadni, ahol $\mu : A \rightarrow Y$ a Moore-automata *jelfüggvénye*. Tetszőleges $a \in A$ állapot esetén azt mondjuk, hogy $\mu(a)$ az $a \in A$ állapot jele.

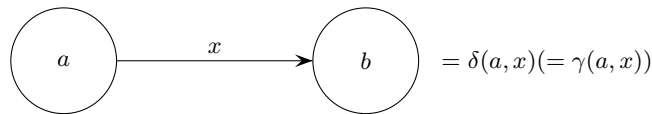


A MOORE-AUTOMATA MŰKÖDÉSI SÉMÁJA

A Moore automata a diszkrét időskála minden egyes időpillanatában egy-egy jól meghatározott $a \in A$ állapotban van (mikor is az állapotjele $\mu(a)$). Ha egy adott időpillanatban ezen $a \in A$ állapotában az $x \in X$ bemenő jelet kapja, akkor a következő időpillanatban $\delta(a, x)$ lesz az állapota, s állapotjele pedig $\mu(\delta(a, x))$ lesz. Ily módon a Moore-automata egy adott időpillanatban kapott bemenő jel hatására a következő időpillanatra átmegy ezen bemenő jel és a belső állapota által egyértelműen meghatározott állapotba, s ezzel egyidejűleg kiadja az új állapot által egyértelműen meghatározott állapotjelet.

Képletesen szólva, amíg a Mealy-automata az olyan embert is modellezheti, aki először cselekszik, azután gondolkodik, addig a Moore-automata az olyan embert modellezi, először gondolkodik azután cselekszik.

A Moore-féle automata a Mealy-féle automata speciális eseteként adódik. A Moore-féle automata további specializálásával jutunk el a kimenő jel nélküli automata fogalmához a következő módon. Amennyiben egy $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automatára $A = Y$ és $\mu : A \rightarrow A$ egy identikus leképezés (azaz minden $a \in A$ -ra $\mu(a) = a$), akkor a *kimenő jel nélküli automata* fogalmához jutunk. Figyelembe véve azt a tényt, hogy a Moore-automatát egy olyan $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatából származtatjuk, melyhez található olyan $\mu : A \rightarrow Y$ függvény, hogy egy tetszőleges $a \in A, x \in X$ pár esetén $\gamma(a, x) = \mu(\delta(a, x))$, továbbá figyelembe véve, hogy a kimenő jel nélküli automata olyan Moore-automata, melynek a jelfüggvénye identikus leképezés, azt is mondhatjuk, hogy a kimenő jel nélküli automata egy olyan $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata, melyre $A = Y$ és $\delta = \gamma$. Ezen okok miatt a kimenő jel nélküli automatát $\mathcal{A} = (A, X, \delta)$ alakban szokás megadni.



A KIMENŐ JEL NÉLKÜLI AUTOMATA MŰKÖDÉSI SÉMÁJA

Egy $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata esetén az X bemenő jelhalmaz minden $x \in X$ eleme egy olyan $x : A \rightarrow A$ egy változós műveletnek (vagyis A

önmagába történő leképezésének) is tekinthető, mely az állapothalmaz tetszőleges $a \in A$ eleméhez az $x(a) = \delta(a, x)$ elemét rendeli. Univerzális algebrai kifejezéssel élve tehát a kimenő jel nélküli automaták unoidok, vagy más szóval unáris algebraik. Ez az egyszerű felismerés lehetővé teszi számunkra a modern algebra módszereinek automataelméleti alkalmazását.

Amint láttuk, a Moore-féle automata speciális Mealy-automataként definiálható. Később látni fogjuk, hogy ez a specializáció látszólagos, ugyanis információ átalakítás szempontjából a két fogalom ekvivalens. (Az információ átalakításán azt értjük, hogy az automata tetszőleges bemenő információ hatására valamilyen "kimenő" információval reagál.) Nevezetesen, absztrakt szempontból a Mealy és a Moore-féle automaták ekvivalensek egymással abban az értelemben, hogy már a speciálisabb Moore-automatákkal előállíthatók azok az információ átalakítások, amelyek Mealy automatákkal megvalósíthatók. Tehát a Moore-automata ebből a szempontból csak látszólag speciálisabb a Mealy-automatánál.

Később látni fogjuk azt is, hogy az elmélet kiépítésénél sok esetben elegendő kimenő jel nélküli automatákra szorítkozni.

Az említett három automata típus mindegyike esetén szokás *véges automatáról* beszélni, ha az állapothalmaz, a bemenő jelhalmaz, s a kimenő jelhalmaz végesek. Szokás véges állapotú automatáról vagy A -véges automatáról beszélni, ha az állapothalmaz véges. Hasonló értelemben beszélünk véges bemenetű vagy X -véges, illetve véges kimenetű vagy Y -véges automatáról, valamint (A, X) -, (A, Y) -, illetve (X, Y) -véges automatáról.

Amennyiben az említett automata-típusok valamelyikénél kijelölünk egy a_0 iniciális, vagy más néven kezdő állapotot, s feltételezzük, hogy az automata működésének van egy kezdő időpontja, mikor is az automata ebben az állapotban van, akkor iniciális automatáról beszélünk. Az iniciális Mealy-féle automatát $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ alakban, az iniciális Moore-féle automatát $\mathcal{A} = (A, a_0, X, Y, \delta, \mu)$ alakban, illetve az iniciális kimenő jel nélküli automatát $\mathcal{A} = (A, a_0, X, \delta)$ alakban szokás megadni, ahol mindhárom esetben a_0 az iniciális állapotot jelöli.

Az említett automatáknak szokás beszélni az alábbi általánosításairól is.

Ha az átmeneti, illetve kimeneti függvény lehet parciális is, azaz nem teljesen definiált, akkor *parciális automatáról* van szó. Teljesen definiált függvényértékek esetén időnként szokás *teljesen definiált automatáról* is beszélni.

Példa 2.2 *Az ajtó szigorú értelemben egy parciális kimenő jel nélküli automatának tekinthető. Bemenő jelei a csukás és a nyitás, s ennek megfelelően két állapota van, nevezetesen csukott és nyitott állapot. Csukott állapotból nyitással lehet nyitott állapotba hozni, nyitott állapotból pedig csukással lehet csukott állapotba hozni. De az ajtó valóban parciális automata, hisz nyitott ajtót kinyitni, vagy csukott ajtót becsukni nem lehetséges.*

Amennyiben az átmeneti és a kimeneti függvények (illetve Moore-automata esetén az átmeneti és a jelfüggvények) nem egyértelműen definiáltak, *nemdeterminisztikus automatáról* van szó. Ekkor valójában ezek nem is tekinthetők függvénynek a hagyományos értelemben. Ahhoz, hogy matematikai értelemben mégis függvényekkel dolgozzunk úgy tekintjük őket, mintha nem az állapot-, illetve a kimenő jelhalmazba

képeznének, hanem ezen halmazok összes részhalmazainak halmazaiiba. Egy nemdeterminisztikus $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata esetén tehát az átmeneti függvény $\delta : A \times X \rightarrow 2^A$, a kimeneti függvény pedig $\gamma : A \times X \rightarrow 2^Y$ alakú, ahol 2^A , illetve 2^Y az állapothalmaz, illetve a kimenő jelhalmaz részhalmazainak halmazát jelöli. Értelemszerűen, egy nemdeterminisztikus $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automata esetén az átmeneti függvény $\delta : A \times X \rightarrow 2^A$, a jelfüggvény $A \rightarrow 2^Y$ alakú, míg egy nemdeterminisztikus kimenő jel nélküli $\mathcal{A} = (A, X, \delta)$ automatánál az átmeneti függvény formája $\delta : A \times X \rightarrow 2^A$.

Példa 2.3 *Nemdeterminisztikus kimenő jel nélküli automatának tekinthető a dobókocka, melynek egyetlen bemenő jele a feldobás. Ezen bemenő jel, azaz a feldobás hatására a dobókocka a hat lehetséges állapotából átmehet a hat lehetséges állapot bármelyikébe annak megfelelően, hogy a feldobás után éppen melyik lapjára esik.*

A nemdeterminisztikus automata ellentétéként beszélünk determinisztikus automatáról is. Determinisztikus automata esetén tehát a szóban forgó függvényértékek mindig pontosan egy meghatározott értéket vesznek fel. (A nemdeterminisztikus terminológiával pedig a függvények értékkészlete csak egyelemű részhalmazokat tartalmaz.)

Példa 2.4 *Ha például egy nemdeterminisztikus $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata esetén a $\delta(a, x)$ függvényérték az A -nak egy hat elemű részhalmaza, $\gamma(a, x)$ pedig az Y egy két elemű részhalmaza, akkor az $\mathcal{A}$ Mealy-automata az a állapotából az x bemenő jel hatására ezen hat elemű részhalmaz bármelyik elemébe átmehet, s kimenő jelként pedig az említett két elemű halmaz bármelyik elemét kiadhatja. S tekintettel arra, hogy egy halmaznak az üres halmaz is részhalmaza, az is előfordulhat, hogy valamely $a \in A$ állapotra és $x \in X$ bemenő jelre $\delta(a, x)$, vagy éppen $\gamma(a, x)$ értéke az üres halmaz. Ha $\delta(a, x)$ az üres halmaz, ez annak felel meg, hogy erre az állapota és bemenő jelre nincs értelmezve egyetlen állapot sem amibe átmenet történhet, ha pedig $\gamma(a, x)$ az üres halmaz, akkor ez azt jelenti, hogy erre az állapota és bemenő jelre nincs értelmezve egyetlen kimenő jel sem.*

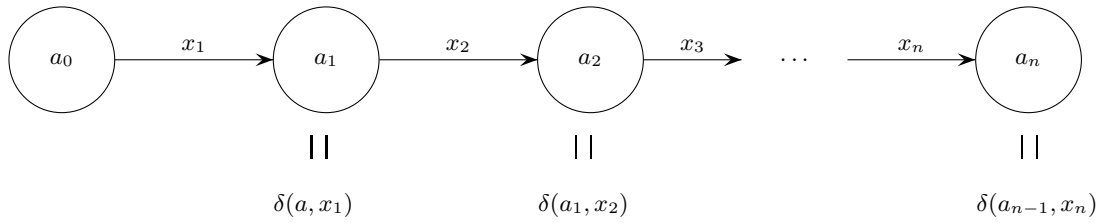
Ezek szerint a parciális automata olyan nemdeterminisztikus automatának tekinthető, ahol a megfelelő függvényértékek vagy egy elemű halmazokat, vagy pedig az üres halmazt szolgáltatják, a teljesen definiált determinisztikus automata pedig egy olyan nemdeterminisztikus automata, ahol ezek a függvényértékek mindig egy elemű halmazok.

Fontos általánosítás a *valószínűségi vagy sztochasztikus automata*, mikoris egy $P((b, y) \mid (a, x))$ feltételes valószínűség adja meg, hogy mi a valószínűsége annak, hogy az automata a b állapotba megy át és az y kimenő jelet adja ki azon feltétel mellett, hogy miközben az a állapotban volt, az x bemenő jelet kapta. Tehát egy valószínűségi automata $\mathcal{A} = (A, X, Y, P)$ alakban adható meg, ahol A az állapotok nem üres halmaza, X a bemenő jelek nem üres halmaza, Y a kimenő jelek nem üres halmaza, P pedig az említett feltételes valószínűség.

Az automaták egy fontos osztályát képezik a *Rabin-Scott féle automaták* (ejtsd rabinszkott), melyeket *felismerő* vagy *elfogadó automatáknak* is hívják. A Rabin-Scott féle automata egy $\mathcal{A} = (A, a_0, X, \delta, A_F)$ ötös, ahol A a nem üres állapothalmaz, $a_0 \in A$

a kezdő állapot, X a nem üres bemenő jelhalmaz, $\delta : A \times X \rightarrow A$ az átmeneti függvény, $A_F \subseteq A$ pedig a végállapotok nem üres halmaza. ($a_0 \in A_F$ megengedett, azaz előfordulhat, hogy a kezdő állapot egyúttal végállapot is.)

A bemenő jelekből felépülő véges hosszúságú láncokat *bemenő szavaknak* hívjuk. Bemenő szónak tekintjük az λ üres szót is, mely nem tartalmaz egyetlen betűt sem. Egy Rabin-Scott automata az λ üres szót definíció szerint akkor *ismeri fel*, ha $a_0 \in A_F$. Egy nem üres, $x_1, \dots, x_n$ (nem feltétlenül különböző) bemenő jelekből álló $x_1 \dots x_n$ (nem üres) bemenő szó esetén akkor mondjuk, hogy a tekintett $\mathcal{A} = (A, a_0, X, \delta, A_F)$ Rabin-Scott féle automata *felismeri*, ha alkalmas $a_1, \dots, a_n$ állapotaira $a_1 = \delta(a_0, x_1), \dots, a_n = \delta(a_{n-1}, x_n)$ teljesülése mellett $a_n \in A_F$.



A RABIN-SCOTT AUTOMATA MŰKÖDÉSI VÁZLATA

Az $\mathcal{A}$ Rabin-Scott automata által felismert $L_{\mathcal{A}}$ nyelvnek hívjuk mindazon bemenő szavak halmazát, melyeket az automata felismer.

Értelmezhetjük a nemdeterminisztikus felismerő automatát is. Ekkor az automata által elfogadott nyelv alatt azon w szavak halmazát értjük, amelyekre az automatának van olyan lehetséges állapot-lánca, amely a kezdőállapotból indul és végállapottal végződik, valamint az átmenetek bemenő jeleit összeolvasva éppen a w szót kapjuk.

2.2 Az automaták megadása

Egy automatát akkor tekintünk adottnak, ha a hozzá tartozó halmazok és függvények adottak. Egy automatát tehát úgy lehet megadni, hogy megadjuk a hozzá tartozó halmazokat és függvényeket. Ezen halmazok és függvények minden olyan típusú megadása lehetséges, ami a halmazok és függvények megadásánál szokásos.

2.2.1 Véges automaták megadása Cayley táblázattal

Véges halmazok és függvények megadásánál szokásos a művelettáblával történő megadás. Automaták művelettáblás megadását automaták Cayley táblázatának (ejtsd: kéjli) is hívjuk Cayley francia matematikus emlékére és tiszteletére, aki véges csoportok művelettábláinak leírására vezette be ezt a táblázatos módszert.

2.2.1.1 Véges Mealy-automata Cayley-táblája

A táblázat bal felső sarkába írjuk az automata nevét, első sorában felsoroljuk az állapotait, első oszlopában pedig a bemenő jeleit. A táblázat $(i + 1)$ -edik sorában és $(j + 1)$ -edik oszlopában szerepel egy két dimenziós vektor, melynek első tagja azt

mondja meg hogy az automata a j -edik állapotból az i -edik bemenő jel hatására melyik állapotába megy át, a másik tagja pedig azt mutatja, hogy a j -edik állapot az i -edik bemenő jel hatására milyen kimenő jelet ad ki.

$\mathcal{A}$	...	a	...
•		•	
•		•	
x	...	$(\delta(a, x), \gamma(a, x))$	...
•		•	
•		•	

VÉGES MEALY-AUTOMATA MŰVELET TÁBLÁZATA

2.2.1.2 Véges Moore-automata Cayley-táblája

A táblázat bal felső sarkába írjuk az automata nevét, első sorában felsoroljuk az állapotait, első oszlopában pedig a bemenő jeleit. Minden állapot fölé beírjuk az állapot jelét. Így az első sor két rész-sorra oszlik. A táblázat $(i + 1)$ -edik sorában és $(j + 1)$ -edik oszlopában szerepel az az állapot, amibe az automata a j -edik állapotból az i -edik bemenő jel hatására átmegy.

	...	$\mu(a)$	...
$\mathcal{A}$	...	a	...
•		•	
•		•	
x	...	$\delta(a, x)$	...
•		•	
•		•	

VÉGES MOORE-AUTOMATA MŰVELET TÁBLÁZATA

2.2.1.3 Véges kimenő jel nélküli automata Cayley-táblája

A táblázat bal felső sarkába írjuk az automata nevét, első sorában felsoroljuk az állapotait, első oszlopában pedig a bemenő jeleit. (Ezesetben az állapotjel maga az állapot, így nem kell az első sorban levő állapotok fölé írni az állapotjelet mint az előző esetben.) Itt is a táblázat $(i + 1)$ -edik sorában és $(j + 1)$ -edik oszlopában szerepel az az állapot, amibe az automata a j -edik állapotból az i -edik bemenő jel hatására átmegy.

$\mathcal{A}$	...	a	...
•		•	
•		•	
•		•	
x	...	$\delta(a, x)$	...
•		•	
•		•	
•		•	

VÉGES KIMENŐ JEL NÉLKÜLI AUTOMATA MŰVELET TÁBLÁZATA

Példa 2.5 Vegyünk egy villamos csengőt. x_1 és x_2 jelöljék azon helyzeteket, mikor is nyomjuk vagy nem nyomjuk a csengőt. A csengő kezdetben az a_0 kezdő állapotban van, ami annak felel meg, hogy nem cseng. Az x_1 jel hatására, vagyis a csengő megnyomására átmegy a csengő az a_0 állapotból az a_1 állapotba. Megszakítva a csengő nyomását, vagyis az x_2 jel hatására a csengő átmegy nem csengő, azaz az a_0 állapotba. Leírásunkat táblázatba foglalva jutunk el a csengő következő absztrakt modelljéhez:

$\mathcal{A}$	a_0	a_1
x_1	a_1	a_1
x_2	a_0	a_0

A táblázat mutatja, hogy mely jel hatására mely állapotból mely állapotba megy át az automata. Például a 2. sor 2. oszlopában levő a_0 azt jelenti, hogy x_2 hatására az a_1 állapotból az a_0 állapotba megy át az automata. Táblázatos megadásnál iniciális automata esetén rendszerint az első oszlop jelzi a kezdő állapot oszlopát (azaz annak megadását, hogy különféle bemenő jelek hatására a kezdő állapotból mely állapotokba megy át az automata).

2.2.2 Véges automaták megadása gráfokkal

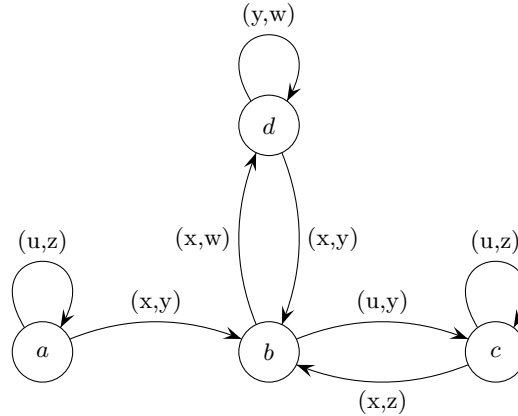
Véges automaták megadásának egy másik szokásos módja a címkézett irányított gráffal történő megadás.

Itt jegyezzük meg, hogy nem-determinisztikus automaták esetén a táblázat cellái az állapotok-, illetve a kimenőjelek halmazának részhalmazait tartalmazhatják.

2.2.2.1 Véges Mealy-automata megadása gráffal

A gráf minden egyes csúcsa meg van címkézve egy állapottal, a csúcsokból kivezető minden irányított él (mely hurokél is lehet) pedig meg van címkézve egy két dimenziós vektorral. Ezen vektor első komponense egy bemenő jel, a második pedig egy kimenő jel. Determinisztikus esetben minden csúcsból pontosan egy él vezet ki, ahány bemenő

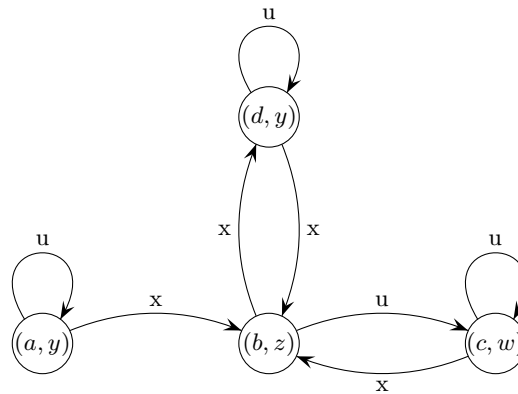
jel van és az élek, valamint azok címkéi adják meg, hogy egy adott állapotból egy adott bemenő jel hatására az automata milyen kimenő jellel reagál és melyik állapotba megy át. Nevezetesen, a bemenő jeleket az élek címkéinek első komponense, a kimenő jeleket az élek címkéinek második komponense, az állapot átmeneteket pedig az élek kezdő- és végcsúcsainak címkéi adják meg. Nevezetesen egy él kezdő csúcsában lévő állapot az él címke első (bemenő jel) komponense hatására épp abba az állapotba megy át, mellyel az él vég csúcsa van megcímkézve.



EGY MEALY-AUTOMATA GRÁFJA

2.2.2.2 Véges Moore-automata megadása gráffal

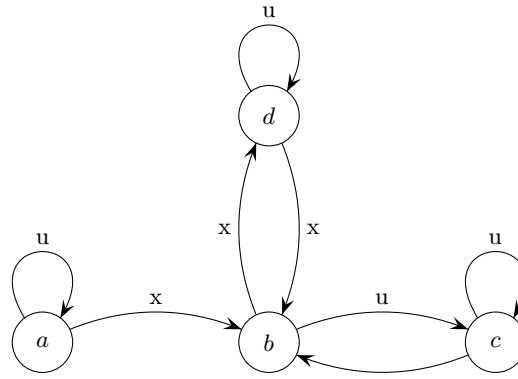
A gráf minden egyes csúcsa meg van címkézve egy két dimenziós vektorral, melynek első komponense egy állapot, a második komponense pedig ezen állapot jele. A csúcsokból kivezető minden irányított él (mely hurokél is lehet) meg van címkézve egy bemenő jellel. (Determinisztikus esetben itt is) minden csúcsból pont annyi él vezet ki, ahány bemenő jel van és az élek, valamint azok címkéi adják meg, hogy egy adott állapotból egy adott bemenő jel hatására az automata melyik állapotba megy át. Nevezetesen, a bemenő jeleket az élek címkéi, az állapot átmeneteket és az állapot jeleket pedig az élek kezdő- és végcsúcsainak címkéi adják meg. Nevezetesen egy él kezdő csúcsában lévő címke első (állapot) komponense az él címke (bemenő jel) hatására épp abba az állapotba megy át, ami az él vég csúcsában levő címke első (állapot) komponense. Az átmenet utáni állapotjel pedig az él vég csúcsában levő címke második (kimenő jel) komponense.



EGY MOORE-AUTOMATA GRÁFJA

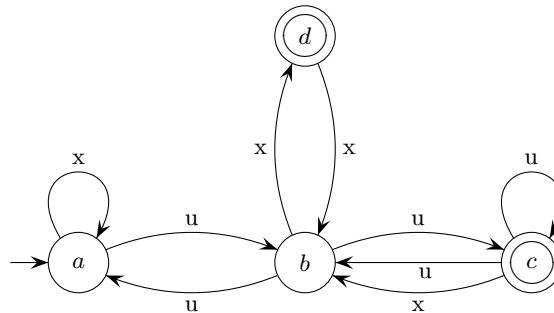
2.2.2.3 Véges kimenő jel nélküli automata megadása gráffal

Csaknem olyan szerkezetű gráffal történik a megadás mint a Moore-automata esetén. Az egyedüli lényeges különbség, hogy a csúcsok itt az állapotokkal vannak megcímkézve (és nem két dimenziós vektorokkal mint a Moore-automata esetén). Tehát a gráf minden egyes csúcsa meg van címkézve egy állapottal, a csúcsokból kivezető minden irányított él (mely hurokél is lehet) pedig meg van címkézve egy bemenő jellel. Most is igaz a determinisztikus esetre, hogy minden csúcsból pont annyi él vezet ki, ahány bemenő jel van és az élek, valamint azok címkei adják meg, hogy egy adott állapotból egy adott bemenő jel hatására az automata melyik állapotba megy át. Nevezetesen, a bemenő jeleket az élek címkei, az állapot átmeneteket pedig az élek kezdő- és végcsúcsainak címkei adják meg. Nevezetesen egy él kezdő csúcsában lévő állapot címke az él címke (bemenő jel) hatására épp abba az állapotba megy át, ami az él vég csúcsában levő állapot címke értéke.



EGY KIMENŐ JEL NÉLKÜLI AUTOMATA GRÁFJA

Itt jegyezzük meg, hogy elfogadó automata esetén szokás a végállapotokat pl. dupla körrel rajzolni, míg a kezdőállapot jelölésére szokás az adott állapotba egy plusz bemenő nyílat rajzolni.



EGY KEZDŐ ÉS VÉGÁLLAPOTOKKAL RENDELKEZŐ NEM-DETERMINISZTIKUS AUTOMATA GRÁFJA

2.3 Az automata mint algebrai struktúra: részautomata, homomorfizmus, izomorfizmus, kompatibilis osztályozás

Az absztrakt automátákat vizsgálhatjuk algebrai struktúráként is. Mint ahogy beszélni szoktunk algebrai struktúrák rész-struktúráiról, homomorfizmusairól, izomorfizmu-

sairól, ugyanígy szokás beszélni ezekről automaták esetén is. Először Mealy-féle automatákra fogjuk megadni a megfelelő részautomata-, homomorfizmus- és izomorfizmus-fogalmakat.

Egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata *részautomatája* alatt értjük azt az $\mathcal{A}' = (A', X', Y', \delta', \gamma')$ Mealy-automatát, melyre $A' \subseteq A, X' \subseteq X, Y' \subseteq Y$ és $\delta' = \delta|_{A' \times X'}$, illetve $\gamma' = \gamma|_{A' \times X'}$. (Szavakban, δ' a δ restrikciója, azaz megszorítása $A' \times X'$ -re, γ' pedig a γ restrikciója, azaz megszorítása $A' \times X'$ -re.) Másként mondva, δ' és γ' úgy van definiálva, hogy alakjuk $\delta' : A' \times X' \rightarrow A'$, illetve $\gamma' : A' \times X' \rightarrow Y'$, s teljesül minden $a' \in A', x' \in X'$ pár esetén a $\delta'(a, x) = \delta(a, x)$, illetve a $\gamma'(a, x) = \gamma(a, x)$ egyenlőség. Amint látjuk, a δ és a γ nem minden $A' \times X'$ -re vett restrikciója alkalmas a részautomata átmeneti, illetve kimeneti függvényének. Kell még az is, hogy a tekintett δ' restrikció az A' -be, illetve hogy a tekintett γ' restrikció az Y' -be képezzen. Azt a tényt, hogy az $\mathcal{A}'$ automata részautomatája az $\mathcal{A}$ automatának, időnként $\mathcal{A}' \subseteq \mathcal{A}$ -val jelöljük.

Amennyiben az $A' \subseteq A, X' \subseteq X, Y' \subseteq Y$ tartalmazások valamelyike valódi tartalmazás, azaz $A' \subset A, X' \subset X, Y' \subset Y$ legalább egyike fennáll, úgy azt is mondjuk, hogy $\mathcal{A}'$ valódi részautomatája $\mathcal{A}$ -nak, s ezt $\mathcal{A}' \subset \mathcal{A}$ -val is jelöljük.

Ha $A' \subseteq A$ és $X = X', Y = Y'$, akkor az $\mathcal{A}'$ -t az $\mathcal{A}$ (valódi vagy nem valódi) állapot-, vagy A -részautomatájának hívjuk (jelekben: $\mathcal{A}' \subseteq_A \mathcal{A}$). Ha $X' \subseteq X$ és $A = A', Y' = Y$, akkor $\mathcal{A}'$ bemenő jel részautomatája vagy X -részautomatája (jelekben: $\mathcal{A}' \subseteq_X \mathcal{A}$), ha pedig $Y' \subseteq Y$, továbbá $A' = A, X' = X$, akkor az $\mathcal{A}'$ kimenő jel részautomatája, vagy Y -részautomatája $\mathcal{A}$ -nak (jelekben: $\mathcal{A}' \subseteq_Y \mathcal{A}$). Hasonló értelemben beszélünk (A, X) -, (A, Y) -, (X, Y) -részautomatákról.

További speciális automata-típus az iniciális részautomata, mely ugyancsak definiálható az említett részautomata-típusok bármelyikére. Akkor mondjuk, hogy egy $\mathcal{A}$ iniciális automatának egy $\mathcal{A}'$ iniciális automata iniciális részautomatája, ha mindamellett, hogy A' az $\mathcal{A}$ -nak részautomatája, az is teljesül, hogy az $\mathcal{A}'$ kezdő állapota épp az $\mathcal{A}$ kezdő állapota lesz. Hasonló értelemben mint a korábbiakban, beszélünk iniciális A -részautomatáról, iniciális X -részautomatáról, iniciális Y -részautomatáról, illetve iniciális (A, X) -, (A, Y) -, (X, Y) -részautomatáról. Minden esetben tehát a megfelelő részautomata-tulajdonság mellett még azt is elvárjuk, hogy a megfelelő részautomata-típus kezdő állapota épp az őt tartalmazó automata kezdő állapota legyen.

Azt mondjuk, hogy az $\mathcal{A}' = (A', X', Y', \delta', \gamma')$ Mealy-automata homomorf képe az $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatának (jelekben $\mathcal{A} \sim \mathcal{A}'$), ha megadható olyan szürjektív (azaz "ra" típusú, vagy más néven ráképező) $\psi_1 : A \rightarrow A', \psi_2 : X \rightarrow X', \psi_3 : Y \rightarrow Y'$ leképezésekből álló $\psi = (\psi_1, \psi_2, \psi_3)$ leképezés-hármas, hogy teljesülnek rá az úgynevezett művelettartó tulajdonságok. Más szóval, képletben kifejeve, tetszőleges $a \in A, x \in X$ pár esetén

$$\psi_1(\delta(a, x)) = \delta'(\psi_1(a), \psi_2(x)),$$

illetve

$$\psi_3(\gamma(a, x)) = \gamma'(\psi_1(a), \psi_2(x)).$$

Világos, hogy a homomorfia mint reláció reflexív ($\mathcal{A} \sim \mathcal{A}$) és tranzitív ($\mathcal{A} \sim \mathcal{A}', \mathcal{A}' \sim \mathcal{A}'' \Rightarrow \mathcal{A} \sim \mathcal{A}''$). Mint ahogy beszélünk speciális részautomatákról, ugyanúgy beszélünk speciális homomorfizmusokról. Azt mondjuk, hogy az $\mathcal{A}' = (A', X', Y', \delta', \gamma')$ Mealy-automata állapot-homomorf képe vagy A -homomorf képe az $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatának (jelekben $\mathcal{A} \sim_{\mathcal{A}} \mathcal{A}'$), ha az előbb definiált ψ_1, ψ_2, ψ_3 leképezés-hármasban a ψ_2 és a ψ_3 választható identikus leképezésnek. Ezen feltétel mellett tehát az előbbi egyenlőségeink tetszőleges $a \in A, x \in X$ pár esetén a következő alakúak lesznek:

$$\psi_1(\delta(a, x)) = \delta'(\psi_1(a), x),$$

illetve

$$\gamma(a, x) = \gamma'(\psi_1(a), x).$$

Ebben az értelemben beszélünk tehát egy $\psi_1 : A \rightarrow A'$ A -homomorfizmusról. Ilyenkor nem egy leképezés-hármas, hanem egyetlen ψ_1 leképezés képviseli az állapot-homomorfizmust (mert eltekintünk az identikus $\psi_2 : X \rightarrow X'$ és $\psi_3 : Y \rightarrow Y'$ leképezések szerepeltetésétől). Hasonló értelemben beszélünk egy $\psi_2 : X \rightarrow X'$ bemenőjel-, vagy X -homomorfizmusról, egy $\psi_3 : Y \rightarrow Y'$ kimenőjel-, vagy Y -homomorfizmusról, illetve egy $\psi' = \{\psi_1, \psi_2\}$ (A, X) -homomorfizmusról (ahol $\psi_1 : A \rightarrow A', \psi_2 : X \rightarrow X'$), egy $\psi'' = \{\psi_1, \psi_3\}$ (A, Y) -homomorfizmusról (ahol $\psi_1 : A \rightarrow A', \psi_3 : Y \rightarrow Y'$), valamint egy $\psi''' = \{\psi_2, \psi_3\}$ (X, Y) -homomorfizmusról (ahol $\psi_2 : X \rightarrow X', \psi_3 : Y \rightarrow Y'$).

További speciális homomorfizmus típus az iniciális homomorfizmus, mely ugyancsak definiálható az említett speciális homomorfizmus-típusok (iniciális A -homomorfizmus, iniciális X -homomorfizmus, stb.) bármelyikére is. Akkor mondjuk, hogy egy $\mathcal{A}$ iniciális automatának egy $\mathcal{A}'$ iniciális automata iniciális homomorf képe (iniciális A -homomorf képe, iniciális X -homomorf képe, stb.), ha mindamellett, hogy A' az $\mathcal{A}$ -nak homomorf képe, az is teljesül, hogy az $\mathcal{A}'$ kezdő állapota épp az $\mathcal{A}$ kezdő állapotának homomorf képe (A -homomorf képe, X -homomorf képe, stb.) lesz. Hasonló értelemben beszélünk tehát iniciális A -homomorfizmusról, iniciális X -homomorfizmusról, iniciális Y -homomorfizmusról, illetve iniciális (A, X) -, (A, Y) -, (X, Y) -homomorfizmusról. (Minden esetben a megfelelő homomorfia-tulajdonság mellett még azt is elvárjuk, hogy a homomorf kép kezdő állapota épp a ráképező automata kezdő állapota legyen.)

Amennyiben a homomorfizmus olyan, hogy az összes szereplő ráképezések bijekciók (azaz kölcsönösen egyértelmű ráképezések), akkor izomorfizmusról beszélünk. Így minden egyes speciális homomorfizmus-típusnak van egy megfelelő izomorfizmus-típusa (A -izomorfizmus, iniciális A -izomorfizmus, stb.). Nyilvánvaló, hogy ha egy $\mathcal{A}$ Mealy-automatának egy $\mathcal{A}'$ Mealy-automata izomorf képe (képletben $\mathcal{A} \simeq \mathcal{A}'$), s a $\varphi = (\varphi_1, \varphi_2, \varphi_3)$ leképezés-hármas az $\mathcal{A}$ izomorfizmusa $\mathcal{A}'$ -re, akkor a $\varphi^{-1} = (\varphi_1^{-1}, \varphi_2^{-1}, \varphi_3^{-1})$ leképezés-hármas az $\mathcal{A}'$ izomorfizmusa lesz $\mathcal{A}$ -ra. Az izomorfizmus tehát mint reláció, szimmetrikus. Mindamellett mint a homomorfizmus általában (és az izomorfizmus a homomorfizmusnak speciális fajtája), az izomorfizmus mint reláció reflexív és tranzitív. Az izomorfizmus mint reláció tehát ekvivalencia reláció, hisz reflexív, szimmetrikus és tranzitív. (Ugyanez természetesen vonatkozik az összes speciális izomorfizmus-típusokra is.)

Az absztrakt algebrai vizsgálatoknál az egymással izomorf struktúrákat azonosnak szokták tekinteni. Ez az úgynevezett *izomorfia elv*. Ezt az elvet automataelméleti vizsgálatoknál sok esetben csak részlegesen szokásos elfogadni a vizsgálatok sajátosságai miatt.

Most részletesebben foglalkozunk még az A -homomorfizmusokkal és a velük kapcsolatban álló kompatibilis osztályozásokkal. Legyen $\mathcal{A} = (A, X, Y, \delta, \gamma)$ tetszőleges Mealy-automata és legyen ϱ tetszőleges ekvivalencia reláció az A állapothalmazon. Ismeretes, hogy minden ekvivalencia reláció egyértelműen indukál egy osztályozást azon a halmazon, amin a reláció értelmezve van. Nevezetesen, pontosan az egymással relációban lévő elemek fognak egy osztályba sorolódni. Így a ϱ reláció is indukálja az A halmaz egy C_ϱ osztályozását: az a és b állapotokat akkor és csak akkor soroljuk egy osztályba, ha egymással relációban vannak, azaz $a\varrho b$ fennáll. Az a elem által reprezentált (vagyis az a elemet tartalmazó) osztályt $C_\varrho[a]$ -val jelöljük. Jelölje $\bar{A}$ az osztályok halmazát, azaz legyen $\bar{A} = \{C_\varrho[a] \mid a \in A\}$. Egy ilyen C_ϱ osztályozást *kompatibilis osztályozásnak* nevezünk, ha a hozzá tartozó ϱ reláció *kongruencia reláció*, vagyis ha ϱ -ra teljesül az a követelmény, hogy minden $a, b \in A$ párra $a\varrho b$ -nek tetszőleges $x \in X$ esetén következménye lesz $\delta(a, x)\varrho\delta(b, x)$ és $\gamma(a, x) = \gamma(b, x)$.

Tegyük fel most, hogy az A halmaz C_ϱ osztályozása pont egy kompatibilis osztályozás. Ekkor tehát minden $a, b \in A$ párra $a\varrho b$ -nek tetszőleges $x \in X$ esetén következménye lesz $\delta(a, x)\varrho\delta(b, x)$ és $\gamma(a, x) = \gamma(b, x)$. Másként fogalmazva, feltételezzük, hogy ha valamely $a, b \in A$ pár a C_ϱ osztályozás szerint egy és ugyanazon C_ϱ osztályba esik, akkor tetszőleges $x \in X$ bemenő jel esetén $\delta(a, x)$ és $\delta(b, x)$ is egy osztályba fognak esni, továbbá $\gamma(a, x) = \gamma(b, x)$ is fennáll. Ily módon definiálható a következő automata:

$$\mathcal{A}/C_\varrho = (\bar{A}, X, Y, \bar{\delta}, \bar{\gamma}),$$

ahol minden $a \in A, x \in X$ párra

$$\bar{\delta}(C_\varrho([a], x) = C_\varrho[\delta(a, x)], \bar{\gamma}(C_\varrho([a], x) = \gamma(a, x).$$

Belátható, hogy ez az automata jól definiált és érvényes a következő:

$$\mathcal{A} \sim_{\Lambda} \mathcal{A}/C_\varrho,$$

aholis a megfelelő A -homomorfizmusokhoz úgy jutunk, hogy minden állapot állapot-homomorf képeként az őt tartalmazó osztály adódik. Más szóval, egy automata faktorautomatái az automatának mindig A -homomorf képei. A következő tétel azt mondja ki, hogy megfordítva, az A -homomorf képek mindig megszerkeszthetők faktorautomataként. Ez a tétel az algebraiban jól ismert Általános Homomorfia Tétel Mealy-automatákra vonatkozó speciális esete.

Tétel 2.1 (Általános Homomorfia Tétel Mealy-Automatákra Vonatkozó Speciális Esete) *Tegyük fel, hogy az $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata az $\mathcal{A}' = (A', X', Y', \delta', \gamma')$ Mealy-automatára valamely állapothomomorfizmussal leképezhető, s tekintsük az A állapothalmaznak azt a C osztályozását, amelynél bármely két $a, b \in A$ állapot akkor és csak akkor van egy osztályban, ha a tekintett állapothomomorfizmus szerinti képük ugyanaz. Ekkor az A állapothalmaz ezen C osztályozása kompatibilis, s a hozzá tartozó $\mathcal{A}/C$ faktorautomata A -izomorf lesz az $\mathcal{A}'$ automatával. Képletben,*

$$\mathcal{A}' \sim_{\mathcal{A}} \mathcal{A}(\varphi) \Rightarrow \mathcal{A}/C \sim_{\mathcal{A}} \mathcal{A}'(C[a] \rightarrow \varphi(a)).$$

□

Most Moore-féle automatákra fogjuk megadni a megfelelő részautomata-, homomorfizmus-, és izomorfizmus-fogalmakat. Ekkor egy $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automata valamely részautomatája alatt értünk egy olyan $\mathcal{A}' = (A', X', Y', \delta', \mu')$ Moore-automatát, melyre $A' \subseteq A, X' \subseteq X, Y' \subseteq Y$ mellett $\delta' = \delta|_{A' \times X'}$ és $\mu' = \mu|_{Y'}$ teljesül. Más szóval tetszőleges $a \in A, x \in X$ pár esetén $\delta'(a, x) = \delta(a, x)$ és $\mu'(a) = \mu(a)$. Felmerül a kérdés, hogy ha egy Moore-automatát Mealy-féle automatának tekintünk, s vesszük ezen Mealy-automata egy (Mealy-automatáknál tárgyalt értelemben tekintett) részautomatáját, vajon ez a részautomata tekinthető lesz-e ugyancsak Moore-féle automatának, illetve részautomata lesz-e abban az értelemben is, ahogy azt a Moore-automaták esetén definiáltuk. A válaszuk az, hogy a két értelemben vett részautomata fogalom (Moore automatáknál) egybeesik. Legyen ugyanis $\gamma : A \times X \rightarrow Y$ az a leképezés, melyre a tekintett Moore-automata, tetszőleges $a \in A', x \in X'$ pár esetén eleget tesz a $\gamma(a, x) = \mu(\delta(a, x))$ összefüggésnek. Tegyük fel, hogy az ezen γ függvénnyel definiált, $\mathcal{A}$ -ból nyert $\mathcal{B} = (A, X, Y, \delta, \gamma)$ Mealy-automatának a $\mathcal{B}' = (A', X', Y', \delta', \gamma')$ Mealy-automata részautomatája. Ekkor $\delta' = \delta|_{A' \times X'}$ és $\gamma' = \gamma|_{A' \times X'}$. Ily módon tetszőleges $a \in A', x \in X'$ párra $\delta'(a, x) = \delta(a, x)$ és $\gamma'(a, x) = \gamma(a, x)$. Viszont $\gamma(a, x) = \mu(\delta(a, x))$, $\gamma'(a, x) = \gamma(a, x)$, valamint $\delta'(a, x) = \delta(a, x)$ miatt ekkor $\gamma'(a, x) = \mu(\delta'(a, x))$. Vagyis definiálhatjuk az $\mathcal{A}'' = (A', X', Y, \delta', \mu)$ Moore-automatát, mely nyilvánvalóan (A, X) -részautomatája lesz $\mathcal{A}$ -nak abban az értelemben, ahogy azt a Moore-automatáknál definiáltuk. Természetesen az is igaz, hogy Y tetszőleges olyan $Y'' \subseteq Y$ részhalmazára, melyre $\{\mu'(a) \mid a \in A'\} \subseteq Y''$, a $\mu'' = \mu|_{Y''}$ feltételnek eleget tevő $\mathcal{A}''' = (A', X', Y'', \delta', \mu'')$ Moore-automata a $\mathcal{A}$ automatának Moore-féle részautomatája lesz a Moore-automatáknál tekintett értelemben. Így valóban, ez a két részautomata-fogalom Moore-automatáknál egybeesik.

A Mealy-automatákra definiált speciális részautomata-fogalmak természetes módon definiálhatók Moore-féle automatákra, s az általános Moore-féle részautomata fogalomnál tárgyaltakhoz hasonló észrevételeket nyerünk, ha a Moore-automatákat speciális Mealy-automatáknak tekintve vesszük ezen Mealy-automaták megfelelő speciális $(A-, X-, Y-, (A, X)-, (A, Y)-, (X, Y)-)$ részautomatáit. Ugyanez érvényben marad az iniciális Moore-automaták iniciális részautomatáira is és azok további speciális (iniciális A -részautomata, iniciális X -részautomata, stb.) eseteire is.

Egy $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automata valamely $\mathcal{A}' = (A', X', Y', \delta', \gamma')$ Moore-automatára történő (általános) Moore-homomorfizmusa alatt egy olyan $\psi = (\psi_1, \psi_2, \psi_3)$ szürjektív leképezésekből álló leképezés-hármast értünk, melynek tagjai $\psi_1 : A \rightarrow A', \psi_2 : X \rightarrow X', \psi_3 : Y \rightarrow Y'$ formájúak, s teljesülnek rájuk minden $a \in A, x \in X$ pár esetén a

$$\psi_1(\delta(a, x)) = \delta'(\psi_1(a), \psi_2(x)),$$

illetve a

$$\psi_3(\mu(a)) = \mu'(\psi_1(a))$$

összefüggés. Igazolható, hogy a Moore-féle homomorfizmus speciális esete a Mealy-féle automatákra értelmezett homomorfizmusnak abban az értelemben, hogy ha szereplő Moore-automatákat speciális Mealy-automatáknak tekintjük, akkor az előbb definiált Moore-féle homomorfizmus egy Mealy-féle automatákra definiált homomorfizmust fog szolgáltatni. De az is igaz, hogy lehetséges példát adni olyan Moore-automatára, melyet ha Mealy-automatának tekintünk, a Mealy-automatáknál vett értelemben homomorfan leképezhető lesz egy olyan Mealy-automatára, mely nem tekinthető Moore-automatának (lásd 2.6. Példa).

Példa 2.6 Legyen adva egy $\mathcal{A} = (\{a, b\}, \{x, y\}, \delta, \mu)$ Moore-automata, melyre $\delta(a, x) = \delta(b, x) = a, \delta(a, y) = \delta(b, y) = b$ és $\mu(a) = w, \mu(b) = z$. Ezt a Moore-automatát Mealy-automatának tekintve nyerjük a $\mathcal{B} = (\{a, b\}, \{x, y\}, \delta, \gamma)$ Mealy-automatát, melyre $\gamma(a, x) = \gamma(b, x) = w, \gamma(a, y) = \gamma(b, y) = z$. Nyilvánvalóan fennáll minden $a' \in \{a, b\}, x' \in \{x, y\}$ párra a $\gamma(a, x) = \mu(\delta(a', x'))$ összefüggés. Most vegyük a $\mathcal{B}' = (\{c\}, \{x, y\}, \delta', \gamma')$ mealy-automatát, melyre $\delta'(c, x) = \delta'(c, y) = c$ fennállása mellett $\gamma'(c, x) = w, \gamma'(c, y) = z$. Azonnal látszik, hogy a $\psi(a) = \psi(b) = c$ összefüggéssel definiált $\psi : \{a, b\} \rightarrow \{c\}$ leképezés a $\mathcal{B}$ egy A -homomorfizmusa lesz $\mathcal{B}'$ -re. De $\delta'(c, x) = \delta'(c, y) = c$ és $\gamma'(c, x) \neq \gamma'(c, y)$ mellett az is látszik, hogy nincs olyan $\mu' : \{c\} \rightarrow \{w, z\}$ leképezés, melyre $\gamma'(c, x') = \mu'(\delta'(c, x'))$ fennállna minden $x' \in \{x, y\}$ mellett. $\mathcal{B}'$ tehát nem Moore-automata.

Ugyanúgy mint a Mealy-automatáknál, itt is tekinthetünk különféle speciális Moore-féle $(A-, X-, Y-, (A, X)-, (A, Y)-, (X, Y)-$ homomorfizmus típusokat, továbbá tekinthetjük mind az általános Moore-féle homomorfizmus-fogalom, mind pedig a speciális Moore-féle homomorfizmus-típusok iniciális változatait iniciális Moore-automatákra. Ugyanúgy, mint az általános esetben, ezekben az esetekben is lehet példát adni arra, hogy (Moore-automaták esetén) a Moore-féle speciális homomorfizmus-fogalmak (Moore-féle $A-, X-, Y-, (A, X)-, (A, Y)-, (X, Y)-$ homomorfizmus és ezek iniciális változatai) is valódi speciális esetei a Mealy-féle változatoknak. Végül megjegyezzük, hogy ha a szereplő leképezések bijektívek (vagy ha egy ilyen leképezés van akkor ha a szereplő leképezés bijektív), akkor Moore-féle izomorfizmusról, illetve annak megfelelő speciális típusairól beszélünk. Izomorfizmusok esetén viszont a két féle (Mealy-féle, illetve Moore-féle) izomorfizmus-fogalmak egybeesnek.

Hasonlóan mint Mealy-automaták esetén, kimondható az algebrában jól ismert Általános Homomorfia Tétel Moore-automatákra vonatkozó speciális esete.

Tétel 2.2 (Általános Homomorfia Tétel Moore-Automatákra Vonatkozó Speciális Esete) Tegyük fel, hogy az $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automata az $\mathcal{A}' = (A', X', Y', \delta', \mu')$ Moore-automatára valamely Moore-féle állapothomomorfizmussal leképezhető, s tekintsük az A állapothalmaznak azt a C osztályozását, amelynél bármely két $a, b \in A$ állapot akkor és csak akkor van egy osztályban, ha a tekintett Moore-féle állapothomomorfizmus szerinti képük ugyanaz. Ekkor az A állapothalmaz ezen C osztályozása kompatibilis, s a hozzá tartozó $\mathcal{A}/C$ faktorautomata A -izomorf lesz az $\mathcal{A}'$ automatával. Képletben,

$$\mathcal{A}' \sim_{\mathcal{A}} \mathcal{A}(\varphi) \Rightarrow \mathcal{A}/C \sim_{\mathcal{A}} \mathcal{A}'(C[a] \rightarrow \varphi(a)).$$

□

Hátramaradt még a megfelelő részautomata-, homomorfizmus-, és izomorfizmus-fogalmak kimenő jel nélküli automatákra történő definiálása. Természetesen az is igaz, hogy némi megszorítással (a kimenő jelhalmazokra vonatkozó összefüggések figyelmen kívül hagyásával) a homomorfizmus, izomorfizmus és annak egyes speciális típusai definiálhatók kimenő jel nélküli automatákra is. Így például az $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automatának részautomatája az $\mathcal{A}' = (A', X', \delta')$ kimenő jel nélküli automata, ha $A' \subseteq A, X' \subseteq X$, valamint $\delta' = \delta_{A' \times X'}$. Továbbá az $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automatának homomorf képe az $\mathcal{A}' = (A', X', \delta')$ kimenő jel nélküli automata, ha alkalmas $\psi_1 : A \rightarrow A', \psi_2 : X \rightarrow X'$ alakú, szürjektív leképezésekből álló $\psi = (\psi_1, \psi_2)$ leképezés-párra minden $a \in A, x \in X$ esetén $\psi_1(\delta(a, x)) = \delta'(\psi_1(a), \psi_2(x))$ fennáll. Az algebrában jól ismert Általános Homomorfia Tétel kimenő jel nélküli automatákra vonatkozó speciális esete formailag csaknem egybeesik a Mealy-féle automatákra vonatkozó speciális esettel.

Tétel 2.3 (Általános Homomorfia Tétel Kimenő Jel Nélküli Automatákra Vonatkozó Speciális Esete) *Tegyük fel, hogy az $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata az $\mathcal{A}' = (A', X', \delta')$ kimenő jel nélküli automatára valamely állapothomomorfizmussal leképezhető, s tekintsük az A állapothalmaznak azt a C osztályozását, amelynél bármely két $a, b \in A$ állapot akkor és csak akkor van egy osztályban, ha a tekintett állapothomomorfizmus szerinti képük ugyanaz. Ekkor az A állapothalmaz ezen C osztályozása kompatibilis, s a hozzá tartozó $\mathcal{A}/C$ faktorautomata A -izomorf lesz az $\mathcal{A}'$ automatával. Képletben,*

$$\mathcal{A}' \sim_A \mathcal{A}(\varphi) \Rightarrow \mathcal{A}/C \simeq_A \mathcal{A}'(C[a] \rightarrow \varphi(a)).$$

□

Az elmélet további kiépítésénél elsősorban Mealy-automatákra szorítkozunk. Így ha mást nem mondunk, automata alatt mindig Mealy-féle automatát fogunk érteni, azaz a Mealy-automaták esetén a "Mealy" jelzőt sok esetben elhagyjuk.

2.4 Az automaták által indukált leképezések

A korábban nem üres és véges halmazokra definiált néhány fogalmat a továbbiakban tetszőleges halmazokra is értelmezni fogjuk. Így valamely (véges vagy végtelen) Z halmaz elemeiből alkotott véges láncot Z -beli *szónak*, Z elemeit pedig időnként *betűknek* hívjuk. Ha Z az üres halmaz, technikai okokból Z^* egy olyan egy elemű halmazt jelöl, melynek egyetlen eleme az üres szó. (Itt jegyezzük meg hogy szemben a formális nyelveknél általánosan elfogadott λ jelöléssel, az automataelméleti könyvekben az üres szót e -vel vagy ε -nal is szokás jelölni. Ezen jegyzetben viszont az egységességre törekvés miatt mi maradunk az eddigi λ jelnél.) Tehát $\emptyset^* = \{\lambda\}$. Amennyiben Z nem üres, Z^* -al az *összes Z feletti szavak halmazát* fogjuk jelölni. Ezen kívül (akár üres a Z akár nem) Z^+ fogja jelölni az *összes nem üres Z feletti szavak halmazát*. Nemcsak

véges, hanem végtelen Z halmazokra is (és az üres halmazra is) érvényes lesz, hogy a Z^* -beli összes szavak a *konkatenációra* (egymás mellé írásra) nézve - mint műveletre - *monoidot*, azaz *egységelemes szabad félcsoporthot* fognak alkotni. Ugyancsak igaz, nemcsak véges hanem végtelen nem üres Z halmazokra is, hogy a Z^+ -beli összes szavak a konkatenációra (egymás mellé írásra) nézve - mint műveletre - *szabad félcsoporthot* fognak alkotni. (Ha Z az üres halmaz, akkor a Z feletti összes nem üres szavak Z^+ halmaza is nyilván üres halmaz. Márpedig egy félcsoporthról fel szokás tételezni, hogy legalább egy eleme van, azaz az üres halmazt nem szokás félcsoporthnak tekinteni. Így Z^+ -t nem tekintjük félcsoporthnak ha Z üres halmaz.) Egy $p \in Z^*$ szó *hosszát* - akkor is ha Z nem véges - $|p|$ -el jelöljük, s mint korábban, a szót alkotó betűk számát értjük alatta (multiplicitásokkal együtt). A Z halmaz számosságát is (akár véges, akár nem, akár üres akár nem) $|Z|$ -el jelöljük. Végül, ha p nem üres, mint korábban, $\prod p$ jelöli a p *utolsó betűjét*.

Legyen $\mathcal{A} = (A, X, Y, \delta, \gamma)$ tetszőleges Mealy-automata. A $\delta : A \times X \rightarrow A$ és a $\gamma : A \times X \rightarrow Y$ függvények értelmezését kiterjesztjük $A \times X^*$ -ra a következő definícióval:

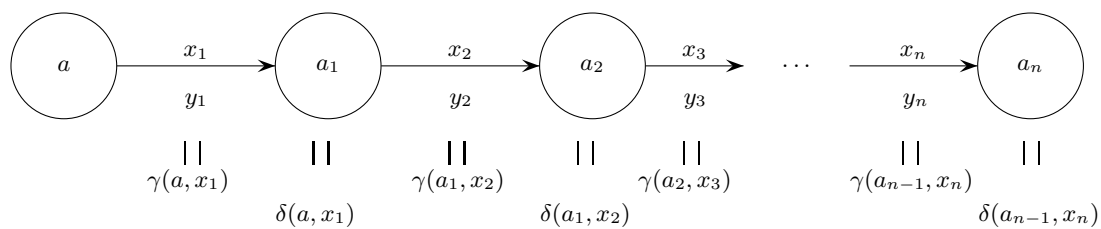
Legyenek $\delta : A \times X^* \rightarrow A^+$, $\gamma : A \times X^* \rightarrow Y^*$ úgy definiálva, hogy tetszőleges $a \in A$ és $x_1, \dots, x_n \in X$ esetén álljanak fenn a

$$\delta(a, x_1 \cdots x_n) = a_1 \cdots a_n$$

és a

$$\gamma(a, x_1 \cdots x_n) = y_1 \cdots y_n$$

összefüggések, ahol $a_1 = \delta(a, x_1), \dots, a_n = \delta(a_{n-1}, x_n)$, illetve $y_1 = \gamma(a, x_1), \dots, y_n = \gamma(a_{n-1}, x_n)$. Emellett legyen $\delta(a, \lambda) = \lambda$, $\gamma(a, \lambda) = \lambda$. Ez a formális definíció annak az interpretációnak felel meg, hogy bármely $a \in A$ állapotból indulva az $\mathcal{A}$ automata egy bemenő jelsorozatnak egy kimenő jelsorozatot feleltet meg (az automata "szekvenciális gép"). Nevezetesen, az automata az üres bemenő szóra üres kimenő szóval reagál.



MEALY-AUTOMATA MINT ÁLTALÁNOSÍTOTT SZEKVENCIÁLIS GÉP MŰKÖDÉSI VÁZLATA

Valamely X halmaz feletti X^* szabad monoidot egy Y halmaz feletti Y^* szabad monoidba leképező $\alpha : X^* \rightarrow Y^*$ leképezést *alfabetikus leképezésnek* hívunk. A bemenő szavak a bemenő információ, a kimenő szavak pedig a kimenő információ hordozói. Az automata az információ átalakítást alfabetikus leképezések segítségével realizálja. Így a fenti $x_1 x_2 \cdots x_n$ szóhoz az automata a fenti $y_1 y_2 \cdots y_n$ szót rendeli hozzá. Speciálisan, az üres szóhoz minden állapot az üres szót felelteti meg, hisz

definícióink értelmében tetszőleges $a \in A$ állapotra $\gamma(a, \lambda) = \lambda$. A továbbiakban egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata minden egyes $a \in A$ állapotához hozzárendeljük a $\varphi_{a, \mathcal{A}}(p) = \gamma(a, p)$, $p \in X^*$ összefüggéssel definiált $\varphi_{a, \mathcal{A}} : X^* \rightarrow Y^*$ leképezést, melyet az $a \in A$ állapot által indukált leképezésnek is fogunk hívni. Ha nem áll fenn a félreértés veszélye, $\varphi_{a, \mathcal{A}}$ helyett a legtöbbször csak φ_a -t írunk. Inicialis $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ Mealy-automata esetén az a_0 kezdő állapottal indukált φ_{a_0} leképezést az $\mathcal{A}$ *inicialis automata által indukált leképezésnek*, vagy röviden az $\mathcal{A}$ *automata leképezésének* is mondjuk és időnként $\varphi_{\mathcal{A}}$ -val jelöljük. A következő tétel George N. Raneytől származik.

Tétel 2.4 (Raney tétele) Egy $\varphi : X^* \rightarrow Y^*$ alfabetikus leképezés akkor és csak akkor automata leképezés, ha eleget tesz a következő két feltételnek:

- (i) hossztartó, azaz tetszőleges $p \in X^*$ -ra $|p| = |\varphi(p)|$,
- (ii) kezdőszelet tartó (kezdőszeletet kezdőszeletbe visz át), azaz minden $p, q \in X^*$ -hoz létezik olyan $r \in Y^*$, hogy $\varphi(pq) = \varphi(p)r$.

Bizonyítás: A szükségesség nyilvánvaló a kiterjesztett átmeneti és kimeneti függvények tulajdonságai miatt. Valóban, legyen $\mathcal{A} = (A, X, Y, \delta, \gamma)$ tetszőleges Mealy-automata, s legyen $a \in A$ tetszőleges állapota. Legyen $p \in X^*$ tetszőleges. Ha $p = \lambda$, azaz p az üres szó, akkor $\gamma(a, \lambda) = \lambda$ a definíciónk szerint, azaz a $\varphi_a(\lambda) = \gamma(a, \lambda)$ és a $\gamma(a, \lambda) = \lambda$ összefüggések miatt ekkor $\varphi_a(\lambda) = \lambda$, amiből $|\varphi_a(\lambda)| = |\lambda|$ nyilvánvalóan következik. φ_a tehát az üres szóra teljesíti a hossztartó tulajdonságot. Most legyen $p \in X^*$ nem üres szó, azaz legyen valamely $x_1, \dots, x_n \in X$ bemenő jelekre $p = x_1 \dots x_n$. Ekkor, amint a δ és γ függvények kiterjesztésénél láttuk, alkalmas $a_1, \dots, a_n \in A$ állapotokra $a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n)$ teljesülése mellett $\gamma(a, x_1 \dots x_n) = \gamma(a, x_1)\gamma(a_1, x_2) \dots \gamma(a_{n-1}, x_n)$. Vagyis, bevezetve az $y_1 = \gamma(a, x_1), y_2 = \gamma(a_1, x_2), \dots, y_n = \gamma(a_{n-1}, x_n)$ jelöléseket, $\varphi_a(x_1 \dots x_n) = \gamma(a, x_1 \dots x_n) = y_1 \dots y_n$. Ebből nyilvánvaló, hogy $|x_1 \dots x_n| = |y_1 \dots y_n| = n$, vagyis $|p| = |\varphi_a(p)|$, azaz φ_a minden X^* -beli (üres vagy nem üres) szóra hossztartó.

Az üres szó X^* -nak és Y^* -nak egységeleme, azaz tetszőleges $q \in X^*, w \in Y^*$ esetén $\lambda q = q\lambda = q$, illetve $\lambda w = w\lambda = w$. Legyen most $q \in X^*$ tetszőleges szó. Ekkor azt kapjuk, hogy $\varphi_a(\lambda q) = \varphi_a(q) = \lambda \varphi_a(q)$. Vagyis a kezdőszelet tartó tulajdonság teljesülni fog ha az üres szó a kezdőszelet, a végszelet pedig maga a tekintett szó. (Írjunk r -t a $\varphi_a(q)$ helyébe a $\varphi_a(\lambda q) = \lambda \varphi_a(q)$ egyenlőség jobboldalán.) Hasonlóan kapjuk tetszőleges $p \in X^*$ -ra a $\varphi_a(p\lambda) = \varphi_a(p) = \varphi_a(p)\lambda$ levezetést. Tehát a kezdőszelet tartó tulajdonság akkor is teljesülni fog, ha a szó kezdőszeletének magát a szót tekintjük, végszeletének pedig az üres szót. (Írjunk r -t az λ helyébe a $\varphi_a(p\lambda) = \varphi_a(p)\lambda$ egyenlőség jobboldalán.) Most válasszuk meg a $p, q \in X^*$ párt úgy, hogy egyikük se legyen üres. Ekkor valamely $x_1, \dots, x_k, x_{k+1}, \dots, x_n \in X$ bemenő jelekre $p = x_1 \dots x_k, q = x_{k+1} \dots x_n$. Vezessük be rendre az $a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n)$ és az $y_1 = \gamma(a, x_1), y_2 = \gamma(a_1, x_2), \dots, y_n = \gamma(a_{n-1}, x_n)$ jelöléseket. Ekkor $\varphi_a(pq) = \gamma(a, pq) = \gamma(a, x_1 \dots x_n) = \gamma(a, x_1)\gamma(a_1, x_2) \dots \gamma(a_{k-1}, x_k)\gamma(a_{k+1}, x_{k+1}) \dots \gamma(a_{n-1}, x_n) = y_1 \dots y_k y_{k+1} \dots y_n$ és $\varphi_a(p) = \gamma(a, p) = \gamma(a, x_1 \dots x_k) = \gamma(a, x_1)\gamma(a_1, x_2) \dots \gamma(a_{k-1}, x_k) = y_1 \dots y_k$ fennállnak. Ebből viszont $\varphi_a(pq) = \varphi_a(p)y_{k+1} \dots y_n$, vagyis az $r = y_{k+1} \dots y_n$ választással kapjuk, hogy alkalmas $r \in Y^*$ -ra $\varphi_a(pq) = \varphi_a(p)r$ ebben az esetben is fennáll. φ_a tehát valóban kezdőszelet tartó.

Az elegendőséghez legyen $\varphi : X^* \rightarrow Y^*$ tetszőleges hossz és kezdőszelet tartó alfabetikus leképezés. Ekkor minden $p, q \in X^*$ párhoz lesz olyan $r \in Y^*$, hogy $\varphi(pq) = \varphi(p)r$.

Rögzített $p \in X^*$ esetén jelölje φ_p azt a $\varphi_p : X^* \rightarrow Y^*$ alfabetikus leképezést, melyre tetszőleges $q \in X^*$ esetén a $\varphi(pq) = \varphi(p)\varphi_p(q)$ egyenlőség fog teljesülni. Először mutassuk meg, hogy a tetszőlegesen választott $p \in X^*$ -hoz tekintett φ_p alfabetikus leképezés is automata leképezés. Valóban, φ hossztartó volta miatt $|pq| = |\varphi(pq)|$ és $|p| = |\varphi(p)|$. Másrészt $\varphi(pq) = \varphi(p)\varphi_p(q)$ miatt $|\varphi(p)\varphi_p(q)| = |\varphi(pq)|$, azaz $|p| + |q| = (|pq| = |\varphi(p)\varphi_p(q)| = |\varphi(p)| + |\varphi_p(q)|)$. Ebből $|p| = |\varphi(p)|$ értelmében $|q| = |\varphi_p(q)|$, azaz φ_p hossztartó. Mutassuk meg, hogy kezdőszelet tartó is. Legyen $q, z \in X^*$ tetszőleges. Ekkor $\varphi(pqz) = \varphi(p)\varphi_p(qz)$ és $\varphi(pqz) = \varphi(pq)\varphi_{pq}(z) = \varphi(p)\varphi_p(q)\varphi_{pq}(z)$ is teljesülni fognak a φ kezdőszelet tartó volta miatt. Így $\varphi(p)\varphi_p(qz) = \varphi(p)\varphi_p(q)\varphi_{pq}(z)$ is teljesül. Ám szabad monoidban a szavak egyenlősége betűről-betűre való egyenlőséget jelent, azaz a legutóbb kapott egyenlőségünkéből $\varphi_p(qz) = \varphi_p(q)\varphi_{pq}(z)$ is következik. Más szóval, tetszőleges $q, z \in X^*$ -hoz létezik olyan $r (= \varphi_{pq}(z)) \in Y^*$, hogy $\varphi_p(qz) = \varphi_p(q)r$ teljesül. Tehát φ_p is kezdőszelet tartó. Azt kaptuk tehát, hogy tetszőleges $p \in X^*$ esetén φ_p is automata leképezés. Tetszőleges $p \in X^*$ esetén a φ_p automata leképezést a továbbiakban a φ leképezés állapotának fogjuk hívni. Legyen $A_\varphi = \{\varphi_p \mid p \in X^*\}$ és definiáljuk az $\mathcal{A}_\varphi = (A_\varphi, \varphi_e, X, Y, \delta_\varphi, \gamma_\varphi)$ Mealy-automatát úgy, hogy minden $\varphi_p \in A_\varphi$, $x \in X$ esetén $\delta_\varphi(\varphi_p, x) = \varphi_{px}$, $\gamma_\varphi(\varphi_p, x) = \varphi_p(x)$. A tételhez azt kell még megmutatni, hogy $\mathcal{A}_\varphi$ automata a φ leképezést indukálja.

Mivel az üres szó az egyetlen olyan szó, melynek hossza nulla, a φ hossztartó volta miatt $\varphi(\lambda) = \lambda$. Így tekintettel arra, hogy az üres szó mint bemenő szó hatására egy Mealy-automata kimenő szóként definíció szerint az üres szót adja ki, $\gamma_\varphi(\varphi_\lambda, \lambda) = \varphi(\lambda) (= \lambda)$ teljesülni fog. Most legyen $p = x_1 \cdots x_n$ egy nem üres bemenő szó, ahol $x_1, \dots, x_n \in X$ tetszőleges bemenő jelek. Az $\mathcal{A}_\varphi$ automata definíciója értelmében, valamint amiatt, hogy tetszőleges $r \in X^*$ szóra $\lambda r = r$ fennáll, $\delta(\varphi_\lambda, p) = \delta(\varphi_\lambda, x_1 \cdots x_n) = \varphi_{\lambda x_1} \varphi_{\lambda x_1 x_2} \cdots \varphi_{\lambda x_1 \cdots x_n} = \varphi_{x_1} \varphi_{x_1 x_2} \cdots \varphi_{x_1 \cdots x_n}$. Ekkor azonban, ugyancsak az $\mathcal{A}_\varphi$ automata definíciója alapján

$$\begin{aligned} \varphi_{\mathcal{A}_\varphi} &= \gamma_\varphi(\varphi_\lambda, p) = \gamma_\varphi(\varphi_\lambda, x_1 \cdots x_n) = \gamma_\varphi(\varphi_\lambda, x_1) \gamma_\varphi(\varphi_{x_1}, x_2) \cdots \gamma_\varphi(\varphi_{x_1 \cdots x_{n-1}}, x_n) = \\ &= \varphi_\lambda(x_1) \varphi_{x_1}(x_2) \cdots \varphi_{x_1 \cdots x_{n-1}}(x_n) = \lambda \varphi_\lambda(x_1) \varphi_{x_1}(x_2) \cdots \varphi_{x_1 \cdots x_{n-1}}(x_n) = \\ &= \varphi(\lambda) \varphi_\lambda(x_1) \varphi_{x_1}(x_2) \cdots \varphi_{x_1 \cdots x_{n-1}}(x_n) = \varphi(\lambda x_1) \varphi_{x_1}(x_2) \cdots \varphi_{x_1 \cdots x_{n-1}}(x_n) = \\ &= \varphi(\lambda x_1 x_2) \varphi_{x_1 x_2}(x_3) \cdots \varphi_{x_1 \cdots x_{n-1}}(x_n) = \cdots = \varphi(\lambda x_1 \cdots x_n) = \varphi(x_1 \cdots x_n) = \varphi(p). \end{aligned}$$

Viszont épp ezt kellett bizonyítani. $\square$

Nyilvánvaló, hogy ha egy automata leképezés egy véges iniciális automatával indukálható, akkor ennek az automata leképezésnek legfeljebb annyi állapota lehet mint az őt indukáló véges automatának. Másrészt az is világos, hogy a Raney tételének bizonyításában szereplő $\mathcal{A}_\varphi$ automata véges, ha a bemenő és kimenő jelhalmazok végeessége mellett a tekintett φ automata leképezés véges állapotú. Így a következő állításhoz jutunk.

Következmény 2.7 *Egy véges halmaz feletti monoidot egy véges halmaz feletti monoidba képező automata leképezés akkor és csak akkor indukálható véges automatában, ha állapotainak száma véges.* $\square$

Megjegyzés 2.8 Egy $\varphi : X^* \rightarrow Y^*$ automata leképezés indukálható a következő Mealy-automatával is: $\mathcal{A}^\varphi = (X^*, \lambda, X, Y, \delta^\varphi, \gamma^\varphi)$, ahol tetszőleges $p \in X^*, x \in X$ párra $\delta^\varphi(p, x) = px, \gamma^\varphi(p, x) = \overline{\varphi(px)}$ (ahol $\overline{\varphi(px)}$ a $\varphi(px)$ utolsó betűjét jelöli).

Az $\mathcal{A}_\varphi$ automatát a φ automata leképezéséhez tartozó alsó automatának, az $\mathcal{A}^\varphi$ automatát pedig a φ automata leképezéséhez tartozó felső automatának hívjuk. Világos, hogy a felső automata mindig végtelen állapotú.

Egy iniciális automatát *iniciálisan összefüggőnek* hívunk, ha a kezdő állapotából minden állapotba visz át bemenő szó. Képletben, az $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ iniciális Mealy automata iniciálisan összefüggő, ha minden $a \in A$ állapothoz létezik olyan $p \in X^*$ bemenő szó, hogy $\delta(a_0, p) = a$. (Megjegyezzük, hogy $\delta(a_0, \lambda) = a_0$ miatt ez a kezdő állapotra mindig fennáll.) Nyilvánvaló, hogy egy automata leképezéséhez tartozó alsó és felső automaták iniciálisan összefüggők. Bizonyítás nélkül megemlítyük a következő tételt.

Tétel 2.5 Ha $\mathcal{A}$ tetszőleges olyan iniciálisan összefüggő automata, amely a φ leképezést indukálja, akkor az $\mathcal{A}$ automata a $\mathcal{A}^\varphi$ felső automatának állapotomorf képe, s ugyanekkor az $\mathcal{A}_\varphi$ alsó automata pedig $\mathcal{A}$ -nak állapotomorf képe. $\square$

Megjegyzés 2.9 Egy φ automata leképezés előállításánál felhasznált automaták közül a hozzá tartozó alsó automata a lehető leggazdaságosabb, a hozzá tartozó felső automata pedig a lehető leggazdaságtalanabb. Így az optimális előállításnál a φ automata leképezéséhez tartozó alsó automatát kell előállítani. Ez így azonban csupán elméleti eredmény, mivel adott $p, q \in X^*$ párra általában nehéz ellenőrizni, hogy fennáll-e a $\varphi_p = \varphi_q$ egyenlőség.

Tétel 2.6 Tetszőleges X halmazra X^* összes önmagába történő automata leképezéseinek K_X halmaza a leképezések szokásos szorzására nézve monoidot, azaz egységelemes félcsoportot alkot. $\square$

Megjegyzés 2.10 Egy $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ által indukált leképezésnek a $\mathcal{B} = (A', a'_0, X', Y', \delta', \gamma')$ automata által indukált leképezéssel való szorzása értelmezhető, ha $Y \subseteq X'$. Ezt a szorzat-leképezést a $\mathcal{B}$ automata indukálja, ha $\mathcal{B}' = (A \times A', (a_0, a'_0), X, Y', \delta'', \gamma'')$ alakú, ahol tetszőleges $(a, a') \in A \times A', x \in X$ párra $\delta''((a, a'), x) = (\delta(a, x), \delta'(a', \gamma(a, x)))$ és $\gamma''((a, a'), x) = \gamma'(a', \gamma(a, x))$. A $\mathcal{B}'$ automatát az $\mathcal{A}$ automata $\mathcal{B}$ automatával történő soros kapcsolásának vagy szuperpozíciójának hívjuk.

Tétel 2.7 Egy X halmaz feletti X^* monoid összes önmagába történő, véges automaták által indukálható leképezéseinek L_X halmaza a leképezések szokásos szorzására nézve részfélcsoportot alkot a K_X félcsoportban. $\square$

Tétel 2.8 Tetszőleges X halmazra az X^* összes önmagába történő bijektív automata leképezéseinek A_X halmaza a leképezések szokásos szorzására nézve részcsoportot alkot a K_X félcsoportban. $\square$

Tétel 2.9 Egy X halmaz feletti X^* monoid összes önmagába történő, véges automaták által indukálható bijektív leképezéseinek G_X halmaza a leképezések szokásos szorzására nézve részcsoportot alkot a K_X félcsoportban és az A_X csoportban. $\square$

Egy S félcsoport generátorrendszerén értjük S -beli elemek egy H részhalmazát, ha S minden eleme előáll H -beli elemek szorzataként. H minimális generátorrendszere, vagy más néven *bázisa* S -nek, ha amellett hogy S -nek generátorrendszere, tetszőleges $h \in H$ mellett $H \setminus \{h\}$ már nem generátorrendszere S -nek. Hasonlóan, egy G csoport generátorrendszerén értjük G -beli elemek egy H részhalmazát, ha G minden eleme előáll olyan szorzatként, melynek minden tényezője vagy egy H -beli elem, vagy pedig egy H -beli elem inverze. Ugyanúgy mint félcsoportok esetén, H minimális generátorrendszere, vagy más néven *bázisa* G -nek, ha amellett hogy G -nek generátorrendszere, tetszőleges $h \in H$ mellett $H \setminus \{h\}$ már nem generátorrendszere G -nek. (Az itt szereplő S és G betűk az angol *group* (=csoport), illetve *semi-group* szavak kezdőbetűi, a generatív nyelvtanoknál szereplő S , illetve G -vel való azonosságuk csak a véletlen műve!)

Nem nehéz belátni, hogy ha X egy egyelemű halmaz vagy X az üres halmaz, akkor K_X, L_X, A_X, G_X mindegyike egy elemű. Így ebben az esetben mindegyiknek önmaga a bázisa. Ha X legalább két elemű, akkor érvényes a következő állítás.

Tétel 2.10 Egy legalább két elemű X halmaz esetén K_X -nek és L_X -nek nincs bázisa. $\square$

Nyitott kérdés, hogy van-e bázisa A_X -nek, illetve G_X -nek egy legalább két elemű X halmaz esetén.

Végül megjegyezzük, hogy Moore-automata esetén az átmeneti és jelfüggvények kiterjesztése hasonlóképp történik mint Mealy-automata esetén. Legyen $\mathcal{A} = (A, X, Y, \delta, \mu)$ tetszőleges Moore-automata. A $\delta : A \times X \rightarrow A$ és a $\mu : A \rightarrow Y$ függvények értelmezését kiterjesztjük $A \times X^*$ -ra, illetve A^* -ra a következő definícióval:

Legyenek $\delta : A \times X^* \rightarrow A^+, \mu : A^* \rightarrow Y^*$ úgy definiálva, hogy tetszőleges $a \in A$ és $x_1, \dots, x_n \in X$ esetén álljanak fenn a

$$\delta(a, x_1 \cdots x_n) = a_1 \cdots a_n$$

és a

$$\mu(a_1 \cdots a_n) = y_1 \cdots y_n$$

összefüggések, ahol $a_1 = \delta(a, x_1), \dots, a_n = \delta(a_{n-1}, x_n)$, illetve $y_1 = \mu(a_1), \dots, y_n = \mu(a_n)$. Emellett legyen $\delta(a, \lambda) = a, \mu(\lambda) = \lambda$. Ekkor az a állapot által indukált φ_a leképezésre $\varphi_a(\lambda) = \mu(\lambda) = \lambda$, illetve tetszőleges $x_1, \dots, x_n \in X$ -re $\varphi_a(x_1 \cdots x_n) = \mu(a_1 \cdots a_n)$, aholis $a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n)$. Később látni fogjuk, hogy minden iniciális Mealy-automatával indukálható automata leképezés indukálható iniciális Moore-automatával is. Sőt az is igaz, hogy ha egy automata leképezés véges iniciális Mealy automatával indukálható, akkor indukálható véges iniciális Moore-automatával is. Így a fejezetben tárgyaltak Moore-automatákra ugyanígy érvényben maradnak. Az átmeneti függvény természetesen kiterjeszthető kimenő

jel nélküli automata esetén is. Nevezetesen, ugyanúgy mint Moore-automata esetén, egy $\mathcal{A} = (A, X, \delta)$ tetszőleges kimenő jel nélküli automatára a $\delta : A \times X \rightarrow A$ és a $\mu : A \rightarrow Y$ függvény értelmezését úgy terjesztjük ki $A \times X^*$ -ra, hogy $\delta : A \times X^* \rightarrow A^+$ -t a következőképp definiáljuk: Legyen $\delta : A \times X^* \rightarrow A^+$ úgy definiálva, hogy tetszőleges $a \in A$ és $x_1, \dots, x_n \in X$ esetén álljanak fenn a

$$\delta(a, x_1 \cdots x_n) = a_1 \cdots a_n$$

összefüggések, ahol $a_1 = \delta(a, x_1), \dots, a_n = \delta(a_{n-1}, x_n)$. Emellett legyen $\delta(a, \lambda) = a$. Ekkor az a állapot által indukált φ_a leképezésre $\varphi_a(\lambda) = \lambda$, illetve tetszőleges $x_1, \dots, x_n \in X$ -re $\varphi_a(x_1 \cdots x_n) = a_1 \cdots a_n$, aholis $a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n)$.

Mint korábban megjegyeztük, egy $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automatát szokás olyan $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatának tekinteni, ahol $Y = A$ és $\gamma = \delta$. Ez a kiterjesztett átmeneti és kimeneti függvényekre csak korlátozottan érvényes. Nevezetesen, ha $p \in X^*$ nem üres, akkor a kiterjesztett kimeneti függvényt is úgy értelmezzük, hogy $\gamma(a, p) = \delta(a, p), a \in A$. Az üres szó esetén viszont a kiterjesztett átmeneti és kimeneti függvényeket ebben az esetben (tehát egy $\mathcal{A} = (A, X, A, \delta, \gamma)$ Mealy automatának tekintett $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automaták esetén) úgy definiáljuk, hogy $\delta(a, \lambda) = a$ és $\gamma(a, \lambda) = \lambda$ minden $a \in A$ -ra. Ekkor egy $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata esetén egy $a \in A$ állapot által indukált φ_a leképezés alatt értjük azt a $\varphi_a : A^* \rightarrow A^*$ leképezést, melyre tetszőleges $p \in X^*$ esetén

$$\varphi_a(p) = \begin{cases} \delta(a, p), & \text{ha } p \neq \lambda, \\ \lambda, & \text{ha } p = \lambda. \end{cases}$$

A kimenő jel nélküli automaták által indukált automata leképezések speciális automata leképezések, s nem minden (Mealy- vagy Moore-féle automatával indukálható) automata leképezés indukálható véges automatával.

Példa 2.11 Legyen $\mathcal{A} = (\{a_0, a\}, a_0, \{x, y\}, \{x, y\}, \delta, \gamma)$ egy iniciális Mealy-automata, ahol $\delta(a_0, x) = \delta(a, x) = a_0, \delta(a_0, y) = \delta(a, y) = a, \gamma(a_0, x) = \gamma(a, x) = \gamma(a_0, y) = x, \gamma(a, y) = y$. Ekkor $\varphi_{a_0}(xxyxyy) = xxxxyy$, amihez akárhogyan is szerkesztünk meg egy iniciális kimenő jel nélküli $\mathcal{B} = (B, b_0, \delta')$ automatát, $\varphi_{b_0}(xxyxyy) \neq xxxxyy$ fog teljesülni. Tehát valóban, nem minden automata leképezés indukálható kimenő jel nélküli automatával.

2.5 Redukált automata. Végese automaták minimalizálása

Az előző fejezetben láttuk, hogy tetszőleges automata alfabetikus leképezések egy sokaságát indukálja. (Minden állapot indukál egy alfabetikus leképezést, melyek közül egyesek egybeeshetnek.) Jelölje $\mathcal{F}_{\mathcal{A}} = \{\varphi_a \mid a \in A\}$ egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ által indukált leképezések sokaságát. Egy ilyen sokaságot az $\mathcal{A}$ automata által indukált leképezések családjának is fogunk hívni. Előfordulhat, hogy $a, b \in A, a \neq b$ és mégis $\varphi_a = \varphi_b$. Most azzal a kérdéssel foglalkozunk, hogy ha egy $\mathcal{F}$ leképezés családhhoz sikerül már olyan $\mathcal{A}$ automatát találnunk, melyre $\mathcal{F}_{\mathcal{A}} = \mathcal{F}$ teljesül, hogyan

tudjuk helyettesíteni $\mathcal{A}$ -t egy ugyanilyen tulajdonságú, de minimális állapotszámú $\mathcal{A}_0$ automatával. (Természetesen ennek a dolognak akkor van igazán értelme, ha az állapothalmaz véges.) Definiáljunk egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automata állapothalmazán egy $\varrho_{\mathcal{A}}$ relációt: $a\varrho_{\mathcal{A}}b \Leftrightarrow \varphi_a = \varphi_b$ (azaz $a\varrho_{\mathcal{A}}b \Leftrightarrow \gamma(a, p) = \gamma(b, p)$ minden p bemenő szóra). Könnyen belátható, hogy az így definiált $\varrho_{\mathcal{A}}$ reláció kongruencia, s ráadásul a $\varrho_{\mathcal{A}}$ -hoz tartozó $C_{\mathcal{A}}$ kompatibilis osztályozás maximális abban az értelemben, hogy minden más kompatibilis osztályozás $C_{\mathcal{A}}$ -nak finomítása. Esetben $C_{\mathcal{A}}$ -t az $\mathcal{A}$ automatához tartozó maximális kompatibilis osztályozásnak is hívjuk, az $\mathcal{A}/\varrho_{\mathcal{A}}$ faktorautomatát pedig az $\mathcal{A}$ -hoz tartozó redukált automatának mondjuk. Általában, egy $\mathcal{A} = (A, X, Y, \delta, \gamma)$ Mealy-automatát redukálnak nevezünk, ha tetszőleges $a, b \in A$ pár esetén

$$a\varrho_{\mathcal{A}}b \Leftrightarrow a = b.$$

Megjegyzés 2.12 Ehhez két fontos megjegyzésünk van:

1. Egy automatához tartozó redukált automata a legkisebb állapotszámmal (illetve végtelen automaták esetén a legkisebb számosságú állapothalmazzal) bíró olyan automata, amely ugyanazt a leképezés családot állítja elő, mint az illető automata.

2. Egy automata akkor és csak akkor redukált, ha A -izomorf saját magával.

Egy $\mathcal{A}$ automata minimalizálásán az $\mathcal{A}$ -hoz tartozó $\mathcal{A}_0$ redukált automata megszerkeszthetőségét értjük. Ebben a vonatkozásban az $\mathcal{A}_0$ redukált automatát *minimalisnak* hívjuk. A következő, véges automaták minimalizálására szolgáló algoritmus D.D. Aufenkamp és F. E. Hohn nevéhez fűződik.

2.5.1 Aufenkamp-Hohn-féle Minimalizációs Algoritmus

Egy véges $\mathcal{A} = (A, X, Y, \delta, \gamma)$ automata esetén az $\mathcal{A}_0$ redukált automatához úgy jutunk el, hogy a

$$\forall a, b \in A : (a\varrho_{\mathcal{A}}b \Leftrightarrow \forall p \in X^* : \gamma(a, p) = \gamma(b, p))$$

(azaz minden $a, b \in A$ pár esetén $a\varrho_{\mathcal{A}}b$ akkor és csak akkor, ha $\gamma(a, p) = \gamma(b, p)$ bármely $p \in X^*$ esetén fennáll) relációhoz tartozó $C_{\mathcal{A}}$ osztályozást osztályozások egy $C_1, C_2, \dots$ sorozatán keresztül szerkesztünk meg, melyeket a következőképp definiálunk:

$$(i) \forall a, b \in A : (C_1[a] = C_1[b] \Leftrightarrow \forall x \in X : \gamma(a, x) = \gamma(b, x))$$

(azaz minden $a, b \in A$ párra a C_1 osztályozás szerint a és b akkor és csak akkor esnek egy osztályba, ha ugyanazon bemenő jelre ugyanazon kimenő jellel reagálnak);

$$(ii) \text{ ha } i \geq 1 \text{ akkor } C_{i+1}[a] = C_{i+1}[b] \Leftrightarrow C_i[a] = C_i[b] \text{ és } \forall x \in X : C_i[\delta(a, x)] = C_i[\delta(b, x)].$$

(Azaz ha $i \geq 1$, úgy a C_{i+1} osztályozás szerint a és b akkor és csak akkor esnek egy osztályba, ha egyrészt már C_i szerint is egy osztályba esnek, másrészt pedig minden bemenő jel hatására egy és ugyanazon C_i szerinti osztályba mennek át.)

Először megszerkesztjük az A állapothalmaz C_1 szerinti osztályait, mikoris pontosan akkor tartozik két állapot egy osztályba, amikor minden egyes bemenő jel hatására ugyanazzal a kimenő jelet adják ki. (Lásd (i).) Ezután minden egyes $m > 1$ -re megszerkesztjük a C_m szerinti osztályokat egészen addig, míg $C_m = C_{m+1}$ teljesül. Látni fogjuk, hogy ilyen m létezik, s nagysága legfeljebb $|A| - 1$. Igazolni

fogjuk, hogy ez a C_m osztályozás épp az A maximális kompatibilis osztályozása. Ezután a redukált automata megszerkesztése van csak hátra, melynek szerkezete $\mathcal{A}/C_m = (C_m, X, Y, \delta_{C_m}, \gamma_{C_m})$, aholis minden $C_m[a] \in C_m, x \in X$ -re $\delta_{C_m}(C_m[a], x) = C_m[\delta(a, x)]$, illetve $\gamma_{C_m}(C_m[a], x) = \gamma(a, x)$. Speciálisan, ha az eredeti automata iniciális volt, és a kezdő állapota a_0 volt, akkor a redukált automata is választható iniciálisnak, éspedig úgy, hogy a kezdő állapotát $C_m[a_0]$ -nak választjuk.

2.5.2 Kiegészítés az Aufenkamp-Hohn Minimalizációs Algoritmushoz

Ha a célunk nem az automatához tartozó (redukált) minimális állapotszámú automata, hanem csupán a φ_{a_0} -t indukáló minimális iniciális automata meghatározása, akkor ehhez az $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ minimalizálandó automata helyett annak a kezdő állapotból elérhető állapotok által meghatározott, azaz az $A' = \{\delta(a_0, p) \mid p \in X^*\}$ állapothalmazzal rendelkező $\mathcal{A}' = (A', a_0, X, Y, \delta', \gamma')$ iniciálisan összefüggő állapot-részautomatáját minimalizáljuk. Az a_0 -ból el nem érhető állapotok ugyanis nem játszanak szerepet a φ_{a_0} leképezés indukálásában (az összes a_0 -ból elérhető állapot viszont igen). Mivel $\mathcal{A}$ végeessége miatt ekkor a kezdő állapotból csak azok az állapotok érhetők el, melyek legfeljebb $|A| - 1$ hosszúságú szavakkal elérhetők, az A állapothalmaz A' részhalmaza algoritmikusan meghatározható. Erre a következő módszer kínálkozik: Legyen először $A_0 = \{a_0\}$. Ezután minden $i \geq 1$ -re legyen $A_{i+1} = \{a \mid \exists a' \in A_i, x \in X : \delta(a', x) = a\}$. Ha valamely $i \geq 1$ -re elérjük, hogy $A_i = A_{i+1}$, akkor $A' = A_i$. Nyilván az ilyen i -re $i \leq |A| - 1$ teljesül.

Most meg fogjuk mutatni, hogy az algoritmus és a kiegészítő megjegyzésünk korrekt. Érvényesek a következő megállapítások:

Megállapítás 2.13 *Tetszőleges $a, b \in A$ párra és $i \geq 1$ természetes számra $C_i[a] = C_i[b]$ akkor és csak akkor áll fenn, ha minden i -nél nem hosszabb $p \in X^*$ bemenő szóra $\gamma(a, p) = \gamma(b, p)$.*

Bizonyítás: Először azt igazoljuk, hogy tetszőleges $a, b \in A$ párra és $i \geq 1$ természetes számra $C_i[a] = C_i[b]$ -nek következménye, hogy minden i -nél nem hosszabb $p \in X^*$ bemenő szóra $\gamma(a, p) = \gamma(b, p)$. Állításunkat teljes indukcióval fogjuk igazolni.

Ha $i = 1$, akkor egy i -nél nem hosszabb szó vagy az üres szó, vagy a bemenő jelhalmaz egy eleme. Az üres szóra $\gamma(a) = \lambda$ minden $a \in A$ pár esetén fennáll, tehát $\gamma(a, \lambda) = \gamma(b, \lambda) (= \lambda)$ akkor is igaz, ha speciálisan $C_1[a] = C_1[b]$ valamely $a, b \in A$ -ra. Amennyiben viszont $x \in X$ és $C_1[a] = C_1[b]$, akkor $\gamma(a, x) = \gamma(b, x)$ is teljesülni fog összhangban a C_1 osztályozás definíciójával.

Tegyük fel most, hogy valamely $i \geq 1$ -re igaz az állítás. Mutassuk meg, hogy ekkor $i + 1$ -re is igaz. Először észrevesszük, hogy minden olyan $a, b \in A$ párra, melyre $C_{i+1}[a] = C_{i+1}[b]$ fennáll, definíciónk értelmében $C_i[a] = C_i[b]$ is igaz. Ekkor viszont az indukciós feltevésünk miatt minden olyan $p \in X^*$ -ra, melyre p nem hosszabb i -nél, $\gamma(a, p) = \gamma(b, p)$. Azt kell tehát csak belátnunk, hogy amennyiben egy p szó hossza $i + 1$, akkor minden olyan $a, b \in A$ párra, melyre $C_{i+1}[a] = C_{i+1}[b]$, fennáll a $\delta(a, p) = \delta(b, p)$ egyenlőség. Igen ám, de ekkor p előáll $p = xq$ alakban, ahol $x \in X$, s ugyanekkor $q \in X^*$ pedig i hosszúságú. Ekkor tehát $\gamma(a, p) = \gamma(a, xq) = \gamma(a, x)\gamma(\delta(a, x), q)$, illetve $\gamma(b, p) = \gamma(b, xq) = \gamma(b, x)\gamma(\delta(b, x), q)$. $C_{i+1}[a] = C_{i+1}[b]$ fennállása mellett $C_1[a] = C_1[b]$ is igaz, amiből $\gamma(a, x) = \gamma(b, x)$ következik (lásd $i = 1$ eset). Másrészt $C_{i+1}[a] = C_{i+1}[b]$ definíció szerint azt is jelenti, hogy minden $x \in X$ esetén

$C_i[\delta(a, x)] = C_i[\delta(b, x)]$. Viszont az indukciós feltevésünk értelmében ekkor minden i hosszúságú $q \in X^*$ szóra $\gamma(\delta(a, x), q) = \gamma(\delta(b, x), q)$. Ehhez figyelembe véve az előbb megállapított $\gamma(a, x) = \gamma(b, x)$ egyenlőséget, fennáll a $\gamma(a, x)\gamma(\delta(a, x), q) = \gamma(b, x)\gamma(\delta(b, x), q)$ egyenlőség is, ami épp azt jelenti, hogy $\gamma(a, xq) = \gamma(b, xq)$. Ez viszont $p = xq$ értelmében a $\gamma(a, p) = \gamma(b, p)$ egyenlőséghez vezet.

Most azt tegyük fel, hogy adott $a, b \in A$ és $i \geq 1$ mellett $\gamma(a, p) = \gamma(b, p)$ teljesül minden $p \in X^*, |p| \leq i$ feltételnek eleget tevő bemenő szóra. Igazoljuk teljes indukcióval, hogy ekkor $C_i[a] = C_i[b]$.

Ha $i = 1$ és minden $x \in X$ -re (azaz minden $p \in X^*, |p| = 1$ -re) $\gamma(a, x) = \gamma(b, x)$, akkor $C_1[a] = C_1[b]$ definíció szerint teljesül. Tegyük fel ezután indukciós feltevésünként, hogy valamely rögzített $i \geq 1$ -re valahányszor egy $a, b \in A$ párra $\gamma(a, p) = \gamma(b, p)$ minden $|p| \leq i$ -nek eleget tevő $p \in X^*$ bemenő szó esetén fennáll, mindannyiszor $C_i[a] = C_i[b]$. Legyen ezután minden $i + 1$ -nél nem hosszabb nem üres $p \in X^*$ bemenő szó esetén $\gamma(a, p) = \gamma(b, p)$. Ekkor $p = xq$, ahol $x \in X$ és $q \in X^*$, ahol $|q| \leq i$. Ez többek között azt jelenti, hogy minden $x \in X$ -re és i -nél nem hosszabb $q \in X^*$ -ra $\gamma(\delta(a, x), q) = \gamma(\delta(b, x), q)$. Indukciós feltevésünk értelmében így $C_i[\delta(a, x)] = C_i[\delta(b, x)]$ tetszőleges $x \in X$ esetén teljesül. Másrészt ha $\gamma(a, p) = \gamma(b, p)$ minden $i + 1$ -nél nem hosszabb $p, q \in X^*$ bemenő szóra teljesül, úgy teljesül minden i -nél nem hosszabb bemenő szóra is. Ez viszont indukciós feltevésünk miatt $C_i[a] = C_i[b]$ -t eredményezi. Ezt összevetve azzal, hogy $C_i[\delta(a, x)] = C_i[\delta(b, x)]$ tetszőleges $x \in X$ esetén teljesül, definíció szerint kapjuk, hogy $C_{i+1}[a] = C_{i+1}[b]$. Ezzel az állítást igazoltuk. $\square$

Megállapítás 2.14 *Ha valamely $m \geq 1$ természetes számra $C_m = C_{m+1}$, akkor minden $u, v \geq m$ természetes szám párra $C_u = C_v$.*

Bizonyítás: Állításunkhoz nyilván elég igazolni, hogy ha $C_m = C_{m+1}$ akkor $C_{m+1} = C_{m+2}$. Legyen tehát valamely $m \geq 1$ természetes számra $C_m = C_{m+1}$, s legyen valamely $a, b \in A$ állapotpárra $C_{m+1}[a] = C_{m+1}[b]$. Ekkor viszont definíció szerint tetszőleges $x \in X$ -re teljesülni fog a $C_m[\delta(a, x)] = C_m[\delta(b, x)]$ egyenlőség. Viszont $C_m = C_{m+1}$ miatt így $C_{m+1}[\delta(a, x)] = C_{m+1}[\delta(b, x)]$ is fenn fog állni. Ez viszont a feltételezett $C_{m+1}[a] = C_{m+1}[b]$ egyenlőséggel együtt azt fogja definíció szerint eredményezni, hogy $C_{m+2}[a] = C_{m+2}[b]$. Vagyis azt kaptuk, hogy ha $C_m = C_{m+1}$, akkor minden olyan $a, b \in A$ párra, melyre $C_{m+1}[a] = C_{m+1}[b]$, fennáll $C_{m+2}[a] = C_{m+2}[b]$ is. Ez pedig pontosan azt jelenti, hogy $C_{m+1} = C_{m+2}$. $\square$

Most igazoljuk, hogy az Aufenkamp-Hohn féle algoritmus valóban korrekt. Ha C_1 szerint minden állapot egy osztályba esik, akkor minden állapot egy és ugyanazon kimenő jellel reagál egy és ugyanazon bemenő jelre. Ekkor $C_1 = C_2$ nyilvánvaló. Ezzetben tehát a 2.14. Megállapítás értelmében a minimális automata egyetlen állapottal fog rendelkezni, s egy-egy bemenő jelre ez a redukált automata egy és ugyanazon kimenő jelet adja ki, mint az eredeti automata bármelyik állapota.

Tegyük fel most, hogy $|C_1| > 1$. Ha $C_1 = C_2$, akkor $m = 1$ és ismét a C_1 a kívánt maximális kongruencia osztályozás. Ellenkező esetben C_2 -nek legalább eggyel több osztályt kell tartalmaznia mint C_1 -nek, vagyis legalább három osztályt. Ezt folytatva lesz egy egyre finomodó osztályozás rendszerünk, ahol minden egyes osztályozás

legalább eggyel több osztályt fog tartalmazni mint az előző. Tekintettel arra, hogy feltételeztük $|C_1| > 1$ -t, továbbá figyelembe véve, hogy legfeljebb $|A|$ számú osztálya lehet egy-egy osztályozásnak (hisz egy adott osztályozásnál minden osztálynak kell legalább egy A -beli elemet tartalmaznia), azt kapjuk, hogy lesz olyan $m \leq |A| - 1$, melyre $C_m = C_{m+1}$. Valóban, ha veszünk valamely $m > 1$ -re egy egyre finomodó olyan osztályozás rendszert, melyre $C_1 \supset C_2 \supset \dots \supset C_m$, akkor ha C_1 -nek legalább két, C_2 -ek legalább három, $\dots$, C_m -nek legalább $m + 1$ eleme van és összesen $|A|$ számú állapotunk van (vagyis legfeljebb $|A|$ számú osztálya lehet egy-egy osztályozásnak), akkor $m + 1 \leq |A|$ fennállása azt is jelenti, hogy $m \leq |A| - 1$. Így viszont valóban lesz olyan $m \leq |A| - 1$, mikoris $C_m = C_{m+1}$, ami a 2.14. Megállapítás értelmében azt is jelenti, hogy teszőleges $i \geq m$ -re $C_i = C_m$. Ez azonban a 2.13. Megállapítás miatt azt is eredményezi, hogy teszőleges olyan $a, b \in A$ állapotpárra, melyre $C_m[a] = C_m[b]$, teljesülni fog a $\gamma(a, p) = \gamma(b, p)$ egyenlőség, akármilyen hosszú is a tekintett $p \in X^*$ bemenő szó. Ezzel beláttuk, hogy C_m egy kongruencia osztályozás, aholis $m \leq |A| - 1$.

Be kell még látnunk, hogy C_m valóban maximális kongruencia osztályozás, vagyis ha két állapot nem tartozik C_m szerint egy osztályba, akkor az általuk indukált leképezések különbözőek. Legyen $a, b \in A$ két állapot, mely C_m szerint nem tartozik egy osztályba. Ha már C_1 szerint sem tartoznak egy osztályba, akkor valamely $x \in X$ -re $\gamma(a, x) \neq \gamma(b, x)$ és akkor valóban igaz, hogy $\varphi_a \neq \varphi_b$. Ellenkező esetben van egy maximális $k < m$, hogy $C_k[a] = C_k[b]$, ám $C_{k+1}[a] \neq C_{k+1}[b]$. Ekkor viszont a 2.13. Megállapítás alapján van legalább egy olyan $k + 1$ hosszúságú $p \in X^*$ szó, hogy $\delta(a, p) \neq \delta(b, p)$. (Nevezetesen, minden $k + 1$ -nél rövidebb $p \in X^*$ szóra $C_k[a] = C_k[b]$ miatt $\gamma(a, p) = \gamma(b, p)$, másrésről pedig $C_{k+1}[a] \neq C_{k+1}[b]$ miatt van legalább egy olyan $k + 1$ -nél nem hosszabb $p \in X^*$ szó, melyre $\gamma(a, p) \neq \gamma(b, p)$. A két állítás együttes fennállása azt jelenti, hogy $\gamma(a, p) \neq \gamma(b, p)$ legalább egy $k + 1$ hosszú szóra teljesül.) Be kell még látnunk azt is hogy a kiegészítő megjegyzésünk is korrekt.

A fentiek alapján az is igaz, hogy ha az $\mathcal{A}$ automata iniciális és kezdő állapota a_0 , akkor a redukált automatát is választhatjuk iniciálisnak oly módon, hogy kezdő állapotának a $C_m[a_0]$ osztályt választjuk. Ez az osztály is a φ_{a_0} leképezést fogja a redukált automatában indukálni ugyanúgy, mint ahogy az a_0 a φ_{a_0} leképezést indukálja $\mathcal{A}$ -ban.

Ha egy iniciális $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ végese automatához keressük azt a minimális automatát, mely φ_{a_0} -t indukálja, először célszerű az A állapothalmaz $A' = \{a' \in A \mid a' = \gamma(a_0, p), p \in X^*\}$ részhalmazát meghatározni, s azután az A' állapothalmaz által meghatározott $\mathcal{A}'$ iniciális állapot-részautomatát minimalizálni. Világos ugyanis, hogy ez az $\mathcal{A}'$ iniciális állapot-részautomata ugyancsak a φ_{a_0} -t indukálja, azaz $\mathcal{A}$ -ban a $A \setminus A'$ -beli, azaz az a_0 -ból el nem érhető állapotok ugyanis φ_{a_0} indukálásában nem játszanak szerepet. Jelölje $\mathcal{B}$ a $\mathcal{A}$ -hoz tartozó iniciális redukált automatát, $\mathcal{B}'$ pedig a $\mathcal{A}'$ -höz tartozó iniciális redukált automatát. Világos, hogy $\mathcal{B}'$ ugyanúgy a φ_{a_0} -t fogja indukálni, mint $\mathcal{B}$. Mivel A' részhalmaza az A -nak, az is világos, hogy a $\mathcal{B}'$ állapothalmaza részhalmaza a $\mathcal{B}$ állapothalmazának. Így ha B' jelöli a $\mathcal{B}'$, továbbá B jelöli a $\mathcal{B}$ állapothalmazát, $|B'| \leq |B|$.

Esetleg lehet olyan $a \in A \setminus A'$ állapotunk, melyre $\varphi_a \notin \{\varphi_{a'} \mid a' \in A'\}$. Ekkor azonban $|B'| < |B|$, vagyis ilyen esetben $\mathcal{B}$ nem egy minimális, φ_{a_0} leképezést indukáló automata. Tehát valóban, egy φ_{a_0} leképezést indukáló minimális automata

keresésénél célszerű először a vizsgált $\mathcal{A}$ iniciálisan összefüggő állapot részautomatáját meghatározni, majd az Aufenkamp-Hohn féle algoritmust erre az iniciális állapot részautomatára alkalmazni.

Kérdés még, hogy az A' meghatározásához javasolt algoritmusunk esetén korrekt-e. Az világos, hogy minden $i \geq 1$ -re az A_i halmaz definíció szerint az a_0 -ból legfeljebb i hosszúságú szavakkal elérhető állapotok halmaza. Így azt kell csak kimutatnunk, hogy ha egy állapot a_0 -ból elérhető, akkor elérhető $|A| - 1$ -nél nem hosszabb bemenő szóval is.

Tetszőleges $p, q, r \in X^*$ esetén $\delta(a_0, pq) = \delta(a_0, p)$ mellett $\delta(a_0, pqr) = \delta(a_0, pr)$ nyilván fennáll. Így ha valamely $a \in A$ állapothoz van olyan p' szó, hogy $\delta(a_0, p') = a$, akkor p' megválasztható úgy, hogy bármely két különböző $p'', p''' \in X^*$ kezdőszületére $\delta(a_0, p'') \neq \delta(a_0, p''')$ teljesüljön. Ekkor viszont p' hossza legfeljebb $|A| - 1$ lehet, különben az ismétlődés elkerülhetetlen. Így megvizsgálva azt, hogy melyek azok az állapotok, melyek elérhetők a kezdő állapotból legfeljebb $|A| - 1$ hosszú szóval, megkapjuk A' -t. Az is világos, hogy ha valamely $k < |A| - 1$ -re az a_0 -ból legfeljebb k hosszú szóval elérhető állapotok A_k halmaza egybeesik az a_0 -ból legfeljebb $k+1$ hosszú szóval elérhető állapotok halmazával, úgy az a_0 -ból elérhető összes állapotok halmaza A_k -val megegyezik. Valóban, ha a legfeljebb k hosszú bemenő szavakkal ugyanazokat az állapotokat érjük el mint a legfeljebb $k+1$ hosszú szavakkal, akkor minden $\ell \geq 0$ -ra a legfeljebb k hosszú bemenő szavakkal ugyanazokat az állapotokat érjük el mint a legfeljebb $k + \ell$, azaz bármilyen hosszú szavakkal. Az Aufenkamp-Hohn-féle minimalizációs algoritmusához vett kiegészítő megjegyzésünk tehát ugyancsak korrekt.

2.5.3 Aufenkamp-Hohn féle Minimalizációs Algoritmus Moore-automatákhoz Adaptált Változata

Megjegyezzük először, hogy ha egy Moore-automatát Mealy-automatának tekintünk, s minimalizáljuk, az eredmény nem feltétlenül lesz Moore-féle automata (lásd 2.6. Példa). Ha az $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automatához azt a minimális állapotszámú Moore-féle automatát akarjuk meghatározni, mely ugyanazt a leképezési családot indukálja, mint az eredeti Moore-automata, akkor az Aufenkamp-Hohn féle algoritmust egy kicsit módosítanunk kell.

Egy véges $\mathcal{A} = (A, X, Y, \delta, \mu)$ Moore-automata esetén az $\mathcal{A}_0$ redukált Moore-automatához úgy jutunk el, hogy definiálunk egy kongruencia relációt a következőképpen:

$$\forall a, b \in A : (a \varrho_{\mathcal{A}} b \Leftrightarrow \forall x_1, \dots, x_n \in X : \mu(a_1 \cdots a_n) = \mu(b_1 \cdots b_n), a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n), b_1 = \delta(b, x_1), b_2 = \delta(b_1, x_2), \dots, b_n = \delta(b_{n-1}, x_n))$$

(azaz minden $a, b \in A$ pár esetén $a \varrho_{\mathcal{A}} b$ akkor és csak akkor, ha $\mu(a_1 \cdots a_n) = \mu(b_1 \cdots b_n)$ fennáll minden olyan $a_1, \dots, a_n, b_1, \dots, b_n \in A$ esetén, melyekre $a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_n = \delta(a_{n-1}, x_n), b_1 = \delta(b, x_1), b_2 = \delta(b_1, x_2), \dots, b_n = \delta(b_{n-1}, x_n)$ fennáll valamely $x_1, \dots, x_n \in X$ bemenő jelek esetén).

Ekkor a $\varrho_{\mathcal{A}}$ relációhoz tartozó $C_{\mathcal{A}}$ osztályozást osztályozások egy $C_1, C_2, \dots$ sorozatán keresztül szerkesztünk meg, melyeket a következőképp definiálunk:

(i) $\forall a, b \in A : (C_1[a] = C_1[b] \Leftrightarrow \forall x \in X : \mu(a) = \mu(b))$
 (azaz minden $a, b \in A$ párra a C_1 osztályozás szerint a és b akkor és csak akkor esnek egy osztályba, ha ugyanaz az állapotjelük);

(ii) ha $i \geq 1$ akkor $C_{i+1}[a] = C_{i+1}[b] \Leftrightarrow C_i[a] = C_i[b]$ és $\forall x \in X : C_i[\delta(a, x)] = C_i[\delta(b, x)]$.

(Azaz ha $i \geq 1$, akkor amint a Mealy-automatáknál, a C_{i+1} osztályozás szerint a és b akkor és csak akkor esnek egy osztályba, ha egyrészt már C_i szerint is egy osztályba esnek, másrészt pedig minden bemenő jel hatására egy és ugyanazon C_i szerinti osztályba mennek át.)

A továbbiakban formailag a Moore-automatákra adaptált Aufenkamp-Hohn féle eljárás csaknem teljesen megegyezik a Mealy-automatáknál tárgyaltakkal. Először megszerkesztjük az A állapothalmaz C_1 szerinti osztályait, mikoris pontosan akkor tartozik két állapot egy osztályba, amikor ugyanaz az állapotjelük. (Lásd (i) pont, fentebb.) Ezután minden egyes $m > 1$ -re megszerkesztjük a C_m szerinti osztályokat egészen addig, míg $C_m = C_{m+1}$ teljesül. Ugyanúgy mint Mealy-automaták esetén, ilyen m létezik, s nagysága legfeljebb $|A| - 1$. Ez a C_m osztályozás épp az A (Moore-változatú) maximális kompatibilis osztályozása. Ezután a Moore-féle redukált automata megszerkesztése van csak hátra, melynek szerkezete $\mathcal{A}/C_m = (C_m, X, Y, \delta_{C_m}, \mu_{C_m})$, aholis minden $C_m[a] \in C_m, x \in X$ -re $\delta_{C_m}(C_m[a], x) = C_m[\delta(a, x)]$, illetve $\mu_{C_m}(C_m[a]) = \mu(a)$. Speciálisan, ha az eredeti automata iniciális volt, és a kezdő állapota a_0 volt, akkor a redukált automata is választható iniciálisnak, éspedig úgy, hogy a kezdő állapotát $C_m[a_0]$ -nak választjuk.

2.5.4 Kiegészítés az Aufenkamp-Hohn-féle Minimalizációs Algoritmus Moore-automatákhoz Adaptált Változatához

Ha a célunk nem az automatahoz tartozó (redukált) minimális állapotszámú Moore-automata, hanem csupán az a φ_{a_0} -t indukáló minimális iniciális Moore-automata meghatározása, akkor ehhez az $\mathcal{A} = (A, a_0, X, Y, \delta, \mu)$ minimalizálandó automata helyett annak a kezdő állapotból elérhető állapotok által meghatározott, azaz az $A' = \{\delta(a_0, p) \mid p \in X^*\}$ állapothalmazzal rendelkező $\mathcal{A}' = (A', a_0, X, Y, \delta', \mu')$ iniciálisan összefüggő állapot-részautomatáját minimalizáljuk. Az a_0 -ból el nem érhető állapotok ugyanis nem játszanak szerepet a φ_{a_0} leképezés indukálásában (az összes a_0 -ból elérhető állapot viszont igen). Mivel $\mathcal{A}$ végeessége miatt ekkor a kezdő állapotból csak azok az állapotok érhetők el, melyek legfeljebb $|A| - 1$ hosszúságú szavakkal elérhetők, az A állapothalmaz A' részhalmaza algoritmikusan meghatározható. Erre ugyanaz a módszer kínálkozik mint Mealy-automaták esetén: Legyen először $A_0 = \{a_0\}$. Ezután minden $i \geq 1$ -re legyen $A_{i+1} = \{a \mid \exists a' \in A_i, x \in X : \delta(a', x) = a\}$. Ha valamely $i \geq 1$ -re elérjük, hogy $A_i = A_{i+1}$, akkor $A' = A_i$. Nyilván az ilyen i -re $i \leq |A| - 1$ teljesül.

2.5.5 Minimalizációs Algoritmus Kimenő Jel Nélküli Automatákra

Tetszőleges $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata esetén valamely $a \in A$ állapot és nem üres p bemenő szó esetén a kimenő szót $\delta(a, p)$ -vel azonosítottuk. Ha a bemenő szó az üres szó, akkor viszont kimenő szónak ugyancsak az üres szót tekintettük minden állapotra vonatkozóan, mint a Mealy-automatáknál.

Ezek szerint egy egymástól különböző $a, b \in A$ állapotpár nem tartozhat egy osztályba, ha valamely (nem feltétlen különböző) $c, d \in A$ állapotpárra és $x, y \in X$ bemenő jelekre $\delta(c, x) = a$ és $\delta(d, y) = b$. Másképp mondva, két $a, b \in A$ állapot csak úgy tartozhat egy osztályba, ha egyrészt minden $x \in X$ -re $\delta(a, x) = \delta(b, x)$, másrészt legalább egy $d \in \{a, b\}$ -re akárhogy is adjuk meg a $c \in A, x \in X$ párt, $\delta(c, x) \neq d$.

Így az Aufenkamp-Hohn féle minimalizációs algoritmust helyett kimenő jel nélküli automatákra a következő algoritmust alkalmazzuk:

Egy véges $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata esetén az $\mathcal{A}_0$ redukált kimenő jel nélküli automatához is úgy jutunk el, hogy először képezzük az A egy olyan B_0 részhalmazát, mely tartalmaz minden olyan $a \in A$ állapotot, melyhez van olyan $b \in A, x \in X$, hogy $\delta(b, x) = a$. Vezessük be továbbá a $C_0 = A \setminus B_0$ jelölést. Mindaddig, míg valamely $i \geq 0$ -ra C_i üres nem lesz, hajtsuk végre a következő algoritmust:

Tegyük fel, hogy valamely $i \geq 0$ -ra B_i és C_i adottak. Legyen valamely tetszőlegesen rögzített $c \in C_i$ -re $B_{i+1} = B_i \cup \{c\}$. Eután C_{i+1} -t C_i -ből úgy képezzük, hogy C_i elemei közül elhagyunk minden olyan $c' \in C_i$ elemet, melyhez van olyan $b \in B_{i+1}$, hogy tetszőleges $x \in X$ mellett $\delta(c', x) = \delta(b, x)$. Ha $C_{i+1} = \emptyset$, akkor készen vagyunk, különben növeljük eggyel i értékét, s ismételjük meg az előző lépést.

Világos, hogy A végeessége miatt ez az eljárás véges lépésben valamely $i \geq 0$ -ra véget ér, s az is könnyen igazolható, hogy a fenti algoritmus végén minden $a \in A$ -hoz tartozik pontosan egy olyan $b \in B_{i+1}$, hogy tetszőleges $x \in X$ mellett $\delta(a, x) = \delta(b, x)$. Ekkor megkapjuk a $\mathcal{B} = (B, X, \delta')$ kimenő jel nélküli automatát, ahol $B = B_{i+1}$ és $\delta' = \delta|_{B \times X}$ mellett $\mathcal{B}$ olyan (kimenő jel nélküli) állapot részautomatája lesz $\mathcal{A}$ -nak, hogy minden $a \in A$ -hoz található olyan $b \in B$, hogy tetszőleges $x \in X$ mellett $\delta(a, x) = \delta(b, x)$. Ez a $\mathcal{B}$ kimenő jel nélküli automata lesz az $\mathcal{A}$ kimenő jel nélküli automatához tartozó redukált automata.

2.5.6 Kiegészítés a Minimalizációs Algoritmus Kimenő Jel Nélküli Automatákra Ismertetett Változatához

Ha a célunk nem a kimenő jel nélküli automatához tartozó (redukált) minimális állapotszámú kimenő jel nélküli automata, hanem csupán az a φ_{a_0} -t indukáló minimális iniciális kimenő jel nélküli automata meghatározása, akkor ehhez az $\mathcal{A} = (A, a_0, X, \delta)$ minimalizálandó kimenő jel nélküli automata helyett annak a kezdő állapotból elérhető állapotok által meghatározott, azaz az $A' = \{\delta(a_0, p) \mid p \in X^*\}$ állapothalmazzal rendelkező $\mathcal{A}' = (A', a_0, X, \delta')$ iniciálisan összefüggő (kimenő jel nélküli) állapot-részautomatáját minimalizáljuk. Az a_0 -ból el nem érhető állapotok ugyanis most sem játszanak szerepet a φ_{a_0} leképezés indukálásában (az összes a_0 -ból elérhető állapotok viszont igen). Mivel $\mathcal{A}$ végeessége miatt ekkor a kezdő állapotból csak azok az állapotok érhetők el, melyek legfeljebb $|A| - 1$ hosszúságú szavakkal elérhetők, az A állapothalmaz A' részhalmaza algoritmikusan meghatározható. Erre itt is a következő módszer kínálkozik: Legyen először $A_0 = \{a_0\}$. Ezután minden $i \geq 1$ -re legyen $A_{i+1} = \{a \mid \exists a' \in A_i, x \in X : \delta(a', x) = a\}$. Ha valamely $i \geq 1$ -re elérjük, hogy $A_i = A_{i+1}$, akkor $A' = A_i$. Nyilván az ilyen i -re $i \leq |A| - 1$ teljesül. Az a_0 -ból elérhető minden $a' \in A'$ állapot nyilván rendelkezik a tulajdonsággal, hogy alkalmas $b \in A', x \in X$ párra $\delta(b, x) = a'$. Így ha a_0 is elérhető önmagából, akkor $\mathcal{A}'$ egy minimális, φ_{a_0} -t indukáló automata lesz. Ugyancsak az lesz, ha minden $a \in A' \setminus \{a_0\}$ esetén van olyan $x \in X$, melyre $\delta(a_0, x) \neq \delta(a, x)$. Ellenkező esetben valamely $a \in A'$ -

re $\delta(a, x) = \delta(a_0, x)$ minden $x \in X$ -re fennáll, továbbá tetszőleges (az üres szótól különböző) $p \in X^+$ bemenő szó esetén $\delta(a_0, p) \neq a_0$. Ha van olyan $p \in X^+$ (üres szótól különböző) bemenő szó, hogy $\delta(a, p) = a$, akkor világos, hogy $\mathcal{A}'$ ismét redukált automata lesz. Ellenkező esetben amellett, hogy az $a, a_0 \in A$ párra $\delta(a, x) = \delta(a_0, x), x \in X$ fennáll, $\{\delta(a_0, p) \mid p \in X^+\} \cap \{a_0, a\} = \emptyset$ és $\{\delta(a, p) \mid p \in X^+\} \cap \{a_0, a\} = \emptyset$ is teljesül. Világos hogy ebben az esetben a $\delta''(b, x) = \delta(b, x), b \in A' \setminus \{a'\}, x \in X$ átmeneti függvénnyel definiált $\mathcal{B} = (A' \setminus \{a\}, a, X, \delta'')$ egy olyan minimális állapotszámú iniciális kimenő jel nélküli automata lesz, melynek $a \in A' \setminus \{a_0\}$ kezdő állapota pont φ_{a_0} -t indukálja.

Csak megemlítjük itt, hogy hasonló módszerrel minimalizálhatóak a Rabin-Scott (felismerő-) automaták is. Ezesetben az Aufenkamp-Hohn algoritmus első osztályozó lépésében két csoportot képzünk, mégpedig a végállapotok, illetve a többi állapot halmazát.

2.6 Automaták Ekvivalenciája, Gill Tétele

A közös X, Y halmazokkal bíró $\mathcal{A} = (A, X, Y, \delta, \gamma)$ és $\mathcal{B} = (B, X, Y, \delta', \gamma')$ automaták $a \in A$ és $b \in B$ állapotait egymással ekvivalens állapotoknak mondjuk (jelekben: $a \equiv b$), ha a és b az $\mathcal{A}$, illetve $\mathcal{B}$ automatában ugyanazt a leképezést indukálja, azaz $\varphi_{a, \mathcal{A}} = \varphi_{b, \mathcal{B}}$. Magukat az $\mathcal{A}$ és $\mathcal{B}$ automatákat ekvivalenseknek nevezzük, ha bármelyikük bármely állapotához van a másiknak ezzel az állapottal ekvivalens állapota, azaz $\mathcal{F}_{\mathcal{A}} = \mathcal{F}_{\mathcal{B}}$. Speciálisan, két iniciális automatát akkor mondunk ekvivalensnek, ha kezdő állapotaik ekvivalensek egymással.

Tétel 2.11 *Tetszőleges $\mathcal{A}$ automatával ekvivalens automaták M halmazában létezik olyan A -izomorfától eltekintve egyértelműen meghatározott $\mathcal{A}'$ automata, amely bármely M -beli automatának A -homomorf képe. $\mathcal{A}'$ -ként választható bármely M -beli automatahoz tartozó redukált automata.*

Mint már korábban láttuk, egy Moore-automata tekinthető speciális Mealy-automatának. Az elmélet kiépítése folyamán fontos szerepet tölt be Arthur Gill következő eredménye, mely azt igazolja, hogy információ átalakítás szempontjából a két automata fogalom ekvivalens.

Tétel 2.12 (Gill Tétele) *Bármely Mealy-automatához létezik vele ekvivalens Moore-automata. Emellett, ha a Mealy-automata véges, akkor a hozzá megkonstruált, vele ekvivalens Moore-automata is választható végesnek.*

Bizonyítás: Legyen $\mathcal{A} = (A, X, Y, \delta, \gamma)$ tetszőleges Mealy-automata. Egy vele ekvivalens Moore-automatát a következőképp konstruálhatunk meg:

Legyen $\mathcal{A}' = (A', X, Y, \delta', \gamma')$, ahol $A' = A \cup A \times X$, továbbá tetszőleges $a' \in A', x \in X$ párra

$$\delta'(a', x) = \begin{cases} (a', x) & \text{ha } a' \in A, \\ (\delta(a, x'), x) & \text{ha } a' = (a, x') \in A \times X, \end{cases}$$

$$\gamma'(a', x) = \begin{cases} \gamma(a, x) & \text{ha } a' \in A, \\ \gamma(\delta(a, x'), x) & \text{ha } a' = (a, x') \in A \times X, \end{cases}$$

Mutassuk meg, hogy minden $a \in A, p \in X^*$ esetén $\gamma(a, p) = \gamma'(a, p)$. Nyilván elegendő nem üres szavakra szorítkoznunk. Legyen tehát $x_1, \dots, x_n \in X$ és tekintsük az $a, a_1 = \delta(a, x_1), a_2 = \delta(a_1, x_2), \dots, a_{n-1} = \delta(a_{n-2}, x_{n-1})$ állapotokhoz az $y_1 = \gamma(a, x_1), y_2 = \gamma(a_1, x_2), \dots, y_n = \gamma(a_{n-1}, x_n)$ kimenő jeleket. Világos, hogy ekkor definíció szerint $\gamma(a, x_1 \dots x_n) = y_1 \dots y_n$. Másrészt, δ' és γ' definíciója alapján $\delta'(a, x_1) = (a, x_1), \delta'((a, x_1), x_2) = (\delta(a, x_1), x_2) = (a_1, x_2), \delta'((a_1, x_2), x_3) = (\delta(a_1, x_2), x_3) = (a_2, x_3)$, illetve $\gamma'(a, x_1) = \gamma(a, x_1) = y_1, \gamma'((a, x_1), x_2) = \gamma(\delta(a, x_1), x_2) = \gamma(a_1, x_2) = y_2, \gamma'((a_1, x_2), x_3) = \gamma(\delta(a_1, x_2), x_3) = \gamma(a_2, x_3) = y_3, \dots, \gamma'((a_{n-2}, x_{n-1}), x_n) = \gamma(\delta(a_{n-2}, x_{n-1}), x_n) = \gamma(a_{n-1}, x_n) = y_n$. Azt kaptuk tehát, hogy minden $a \in A$ -ra és $p \in X^*$ -ra $\gamma(a, p) = \gamma'(a, p)$. Így minden $\mathcal{A}$ -beli állapothoz létezik vele ekvivalens $\mathcal{A}'$ -beli állapot. Természetesen az is igaz, hogy $\mathcal{A}'$ minden A -ba eső állapotához van vele ekvivalens $\mathcal{A}$ -beli állapot. $\mathcal{A}$ és $\mathcal{A}'$ ekvivalenciájához azt kell még megmutatnunk, hogy az $\mathcal{A}'$ minden $A \times X$ -beli állapotához is van az $\mathcal{A}$ automatának vele ekvivalens állapota. Ehhez legyen $(a, x) \in A \times X$. Mutassuk meg, hogy $\mathcal{A}'$ ezen állapota ekvivalens az $\mathcal{A}$ automata $\delta(a, x)$ állapotával, azaz $\gamma'((a, x), p) = \gamma(\delta(a, x), p)$ minden $p \in X^*$ mellett fennáll. Nyilván most is elegendő nem üres szavakra szorítkoznunk. Legyen tehát $x_1, \dots, x_n \in X$.

Ekkor, az előbb kapott $\gamma(a, x_1 \dots x_n) = \gamma'(a, x_1 \dots x_n)$ egyenlőség fennállásának igazolásához hasonlóan bizonyítható, hogy $\gamma(a, xx_1 \dots x_n) = \gamma'(a, xx_1 \dots x_n)$. Viszont $\gamma(a, xx_1 \dots x_n) = \gamma(a, x)\gamma(\delta(a, x), x_1 \dots x_n)$ és $\gamma'(a, xx_1 \dots x_n) = \gamma(a, x)\gamma'(\delta'(a, x), x_1 \dots x_n)$ definíció szerint fennáll. δ' definíciója szerint $\delta'(a, x) = (a, x)$, így $\gamma'(a, xx_1 \dots x_n) = \gamma(a, x)\gamma'((a, x), x_1 \dots x_n)$. Másrészt γ' definíciója értelmében $\gamma'(a, x) = \gamma(a, x)$. Mivel két kimenő szó pontosan akkor egyezik meg, ha betűről-betűre megegyezik, $\gamma'(a, x) = \gamma(a, x)$ és $\gamma(a, xx_1 \dots x_n) = \gamma'(a, xx_1 \dots x_n)$ teljesülése ekkor $\gamma(a, xx_1 \dots x_n) = \gamma(a, x)\gamma(\delta(a, x), x_1 \dots x_n)$ és $\gamma'(a, xx_1 \dots x_n) = \gamma(a, x)\gamma'((a, x), x_1 \dots x_n)$ miatt azt jelenti, hogy $\gamma(\delta(a, x), x_1 \dots x_n) = \gamma'((a, x), x_1 \dots x_n)$. Ezzel kimutattuk, hogy az $\mathcal{A}$ automata $\delta(a, x)$ állapota ekvivalens az $\mathcal{A}'$ automata (a, x) állapotával. Tehát, valóban, az $\mathcal{A}$ automata ekvivalens az $\mathcal{A}'$ automatával.

A tételhez azt fogjuk még igazolni, hogy $\mathcal{A}'$ tekinthető Moore-automatának. Legyen $\mu : A' \rightarrow Y$ tetszőleges olyan leképezés, melyre minden $(a, x') \in A \times X$ esetén $\mu((a, x')) = \gamma(a, x)$. ($\mu(a)$ értéke $a \in A$ mellett tehát lényegtelen azon túlmenően, hogy egy rögzített Y -beli elemnek kell lennie.) Ekkor tetszőleges $a \in A, x \in X$ párra $\gamma'(a, x) = \gamma(a, x)$, ahol $(a, x) = \delta'(a, x)$. Mindemellett ha $(a, x') \in A \times X$, úgy minden $x \in X$ -re $\gamma'((a, x'), x) = \gamma(\delta(a, x'), x)$, ahol $(\delta(a, x'), x) = \delta'((a, x'), x)$. Tehát $\mu : A' \rightarrow Y$ speciális megválasztásával kaptuk, hogy minden $a \in A', x \in X$ párra $\gamma'(a, x) = \mu(\delta'(a, x))$. $\mathcal{A}'$ így valóban Moore-automata. (Érdekességgént megjegyezzük, hogy $\mu(a)$ értékére $a \in A$ esetén annyit kellett csak kikötni, hogy egy tetszőlegesen rögzített Y -beli elem legyen.)

Végül, ha $\mathcal{A}$ véges, akkor A, X, Y rendre véges halmazok. Ekkor viszont $A' = A \cup A \times X$ is véges, ami (X és Y végeessége mellett) azt jelenti, hogy $\mathcal{A}'$ véges. Ezzel a tételt teljesen bebizonyítottuk. $\square$

Megjegyzés 2.15 *Gill tétele lehetővé teszi, hogy bizonyos kérdésekben csupán Moore-automatákra szorítkozzunk, hiszen a tétel azt fejezi ki, hogy már Moore-automatákkal előállíthatók mindazok a leképezések, amelyek előállíthatók az általánosabb Mealy-automatákkal. Másrészt azt is meg kell jegyeznünk, hogy a Moore-automaták osztálya számos fontos konstrukcióra nézve nem zárt (állapothomomorf kép képzése, redukált automata meghatározása, stb.). Emiatt mégsem szorítkozhatunk minden vizsgálatnál csak a Moore-automaták osztályára. Végül megjegyezzük azt is, hogy hasonló konstrukcióval igazolható Gill tétele iniciális automatákra is.*

2.7 Reguláris Kifejezések

Legyen $X = \{x_1, \dots, x_n\}$ tetszőleges nem üres és véges halmaz. Ekkor az $\mathcal{L}_X = \{L \mid L \subseteq X^*\}$ halmazt X feletti nyelvek halmazának, elemeit X feletti nyelveknek nevezzük.

Speciális X feletti nyelvek:

- (a) *üres nyelv*: $\emptyset$ (üres halmaz);
- (b) *kezdő nyelv*: $\{\lambda\}$;
- (c) *alapnyelvek*: $\{x_1\}, \{x_2\}, \dots, \{x_n\}$;
- (d) *univerzális nyelv*: X^* .

Az üres nyelvet, a kezdő nyelvet, valamint az alapnyelveket *elemi nyelveknek* is hívjuk.

$\mathcal{L}_X$ -ben a következő alpműveleteket (*nyelvműveleteket*) definiáljuk:

(i) *összeadás*: $L_1, L_2 \in \mathcal{L}_X$ *összegén* értjük és $L_1 + L_2$ -vel jelöljük az $L_1 + L_2 = \{p \mid p \in L_1 \text{ vagy } p \in L_2\}$ nyelvet, mely nem más mint L_1 és L_2 halmazelméleti összege;

(ii) *szorzás*: $L_1, L_2 \in \mathcal{L}_X$ *szorzatán* értjük és $L_1 L_2$ -vel jelöljük az $L_1 L_2 = \{p_1 p_2 \mid p_1 \in L_1, p_2 \in L_2\}$ nyelvet;

(iii) *iteráció*: $L \in \mathcal{L}_X$ *iteráltján* vagy más néven *lezártján* értjük és L^* -al jelöljük az $L^* = \{p_1 p_2 \dots p_k \mid p_1, \dots, p_k \in L, k \geq 1\} \cup \{\lambda\}$ nyelvet. Másként kifejezve, $L \in \mathcal{L}_X$ iteráltja az $L^* = \sum_{i=0}^{\infty} L^i$, ahol $L^0 = \{\lambda\}, L^i = L L^{i-1}$. Az $\mathcal{L}_X = \{L \mid L \subseteq X^*\}$ halmazt a rajta értelmezett ezen három művelettel (összeadás, szorzás, iteráció) X feletti *nyelvalgebrának* hívjuk. A nyelvalgebrában érvényesek az absztrakt algebrából ismert, vagy azokhoz hasonló műveleti tulajdonságok:

- $L_1 + L_2 = L_2 + L_1$ (összeadás kommutativitása);
- $(L_1 + L_2) + L_3 = L_1 + (L_2 + L_3)$ (összeadás asszociativitása);
- $L + L = L$ (összeadás idempotenciája);
- $L + \emptyset = L$ (additív egységelem létezése);
- $(L_1 L_2) L_3 = L_1 (L_2 L_3)$ (szorzás asszociativitása);
- $L \{\lambda\} = \{\lambda\} L = L$ (multiplikatív egységelem létezése);
- $(L_1 + L_2) L_3 = L_1 L_3 + L_2 L_3$ (baloldali disztributivitás);
- $L_1 (L_2 + L_3) = L_1 L_2 + L_1 L_3$ (jobboldali disztributivitás);
- $\{\lambda\}^* = \{\lambda\}$;
- $\emptyset^* = \{\lambda\}$;
- $L^* = (L^k)^* (\sum_{i=0}^{k-1} L^i), k \geq 1$;
- $L L^* = L^* L$;
- $L^* = \{\lambda\} + L L^*$;

$$(L_1 + L_2)^* = (L_1^* L_2^*)^*;$$

Két kivétellel minden felsorolt azonosság egyszerű számítással adódik a definíciók alapján. Ezért csak ezt a két összefüggést fogjuk bizonyítani.

Mutassuk meg először, hogy $L^* = (L^k)^* (\sum_{i=0}^{k-1} L^i)$, $k \geq 1$. Ehhez azt kell belátnunk, hogy a baloldali halmaznak pontosan azok az elemei mint a jobboldalinak.

A baloldali halmaznak és a jobboldali halmaznak is eleme az üres szó, így csak az üres szótól különböző elemek vizsgálatára kell szorítkoznunk. A baloldal az üres szón kívül tartalmazza azokat a p szavakat, melyek előállnak valamely $n \geq 1$ -re $p = p_1 \cdots p_n$ alakban, ahol $p_1, \dots, p_n \in L$. Ha minden tényező az üres szó, akkor $p = \lambda$, s ezzel az esettel már nem kell tovább foglalkoznunk. Tehát feltehető, hogy van olyan $1 \leq i \leq n$, hogy $p_i \neq \lambda$, azaz $p \neq \lambda$. Legyen valamely $u \geq 0$ nemnegatív egészre $n = uk + v$, ahol $0 \leq v < k$. Ekkor $p = q_1 q_2$, ahol vagy $q_1 = \lambda$ (ekkor $u = 0$) vagy pedig $q_1 = p_1 \cdots p_{uk}$, ahol $p_1, \dots, p_{uk} \in L$ (ha $u \neq 0$), s ugyanekkor vagy $q_2 = \lambda$ (ekkor $v = 0$), vagy $q_2 = p_{uk+1} \cdots p_{uk+v}$, ahol $p_{uk+1}, \dots, p_{uk+v} \in L$ (ha $v \neq 0$). Mindenképp fennáll, hogy $q_1 \in (L^k)^*$, illetve $q_2 \in L^v$, $0 \leq v < k$, azaz $p = q_1 q_2 \in (L^k)^* L^v$. Tehát p eleme mindkét oldalnak, amivel a két oldal egyenlőségét kimutattuk.

Igazoljuk most az $(L_1 + L_2)^* = (L_1^* L_2^*)^*$ egyenlőséget. Itt is igaz, hogy a baloldali halmaznak és a jobboldali halmaznak is eleme az üres szó, így csak az üres szótól különböző elemek vizsgálatára kell szorítkoznunk. Hasonló a helyzet mint az előző esetben. Mindkét oldal az üres szón kívül pontosan azokat a p szavakat tartalmazza, melyek előállnak valamely $n \geq 1$ -re $p = p_1 \cdots p_n$ alakban, ahol $p_1, \dots, p_n \in L_1 + L_2$. A két halmaz tehát egyenlő.

Reguláris nyelvnek hívjuk az üres nyelvet, továbbá mindazon nyelveket, amelyek elemi nyelvekből az alapműveletek véges számú alkalmazásával előállíthatók. (Így például reguláris nyelvek: $\emptyset, \{\lambda\}, \{x_1\}, X^*, (\{x_1\}\{x_2\})^* + \{x_3\}$.) Később igazolni fogjuk, hogy a reguláris nyelv fogalmának másik jól ismert definíciója ezzel a fogalommal ekvivalens. *Reguláris kifejezésnek* hívjuk az $\emptyset$ -t, továbbá mindazon elemi nyelvekből, műveleti jelekből és zárójelekből álló véges láncokat, melyek megmutatják, hogy a reguláris nyelv hogyan áll elő az elemi nyelvek és az alapműveletek segítségével. A reguláris nyelv és a reguláris kifejezés általában nem egyértelműen meghatározott. (Például $L\{\lambda\} = \emptyset^* L = (\{\lambda\} + \{\lambda\})L = \dots$.) Ezért külön érdekes, hogy hogyan lehet eldönteni az $\mathcal{F}_X$ nyelvalgebrában, hogy két különböző reguláris kifejezés ugyanazt a reguláris nyelvet állítja-e elő. Később látni fogjuk, hogy ez a kérdés eldönthető. Sőt, ha pontosan megmondjuk, hogy mit is értünk egy reguláris nyelvet megadó minimális reguláris kifejezésen (például lehet ez alatt érteni (a karakterszámra) a legrövidebbet), akkor annak meghatározására is megadható algoritmus.

Az alapműveletekre fentebb felsorolt műveleti azonosságok lehetővé teszik, hogy alkalmazásukkal a reguláris nyelv reguláris kifejezését viszonylag egyszerűbb, vagy egyéb szempontból kedvezőbb (például rövidebb) alakra hozzuk. Megállapodunk abban, hogy ha a kijelölt műveletek sorrendjét műveleti zárójelek nem írják elő, akkor az alapműveleteket iteráció, szorzás, összeadás sorrendben végezzük el.

Végül megjegyezzük, hogy reguláris kifejezés megadásánál a legtöbb esetben az (egy elemű) elemi nyelveket az elemükkel jelöljük, s a továbbiakban mi is ezt a konvenciót fogjuk követni. (Például az $\{x\}^* + (\{x\} + \{\lambda\})\{y\}^*$ helyett a továbbiakban egyszerűen csak $x^* + (x + \lambda)y^*$ -t írunk.)

2.8 Automaták Analízise és Szintézise. Nyelvek Előállítása Automatákban

Az automaták és az automata leképezések kapcsolatát illetően két ellentétes kérdés fogalmazható meg:

I. Ha adva van egy iniciális automata, meghatározandó az általa indukált leképezés (analízis);

II. Ha adva van egy iniciális automata leképezés, meghatározandó legalább egy olyan iniciális automata, amely ezt a leképezést indukálja (szintézis).

A szintézis feladata nem egyértelmű ugyan, de egyértelművé tehető a minimalizálás feladatával. Az analízis és szintézis feladata csak akkor fogalmazható meg egzakt módon, ha megállapodunk abban, hogy mit értünk automata és automata leképezés megadásán. Véges automatát például táblázattal adunk meg (lásd 2.2. fejezet). Véges automata leképezések megadása általában problémát jelent a végtelen értelmezési tartomány miatt. Bizonyos esetekben viszont közönséges módon is lehetséges. Így például $X = \{x\}$, $Y = \{u, v\}$ mellett az $\lambda \rightarrow \lambda$, $x^n \rightarrow uv^{n-1}$, $n \geq 1$ kifejezésekkel nyilván egy automata leképezést adtunk meg. Az analízis és szintézis probléma ebben a megfogalmazásban túl általános. Ezért ezt a két feladatot csak véges automatákra fogalmazzuk meg pontosabban. Ezzel kapcsolatban egy további probléma is fellép:

0. Meg kell adnunk a véges automaták által indukálható automata leképezéseknek az automatáktól független leírását. Más szóval, keresnünk kell olyan írásmódot, amelyben el tudjuk dönteni, hogy valamely, ebben az írásmódban megadott automata leképezés indukálható-e véges automatával vagy sem. Ezek után véges automatákra az analízis és szintézis problémája így fogalmazható meg:

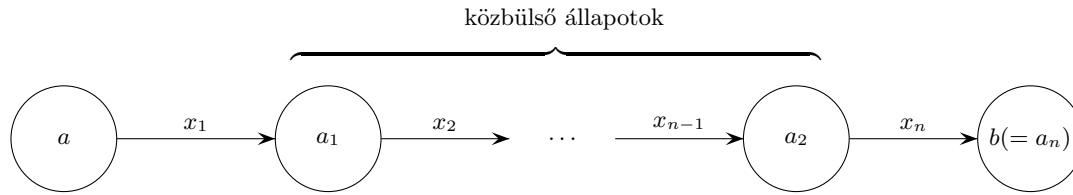
I'. Bármely adott véges automata esetén meg kell tudnunk adni a szóban forgó írásmódban azt a leképezést, amelyet az automata indukál;

II'. Ha ebben az írásmódban meg van adva egy véges automata által indukálható leképezés, akkor meg kell tudnunk adni legalább egy olyan automatát, mely ezt a leképezést indukálja és állapotthalmaza véges.

Egy olyan leképezés megadási módszer, mely eleget tesz a 0., I', II.' feltételeknek, először S. C. Kleene talált 1956-ban, bevezetve a reguláris kifejezés fogalmát.

A fejezet további részében kimenő jel nélküli automatákkal foglalkozunk. Az itt leírt eredmények lehetővé teszik, hogy az analízis és szintézis foglamát módosítsuk és e kettős feladat megoldását kimenő jel nélküli automaták vizsgálatára vezessük vissza. Amint láttuk, a generatív rendszerek egyik fontos típusát képezik a generatív nyelvtanok, melyek a formális nyelvek előállítására alkalmas eszközök. A formális nyelvek felismerésére viszont nem az analitikus nyelvtanokat, hanem kimenő jel nélküli automatákat szokás alkalmazni.

Tetszőleges $\mathcal{A} = (A, X, \delta)$ kimenő jel nélküli automata esetén egy $a \in A$, $p \in X^*$ pár esetén jelölje ap a $\delta(a, p)$ állapotszó utolsó betűjét. (Így $a\lambda = a$.) Azt mondjuk, hogy az $\mathcal{A}$ automata az $a \in A$ állapotból a $p = x_1 \cdots x_n$ (, ahol $x_1, \dots, x_n \in X$) szó hatására *átmegy* a $b \in B$ állapotba az $a_1, \dots, a_{n-1}$ *állapotsorozaton keresztül*, ha $\delta(a, p) = a_1 \cdots a_n$ és $a_n = b$. A *közbülső állapotok* ennél az átmenetnél az $a_1 = ax_1$, $a_2 = ax_1x_2$, $\dots$, $a_{n-1} = ax_1 \cdots x_{n-1}$ lesznek.



Azt mondjuk, hogy egy X feletti L nyelv az $\mathcal{A} = (A, a_0, X, \delta)$ kimenő jel nélküli automatában előállítható az A állapothalmaz A' részhalmazával, vagy az $\mathcal{A}$ automata az L nyelvet *felismeri* az A' halmazzal - jelekben: $L = L_{\mathcal{A}}^{A'}$ - ha L azokból és csak azokból a bemenő szavakból áll, amelyek hatására az a_0 kezdő állapotból A' -beli állapotba megy át, vagyis $p \in L \Leftrightarrow a_0 p \in A'$. Speciálisan, $L_{\mathcal{A}}^A = X^*$ és $L_{\mathcal{A}}^{\emptyset} = \emptyset$. Itt utalunk a 2.1. fejezetre, ahol a Rabin-Scott automatát leírtuk. Most tulajdonképpen ezekről a felismerő automatákról van szó, amelyekre a végállapotok halmaza éppen pl. az A' .

Tétel 2.13 (Kleene Tétele) *Bármely adott $\mathcal{A}$ véges kimenő jel nélküli iniciális automatához és állapotai M részhalmazához meg tudjuk adni annak a nyelvnek egy reguláris kifejezését, amely $\mathcal{A}$ -ban az M halmazzal előállítható. Megfordítva, minden reguláris kifejezéshez megadható egy $\mathcal{A}$ véges kimenő jel nélküli iniciális automata és állapotai M részhalmaza úgy, hogy a tekintett reguláris kifejezés $\mathcal{A}$ -ban az M halmazzal előállítható.* $\square$

Kleene tételének konstruktív bizonyítását fogjuk adni a 2.9. és 2.10. Fejezetekben.

Azt mondjuk, hogy az X^* monoidon (azaz egységelemes félcsoporton) definiált valamely ϱ ekvivalencia reláció *jobbkongruencia*, ha minden $p, q, r \in X^*$ hármásra igaz, hogy $p \varrho q \Rightarrow pr \varrho qr$. Az X^* monoid egy C_{ϱ} osztályozását *jobbról stabilnak* (vagy *jobbról kompatibilisnek*) hívjuk, ha a ϱ reláció jobb kongruencia. Végül, a ϱ relációt, illetve a C_{ϱ} osztályozást *véges indexűnek* mondjuk, ha véges sok osztályból áll. Analóg módon definiálható a balkongruencia és a balról kompatibilis osztályozás. Egy, az X^* monoidon megadott relációt *kongruenciának* hívunk, ha egyidejűleg bal- és jobbkongruencia. Másként kifejezve, az X^* monoidon értelmezett ϱ reláció kongruencia, ha bármely $p, q, r, s \in X^*$ -ra $p \varrho q \Rightarrow rps \varrho rqs$. (Megjegyezzük, hogy természetesen p, q, r, s bármelyike lehet az üres szó.) Egy ϱ kongruenciához tartozó C_{ϱ} osztályozást *kompatibilisnek* mondunk. Másként kifejezve, kompatibilisnek hívjuk az egyidejűleg balról és jobbról stabil osztályozást.

A véges automatákban előállítható nyelvek egy relációelméleti jellemzését adja Myhill és Neroda tétele.

Tétel 2.14 (Myhill és Neroda tétele) *Egy véges X ábécé feletti L nyelv akkor és csak akkor állítható elő egy véges kimenő jel nélküli iniciális automatában az állapotok valamely részhalmazával, ha megadható X^* -nak olyan véges indexű kompatibilis C osztályozása, hogy L előáll bizonyos C -beli osztályok egyesítési halmazaként.*

Bizonyítás: Mindenekelőtt megjegyezzük, hogy X^* -nak az a triviális osztályozása, amelynél minden elem egymagában alkot egy osztályt, nyilván kompatibilis és egy L nyelv mindig előáll ennél az osztályozásnál bizonyos osztályok egyesítési halmazaként. Ennél fogva egy tetszőleges L nyelv esetén mindig van olyan kompatibilis osztályozása X^* -nak, hogy az L előáll bizonyos osztályok egyesítési halmazaként.

A tétel szükségességének bizonyításához tegyük fel, hogy egy L nyelv előállítható az $\mathcal{A} = (A, a_0, X, \delta)$ kimenő jel nélküli véges automata állapotainak egy A' részhalmazában, vagyis az (A, a_0, X, δ, A') Rabin-Scott automata éppen az L nyelvet ismeri fel. Defináljunk X^* -on egy kongruenciát a következőképp:

$$\forall p, q \in X^* : (p \varrho_{\mathcal{A}} q \Leftrightarrow \forall a \in A : ap = aq).$$

(Vagyis minden $p, q \in X^*$ esetén $p \varrho_{\mathcal{A}} q$ akkor és csak akkor teljesüljön, ha tetszőleges A -beli állapotra a p szó az a állapotot ugyanabba az állapotba viszi át mint a q szó.) Ekkor a ϱ -hoz tartozó C_ρ kompatibilis osztályozásnál egy-egy osztályba pontosan azok az X^* -beli elemek tartoznak, melyek az $\mathcal{A}$ iniciális kimenő jel nélküli automata kezdő állapotát egy meghatározott állapotba viszik. Nyilvánvaló, hogy az $\mathcal{A}$ végeessége miatt az A halmaz véges, azaz a C_ρ kompatibilis osztályozás véges sok osztályt tartalmaz. Tehát C_ρ valóban véges indexű. Azt kell még kimutatnunk, hogy L előáll bizonyos C_ρ -beli osztályok egyesítési halmazaként. Jelölje most tetszőleges $a \in A$ mellett $C^{(a)}$ ezen C_ρ kompatibilis osztályozás azon osztályát, melynek elemei az a_0 kezdő állapotot az a állapotba viszik át. Mivel feltettük, hogy a $\mathcal{A}$ az L nyelvet állapotainak A' halmazával ismeri fel, ekkor $L = \cup \{C^{(a)} \mid a \in A'\}$. (Természetesen $L = \emptyset$ előfordulhat. Ilyen esetben $A' = \emptyset$.) Ezzel a tétel szükségességét kimutattuk.

Most igazoljuk az elegendőséget. Tegyük fel, hogy megadható X^* -nak olyan véges indexű kompatibilis C osztályozása, hogy L előáll bizonyos C -beli osztályok egyesítési halmazaként. Korábbi konvencionknak megfelelően tetszőleges $p \in X^*$ mellett $C[p]$ -vel fogjuk jelölni azt az osztályt, melynek egy adott $p \in X^*$ eleme. Ha $L = \emptyset$, akkor L bármely véges iniciális kimenőjel nélküli automatában előállítható az állapotok egy üres halmazával. Így a továbbiakban feltehetjük, hogy L nem üres. Mivel C kompatibilis, ez azt jelenti, hogy tetszőleges $p, q \in X^*$ mellett $C[p] = C[q] \Rightarrow \forall x \in X : C[px] = C[qx]$ (vagyis tetszőleges p, q^* esetén a p és a q egy osztályba esése azt eredményezi, hogy minden $x \in X$ esetén a px és a qx is egy osztályba fognak esni.) Ekkor viszont jól definiált (azaz az értelmezési tartomány minden elemének valóban pontosan egy elem felel meg az értékkészletben) az a $\delta : C \times X \rightarrow C$ leképezés, melyre minden $p \in X^*, x \in X$ pár mellett $\delta(C[p], x) = C[px]$. Tekintsük most az ezen δ átmeneti függvényvel rendelkező $\mathcal{A} = (C, C[\lambda], X, \delta)$ véges iniciális kimenő jel nélküli automatát (ahol $C[\lambda]$ jelöli a C azon osztályát, mely az üres szót tartalmazza). Jelölje C' a C azon részhalmazát, melyre teljesül, hogy az L előáll a C' -beli osztályok egyesítéseként. Az $\mathcal{A}$ definíciója értelmében minden $p \in X^*$ -ra $\delta(C[\lambda], p) = C[p]$. Világos, hogy ekkor minden $p \in X^*$ esetén $C[p] \in C'$ pontosan akkor áll fenn, ha $C[\lambda]$ -t a p szó az $\mathcal{A}$ automatában valamely C' -beli állapotába viszi át. Vagyis $\mathcal{A}$ állapotainak C' részhalmazával épp az L nyelvet ismeri fel. Ezzel a tétel bizonyítását befejeztük. $\square$

Valamely $\mathcal{A} = (A, a_0, X, \delta)$ kimenő jel nélküli (nem feltétlenül véges) automata esetén a $\forall p, q \in X^* : (p \varrho_{\mathcal{A}} q \Leftrightarrow \forall a \in A : ap = aq)$ -val definiált ϱ kongruenciát az $\mathcal{A}$

Myhill-Nerode féle kongruenciájának hívjuk, a hozzá tartozó X^*/ϱ fakorfélcsoportot pedig az $\mathcal{A}$ automata *karakterisztikus félcsoporthjának* nevezzük.

Az $\mathcal{A}$ karakterisztikus félcsoporthja nyilván úgy is megadható, mint az A állapothalmaz feletti teljes transzformáció félcsoporth azon részfélcsoporthja, amelyet az összes, az $\delta_x(a) = \delta(a, x)$, $a \in A$ összefüggésnek eleget tevő $\delta_x : A \rightarrow A$ leképezések $\{\delta_x \mid x \in X\}$ halmaza generál. Ilyen interpretációban az $S(\mathcal{A})$ elemei azok az $\eta_p : A \rightarrow A$, $p \in X^*$ alakú leképezések lesznek, melyekre $\eta_p(a) = ap$, $a \in A$.

Példa 2.16 Az $L = \{x^m y^n \mid m > n \geq 0\}$ nyelv véges kimenő jel nélküli iniciális automatában nem állítható elő. (x^n egy n hosszúságú, csupa x betűből álló szót jelöl.) Valóban, ha L véges kimenő jel nélküli iniciális automatában előállítható lenne, akkor a Myhill-Nerode tétel szerint lenne az X^* -nak véges indexű olyan C kompatibilis osztályozása, hogy L előáll véges sok osztály egyesítéseként. Mivel L végtelen elemű, s véges sok osztály egyesítéseként előáll, van olyan osztály, mely valamely $u - v > k - \ell > 0$ feltételnek eleget tevő u, v, k, ℓ természetes számokra $x^u y^v$ -t és $x^k y^\ell$ -t tartalmazza. Viszont $x^u y^{v+k-\ell} (= x^u y^v y^{k-\ell}) \in L$, s ugyanekkor $x^k y^k (= x^k y^\ell y^{k-\ell}) \notin L$. Vagyis van olyan $p \in X^*$, hogy $x^u y^v p \in L$ mellett $x^k y^\ell p \notin L$. Mivel feltételezésünk szerint L előáll (véges sok) C -beli osztály egyesítéseként, L -nek azok és csak azok az osztályok lehetnek a részhalmazai, melyek egyesítéseként L előáll. Ez viszont azt is jelenti, hogy $x^u y^v p$ és $x^k y^\ell p$ nem tartozhatnak a C osztályozás egy és ugyanazon osztályába. Tehát, feltételezésünknek ellentmondva a C nem kongruencia osztályozás. Ekkor viszont nem lehet az L nyelvet véges kimenő jel nélküli iniciális automatában előállítani.

2.9 Véges Automaták Analízise

Az automaták analízisének problémáját véges automatákra így módosítjuk:

A véges automaták analízise jelentse olyan univerzális algoritmus megadását, amelynek alkalmazásával bármely adott $\mathcal{A} = (A, a_0, X, \delta)$ véges kimenő jel nélküli iniciális automatahoz és állapotai M részhalmazához meg tudjuk adni annak a nyelvnek egy reguláris kifejezését, amely $\mathcal{A}$ -ban az M halmazzal előállítható. Ilyen algoritmus létezését S. C. Kleene bizonyította be 1956-ban. Mi a továbbiakban McNaughton és Yamada algoritmusát ismertetjük, amely egyben konstruktív bizonyítást is szolgáltatja a következő tételnek.

Tétel 2.15 Ha a véges X halmaz feletti L nyelv előállítható véges kimenő jel nélküli iniciális automata állapotai valamely M részhalmazával, akkor az L nyelv reguláris.

Bizonyítás: A kívánt algoritmus megadásához nyilván elég olyan eljárást megkonstruálni, amely lehetővé teszi az olyan nyelv egy reguláris kifejezésének felírását, amely $\mathcal{A}$ -ban tetszőleges, egyetlen állapottal előállítható, hisz ha $M = \{a_1, a_2, \dots, a_n\}$, akkor $L_{\mathcal{A}}^M = L_{\mathcal{A}}^{a_1} + L_{\mathcal{A}}^{a_2} + \dots + L_{\mathcal{A}}^{a_n} (= L_{\mathcal{A}}^{a_1} \cup L_{\mathcal{A}}^{a_2} \cup \dots \cup L_{\mathcal{A}}^{a_n})$. Így a továbbiakban csupán az olyan

reguláris nyelvek reguláris kifejezéssel történő megadását vizsgáljuk, melyek mindegyike $\mathcal{A}$ -ban egy-egy állapottal (azaz az állapotok egy-egy elemű részhalmazával) előállítható. Legyen $\mathcal{A} = (\{1, \dots, n\}, 1, X, \delta)$ egy véges iniciális kimenő jel nélküli automata, melyben az állapothalmaz 1-től alkalmas n -ig terjedő természetes számok halmaza, s az 1 természetes szám a kezdő állapot. Az állapothalmaz és a kezdő állapot ilyen megválasztása nem jelent igazi megszorítást, hisz tetszőleges véges iniciális kimenő jel nélküli automata esetén jelölhetjük 1-el a kezdő állapotot, s amennyiben az állapotok száma n , az $\{1, 2, \dots, n\}$ -el az állapothalmazt (aszerint hogy az állapotok egy tetszőleges, kezdő állapottal kezdődő felsorolásánál beszélünk első, második, ..., n -edik állapotról). Jelölje $L_{i,j}^k$ tetszőleges $i, j \in \{1, \dots, n\}$ -re és $k \in \{0, 1, \dots, n\}$ -re azt a nyelvet, mely azokból és csak azokból a bemenő szavakból áll, amelyek hatására az $\mathcal{A}$ az $i \in \{1, \dots, n\}$ állapotból átmegy a $j \in \{1, \dots, n\}$ állapotba úgy, hogy $k = 0$ esetén nincs közbülső állapot, $k > 0$ esetén pedig legfeljebb az $1, \dots, k$ állapotok léphetnek fel közbülső állapotként. Speciálisan, $L_{i,j}^0$ jelöli azt a nyelvet, amelynek elemei hatására $\mathcal{A}$ az $i \in \{1, \dots, n\}$ állapotból közvetlenül, közbülső állapotok nélkül megy át a $j \in \{1, \dots, n\}$ állapotba. (Itt tehát k nem kitevőt hanem egyszerűen indexet jelöl!)

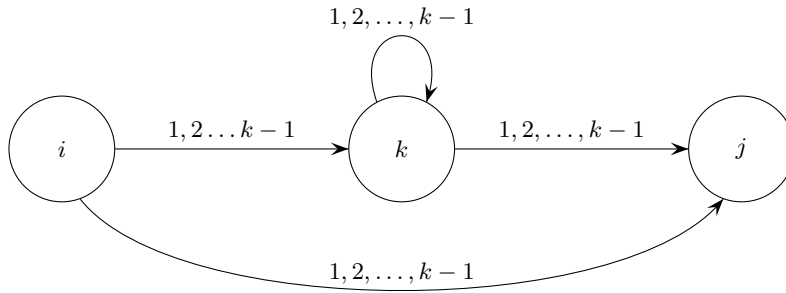
Elegendő megmutatni, hogy bármely $m \in \{1, \dots, n\}$ esetén az $L_{1,m}^n$ reguláris, hiszen $L_{\mathcal{A}}^m = L_{1,m}^n$, s elegendő ezen nyelvek egy reguláris kifejezését megadni. Ennél valamivel többet bizonyítunk: Igazolni fogjuk, hogy minden $L_{i,j}^k$ nyelv reguláris és rekurzív formulát adunk az ilyen nyelvek egy reguláris kifejezésének felírásához. Ez egyben azt is jelenti, hogy tételünknek egy teljes indukciós bizonyítását adjuk. Újra megjegyezzük itt, hogy a konvencióknak megfelelően egy reguláris kifejezésben az (egy elemű) elemi nyelveket az elemükkel jelöljük. (Például az $\{x_1\} + \{x_2\}^*$ reguláris kifejezés helyett egyszerűen csak $x_1 + x_2^*$ -t írunk.)

Legyen $k = 0$. Mivel az $L_{i,j}^k$ nyelvek mindegyike vagy az üres nyelv, vagy pedig X bizonyos elemeinek egyesítési halmaza, ezért minden ilyen nyelv reguláris és az automata átmeneti táblázata alapján felírható. Nevezetesen, a táblázat i állapothoz tartozó oszlopában kigyűjtjük azokat az állapotokat, melyek j -vel megegyeznek, s tekintjük az összes olyan $x_{i_1}, \dots, x_{i_s} \in X$ bemenő jeleket, melyek az $\mathcal{A}$ átmeneti táblázatában ezen j -vel megegyező elemekkel egy sorba esnek. Ezek a bemenő jelek lesznek azok, melyek hatására az $\mathcal{A}$ átmegy az i állapotából a j állapotába. Ha ilyen bemenő jel nincs és $i \neq j$, akkor $L_{i,j}^0 = \emptyset$ lesz a tekintett reguláris kifejezés. Amennyiben van olyan bemenő jel, mely az i állapotot a j állapotba viszi át és $i \neq j$, akkor $L_{i,j}^0 = x_{i_1} + x_{i_2} + \dots + x_{i_s}$ lesz a megfelelő reguláris kifejezés. Ha nincs az i -t a j -be átvivő bemenő jel, de $i = j$, akkor $L_{i,j}^0 = \lambda$, hisz az üres szó hatására minden állapot át tud menni önmagába közbülső állapot nélkül, definíció szerint. Végül, ha van olyan bemenő jel, mely az i állapotot a j állapotba viszi át és $i = j$, akkor nyilvánvalóan igaz, hogy $L_{i,j}^0 = \lambda + x_{i_1} + x_{i_2} + \dots + x_{i_s}$ megfelelő reguláris kifejezés.

Legyen most $k > 0$, s tegyük fel, hogy az $L_{i,j}^0, \dots, L_{i,j}^{k-1}, i, j \in \{1, \dots, n\}$ nyelvek regularitását már bebizonyítottuk és fel is írtuk ezen nyelvek egy-egy reguláris kifejezését. Mutassuk meg, hogy érvényes az

$$L_{i,j}^k = L_{i,j}^{k-1} + L_{i,k}^{k-1} (L_{k,k}^{k-1})^* L_{k,j}^{k-1} (i, j, k \in \{1, \dots, n\})$$

formula. Az egyenlőség igazolásához először azt gondoljuk végig, hogy milyen módon tudunk egy $p \in X^*$ szóval eljutni az i állapotból a j állapotba úgy, hogy legfeljebb $1, \dots, k$ léphetnek fel közbülső állapotként.



Ha valamely $p \in X^*$ szóval az i állapotból a j állapotba el tudunk jutni úgy, hogy legfeljebb az $1, 2, \dots, k$ léphetnek fel közbülső állapotokként, akkor vagy el tudunk jutni ezen p szóval az i állapotból a j állapotba úgy, hogy legfeljebb $1, \dots, k-1$ léphet fel közbülső állapotként (azaz ha $k = 1$ akkor ezesetben nem lép fel közbülső állapot) és ekkor $p \in L_{i,j}^{k-1}$, vagy nem. Utóbbi esetben fel kell lépnie a k állapotnak legalább egyszer, de persze véges sokszor közbülső állapotként. Vagyis ekkor először eljutunk a k állapotba úgy, hogy legfeljebb $1, \dots, k-1$ léphet fel közbülső állapotként, s ha k csak egyszer lépett fel közbülső állapotként, akkor k -ból máris tovább juthatunk j -be úgy, hogy ismét legfeljebb $1, \dots, k-1$ léphet fel közbülső állapotként. Ezesetben tehát $p = p_1 p_2$, ahol $p_1 \in L_{i,k}^{k-1}$, $p_2 \in L_{k,j}^{k-1}$. Az az eset van még hátra, mikor k egynél többször (de persze véges sokszor) lép fel közbülső állapotként. Ekkor is először eljutunk a k állapotba úgy, hogy legfeljebb $1, \dots, k-1$ léphet fel közbülső állapotként, majd véges sokszor k -ból elindulva visszatérünk k -ba úgy, hogy ismét legfeljebb $1, \dots, k-1$ léphessenek fel közbülső állapotként, s végül mint az előző esetben, k -ból tovább juthatunk j -be úgy, hogy ismét legfeljebb $1, \dots, k-1$ léphet fel közbülső állapotként. Ilyenkor p előáll valamely $t > 2$ természetes számra $p = p_1 p_2 \dots p_{t-1} p_t$ alakban úgy, hogy $p_1 \in L_{i,k}^{k-1}$, $p_2, \dots, p_{t-1} \in L_{k,k}^{k-1}$, illetve $p_t \in L_{k,j}^{k-1}$. Minden lehetséges esetben azt kaptuk, hogy ha p benne van az egyenlőségünk baloldalán szereplő nyelvben, akkor benne van az egyenlőség jobboldalán álló nyelvben. Az egyenlőség fennállásához be kell még látni az is, hogy ha $p \in X^*$ benne van az egyenlőség jobboldalán lévő nyelvben akkor benne van a baloldalán lévőben is. $L_{i,j}^{k-1} \subseteq L_{i,j}^k$ definíció szerint fennáll, így csak azzal az esettel kell foglalkoznunk, mikor $p \in L_{i,k}^{k-1} (L_{k,k}^{k-1})^* L_{k,j}^{k-1}$. Ekkor $p = p_1 \dots p_s$, $s \geq 1$, ahol p_1 az i állapotot a j állapotba, p_s a k állapotot a j állapotba, s ha $s > 2$, akkor minden egyes p_ℓ , $\ell \in \{2, \dots, s-1\}$ tényező a k állapotból önmagába viszi át az $\mathcal{A}$ automatát úgy, hogy legfeljebb $1, \dots, k-1$ léphetnek fel közbülső állapotként. Világos, hogy ekkor p hatására úgy juthatunk el az i állapotból a j állapotba, hogy legfeljebb $1, \dots, k$ léphetnek fel közbülső állapotként. Ezzel a tétel teljes bizonyítást nyert. $\square$

2.10 Végés Automaták Szintézise

Az automaták szintézisének problémáját végés automatákra a következőképp módosítjuk: A végés automaták szintézise jelentse olyan univerzális algoritmus megadását, amelynek alkalmazásával bármely végés X halmaz feletti, reguláris kifejezéssel megadott L reguláris nyelvhez meg tudunk konstruálni olyan végés kimenő jel nélküli iniciális $\mathcal{A} = (A, a_0, X, \delta)$ automatát, amelyben az adott L nyelv az állapothalmaz valamely M részhalmazával előállítható (, vagyis legyen az (A, a_0, X, δ, M) az L reguláris nyelv felismerő automatája). Egy ilyen algoritmus megadása egyben a következő tétel konstruktív bizonyítását szolgáltatja:

Tétel 2.16 *Egy végés X halmaz feletti tetszőleges reguláris nyelv előállítható végés kimenő jel nélküli iniciális automatában az állapothalmaz valamely részhalmazával.*

Bizonyítás: A tétel első bizonyítása ugyancsak S. C. Kleene eredménye 1956-ból. Mi egy V. M. Gluskovtól származó egyszerű eljárást ismertetünk, amely eléggé gazdaságos is abban az értelemben, hogy az esetek többségében a minimálisához közeli automatát eredményez. Másrészt lehetővé teszi, hogy az adott végés sok, reguláris kifejezéssel megadott reguláris nyelv esetén egyszerre szerkesszünk meg egy olyan végés automatát, amelyben a nyelvek mindegyike előállítható (általában az állapothalmaz más-más részhalmazával).

A módszer lényege, hogy számbavesszük azoknak a szavaknak a struktúráját, amelyek az adott nyelvek közül legalább egyikben előfordulnak. Az eljárás ismertetése során a könnyebb megértés kedvéért egy példát is kidolgozunk. Legyenek $L_1, L_2, \dots, L_k$ olyan nyelvek az $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ ábécé felett (az $1, 2, \dots, n$ itt nem kitevőt, hanem indexet jelentenek), amelyek reguláris kifejezésekkel vannak megadva. A módszer alkalmazása előtt a nyelvgebra alapműveleteire megismert műveleti azonosságok alkalmazásával hozzuk a reguláris kifejezéseket minél rövidebb alakra. Ezt követően indexeljük a reguláris kifejezésekben előforduló betűket balról jobbra haladva úgy, hogy az azonos betűk különböző előfordulási helyükön más-más indexet kapjanak.

Példa 2.17 *Legyenek az előállítandó nyelvek*

$$L_1 = xx^*, L_2 = yy^*, L_3 = (xx^*y + yy^*x)(x + y)^*.$$

Ekkor egy lehetséges indexelés eredménye:

$$L_1 = x_1x_2^*, L_2 = y_3y_4^*, L_3 = (x_5x_6^*y_7 + y_8y_9^*x_{10})(x_{11} + y_{12})^*.$$

Legyenek u és v indexelt betűk. Azt mondjuk, hogy az u indexelt betű megelőzi a v indexelt betűt, jelekben: $u < v$, ha az u index nélküli u' és v index nélküli v' formájához létezik olyan p szó, mely benne van legalább egy adott $L_i (1 \leq i \leq k)$ nyelvben és előáll $p = wu'v'z$ alakban valamely $w, z \in X^*$ -ra. Az u indexelt betűt kezdő betűnek nevezzük, ha index nélkül formája első betűje legalább egy olyan szónak, amely legalább egy $L_i (1 \leq i \leq k)$ -ben benne van. Végül, az u indexelt bezűt záró betűnek hívunk, ha index nélküli formája utolsó betűje olyan szónak, mely legalább egy $L_i (1 \leq i \leq k)$ -ben benne van.

Példa 2.18 *Példánkban a kezdő betűk: x_1, y_3, x_5, y_8 .*

Záró betűk: $x_1, x_2, y_3, y_4, y_7, x_{10}, x_{11}, y_{12}$.

Emellett:

$$\begin{array}{ll} x_1 \prec x_2; & y_7 \prec x_{11}, y_{12}; \\ x_2 \prec x_2; & y_8 \prec y_9, x_{10}; \\ y_3 \prec y_4; & y_9 \prec y_9, x_{10}; \\ y_4 \prec y_4; & x_{10} \prec x_{11}, y_{12}; \\ x_5 \prec x_6, y_7; & x_{11} \prec x_{11}, y_{12}; \\ x_6 \prec x_6, y_7; & y_{12} \prec x_{11}, y_{12}; \end{array}$$

Ezek után a keresett véges kimenő jel nélküli iniciális $\mathcal{A} = (A, a_0, X, \delta)$ automatát a következő módon definiáljuk:

Legyen a_0 tetszőleges szimbólum. Legyen ezután $\delta(a_0, x^{(i)})$ minden $i = 1, \dots, n$ -re az $x^{(i)}$ betű azon indexelt változatainak azon $a_1^{(i)}$ halmaza, melyek kezdő betűk. Így módon definiáljuk először az $a_1^{(1)}, a_1^{(2)}, \dots, a_1^{(n)}$ halmazokat mint $\mathcal{A}$ -beli állapotokat. Megjegyezzük, hogy ha valamelyik X -beli $x^{(i)}$ betű egyik indexelt változata sem kezdő betű, akkor $a^{(i)}$ az üres halmaz.

A többi állapotot rekurzióval definiáljuk a következő módon: Ha már egy a állapot definiálva van, akkor minden $i = 1, \dots, n$ -re a $\delta(a, x^{(i)})$ állapot legyen az $x^{(i)}$ betű azon indexelt változatainak halmaza, melyeket megelőz legalább egy a -beli betű. (Ez az a -beli indexelt betű természetesen nem szükségképp az $x^{(i)}$ betű indexelt változata.) Természetesen előfordulhat az is, hogy az $x^{(i)}$ betűnek nincs egyetlen ilyen indexelt változata sem, s ekkor $\delta(a, x^{(i)})$ az üres halmaz. Az is előfordulhat, hogy a maga is az üres halmaz. Ezesetben $\delta(a, x^{(i)}) = a$ fog fennállni a konstrukciónk értelmében.

Világos, hogy az így definiált $\mathcal{A}$ iniciális kimenő jel nélküli automata iniciálisan összefüggő, azaz a kezdő állapotából alkalmas bemenő szavakkal bármely állapota elérhető. Az is világos, hogy véges, hisz állapotainak száma legfeljebb annyi mint a bemenő ábécé összes részhalmazainak halmaza, azaz $2^n + 1$. Végül, konstrukciónk alapján minden $p \in L_i, i = 1, \dots, k$ nem üres szó esetén a p szó az így megkonstruált automatát a kezdő állapotból egy olyan állapotba viszi át, melynek legalább egy indexelt betűje az L_i nyelv záróbetűje. Ugyancsak konstrukciónk alapján látható, hogy ha $q \notin L_i$ akkor ez nem áll fenn. Tehát minden $i = 1, \dots, k$ -ra az $L_i \setminus \{\lambda\}$ nyelv előállítható az $\mathcal{A}$ automatában azzal az A_i halmazzal, amely azokból az állapotokból áll, melyek legalább egy L_i -beli záró betűt tartalmaznak. Amennyiben az üres szó eleme L_i -nek, akkor A_i -hez még hozzá kell venni az a_0 kezdő állapotot is. Példánkban az $\mathcal{A}$ kimenő jel nélküli iniciális automata megszerkesztése így alakul:

$$\begin{array}{ll} \delta(a_0, x) = \{x_1, x_5\} = a_1, & \delta(a_4, y) = \{y_{12}\} = a_8, \\ \delta(a_0, y) = \{y_3, y_8\} = a_2, & \delta(a_5, x) = \{x_{11}\} = a_7, \\ \delta(a_1, x) = \{x_2, x_6\} = a_3, & \delta(a_5, y) = \{y_{12}\} = a_8, \\ \delta(a_1, y) = \{y_7\} = a_4, & \delta(a_6, x) = \{x_{11}\} = a_7, \\ \delta(a_2, x) = \{x_{10}\} = a_5, & \delta(a_6, y) = \{y_{12}\} = a_8, \\ \delta(a_2, y) = \{y_4, y_9\} = a_6, & \delta(a_7, x) = \{x_{11}\} = a_7, \\ \delta(a_3, x) = \{x_2, x_6\} = a_3, & \delta(a_7, y) = \{y_{12}\} = a_8, \\ \delta(a_3, y) = \{y_7\} = a_4, & \delta(a_8, x) = \{x_{11}\} = a_7, \\ \delta(a_4, x) = \{x_{11}\} = a_7, & \delta(a_4, y) = \{y_{12}\} = a_8. \end{array}$$

A kapott automata átmenet táblázata:

$\mathcal{A}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x	a_1	a_3	a_5	a_3	a_7	a_7	a_5	a_7	a_7
y	a_2	a_4	a_6	a_4	a_8	a_8	a_6	a_8	a_8

Ebben az automatában előállítható
 az L_1 nyelv az $A_1 = \{a_1, a_3\}$ halmazzal,
 az L_2 nyelv az $A_2 = \{a_2, a_6\}$ halmazzal,
 az L_3 nyelv az $A_3 = \{a_4, a_5, a_7, a_8\}$ halmazzal.

2.11 Végés Automaták Által Indukálható Leképezések

Érvényes a következő állítás.

Észrevétel 2.19 *Tetszőleges $\varphi : X^* \rightarrow Y^*$ automata leképezés esetén egy $y \in Y$ akkor és csak akkor lesz valamely $p \in X^*$ -ra a $\varphi(p)$ képszó valamely betűje, ha van olyan $q \in X^*$, hogy y a $\varphi(q)$ képszó utolsó betűje.*

Bizonyítás: Legyen $\varphi : X^* \rightarrow Y^*$ tetszőleges automata leképezés, s tételezzük fel, hogy valamely $p \in X^*$ -ra az $y \in Y$ betű előfordul a $\varphi(p)$ képszóban. Ekkor $\varphi(p)$ előáll $\varphi(p) = wyz$ alakban, ahol $w, z \in Y^*$. Mivel φ hossztartó, $|p| = |\varphi(p)|$. Így p előáll $p = uxv$ alakban, ahol $|u| = |w|$, $x \in X$ és $|v| = |z|$. Ráadásul a hossztartó tulajdonság miatt $\varphi(ux)$ és wy hossza is szükségképp megegyezik. Másrészt φ kezdőszelet tartó, azaz $\varphi(ux) = wy$. Ha tehát $y \in Y$ előfordul egy képszóban, akkor előfordul egy (esetleg másik) képszó utolsó betűjeként is. $\square$

A továbbiakban olyan $\varphi : X^* \rightarrow Y^*$ automata leképezésekre szorítkozunk, ahol minden $y \in Y$ előfordul legalább egy $p \in X^*$ szó képszavában. 2.19. Észrevételünk értelmében ekkor minden $y \in Y$ elő fog fordulni legalább egy $p \in X^*$ szó képszavának utolsó betűjeként.

Legyenek X és Y végés halmazok és legyen $\varphi : X^* \rightarrow Y^*$ tetszőleges (nem feltétlen végés állapotú) automata leképezés. Megmutatjuk, hogy φ jellemezhető nyelvek egy alkalmas rendszerével. Tetszőleges $y \in Y$ -ra legyen $L_y = \{p \in X^* \mid \overline{\varphi(p)} = y\}$. Más szóval, az L_y legyen azoknak az X^* -beli p szavaknak a halmaza, amelyek a φ leképezésnél az y -ra végződő szóba esnek át. Tekintsük az ilyen L_y nyelvek halmazát: $H_\varphi = \{L_y \mid y \in Y\}$. H_φ eleget tesz a következő feltételeknek:

- (a) H_φ végés sok elemből áll.
- (b) $\forall y, y' \in Y : y \neq y' \Rightarrow L_y \cap L_{y'} = \emptyset$.
- (c) Minden nem üres $p \in X^*$ -hoz van olyan $y \in Y$, hogy $p \in L_y$.
- (d) A λ üres szó nincs benne egyik L_y -ban sem.

Az (a)-(d) feltételek jelentése az, hogy az $L_y (y \in Y)$ nyelvek az $X^* \setminus \{\lambda\}$ halmaz egy végés indexű osztályozását alkotják. Az (a)-(d) feltételeknek eleget tevő nyelvrendszereket *automata nyelvrendszereknek* nevezzük.

Tétel 2.17 *Ha az X és Y véges halmazok, akkor minden $\varphi : X^* \rightarrow Y^*$ automata leképezéshez hozzárendelhető az X feletti nyelvvalgebra egy H_φ automata nyelvrendszere. Megfordítva, a véges X ábécé feletti nyelvvalgebra bármely H automata nyelvrendszere a véges Y halmaz elemeinek jelölésétől eltekintve egyértelműen meghatározza azt a $\varphi : X^* \rightarrow Y^*$ automata leképezést, melyre $H_\varphi = H$.*

Bizonyítás: A tétel első felét már beláttuk. Második felének bizonyításához legyen H az $\mathcal{L}_Y$ nyelvvalgebra egy tetszőleges automata nyelvrendszere és legyen $H = \{H_1, \dots, H_n\}$. Legyen továbbá $Y = \{1, \dots, n\}$ és határozzuk meg azt a $\varphi : X^* \rightarrow Y^*$ automata leképezést, melyre tetszőleges $p = x_1 \cdots x_k \in X^*$, $x_1, \dots, x_k \in X$ szó esetén $\varphi(p) = i_1 \cdots i_k$, ahol $x_1 \in H_{i_1}, x_1 x_2 \in H_{i_2}, \dots, x_1 \cdots x_k \in H_{i_k}$. Ez a φ leképezés nyilván automata leképezés és rá $H_\varphi = H$ teljesül. $\square$

A tétel azt fejezi ki, hogy a véges X és Y halmazokhoz tartozó $\varphi : X^* \rightarrow Y^*$ alakú automata leképezések az Y elemeinek jelölésétől eltekintve egy-egy értelmű módon jellemezhetők nyelvek bizonyos rendszerével. Így valóban eljutottunk az automata leképezések olyan megadási módjához, mely a korábban kirótt feltételeinket teljesíti.

Egy $\mathcal{A}$ automatához tartozó $K_{\mathcal{A}}$ kanonikus nyelvrendszeren az $\mathcal{A}$ automata által indukált leképezéshez tartozó automata nyelvrendszert értjük. Képletben: $K_{\mathcal{A}} = H_{\varphi_{\mathcal{A}}}$. (Itt feltesszük, hogy $\mathcal{A}$ iniciális automata, melynek bemenő és kimenő jelhalmaza véges.) A definícióból közvetlenül adódik:

Tétel 2.18 *Egy véges X és Y halmazokkal rendelkező $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ automata akkor és csak akkor indukálja a $\varphi : X^* \rightarrow Y^*$ automata leképezést, ha a φ -hez tartozó automata nyelvrendszer egybeesik az $\mathcal{A}$ -hoz tartozó kanonikus nyelvrendszerrel.* $\square$

Ebben a fejezetben eddig végig feltételeztük, hogy az X és Y halmazok végesek. Most tovább szűkítjük vizsgálataink körét és az állapothalmaz végeességét is kikötjük. Más szóval, a továbbiakban csupán véges automatákkal foglalkozunk. Arra vonatkozóan, hogy milyen leképezések indukálhatóak véges automatákkal, érvényes a következő tétel.

Tétel 2.19 (Véges Automaták Alaptétele) *A véges X és Y halmazok esetén egy $\varphi : X^* \rightarrow Y^*$ automata leképezés akkor és csak akkor indukálható véges automatában, ha a φ -hez tartozó H_φ automata nyelvrendszer minden eleme reguláris nyelv.*

Bizonyítás: A szükségesség igazolásához tegyük fel, hogy X és Y véges halmazok, s a $\varphi : X^* \rightarrow Y^*$ automata leképezés indukálható a véges $\mathcal{A} = (A, a_0, X, Y, \delta, \gamma)$ automatával. Akkor Gill tétele szerint van olyan véges $\mathcal{A}' = (A', a_0, X, Y, \delta', \mu)$ Moore-automata is, mely ugyancsak indukálja a φ leképezést. Minden $y \in Y$ -ra legyen $A'_y = \{a' \in A' \mid \mu(a') = y\}$, azaz legyen A'_y az $\mathcal{A}'$ automata y állapotjelű állapotainak halmaza és jelölje L'_y azt a nyelvet, mely az $\mathcal{A}'' = (A', a_0, X, \delta')$ kimenő jel nélküli automatában az A'_y halmazzal előállítható, azaz legyen $L'_y = L_{\mathcal{A}''}^{A'_y}$. Mivel az $\mathcal{A}$ véges, az analízisre vonatkozó tétel alkalmazásával kapjuk, hogy L'_y minden $y \in Y$ -ra reguláris nyelv. Ezért a szükségességhez azt kell csupán megmutatni, hogy ha $H_\varphi = \{L_y \mid y \in Y\}$, akkor minden $y \in Y$ -ra $L_y = L'_y$. Ez pedig a következő számolással adódik:

$p \in L_y \Leftrightarrow \overline{\varphi(p)} = y \Leftrightarrow \overline{\varphi_{\mathcal{A}}(p)} = y \Leftrightarrow \varphi_{\mathcal{A}'}(p) = y \Leftrightarrow \overline{\mu(\delta(a_0, p))} = y \Leftrightarrow \overline{\delta(a_0, p)} \in A'_y \Leftrightarrow p \in L'_y$. Ezzel a szükségesség igazolását befejeztük.

Az elegendőség kimutatásához tegyük fel, hogy a véges X és Y halmazokra megadott $\varphi : X^* \rightarrow Y^*$ automata leképezés $H_\varphi = \{L_{y_1}, \dots, L_{y_m}\}$ automata nyelvrendszeréhez létezik olyan $\mathcal{A}'' = (A, a_0, X, \delta)$ iniciális kimenő jel nélküli automata, amelyben az $L_{y_1}, \dots, L_{y_m}$ nyelvek rendre előállíthatók az A állapothalmaz $A_1, \dots, A_m$ részhalmazzaival. Bővítsük ki az $\mathcal{A}'' = (A, a_0, X, \delta)$ automatát az $\mathcal{A} = (A, a_0, X, Y, \delta, \mu)$ kimenő jellel bíró Moore-automatává a μ jelfüggvény következő definíciójával: Legyen $\mu(a_0)$ egy tetszőlegesen rögzített Y -beli elem, továbbá legyen minden $a \in A_i, i \in \{1, \dots, m\}$ mellett $\mu(a) = y_i$. Minthogy H_φ automata nyelvrendszer, azért az $A_1, \dots, A_m$ páronként idegen halmazok. Így μ definíciója egyértelmű. Megmutatjuk, hogy az $\mathcal{A}$ Moore-automata indukálja a φ leképezést. Ehhez az előzőek alapján elegendő kimutatni, hogy a $K_{\mathcal{A}} = \{R_{y_i} \mid y_i \in Y\}$ kanonikus nyelvrendszer egybeesik a H_φ automata nyelvrendszerrel. Ez pedig a következő számolással adódik: $p \in R_{y_i} \Leftrightarrow \overline{\varphi_{\mathcal{A}}(p)} = y_i \Leftrightarrow \overline{\mu(\delta(a_0, p))} = y_i \Leftrightarrow \overline{\delta(a_0, p)} \in A_i \Leftrightarrow p \in L_{y_i}$. Ezzel az elegendőség igazolását is befejeztük. $\square$

Az analízisre és a szintézisre vonatkozó tétel a most bebizonyított tétellel együtt - konstruktív bizonyításukat konkrét esetre alkalmazva - algoritmust szolgáltat az analízis és a szintézis eredeti értelemben vett megoldására.

Példa 2.20 Legyen $X = \{x, y\}$ és az $X^* \setminus \{e\}$ félcsoporthat bontsuk fel a következő módon:

- L_1 legyen az összes, csupa x jelből álló szavak halmaza,
- L_2 legyen az összes, csupa y jelből álló szavak halmaza,
- L_3 legyen az összes többi szavak halmaza.

Ez a nyelvrendszer automata nyelvrendszer, mégpedig minden eleme reguláris nyelv, hiszen nyilván $L_1 = xx^*, L_2 = yy^*, L_3 = (xx^*y + yy^*x)(x + y)^*$. Vegyük észre, hogy a véges automaták szintézisének tárgyalt Gluskov módszert épp erre a három nyelvre alkalmaztuk, így az ott kapott 9 állapotú $\mathcal{A}$ automata játszhatja az alaptétel elegendősége bizonyítása során $\mathcal{A}''$ -vel jelölt kimenő jel nélküli automata szerepét. Ha most ezt az automatát kimenő jellel bíró automatává egészítjük ki úgy hogy az L_1, L_2, L_3 nyelvekhez rendre hozzárendeljük az u, v, w szimbólumokat, s a jelfüggvényt az alaptétel elegendősége bizonyításánál látott módon definiáljuk, akkor az alábbi Moore-automatához jutunk, mely az L_1, L_2, L_3 nyelvrendszerrel megadott leképezést indukálja. Az Aufenkamp-Hohn algoritmussal meghatározható az ezen leképezést indukáló minimális (négy állapotú) automata. Ennek ismertetését mellőzzük.

$\mathcal{A}$	tetsz.	u	v	u	w	w	v	w	w
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
x	a_1	a_3	a_5	a_3	a_7	a_7	a_5	a_7	a_7
y	a_2	a_4	a_6	a_4	a_8	a_8	a_6	a_8	a_8

Eredményeink alapján algoritmust tudunk konstruálni annak eldöntésére is, hogy két reguláris kifejezés ekvivalens-e, azaz egy és ugyanazon reguláris nyelv regulá-

ris kifejezéseiről van-e szó. Gluskov módszerével ugyanis mindkettőhöz meg tudunk szerkeszteni egy-egy iniciális kimenő jel nélküli automatát, melyek állapotaik alkalmas részhalmazával ezeket a nyelveket felismerik. Mindkét automatát átalakíthatjuk Moore-automatává úgy, hogy az állapothalmaz elemeihez két fajta állapotjelet rendelünk hozzá: az egyik fajta állapotjelet azon állapotokhoz, melyek az állapotoknak a reguláris kifejezést felismerő részhalmazához tartoznak, a másikat pedig a többi, ezen részhalmazhoz nem tartozó állapotokhoz. Mindkét automatára hajtsuk végre az Aufenkamp-Hohn féle minimalizációs algoritmus Moore-féle automatákra vonatkozó változatát. (Eljárhatunk úgy is, hogy mindkét automatát Mealy-automatának tekintjük és alkalmazzuk mindkettőre az Aufenkamp-Hohn algoritmus Mealy-automatákra vonatkozó változatát. A lényeg az, hogy mindkét automatára az algoritmus egy és ugyanazon automata típusokra tekintett változatát alkalmazzuk.) Ezután már csak azt kell eldöntenünk, hogy a kapott két (Moore-féle, vagy ha az Aufenkamp-Hohn algoritmus Mealy-automatákra tekintett változatát vettük, akkor a kapott két Mealy-féle) iniciális automata állapot-izomorf-e egymással úgy, hogy egy alkalmas izomorfizmus a kezdő állapotokat egymáshoz rendeli. Véges automatákra ez nyilván eldönthető. Ha igen a válasz, akkor a két reguláris kifejezés ekvivalens, különben nem.

A fentiek alapján algoritmus adható egy reguláris kifejezéssel megadott reguláris nyelvhez a legrövidebb vele ekvivalens reguláris kifejezés meghatározásához is: Végig kell vizsgálni az összes, a mi reguláris kifejezésünknel nem hosszabb olyan reguláris kifejezéseket, melyek összes változói legalább egyszer előfordulnak az adott reguláris kifejezésben. A vizsgálat során ki kell választani azokat a reguláris kifejezéseket, melyek ekvivalensek az adott reguláris kifejezéssel, s a legrövidebbet (vagy ha több ilyen van akkor az egyik legrövidebbet) meg kell tartanunk.

Cél lehet egy adott reguláris kifejezéshez megtalálni egy vele ekvivalens olyan reguláris kifejezést, melyben változók a lehető legkevesebbszer fordulnak elő. Nyilván csak azok a reguláris kifejezések jöhetnek számításba, melyekben a változók előfordulási száma nem nagyobb, mint az adott reguláris kifejezésben. Ha ez a szám és a zárójel-párok száma ismert, akkor felső korlátot tudunk adni a vizsgálandó reguláris kifejezések hosszára. Nevezetesen, legfeljebb annyi iterációs jelet tartalmazhat, mint a zárójel-párok száma plusz a változók előfordulási száma. Az összeadásjelek és a szorzások számának összege ennél az értéknél kisebb, mert a legelső előfordulás elé (ami vagy egy kezdő zárójel, vagy egy változó) nem teszünk műveleti jelet. Elvileg akárhány zárójelpár-lehet, de célszerű kizárnunk azokat a lehetőségeket, mikor felesleges, elhagyható zárójelpárok szerepelnek a reguláris kifejezésben. Belátható, hogy a nem elhagyható zárójelpárok száma kisebb mint a változók előfordulási száma. Vagyis ha csupán nem elhagyható zárójeleket tételezünk fel, a minimális változó előfordulási számú reguláris kifejezés (vagy ha több van akkor ezek egyike) meghatározható.

3. Formális Nyelvek: Definíciók és jelölések

Ebben a fejezetben összefoglaljuk a legfontosabb fogalmakat. A teljesség kedvéért időnként ismétlésekbe bocsátkozunk.

3.1 Alapfogalmak

Szimbólumok tetszőleges nemüres, véges halmazát *ábécének* nevezzük, és V -vel jelöljük. A V elemeit az ábécé *betűinek* mondjuk.

A V elemeiből álló véges jelsorozatokat az adott ábécéből alkotott *szavaknak* nevezzük. A V elemeiből álló szavak halmazát V^* -gal jelöljük. Egy $P \in V^*$ szó *hosszán* a szót alkotó betűk számát értjük, és ezt $|P|$ -vel jelöljük. Az egyetlen betűt sem tartalmazó sortozatot *üres szónak* nevezzük, és λ -val jelöljük. A P és Q szavak *konkatenációján* azt a PQ szót értjük, amely úgy jön létre, hogy Q szót a P szó után írjuk. A P szó a Q szó *részszava*, ha léteznek olyan P_1, P_2 szavak, amelyekre $Q = P_1PP_2$. Legyen i nemnegatív egész szám. Ekkor a P szó i -edik *hatványán* az önmagával vett i -szeres konkatenációját értjük. Megállapodás szerint $P^0 = \lambda$ minden P szóra. A V ábécé feletti szavak egy tetszőleges L halmazát a V ábécéből alkotott (*formális*) *nyelvnek* nevezzük.

Egy *nyelv véges*, ha csak véges sok szót tartalmaz, egyébként *végtelen*. Az így definiált nyelvfogalom túl általános, magában foglalja mind a mesterséges, mind a természetes nyelvek halmazát. A kérdés viszont az, hogyan lehet ténylegesen megadni egy konkrét nyelvet. Az egyszerű tulajdonságokkal rendelkező nyelveket máris megadhatjuk a halmazok megadásának különböző módjai szerint. Legyen például a véges ábécénk mindössze két elemű: $V = \{0, 1\}$. Ekkor az

$$L_1 = \{\lambda\}, L_2 = \{1, 10, 0010, 111\}, L_3 = \{1^i \mid i \text{ prím}\}$$

halmazok mindegyike egy-egy nyelv a fenti definíció értelmében. Szükségünk van azonban olyan eszközökre, amelyekkel a fentieknél lényegesen bonyolultabb tulajdonságú nyelveket is definiálhatunk. Ebből a célból vezetjük be a generatív nyelvtan fogalmát.

A $G = \langle V_N, V_T, S, H \rangle$ rendezett négyest *generatív nyelvtannak* nevezzük, ha V_N és V_T diszjunkt véges ábécék, $S \in V_N$, $H \subseteq (V_N \cup V_T)^* V_N (V_N \cup V_T)^* \times (V_N \cup V_T)^*$. A V_N elemeit *nemterminális jeleknek* vagy *változóknak* nevezzük, és általában nagybetűkkel jelöljük. A V_T elemeit *terminális jeleknek* nevezzük, és általában kisbetűkkel jelöljük. A H elemeit képező $\langle P, Q \rangle$ rendezett párokat *helyettesítési szabályoknak* nevezzük, és általában $P \rightarrow Q$ alakban írjuk. Az S egy kitüntetett nemterminális jel, amely a G nyelvtanban a generálás kiinduló vagy kezdő eleme, másnéven *mondatszimbóluma*. Most még definiálnunk kell, hogy állítunk elő egy nyelvet egy generatív nyelvtan segítségével.

Egy G generatív nyelvtanban az X szóból egy *lépésben levezethető* az Y szó, ha van olyan $P \rightarrow Q$ helyettesítési szabály a H -ban és $P_1, P_2 \in (V_N \cup V_T)^*$ úgy, hogy $X = P_1PP_2$ és $Y = P_1QP_2$. Jelölés: $X \Rightarrow Y$.

Egy G generatív nyelvtanban az X szóból *levezethető* az Y szó, ha van olyan $X_0, X_1, \dots, X_n$ véges szósorozat a $(V_N \cup V_T)^*$ -ban, amelyre teljesül, hogy $X_0 = X$,

$X_n = Y$ és $X_i \Rightarrow X_{i+1}$ ($i = 0, 1, \dots, n-1$). Ezt a relációt $X \Rightarrow^* Y$ szimbólummal jelöljük. Megegyezés szerint $\forall X \in (V_N \cup V_T)^*$ szóra $X \Rightarrow^* X$ teljesül.

A $G = \langle V_N, V_T, S, H \rangle$ nyelvtan által generált nyelven az $L(G) = \{P \mid S \Rightarrow^* P, P \in V_T^*\}$ halmazt értjük.

Példa 3.1 Legyen a $G = \langle V_N, V_T, S, H \rangle$ nyelvtanban $V_N = \{S, C, B\}$, $V_T = \{a, b, c\}$, a H -beli szabályok pedig a következők: $S \rightarrow aBC$, $S \rightarrow aSBC$, $CB \rightarrow BC$, $aB \rightarrow ab$, $bB \rightarrow bb$, $bC \rightarrow bc$, $cC \rightarrow cc$. Belátható, hogy ez a nyelvtan azt az L nyelvet generálja, amely a következő alakban adható meg: $L = \{a^i b^i c^i \mid i \geq 1\}$.

3.2 A Chomsky-féle nyelvosztályok

A nyelvtan és az általa generált nyelv definíciója szerint minden nyelvtanhoz egy egyértelműen meghatározott nyelv tartozik, de megfordítva, egy nyelvet nem csak egy nyelvtannal generálhatunk.

Két nyelvtant (*gyengén*) ekvivalensnek nevezzük, ha ugyanazt a nyelvet generálják.

Az ekvivalencia fogalmának ismeretében kézenfekvőnek látszik a különböző nyelvtanokat bizonyos formai tulajdonságok alapján osztályokba sorolni. Az osztályozás alapját a helyettesítési szabályok alakjára vonatkozó kikötések képezik abban a hierarchiában, amelyet az elmélet egyik megalapozója, N. Chomsky vezetett be, és amelyet alább ismertetünk.

A $G = \langle V_N, V_T, S, H \rangle$ -t i -típusú nyelvtannak nevezzük, ha az alábbi megkötések közül az i -edik teljesül:

- $i = 0$: Nincs semmilyen további megkötés. (*Mondatszerkezetű nyelvtan*)
- $i = 1$: Minden H -beli szabály $P_1QP_2 \rightarrow P_1RP_2$ alakú, ahol $P_1, P_2 \in (V_N \cup V_T)^*$, $Q \in V_N$, $R \in (V_N \cup V_T)^* \setminus \{\lambda\}$; vagy pedig $S \rightarrow \lambda$ alakú, de ekkor S nem fordulhat elő egyetlen H -beli szabály jobb oldalán sem. (*Környezetfüggő nyelvtan*)
- $i = 2$: Minden H -beli szabály $P \rightarrow Q$ alakú, ahol $P \in V_N$, $Q \in (V_N \cup V_T)^*$. (*Környezetfüggetlen nyelvtan*)
- $i = 3$: Minden H -beli szabály $P \rightarrow aQ$ vagy $P \rightarrow a$ alakú, ahol $P, Q \in V_N$, $a \in V_T^*$. (*Reguláris nyelvtan*)

Itt jegyezzük meg, hogy a fenti nyelvtanokon kívül szokás még a

- Minden H -beli szabály $P \rightarrow aQb$ vagy $P \rightarrow a$ alakú, ahol $P, Q \in V_N$, $a, b \in V_T^*$. (*Lineáris nyelvtan*)

A környezetfüggő elnevezés az 1-típus esetén arra mutat, hogy egy szabály egyetlen Q nemterminális helyébe egy nem üres szónak a beírását jelenti, de csak akkor ha a nemterminális közvetlen környezete egy adott P_1, P_2 szópár. (Természetesen megengedett az is, hogy P_1, P_2 bármelyike, vagy akár mind a kettő üres legyen.) Üres szavat csakis a mondat-szimbólum helyébe tehetünk, s azt is csak egy levezetés legelső lépésében teszi lehetővé az 1-típusú nyelvtan. Ezt biztosítjuk azon megszorítással, hogy ha 1-nyelvtanunkban egy $P_1QP_2 \rightarrow P_1RP_2$ szabályban $Q = \lambda$, csak úgy lehessen, ha $X = S, P_1 = P_2 = \lambda$ (azaz $S \rightarrow \lambda$ szabályról van szó). S hogy egy levezetés során az S helyébe csak az első lépésben lehessen üres szót tenni (a többi nemterminálisra ez a lehetőség is ki van zárva), az $S \rightarrow \lambda$ szabály előfordulása esetén

kizárjuk annak lehetőségét, hogy S szabályok jobboldalán szerepeljen. Utalva arra, hogy a 2-típusú nyelvtanok esetén egy szabály baloldalán álló nemterminális bármilyen környezetben helyettesíthető a szabály jobboldalán álló szóval, innen az elnevezés. Ha egy 2-típusú nyelvtan minden szabálya legfeljebb egy nemterminálist tartalmaz a jobb oldalán, akkor eljutunk a lineáris grammatika fogalmához. A lineáris nyelvtanok a hierarchiában a 2- és a 3-típus között helyezkednek el. Ha egy lineáris nyelvtanban speciálisan az összes $P \rightarrow aQb$ alakú szabályban $a = \lambda$, akkor 3-típusú nyelvtanhoz jutunk, amik a *jobb-lineáris nyelvtan* elnevezést innen kapták. Hasonló módon, ha az összes $P \rightarrow aQb$ alakú szabályban $b = \lambda$, akkor *bal-lineáris nyelvtanokról* beszélünk. A bal-lineáris grammatikával generálható nyelvek generálhatók jobblinéáris grammatikával is.

Az $i = 0, 1, 2, 3$ értékek esetén azt mondjuk, hogy egy *nyelv i -típusú*, ha van olyan i -típusú nyelvtan, amely azt generálja. Az i -típusú nyelvek osztályát $\mathcal{L}_i$ -vel jelöljük. Rögtön látható, hogy minden 3-típusú nyelvtan egyben 2-típusú, és minden 1-típusú egyben 0-típusú is. Az is nyilvánvaló, hogy minden 2-típusú vagy 3-típusú nyelvtan egyben 0-típusú is. Nézzük meg az 1-típusú és a 2-típusú nyelvtanok közötti kapcsolatot.

Tétel 3.1 (Üres-szó lemma) *Minden $G = \langle V_N, V_T, S, H \rangle$ 2-típusú nyelvtanhoz létezik olyan vele ekvivalens 1-típusú $G' = \langle V_N, V_T, S, H' \rangle$ nyelvtan. (Természetesen így H' szabályai jobb oldalán nem szerepel λ , illetve ha igen, akkor csak $S \rightarrow \lambda$ alakú szabály esetén. Ekkor azonban S nem fordul elő egyetlen H' -beli szabály jobb oldalán sem.)*

Bizonyítás: Legyen $G = \langle V_N, V_T, S, H \rangle$ környezetfüggetlen nyelvtan. Generáljuk az U_i halmazokat az alábbiak szerint:

$$U_1 = \{P \mid P \rightarrow \lambda \in H\}$$

$$U_{k+1} = U_k \cup \{P \mid P \rightarrow Q \in H \text{ és } Q \in U_k^*\}$$

Ekkor $\exists i$, amelyre teljesül, hogy $U_i = U_{i+1}$. Ezt a V_N végeessége biztosítja. Legyen $P \rightarrow Q \in H'$, ha $P \rightarrow R \in H$, $Q \neq \lambda$, és vagy $Q = R$, vagy Q úgy áll elő, hogy R -ből U_i -beli elem(ek)et törölünk. Legyen $G' = \langle V_N, V_T, S, H' \rangle$. Ekkor $L(G) \setminus \{\lambda\} = L(G')$. Ilyenkor

- ha $\lambda \notin L(G)$, akkor $L(G) = L(G')$, valamint nyelvtanunk 1-típusú.
- ha $\lambda \in L(G)$, akkor $V'_N = V_N \cup \{S'\}$, $S' \notin (V_N \cup V_T)$ és $H'' = H' \cup \{S' \rightarrow S, S' \rightarrow \lambda\}$ választásokkal megkonstruált $G'' = \langle V'_N, V_T, S', H'' \rangle$ nyelvtan 1-típusú, és $L(G) = L(G')$. $\square$

A fenti tételünk alapján könnyen látható, hogy $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$. Ez a hierarchia valójában élesebb, de ezt a későbbiekben fogjuk bizonyítani.

3.3 Nyelvekre értelmezett műveletek

Formális nyelvekre, mint szóhalmazokra közvetlenül értelmezhetők a halmazelméleti alapműveletek:

$$L_1 \cup L_2 = \{P \mid P \in L_1 \text{ vagy } P \in L_2\}$$

$$\begin{aligned} L_1 \cap L_2 &= \{P \mid P \in L_1 \text{ és } P \in L_2\} \\ L_1 \setminus L_2 &= \{P \mid P \in L_1 \text{ és } P \notin L_2\} \\ \overline{L} &= V^* \setminus L \end{aligned}$$

Ahogy a 2.7. fejezetben már említettük, a nyelveken az unió műveletén kívül a konkatenációt és az iterációt alpműveleteknek tekintjük. Emlékeztetőül, két nyelv *konkatenációján* a következő nyelvet értjük: $L_1 \cdot L_2 = \{PQ \mid P \in L_1 \text{ és } Q \in L_2\}$. Legyen $i = 1, 2, \dots$. Ekkor egy L nyelv i -edik *hatványán* a nyelv i -szer egymás utáni, önmagával való konkatenációját értjük. Jelölés: L^i . Megállapodás szerint $L^0 = \{\lambda\}$. A *konkatenáció lezárását* az $L^* = \bigcup_{i=0}^{\infty} L^i$ összefüggéssel értelmezzük. Használatos még az $L^+ = L^* \setminus \{\lambda\}$ jelölés is.

Legyen adva két véges ábécé, V_1 és V_2 . A V_1^* -nak a V_2^* -ba való h leképezését *homomorfizmusnak* nevezzük, ha:

1. injektív,
2. és művelettartó, azaz $h(PQ) = h(P)h(Q)$, tetszőleges $P, Q \in V_1^*$ -ra.

A fenti két tulajdonságból rögtön következik, hogy az üres szó képe az üres szó lesz, ugyanis $h(P) = h(P\lambda) = h(P)h(\lambda)$ minden $P \in V_1^*$ szóra.

Egy homomorf leképezést *λ -mentesnek* nevezünk, ha $h(P) = \lambda$ esetén $P = \lambda$.

Azt mondjuk, hogy egy nyelvtan *normális alakban* van, ha a helyettesítési szabályokban terminális jelek csak $X \rightarrow a$ ($X \in V_N, a \in V_T$) alakú szabályokban fordulnak elő.

Tétel 3.2 Minden $G = \langle V_N, V_T, S, H \rangle$ nyelvtanhoz megadható egy vele ekvivalens $G' = \langle V'_N, V_T, S, H' \rangle$ nyelvtan úgy, hogy minden H' -beli szabály, amely terminális jelet tartalmaz, $X \rightarrow a$ alakú, ahol $X \in V_N, a \in V_T$. Emellett G' nyelvtan ugyanolyan típusú, mint G , kivéve, ha G 3-as típusú.

Bizonyítás: Minden egyes $a_k \in V_T$ terminális jelhez vezessünk be új $A_k \notin V_N$ nemterminálist, és legyen

$$V'_N = V_N \cup \{A_1, A_2, \dots, A_n\}.$$

A H' -beli szabályokat adjuk meg úgy, hogy a H szabályaiban minden a_k terminális helyére a megfelelő A_k változót írjuk. Ezenkívül a H' -beli szabályok közé még felvesszük az $A_k \rightarrow a_k$ alakú új szabályokat, minden $k = 1, \dots, n$ -re. Az így kapott H' nyilván rendelkezik a kívánt tulajdonsággal.

Az ekvivalencia bizonyításához először megmutatjuk, hogy $L(G) \subseteq L(G')$. Legyen $P = a_{k_1}a_{k_2}\dots a_{k_l} \in L(G)$. Ekkor a G' nyelvtanban levezethető a megfelelő $A_{k_1}A_{k_2}\dots A_{k_l}$ szó, és az $A_k \rightarrow a_k$ szabályok segítségével ebből a P -t megkaphatjuk. Ha továbbá $\lambda \in L(G)$, akkor a G -ben a megfelelő szabályok alkalmazásával ugyancsak megkapjuk λ -t.

Most lássuk be, hogy $L(G') \subseteq L(G)$. Legyen $h : (V'_N \cup V_T)^* \rightarrow (V_N \cup V_T)^*$ homomorf leképezés a következő módon értelmezve:

- $h(A_k) = a_k$, illetve
- $h(x) = x, x \in (V_N \cup V_T)$.

Ekkor bármely két $P, Q \in (V'_N \cup V_T)^*$ szóra a $P \xrightarrow{G} Q$ relációból következik, hogy vagy $h(P) = h(Q)$, vagy $h(P) \xrightarrow{G} h(Q)$. Ha ugyanis a G nyelvtanban a P -ből a Q

úgy adódik, hogy valamelyik $A_k \rightarrow a_k$ szabályt alkalmazzuk, akkor $h(P) = h(Q)$. Ha pedig H' -nek egy olyan szabályát alkalmazzuk, amelyet egy H -beli szabályból nyertünk, akkor nyilván $h(P) \xrightarrow{G} h(Q)$. Tehát mindkét esetben $P \xrightarrow{G} Q$ relációból a $h(P) \xrightarrow{G}^* h(Q)$ következik. Legyen most $P \in L(G')$, akkor $S \xrightarrow{G}^* P$, mert $S = h(S) \xrightarrow{G}^* h(P) = P$, amivel a tételt bebizonyítottuk. $\square$

A nyelvekre mint szóhalmazokra értelmezett alpműveleteket (a halmazelméleti egyesítést, a konkatenációt, és a konkatenáció-lezárását) *reguláris műveleteknek* nevezzük.

Tétel 3.3 *Az $\mathcal{L}_i$ nyelvosztályok mindegyike ($i = 0, 1, 2, 3$) zárt a reguláris műveletekre nézve.*

Bizonyítás: A bizonyítás során tételezzük fel, hogy $L = L(G)$, $L' = L(G')$, ahol $G = \langle V_N, V_T, S, H \rangle$, $G' = \langle V'_N, V'_T, S', H' \rangle$. Az előző tétel alapján feltehetjük, hogy az H és H' szabályokban terminális jelek nem szerepelnek a bal oldalon. Az általánosság megszorítása nélkül feltehetjük, hogy $V_N \cap V'_N = \emptyset$. A bizonyítást az egyes műveletekre külön-külön végezzük el.

Egyesítés. Ha $i = 0, 2, 3$, akkor S_0 legyen egy olyan nemterminális, amely eddig nem szerepelt sem V_N -ben, sem V'_N -ben. A

$$G_e = \langle V_N \cup V'_N \cup \{S_0\}, V_T \cup V'_T, S_0, H \cup H' \cup \{S_0 \rightarrow S, S_0 \rightarrow S'\} \rangle$$

nyelvtan ekkor az $L \cup L'$ nyelvet generálja, és egyben i -típusú, ha mind G , mind G' i -típusú. Egyrészt ugyanis $L \cup L' \subseteq L(G_e)$, másrészt az $L(G_e) \subseteq L \cup L'$ igazolásához azt kell megjegyeznünk, hogy az $S_0 \rightarrow S$ ($S_0 \rightarrow S'$) szabály kizárja az H' (H) halmazban található szabályok alkalmazását. Ha $i = 1$ és $\lambda \notin L \cup L'$, akkor a fenti konstrukció ugyanúgy megfelel. Ha $\lambda \in L \cup L'$, akkor vegyük az $L_1 = L \setminus \{\lambda\}$ és az $L_2 = L' \setminus \{\lambda\}$ nyelveket (, vagyis hagyjuk el az $S \rightarrow \lambda$, illetve $S' \rightarrow \lambda$ szabályok közül előforduló(ka)t), és alkalmazzuk a korábbi konstrukciót, majd vegyünk fel az új mondatshimbólummal az új szabályokkal ($S_0 \rightarrow S, S_0 \rightarrow S'$) az $S_0 \rightarrow \lambda$ szabályt is.

Konkatenáció. Legyen $i = 0, 2$ és $S_0 \notin V_N \cup V'_N$. Ekkor a

$$G_k = \langle V_N \cup V'_N \cup \{S_0\}, V_T \cup V'_T, S_0, H \cup H' \cup \{S_0 \rightarrow SS'\} \rangle$$

nyelvtan az LL' nyelvet generálja, és egyben i -típusú, ha mind G , mind G' i -típusú. Az nyilvánvaló, hogy $LL' \subseteq L(G_k)$. Az $L(G_k) \subseteq LL'$ bizonyításához tegyük fel, hogy $S_0 \Rightarrow^* P$. Az adott levezetést az $S_0 \Rightarrow P_1 \Rightarrow \dots \Rightarrow P_m = P$ lépésekre bontva belátható, hogy minden P_j -hez van olyan Q_j és Q'_j , amelyekre $P_j = Q_j Q'_j$ és $S \Rightarrow^* Q_j$, $S' \Rightarrow^* Q'_j$ ($j = 1, 2, \dots, m$). Nyilvánvalóan $P_1 = SS'$. Ha pedig valamely P_j -re ilyen alakban írható, akkor P_{j+1} -re is teljesül. Eszerint a nyelvtenban levezethető szavak mindegyike benne van az LL' -ban. Ha $i = 1$ és $\lambda \notin L \cup L'$, akkor a fenti konstrukció megfelel a céljainknak. Ha $i = 1$ és $\lambda \in L \cup L'$, akkor a korábbiakhoz hasonlóan vegyük először az $L_1 = L \setminus \{\lambda\}$ és $L_2 = L' \setminus \{\lambda\}$ nyelveket. A halmazműveletek disztributivitása értelmében az LL' megegyezik az alábbi három nyelv valamelyikével:

$$L_1 L_2 \cup L_2 \quad L_1 L_2 \cup L_1 \quad L_1 L_2 \cup L_1 \cup L_2 \cup \{\lambda\}$$

Minthogy az L_1 , L_2 és $\{\lambda\}$ nyelvekre mind az egyesítés, mind a konkatenáció zártságát beláttuk, ebből állításunk következik. Könnyű látni, hogy $i = 3$ esetén a következő konstrukció megfelel céljainknak. Helyettesítsük H -ban minden $A \rightarrow P$ ($A \in V_N$, $P \in V_T^*$) alakú szabályt az $A \rightarrow PS'$ szabállyal. Az így módosított H szabályhalmazt H_1 -gyel jelölve képezzük a

$$G_k = \langle V_N \cup V'_N, V_T \cup V'_T, S, H_1 \cup H' \rangle$$

nyelvtant, amely az LL' nyelvet generálja és szintén 3-típusú.

Konkatenáció lezárása. Legyen $i = 2$ és $S_0 \notin V_N$. Ekkor a

$$G_* = \langle V_N \cup \{S_0\}, V_T, S_0, H \cup \{S_0 \rightarrow \lambda, S_0 \rightarrow S_0 S\} \rangle$$

nyelvtan az L^* nyelvet generálja és ugyancsak 2-típusú.

Legyen $i = 3$. Ekkor legyen H_2 az a szabályrendszer, amiben helyettesítjük a H minden $A \rightarrow P$ ($A \in V_N$, $P \in V_T^*$) szabályát az $A \rightarrow PS$ szabállyal. Az így kapott H_2 szabályhalmazzal és $S_0 \notin V_N$ -nel képezzük

$$G_* = \langle V_N \cup \{S_0\}, V_T, S_0, H \cup H_2 \cup \{S_0 \rightarrow \lambda, S_0 \rightarrow S\} \rangle$$

nyelvtan az L^* nyelvet generálja és 3-típusú.

Legyen végül $i = 0, 1$ és $\lambda \notin L$. Ekkor a

$$G_* = \langle V_N \cup \{S_0, S_1\}, V_T, S_0, H \cup \{S_0 \rightarrow \lambda, S_0 \rightarrow S, S_0 \rightarrow S_1 S\} \cup \{S_1 a \rightarrow S_1 S a \mid a \in V_T\} \cup \{S_1 a \rightarrow S a \mid a \in V_T\} \rangle$$

nyelvtan, ahol $S_0 \notin V_N$, $S_1 \notin V_N$ új nemterminális jelek, az L^* nyelvet generálja és egyben i -típusú. Nyilván $L^* \subseteq L(G_*)$. Az ellenkező irányú tartalmazáshoz legyen

$$S_0 \Rightarrow P_1 \Rightarrow \dots \Rightarrow P$$

egy tetszőleges levezetés G_* -ban. Ha $P_1 = \lambda$, akkor $P_1 \in L^*$, ha pedig $P_1 = S$, akkor ez a levezetés csak olyan V_T^* -beli szóhoz vezethet, amely benne van az L -ben. Egyébként P_1 csak $S_1 S$ alakú lehet, és ekkor minden P_j vagy

- $S_1 Q_1 \dots Q_k$ alakú, ahol $k \geq 1$ és a $Q_2, \dots, Q_k$ szavak mindegyikének az első betűje terminális, és $S \Rightarrow^* Q_l$ ($l = 1, \dots, k$) vagy
- $Q_0 Q_1 \dots Q_k$ alakú, ahol $k \geq 1$ és a $Q_1, Q_2, \dots, Q_k$ szavak mindegyikének az első betűje terminális és $S \Rightarrow^* Q_l$ ($l = 0, 1, \dots, k$).

Ezt teljes indukcióval bizonyíthatjuk. Így beláttuk, hogy $L(G_*) \subseteq L^*$.

Abban az esetben, ha $i = 0, 1$ és $\lambda \in L$, akkor elsőként egy olyan G_1 nyelvtant konstruálunk, amelyre $L(G_1) = L(G) \setminus \{\lambda\}$. Ezt $i = 1$ esetben egyszerűen úgy kapjuk, hogy az $S \rightarrow \lambda$ szabályt elhagyjuk. Az $i = 0$ esetben pedig minden $P \rightarrow \lambda$ alakú szabályt helyettesítünk az $xP \rightarrow x$ és $Px \rightarrow x$ alakú szabályok halmazával, ahol az x végigfut az összes terminális és nemterminális jelen. Az így kapott G_1 nyelvtan típusa tehát megegyezik a G típusával, és nyilván $L_1^* = L^*$. Elég tehát a konstrukciót a $\lambda \notin L$ esetre megadni. Ezt az esetet már korábban beláttuk. $\square$

4. Nyelvek és Automaták

Mielőtt a különböző típusú nyelvtanokat, illetve nyelveket más szempontok alapján is megvizsgálánk, megnézzük kapcsolatukat az automatákkal.

Ebben a részben, hacsak mást nem mondunk, automatán mindig iniciális kimenő jel nélküli automatát értünk.

4.1 Reguláris Nyelvtanok és Véges Automaták

E fejezetben megmutatjuk, hogy a 3-as típusú, azaz reguláris nyelvtanokkal generálható $\mathcal{L}_3$ osztálya megegyezik a véges automaták által felismerhető nyelvek osztályával, vagyis Kleene tétele értelmében, a reguláris nyelvek osztályával. Más szóval, a 3-as típusú, azaz reguláris nyelvtan, mint generatív eszköz, azonos értékű a véges automatával mint felismerő eszközzel.

Amint azt a 2.1. fejezetben már láttuk, a *nemdeterminisztikus véges automata* olyan $\mathcal{A} = (A, a_0, X, \delta)$ rendszer, amely hasonlóan működik mint az iniciális kimenő jel nélküli (determinisztikus) automata, azzal a különbséggel, hogy a δ átmeneti függvény az $A \times X$ szorzathalmazt az A állapothalmaz részhalmazainak halmazába, vagyis a 2^A hatványhalmazba képezi le: $\delta : A \times X \rightarrow 2^A$. Minden $a \in A, x \in X$ párra $\delta(a, x) \subseteq A$ az átmeneti lehetőségek halmaza. Feltételezzük viszont, hogy az $\mathcal{A}$ egy $a \in A$ állapotból az $x \in X$ jel hatására mindig egy állapotba megy át (kivéve ha $\delta(a, x) = \emptyset$, mikor is átmenet nincs értelmezve), az átmenet azonban $\mathcal{A}$ által nincs egyértelműen meghatározva.

Példa 4.1

$\mathcal{A}$	a	b	c
x	$\{a, b\}$	$\emptyset$	$\{b\}$
y	$\{c\}$	$\{b, c\}$	$\{c\}$

A fenti táblázattal definiált $\mathcal{A}$ nemdeterminisztikus véges automata például az x bemenő jel hatására az a állapotból átmehet b -be, de maradhat az a állapotban is. A b állapotban az $\mathcal{A}$ az x bemenő jelet nem képes feldolgozni.

A δ függvény értelmezését kiterjesztjük úgy, hogy legyen $\delta : A \times X^* \rightarrow 2^A$, ahol minden $a \in A$ -ra $\delta(a, \lambda) = \{a\}$ és minden $x \in X, p \in X^*$ párra $\delta(a, px) = \cup_{b \in \delta(a, p)} \delta(b, x)$. (Természetesen előfordulhat, hogy $\delta(a, px) = \emptyset$.) Vegyük észre, hogy ez a kiterjesztés lényegesen különbözik az automataelméleti vizsgálatoknál korábban alkalmazott kiterjesztéstől, hisz ott képelemként $(2^A)^+$ -beli elemek léptek fel, míg itt csak 2^A -beliek.

Azt mondjuk, hogy az L nyelv előállítható vagy felismerhető $\mathcal{A}$ -ban az $A' \subseteq A$ halmazzal, jelekben: $L = L_{\mathcal{A}}^{A'}$, ha $p \in L \Leftrightarrow \delta(a_0, p) \cap A' \neq \emptyset$.

Tétel 4.1 A nemdeterminisztikus véges automatákban előállítható nyelvek $\mathcal{L}_{NFA}$ osztálya megegyezik a véges automatákban előállítható nyelvek $\mathcal{L}_{FA}$ osztályával.

Bizonyítás: $\mathcal{L}_{FA} \subseteq \mathcal{L}_{NFA}$ triviális. A fordított irányú tartalmazás igazolásához legyen $L = L_{\mathcal{A}}^{A'}$, ahol $\mathcal{A} = (A, a_0, X, \delta)$ most valamilyen nemdeterminisztikus véges automata.

Alkossuk meg a $\mathcal{B} = (B, b_0, X, \delta')$ véges automatát, amelyre $B = 2^A, b_0 = a_0$ és bármely $b \in B, x \in X$ párra $\delta'(b, x) = \cup_{a \in b} \delta(a, x)$. Könnyű belátni, hogy $L = L_{\mathcal{B}}^{B'}$, ahol $B' = \{\overline{b} = \delta(b_0, p) \mid b \cap A' \neq \emptyset\}$, amivel az $\mathcal{L}_{NFA} \subseteq \mathcal{L}_{FA}$ tartalmazást is igazoltuk. $\square$

Egyszerűen bizonyíthatók a következő tételek.

Tétel 4.2 Minden G 3-as típusú nyelvtanhoz létezik egy vele ekvivalens G' 3-as típusú nyelvtan, amelynek szabályai jobb oldalán a mondatzimbólum nem lép fel.

Bizonyítás: Valóban, ha $G = (V_N, V_T, S, H)$ 3-as típusú nyelvtan, akkor egy ilyen G' nyelvtant megadhatunk $G' = (V_N \cup \{S'\}, V_T, S', H')$ alakban, ahol $S' \notin V_N$ és $H' = H \cup \{S' \rightarrow P \mid S \rightarrow P \in H\}$. $\square$

Tétel 4.3 Ha az L nyelv 3-as típusú, akkor $L \cup \{\lambda\}$ is.

Bizonyítás: Legyen $G = (V_N, V_T, S, H)$ egy 3-as típusú nyelvtan, melyre $L = L(G)$. Ha L tartalmazza az üres szót, $L \cup \{\lambda\}$ 3-as típusú volta automatikusan teljesül. Ha nem tartalmazza, akkor tekintsünk egy G' nyelvtant, ahol $G' = (V_N, V_T, S, H')$ és $H' = H \cup \{S \rightarrow \lambda\}$. Ekkor a G' 3-as típusú nyelvtan generálja $L \cup \{\lambda\}$ -t. $\square$

Tétel 4.4 A 3-as típusú nyelvek $\mathcal{L}_3$ osztálya egybeesik a véges automatákkal előállítható nyelvek $\mathcal{L}_{FA}$ osztályával.

Bizonyítás: Legyen $L \in \mathcal{L}_{FA}$ és legyen $L = L_{\mathcal{A}}^{A'}$, ahol $\mathcal{A} = (A, a_0, X, \delta)$ véges automata. Tekintsük azt a $G = (A, X, a_0, H)$ nyelvtant, amelyre bármely $a, b \in A$ és $x \in X$ esetén $a \rightarrow xb \in H \Leftrightarrow \delta(a, x) = b$ és $a \rightarrow x \in H \Leftrightarrow \delta(a, x) \in A'$. Világos, hogy G 3-as típusú (sőt, jobbról reguláris) nyelvtan, mely $a_0 \notin A'$, azaz $\lambda \notin L$ esetén az L nyelvet generálja. Ha pedig $a_0 \in A'$, vagyis $\lambda \in L$, akkor $L(G) = L \setminus \{\lambda\}$, tehát $L \setminus \{\lambda\}$ jobbról lineáris. Így az 4.3. Tétel alapján az $L = (L \setminus \{\lambda\}) \cup \{\lambda\}$ is jobbról lineáris. Tehát mindenképp $L \in \mathcal{L}_3$, amivel az $\mathcal{L}_{FA} \subseteq \mathcal{L}_3$ igazolást nyert.

Legyen most $L \in \mathcal{L}_3$. Ekkor a 4.2. Tétel szerint létezik olyan $G = (V_N, V_T, S, H)$ 3-as típusú nyelvtan, amelyre $L = L(G)$ és amelynek szabályai jobboldalán az S mondatzimbólum nem lép fel. A 4.1.cTétel alapján elegendő azt megmutatnunk, hogy L előállítható nemdeterminisztikus véges automatában. Egy ilyen nemdeterminisztikus véges automatát $\mathcal{A} = (V_N \cup \{*\}, S, V_T, \delta)$ alakban adhatunk meg, ahol $*$ $\notin V_N$ tetszőleges új szimbólum, a δ átmeneti függvény pedig tetszőleges $x \in X$ bemenő jelre a $\delta(*, x) = \emptyset$ és

$$\delta(B, x) = \begin{cases} \{C \in V_N \mid B \rightarrow xC \in H\} & \text{ha } B \rightarrow x \notin H, \\ \{*\} \cup \{C \in V_N \mid B \rightarrow xC \in H\} & \text{ha } B \rightarrow x \in H \end{cases}$$

összefüggésekkel van definiálva. A konstrukció alapján látható, hogy $L = L_{\mathcal{A}}^M$, ahol

$$M = \begin{cases} \{S, *\}, & \text{ha } S \rightarrow \lambda \in H, \\ \{*\}, & \text{ha } S \rightarrow \lambda \notin H. \end{cases}$$

Tehát $L \in \mathcal{L}_{FA}$, amivel az $\mathcal{L}_3 \subseteq \mathcal{L}_{FA}$ tartalmazást is igazoltuk. $\square$

Itt jegyezzük meg, hogy tehát egy reguláris nyelvet három (ekvivalens) módon adhatunk meg: reguláris kifejezéssel, véges (determinisztikus vagy nemdeterminisztikus) automatával, illetve 3. típusú nyelvtannal.

4.2 Környezetfüggetlen Nyelvtanok és Veremautomaták

Ha a véges automata definíciójában az állapothalmaz végességére vonatkozó követelményt elhagyjuk, akkor (a véges ábécéjű) végtelen automata fogalmához jutunk. Ez a fogalom e formájában túl általánosnak bizonyult, ezért bevezették a végtelen automaták speciális fajtáit.

A veremautomatát eredetileg aritmetikai kifejezések számítógéppel történő kiértékelésére találták ki, s történeti érdekessége, hogy egy veremautomatát realizáló szoftver volt az első olyan számítógépes szoftver termék, mely szabadali oltalmat kapott az USA-ban. Egyben ez volt a világon az első szoftver szabadalom. (A szabadalmaztatás a hatvanas években történt.)

A veremautomata esetén egy potenciálisan végtelen befogadóképességű veremmemória összes lehetséges tartalma eredményez végtelen sok állapotot. A veremmemória pozíciókra felosztott egy irányban végtelen szalag. Minden pozícióba egy-egy jel írható. A jelek kiolvasása, illetve törlése a bevitelhez képest fordított sorrendben történik. Más szóval, a verem belsejének tartalmához közvetlenül nem férünk hozzá, hanem mindig csak a verem tetején lévő jelet tudjuk kiolvasni, illetve módosítani (felülrni, vagy törölni) és csak a verem tetejére (a benn lévők fölé) helyezhetünk el újabb jelet. A verem alján van egy speciális "verem alja" jel, mely nem tartozik szervesen a verem ábécéhez, s ezt a jelet a veremautomata sem felülrni, sem törölni nem tudja.

A veremautomata a véges bemenő szót egy input szalagon kapja meg, melyről egy olvasófejjel, irányban balról jobbra haladva betűnként tudja leolvasni a tekintett bemenő szót. Az is megengedett, hogy az input szalag üres legyen, azaz az üres szót tartalmazza.

A veremautomatához a potenciálisan végtelen veremmemórián és az input szalagon kívül tartozik még egy véges iniciális nemdeterminisztikus kimenő jel nélküli automata, mely az input szalagról beolvasott betű és a veremautomata tetején levő veremábécé betű, valamint a belső állapota alapján fog a verem tetején operálni, belső állapotát változtatni. Az is megengedett, hogy egy-egy ilyen elemi lépés során az input szalagon a soron következő betűt, vagy a verem tetején levő betűt vagy egyiket se vegye figyelembe. (Azt az elemi lépést, mikor a veremautomata az input szalagon soron következő betűt nem olvassa be (az olvasó fej helyben marad), λ -lépésnek is szokás hívni.) Ezt a véges iniciális nemdeterminisztikus kimenő jel nélküli automatát a fejezet további részében a rövidség kedvéért a veremautomata véges automatájának hívjuk.

A veremautomata működésének kezdetekor a verem üres (csak a verem alja jel van a verem alján), az input szalag olvasófeje a szalag első betűjére mutat (vagy ha az

input szalag tartalma az üres szó, ezt érzékeli), s a veremautomatához tartozó véges automata pedig a kezdő állapotában van.

A működés veremautomata esetén is diszkrét időskála mentén haladva történik. Ha a veremautomatához tartozó véges automata a működés során eljut egy végállapotba, akkor veremautomata megáll. A veremautomata megáll akkor is, ha a hozzá tartozó véges automata egy olyan állapotban van, melyhez nem tartozik egyetlen alkalmas átmenet sem. Az alkalmas átmenet létezése azt jelenti, hogy a verem tetején lévő jel figyelembe vételével vagy anélkül és az input szalagon lévő következő betű (ha van olyan) figyelembe vételével vagy anélkül az adott állapotból van lehetséges átmenet egy másik állapotba a veremautomata véges automatájában. A veremautomatához tartozó véges automata belső állapotát a továbbiakban röviden a veremautomata belső állapotának, vagy a veremautomata állapotának mondjuk. (Szigorúan véve a veremautomata belső állapotát egy pár határozza meg, melynek első eleme a verem tartalma, a második pedig a veremautomatához tartozó véges automata belső állapota, s így tekintve a veremautomata végtelen.)

Ha az input szalagon egy nem üres szó van, s a veremautomata úgy áll meg, hogy előzőleg az input szalag utolsó betűjét is beolvasta, akkor azt mondjuk, hogy a veremautomata az input szalagon levő szót elfogadta. Ha a veremautomata úgy áll meg, hogy az input szalag tartalma az üres szó, akkor azt mondjuk, hogy a veremautomata az üres szót elfogadta. Akkor mondjuk, hogy a (nemdeterminisztikus működésű) veremautomata egy bemenő szót elfogad, ha van olyan működésmódja, hogy a bemenő szót elfogadja. A veremautomata által elfogadott bemenő szavak összességét a veremautomata által elfogadott nyelvek hívjuk. Látni fogjuk, hogy a veremautomaták által elfogadott nyelvek osztálya épp a környezetfüggetlen nyelvek osztálya.

Példa 4.2 *Az egyetemi menzán az ételeket tálcán szokás a kiszolgáló pultoktól az asztalokhoz vinni. A tálcák a kiszolgáló pultoknál egymás tetején vannak elhelyezve, s mindig csak a legfelső tálcát lehet elvenni. Ha a legfelső tálca valamiért nem szimpatikus (színe vagy más miatt), ahhoz hogy az alatta levő valamelyik tálcához hozzáférjünk, le kell venni a legfelsőt, s félretenni. Ezt a mozdulatot mindaddig folytatnunk kell, míg a kívánt (valamiért szimpatikus) tálcához nem jutunk. Ezután (ha rendeseknek akarunk látszani) akkor a félretett tálcákat visszapakoljuk (többnyire fordított sorrendben mint ahogy elvettük) úgy, hogy minden visszapakolt tálca fölé helyezzük a következő visszapakolandót.*

Formálisan, egy veremautomata a következő hetes: $\mathcal{A} = (Z, A, X, \delta, z_0, a_0, A_F)$, ahol Z a veremábécé, A a belső állapotok nem üres és véges halmaza, vagy más néven állapot ábécé, $\delta : (Z \cup \{\lambda\}) \times (A \setminus A_F) \times (X \cup \{\lambda\}) \rightarrow 2^{Z^*} \times A$ a (nemdeterminisztikus) átmeneti függvény, $z_0 \notin Z$ a verem kezdőjele ("alja"), $a_0 \in A$ a veremautomata kezdő állapota, A_F a veremautomata végállapotainak halmaza.

Egy veremautomata pillanatnyi konfigurációján értjük azt a Wa szót, ahol $W \in Z^*$ a veremmemória pillanatnyi tartalma (melyhez nem értjük hozzá a verem alján lévő kezdő jelet), $a \in A$ pedig a veremautomata pillanatnyi belső állapota. Egy veremautomata minden lépésben a pillanatnyi konfigurációból a δ átmeneti függvény definíciója értelmében egy vagy több pillanatnyi konfigurációba mehet át (vagy épp nem mehet

át egybe sem). Mint már említettük, az átmenet történhet oly módon, hogy közben a veremautomata nem kéri a bemenő jelsorozat következő jelét (λ -lépés).

Az $\mathcal{A}$ veremautomata egy $P \in (Z \cup \{\lambda\})^* A (X \cup \{\lambda\})^*$ szót *egy lépésben átalakít* a $Q \in (Z \cup \{\lambda\})^* A (X \cup \{\lambda\})^*$ szóba (jelekben: $P \xrightarrow{A} Q$, vagy csak egyszerűen: $P \Rightarrow Q$, ha van olyan $z \in Z \cup \{\lambda\}$ és $a, b \in A$, valamint $W_1, W_2 \in Z^*$, $r \in X^*$, hogy a következő összefüggések valamelyike fennáll:

- (a) alkalmas $x \in X$ -re $P = W_2 z a x r$, $Q = W_2 W_1 b r$ mellett $(W_1, b) \in \delta(z, a, x)$;
- (b) $P = W_2 z a r$, $Q = W_2 W_1 b r$ mellett $(W_1, b) \in \delta(z, a, \lambda)$.

Egy veremautomata a $P \in Z^* A X^*$ szót (véges lépésben) *átalakít* (átalakíthat) a $Q \in Z^* A X^*$ szóba - jelekben: $P \xrightarrow{A}^* Q$ - ha van olyan $P_1, \dots, P_n$ sorozat, hogy $P = P_1$, $Q = P_n$ és $P_i \xrightarrow{A} P_{i+1}$, $i = 1, \dots, n-1$. A veremautomata által elfogadott nyelv:

$$L_{\mathcal{A}} = \{p \in X^* \mid \lambda a_0 p \xrightarrow{A}^* W a \text{ valamely } W \in Z^* \text{ és } a \in A_F \text{ esetén } \}.$$

Látható, hogy a veremautomatán egy lépésben egy jelpáron tud operációt végrehajtani, aholis a jelpár első tagja a veremábécé egy jele, vagy pedig az üres szó, a második tagja pedig az input ábécé egy betűje, vagy az üres szó. A veremautomata alkalmas átalakításával elérhető, hogy a veremábécé bizonyos jelei ne kerüljenek a működés során további átalakításra. Ezen veremjeleket terminálisoknak, a többit pedig a veremautomata belső állapotaival együtt nemterminálisnak választva igazolható a következő

Tétel 4.5 *A környezetfüggetlen nyelvek osztálya egybeesik a veremautomaták által elfogadott nyelvek osztályával.* $\square$

Amennyiben az átmeneti függvény képhalmaza a $Z^* \times A$ halmaz (és nem annak részhalmazai halmaza), determinisztikus veremautomatáról beszélünk. Érvényes a következő

Tétel 4.6 *A determinisztikus veremautomaták által elfogadott nyelvek osztálya valódi részosztálya a nemdeterminisztikus veremautomaták által elfogadott nyelvek osztályának.* $\square$

A determinisztikus veremautomaták által elfogadott nyelvek osztályát determinisztikus környezetfüggetlen nyelveknek hívjuk.

Itt jegyezzük meg, hogy a lineáris (nyelvtanok által generált) nyelvek éppen az *egyszerforduló veremautomaták* által elfogadott nyelvek osztálya. Az egyszerforduló veremautomaták olyan speciális veremautomaták amelyek működésük közben előbb csak "töltik" a vermet, aztán pedig csak "ürítik". (Ha volt már törlés a veremből, akkor már nem tesznek bele újabb szimbólumokat.)

4.3 Mondatszerkezetű Nyelvek s a Turing-gépek

A *Turing-gép* fogalmát Alan Turing vezette be bizonyos automatikusan végrehajtható számítások tanulmányozására 1936-ban, jóval az első programvezérlésű elektronikus számítógépek megjelenése előtt. Egyszerűsége ellenére a Turing-gép elég jó modellnek bizonyult bizonyos számítógépekkel kapcsolatos vizsgálatokban, így például a számítógépek számítási kapacitásának elvi korlátai kutatásában.

A Turing-gép egy potenciálisan végtelen *szalagmemóriával* és egy *író-olvasó fej* ellátott véges automata. A szalagmemória *pozíciókra* van osztva, s minden egyes pozíció mint memória-egység az úgynevezett *szalagábécé* pontosan egy betűjének tárolására képes. Kezdetben a Turing-gép egy specifikált kezdőállapotában van, s a szalagon egy véges hosszúságú *startszó* helyezkedik el. Működésének kezdetekor a Turing-gép író-olvasó feje a startszó első betűjén áll. A startszó előtti és utáni (végtelen sok) szalagpozíció egy speciális betűvel, a *szóközzel* van feltöltve, ami nem tévesztendő össze az üres szóval. Többek között azért is, hogy a startszó elkülöníthető lehessen a szalag többi részén tárolt mindkét irányban végtelen számú szóköztől, feltételezzük, hogy a startszó utolsó betűje nem lehet szóköz. A startszó tehát az író-olvasó fej alatti betűtől (jobbra haladva) tart a szalag utolsó nem üres betűjéig. Speciálisan, üres startszó is elképzelhető. Ezt úgy interpretáljuk, hogy ezesetben a szalag minden egyes pozíciója szóközzel van feltöltve, s az író-olvasó fej ezek egyikére mutat. (Utolsó szóköztől különböző betű pedig ekkor értelemszerűen nincs.) A Turing-gép diszkrét időskála mentén, elkülönített időpillanatokban hajt végre egy-egy elemi műveletet kiindulva egy kezdeti időpontból. A Turing-gép egy ilyen elemi operációja az író-olvasó fej alatti betű olvasásából, ezen betű felülírásából, a belső állapot változtatásából, s az író-olvasó fej egy pozícióval való balra avagy jobbra mozgatásából, vagy éppen a fej helybenhagyásából áll. Amennyiben a Turing-gép eljut egy végállapotba, megáll.

Formálisan, a Turing-gép egy rendezett $\mathcal{M} = (A, a_0, X, b, A_F, \mu)$ hatos, ahol A a gép *belső állapotainak* (véges) halmaza, $a_0 (\in A)$ a *kezdő állapot*, X a *szalagábécé*, $b (\in X)$ a *szóköz betű*, $A_F (\subseteq A)$ a *végállapotok halmaza*, $\mu : (A \setminus A_F) \times X \rightarrow 2^{A \times X \times \{Bal, Jobb, Helyben\}}$ a gép *mozgásfüggvénye*, aholis, mint szokásos, $2^{A \times X \times \{Bal, Jobb, Helyben\}}$ jelöli az $A \times X \times \{Bal, Jobb, Helyben\}$ halmaz összes részhalmazait.

Ha tehát az $\mathcal{M}$ Turing-gép egy $a \in A \setminus A_F$ állapotában van és az író-olvasó fej alatt valamely $x \in X$ jel áll, akkor - ha $\mu(a, x)$ nem üres - a $\mu(a, x)$ -beli hármasok egyike szolgáltatja a gép operáció utáni új állapotát, a szalagjelet felülíró szimbólumot (mely nem feltétlen különböző a felülírt szimbólumtól), illetve az elmozdulás irányát. Ha $\mu(a, x) = \emptyset$, azt úgy interpretáljuk, hogy ha a gép az a állapotban az író-olvasó fej alatt az x betűt találja, további működését felfüggeszti.

Az egyszerűbb írásmód kedvéért a továbbiakban fel fogjuk tételezni azt az egyébként megfelelő írásmóddal elérhető, ám jelentéktelen megszorítást, hogy $A \cap X = \emptyset$. Egy *pillanatnyi konfiguráció* $W = PaQ$ alakú, ahol $P, Q \in X^*$ és $a \in A$, aholis $P \in X^+$ esetén P nem kezdődhet, $Q \in X^+$ esetén pedig Q nem végződik szóközzel. (Természetesen $P = \lambda, Q = \lambda, PQ = \lambda$ bármelyike előfordulhat.) Tetszőleges Z pillanatnyi konfigurációt *ekvivalensnek* hívunk egy X^*AX^* -beli PaQ szóval és viszont,

és ezt $Z \equiv PaQ$ -val, illetve $PaQ \equiv Z$ -vel jelöljük, ha alkalmas k, ℓ nemnegatív egész számpárra $b^k Z b^\ell = PaQ$. Tehát egy $X^* A X^*$ -beli szóval ekvivalens pillanatnyi konfigurációt úgy kapunk meg, hogy mind a bal szélétől jobbra haladva, mind pedig a jobb szélétől balra haladva letöröljük a b betűket egészen addig míg egy b -től különböző betűig nem jutottunk. Ha $a = a_0$ és $P = \lambda$ akkor *kezdő konfigurációról*, ha pedig $a \in A_F$, akkor *végkonfigurációról* beszélünk. (Sok esetben nem is szokták elkülöníteni a Turing-gép végállapotait a többi állapottól. Ilyenkor végkonfiguráció alatt azokat a $W = PaxQ, x \in X \setminus \{b\}$, illetve Pa konfigurációkat szokás érteni, amikor $\mu(a, x) = \emptyset$, illetve $\mu(a, b) = \emptyset$.) Amennyiben valamely $PaxQ, a \in A \setminus A_F, x \in X$, illetve $Pa, a \in A \setminus A_F$ pillanatnyi konfiguráció esetén $\{(a, x, Helyben)\} = \mu(a, x)$, illetve $\{(a, b, Helyben)\} = \mu(a, b)$, akkor azt mondjuk, hogy az $\mathcal{M}$ Turing-gép a tekintett pillanatnyi konfigurációban *egyszerű végtelen ciklusba esik*.

Minden egyes $W = PxyQ, a \in A, P, Q \in X^*, x, y \in X \cup \{\lambda\}$ pillanatnyi konfigurációhoz (aholis ha $Px \neq \lambda$, akkor Px nem kezdődhet, ha pedig $yQ \neq \lambda$, akkor yQ nem végződhet szóközzel) rendeljük hozzá a következőképp definiált $\varphi(W)$ kifejezést.

$$\varphi(W) = \begin{cases} PxyQ & , \text{ ha } x, y \in X \\ Pxab & , \text{ ha } x \in X, yQ = \lambda \\ bayQ & , \text{ ha } Px = \lambda, y \in X \\ bab & , \text{ ha } Px = yQ = \lambda. \end{cases}$$

Világos, hogy ez a hozzárendelés kölcsönösen egyértelmű. Mondjuk azt, hogy a W pillanatnyi konfigurációból a Z pillanatnyi konfiguráció ($\mathcal{M}$ -ben) *közvetlenül (vagy egy lépésben) levezethető* (jelekben $W \xrightarrow{\mathcal{M}} Z$), ha a $\varphi(W) = PxyQ, a \in A, x, y \in X, P, Q \in X^*$ előállítás esetén a következő feltételek valamelyike teljesül.

- (I) $Z \equiv Pa'xy'Q$ és $(a', y', Bal) \in \mu(a, y)$,
- (II) $Z \equiv Pxy'a'Q$ és $(a', y', Jobb) \in \mu(a, y)$,
- (III) $Z \equiv Pxa'y'Q$ és $(a', y', Helyben) \in \mu(a, y)$.

A P pillanatnyi konfigurációból a Q pillanatnyi konfiguráció ($\mathcal{M}$ -ben) *levezethető* (jelekben $W \xrightarrow{\mathcal{M}}^* Z$), ha van a pillanatnyi konfigurációknak olyan $W_0, \dots, W_t$ sorozata, hogy $W_i \xrightarrow{\mathcal{M}} W_{i+1}, i = 0, \dots, t-1$ mellett $W_0 = W, W_t = Z$. Speciálisan, minden W pillanatnyi konfigurációra feltesszük $W \xrightarrow{\mathcal{M}}^* W$ fennállását. Szokásosan, ha ki akarjuk hangsúlyozni, hogy $W \xrightarrow{\mathcal{M}}^* Z$ és $W \neq Z$, használjuk a $W \xrightarrow{\mathcal{M}}^+ Z$ jelölést is. Az $\mathcal{M}$ Turing gép által elfogadott nyelven értjük az

$L_{\mathcal{M}} = \{P \in X^* \mid a_0 P \xrightarrow{\mathcal{M}}^* W, W \text{ végkonfiguráció}\}$
nyelvet.

Megjegyzés 4.3 Jelölje $C_{\mathcal{M}}$ az $\mathcal{M}$ Turing-gép összes pillanatnyi konfigurációi halmazát. Vegyük észre, hogy a fentiekben tekintett $\mathcal{M}$ Turing-gép által elfogadott nyelv egybeesik a következő G nyelvtan által elfogadott nyelvvel: $G = (A, X, a_0, H)$, ahol $H = H_1 \cup H_2$, $H_1 = \{a \rightarrow \lambda \mid a \in A_F\}$, $H_2 = \{xay \rightarrow Z \mid a \in A, x, y \in (X \setminus \{b\}) \cup \{\lambda\}, Z \in C_{\mathcal{M}}, xay \xrightarrow{\mathcal{M}} Z\}$.

Tekintsünk egy $\mathcal{M} = (A, a_0, X, \flat, A_F, \mu)$ és egy $\mathcal{M}' = (A', a_0', X', \flat', A_F', \mu')$ Turing-gépet. Akkor mondjuk, hogy $\mathcal{M}$ *izomorf* $\mathcal{M}'$ -vel, ha az állapotok és a szalagábécé betűinek alkalmas kölcsönösen egyértelmű átjelölésével a két gép meg fog egyezni. Formálisan, ha létezik olyan $\varphi_1 : A \rightarrow A'$ és $\varphi_2 : X \rightarrow X'$ kölcsönösen egyértelmű leképezés-pár, hogy fennállnak a következők:

- (i) $\varphi_1(a_0) = a_0', \varphi_2(\flat) = \flat'$;
- (ii) minden $a \in A$ -ra $a \in A_F$ akkor és csak akkor ha $\varphi_1(a) \in A_F'$;
- (iv) valahányszor $(b, y, \text{Irány}) \in \mu(a, x)$, mindannyiszor $(\varphi_1(b), \varphi_2(x), \text{Irány}) \in \mu'(\varphi_1(a), \varphi_2(x))$ és viszont.

Érvényes a következőn tétel:

Tétel 4.7 *A monatszerkezetű nyelvek osztálya egybeesik a Turing gépek által elfogadott nyelvek osztályával.* $\square$

Amennyiben a mozgásfüggvény képhalmaza a $A \times X \times \{Bal, Jobb, Helyben\}$ halmaz (és nem annak részhalmazai halmaza), determinisztikus Turing gépről beszélünk. Érvényes a következő

Tétel 4.8 *A determinisztikus Turing gépek által elfogadott nyelvek osztálya egybeesik a nemdeterminisztikus Turing gépek által elfogadott nyelvek osztályával.* $\square$

4.4 Környezetfüggő Nyelvek s a Lineárisan Korlátolt Automaták

Ha kikötjük, hogy a nemdeterminisztikus Turing gép működése során legfeljebb egy konstansszor annyi szalagpozíciót használjon mint a startszó hossza, a lineárisan korlátolt automata fogalmához jutunk. Érvényes a következő

Tétel 4.9 *A környezetfüggő nyelvek osztálya egybeesik a lineárisan korlátolt automata által elfogadott nyelvek osztályával.* $\square$

Megjegyzés 4.4 *Mindezideig nevezetes megoldatlan probléma, hogy a determinisztikus lineárisan korlátolt automata által felismert nyelvek osztlya valódi részosztálya-e a nemdeterminisztikus lineárisan korlátolt automata által felismert nyelvek osztályának.*

Igazolható az is, hogy a lineárisan korlátolt automata alkalmas átdefiniálásával elérhető, hogy működése során legfeljebb annyi szalagpozíciót vegyen igénybe, mint a startszó hossza, s ugyanakkor az általa elfogadott nyelv egybeessen az eredeti lineárisan korlátolt automata által elfogadott nyelvvel.

4.5 A Turing-gépek megállási problémája és az algoritmikusan eldönthetetlen feladatosztályok kapcsolata

A matematikában és az informatikában gyakran foglalkozunk igen\nem problémaosztályokkal, vagyis olyan problémák osztályával, amelyeknél a megoldás mindig vagy "igen," vagy "nem." Ezeknél azt vizsgáljuk, hogy létezik-e valamilyen algoritmus (kiszámítási mód) az illető osztály összes problémájának megoldására. Church tézise szerint egy Turing-géppel ekvivalens erejű számítási modellel, a *parciális rekurzív függvényekkel* kiszámítható feladatok tekinthetők effektíve kiszámíthatónak. Ezek szerint a *Turing-gép a lehető legáltalánosabb számítási eszköznek tekinthető, azaz minden, ami effektíve kiszámítható, kiszámítható Turing-géppel is*. Ez utóbbi átfogalmazást Church-Turing tézisnek is hívják a szakirodalomban. A Church tézist, miszerint az *effektíve kiszámítható függvények osztálya megegyezik a parciális rekurzív függvények osztályával*, Alonzo Church fogalmazta meg 1936-ban. (Még azt is hozzátette, hogy aki ezzel nem értene egyet, annak ez a tézis legyen kihívás.) Ezt a tézist, illetve annak ekvivalens megfelelőit, így a Church-Turing tézist is ma a szakemberek többsége elfogadja. Tézisről, tehát egy olyan nem bizonyított állításról van szó, melynek igazolása formális matematikai eszközökkel nem lehetséges. Érvényességét a gyakorlat verifikálja.

A Church-Turing tézis értelmében az igen\nem problémaosztályt *megoldhatónak* hívjuk, ha létezik olyan rögzített algoritmus (Turing-gép), mely az osztály tetszőleges problémája mint bemenő adat esetén eredményként megadja a helyes "igen" vagy "nem" választ. 1936-ban Turing azt az akkor meglepő eredményt kapta, hogy létezik ebben az értelemben megoldhatatlan feladatosztály. Az eredeti bizonyítás helyett mi Minsky 1961-ben adott bizonyítását tárgyaljuk.

Akkor mondjuk, hogy egy Turing-gép valamely startszó hatására *megáll*, ha a startszó eleme a tekintett Turing-gép által felismert nyelvnek, azaz a startszóhoz tartozó kezdő konfigurációból kiindulva eljut egy végkonfigurációba. Mondjuk azt, hogy a *Turing-gépek megállási problémája megoldható*, ha létezik olyan $\mathcal{A}$ Turing-gép, melynek egy alkalmas kódolási algoritmussal egy tetszőleges $\mathcal{M}$ Turing-gép leírását és az $\mathcal{M}$ gép egy p input szavának kódolt alakját startszóként megadva megáll, s megállva az $\mathcal{A}$ szalagján olyan szó keletkezik, melyre alkalmas dekódolási eljárást alkalmazva egyértelműen megállapítható, hogy $p \in L_{\mathcal{A}}$, avagy sem. Más szóval, egyértelműen megállapítható, hogy a leírt $\mathcal{M}$ Turing-gép a kérdéses p szó mint startszó hatására el tud-e jutni egy végkonfigurációba, avagy sem. Ha ilyen $\mathcal{A}$ Turing-gép nem létezik, akkor mondjuk azt, hogy a *Turing-gépek megállási problémája megoldhatatlan*.

Tétel 4.10 (Turing tétel) *A Turing-gépek megállási problémája megoldhatatlan.*

Bizonyítás: Először eltekintünk a bekódolás kérdésének vizsgálatától, s feltesszük, hogy létezik olyan általános kódolási algoritmus, mely tetszőleges $\mathcal{M}$ Turing-gép és annak w startszava esetén megad a gép egy $\ell(\mathcal{M})$ leírását és a w által meghatározott alkalmas $c(\ell(\mathcal{M}), w)$ kódolt alakot mint egy rögzített ábécé feletti szót.

A bizonyítás indirekt. Tegyük fel, hogy létezik olyan $\mathcal{A}$ Turing-gép, mely a $c(\ell(\mathcal{M}), w)$ szót startszóként megkapva

(a) megáll és ekkor az "IGEN" válasz kódolt alakja olvasható a szalagján, ha $\mathcal{M}$ megáll a w inputra,

(b) megáll és ekkor a "NEM" válasz kódolt alakja olvasható a szalagján, ha $\mathcal{M}$ nem áll meg a w inputra.

Ha $\mathcal{A}$ létezik, megszerkeszthető az a $\mathcal{B}$ Turing-gép, mely az $\ell(\mathcal{M})$ startszó hatására előállítja a $c(\ell(\mathcal{M}), \ell(\mathcal{M}))$ startszót, majd ezután erre a startszóra az $\mathcal{A}$ Turing-gép működését utánozza egyetlen módosítással: valahányszor az $\mathcal{A}$ igen megállást ér el, a $\mathcal{B}$ gép egyszerű végtelen ciklusba esik. Figyelembe véve az $\mathcal{A}$ eredeti viselkedését, kapjuk:

$\mathcal{B}$ az $\ell(\mathcal{B})$ startszó hatására pontosan akkor áll meg, ha $\mathcal{B}$ a $\ell(\mathcal{B})$ startszó hatására nem áll meg. Ez nyilvánvaló ellentmondás, amivel (feltételezve, hogy létezik az univerzális kódolási eljárás,) a tétel igazolást nyert. $\square$

Tétel 4.11 *Létezik univerzális algoritmus, mely izomorfizmustól eltekintve egyértelműen megadja bármely Turing-gép és annak startszava leírását.*

Bizonyítás: A bizonyítás során alkalmazott meggondolást Gödel-számozásnak, a kódolt alakot pedig a Turing-gép Gödel számának hívjuk. Megjegyezzük, hogy ismeretesen ezen módszernél jóval hatékonyabb univerzális kódolási módszerek is. (Az ismertetett módszert Gödel eredetileg axiómarendszerek vizsgálatára használta fel.)

Ismeretes, hogy minden természetes szám sorrendtől eltekintve egyértelműen felírható prímhatalvány tényezőssé alakba (azaz páronként különböző prímszámok hatványainak szorzataként). Ezt a tényt fogjuk felhasználni. Tekintsünk egy $\mathcal{M} = (A, a_0, X, b, A_F, \mu)$ Turing-gépet, s legyen $a_0, \dots, a_{m-1}$ az állapotok egy olyan (kezdő állapottal kezdődő) felsorolása, aholis alkalmas $0 \leq k \leq m-1$ mellett $\{a_0, \dots, a_k\} = A \setminus A_F$. Jelölje továbbá $x_1, \dots, x_n$ a szalagábécé betűinek egy felsorolását, s végül legyen $d_1 = (a_0, x_1), \dots, d_{k+1} = (a_k, x_1), d_{k+2} = (a_0, x_2), \dots, d_{2(k+1)} = (a_k, x_2), \dots, d_{n(k+1)} = (a_k, x_n)$.

(Megtesszük azt a nem túl lényeges megjegyzést, hogy az $a_0, \dots, a_k$ és az $x_1, \dots, x_n$ felsorolások már egyértelműen meghatározzák a $d_1, \dots, d_{n(k+1)}$ felsorolást. Utóbbi alkalmazását csupán a módszer könnyebb megértése kedvéért vezettük be.)

Képezzük az $\mathcal{M}$ Turing-géphez és ezekhez az elrendezésekhez a

$$g_{\mathcal{M}} = 2^m \cdot 3^{k-1} \cdot 5^n \cdot 7^{u_1} \cdot 11^{v_1} \cdot 13^{w_1} \cdot \dots \cdot p_{n(k+1)+1}^{u_{n(k+1)}} \cdot p_{n(k+1)+2}^{v_{n(k+1)}} \cdot p_{n(k+1)+3}^{w_{n(k+1)}}$$

(tetszőlegesen rögzített számrendszerbeli, például tízes számrendszerbeli) természetes számot, aholis 2 hatványa az állapotok számát, 3 hatványa a nem-végállapotok számát, 5 hatványa a szalagábécé betűinek számát jelöli, s minden további $p_{i+1}^{u_i}, p_{i+2}^{v_i}, p_{i+3}^{w_i}, i = 1, \dots, n(k+1)$ prímhatalvány-hármas esetén, aholis alkalmas $a_s \in \{a_0, \dots, a_k\} (= A \setminus A_F), x_t \in \{x_1, \dots, x_n\}$ mellett $d_i = (a_s, x_t)$ áll fenn, $(u_i, v_i, w_i) = (0, 0, 0)$ ha $\mu(a_s, x_t)$ nincs értelmezve, ha pedig $\mu(a_s, x_t)$ értelmezve van, akkor $\mu(a_s, x_t) = (a_{s'}, x_{t'})$, Merre) esetén $u_i = s' + 1, v_i = t'$, továbbá, mondjuk, $w_i = 1$ ha Merre=Bal, $w_i = 2$ ha Merre=Jobb, illetve $w_i = 3$ ha Merre=Helyben. Az így meghatározott $g_{\mathcal{M}}$ számot az

$\mathcal{M}$ Turing-gép *Gödel-számának* fogjuk hívni. Amennyiben a w startszó $w = x_{i_1} \dots x_{i_d}$ alakú, a $c(\ell(\mathcal{M}), w)$ kódolt alak legyen (a rögzített számrendszerbeli)

$$g_{\mathcal{M}}(p_{n(k+1)+4})^{i_1} \dots (p_{n(k+1)+d})^{i_d}$$

természetes szám. Látható, hogy a szóban forgó kódolás (izomorfizmustól eltekintve) egyértelmű, s ha a nyert $c(\ell(\mathcal{M}), w)$ természetes szám például tízes számrendszerben van megadva, akkor egy alkalmas tizenegy bemenő jeles gépnek inputként megadható (tizenegyedik bemenő jel a szóköz jel). $\square$

5. Környezetfüggetlen nyelvek

Ebben a fejezetben a 2-típusú nyelvek sajátosságait tárgyaljuk.

5.1 Chomsky-féle normál alak

Egy nyelvtant λ -mentesnek nevezünk, ha a szabályok jobb oldalán egyáltalán nem fordul elő a λ .

Itt jegyezzük meg, hogy minden 2-típusú $\lambda \notin L$ nyelv esetén megadható olyan 2-típusú λ -mentes grammatika ami L -et generálja (lásd Üresszó Lemma). Ha viszont $\lambda \in L$, akkor igaz az, hogy megadható olyan 2-típusú λ -mentes G grammatika ami a $L \setminus \{\lambda\}$ nyelvet generálja. (Ilyenkor G szabályrendszerét kiegészítve az $S \rightarrow \lambda$ szabállyal kapunk egy L -et generáló nyelvtant.)

Egy környezetfüggetlen nyelvtanról azt mondjuk, hogy *Chomsky-féle normál alakú*, ha minden szabálya $X \rightarrow a$ vagy $X \rightarrow YZ$ alakú, ahol $X, Y, Z \in V_N$ és $a \in V_T$.

Tétel 5.1 *Minden λ -mentes környezetfüggetlen G nyelvtanhoz van olyan vele ekvivalens környezetfüggetlen G' nyelvtan, amely Chomsky-féle normál alakú.*

Bizonyítás: Hozzuk a nyelvtanunkat normális alakra, ekkor terminális jel csak $X \rightarrow a$ alakú szabályokban fog előfordulni ($X \in V_N$, $a \in V_T$). Ekkor az összes többi szabályunk $X \rightarrow P$ alakú ($X \in V_N$, $P \in V_N^* \setminus \{\lambda\}$). Tekintsük ezek közül azokat a szabályokat, amelyekben $|P| > 2$. Egy ilyen $X \rightarrow Y_1 Y_2 \dots Y_k$ ($k > 2$) alakú szabályt helyettesíthetünk az

$$\begin{aligned} X &\rightarrow Y_1 Z_1 \\ Z_1 &\rightarrow Y_2 Z_2 \\ &\dots \\ Z_{k-2} &\rightarrow Y_{k-1} Y_k \end{aligned}$$

szabályok halmazával, ahol $Z_1, Z_2, \dots, Z_{k-2}$ újonnan bevezetett nemterminálisok. Ezt a helyettesítést minden kettőnél hosszabb jobb oldalú szabályra külön-külön végrehajtva a nyelvtanban csak az alábbi háromféle szabály marad:

1. $X \rightarrow a$,
2. $X \rightarrow Y$,
3. $X \rightarrow YZ$.

Az így kapott $G_1 = \langle V'_N, V_T, S, H_1 \rangle$ nyelvtanra $L(G) = L(G_1)$. Most ki kell küszöbölni a H_1 -ből az $X \rightarrow Y$ alakú szabályokat. Defináljuk minden $X \in V'_N$ változóhoz a következő halmazokat:

- $U_1(X) = \{X\}$
- $U_{i+1}(X) = U_i(X) \cup \{Y \in V'_N \mid \exists Z \in U_i(X) \mid Z \rightarrow Y \in H_1, i > 0\}$

Ekkor V'_N véges volta miatt létezik olyan minimális k index, hogy $U_k(X) = U_{k+l}(X)$, ha $l = 1, 2, \dots$. Jelöljük minden $X \in V'_N$ nemterminális jelhez tartozó $U_k(X)$ -t $U(X)$ -el. Most tetszőleges $X, Y \in V'_N$ változókra $X \Rightarrow^* Y$ pontosan akkor, ha $Y \in U(X)$. Ezek után definiáljuk a H' szabályhalmazt a következőképpen:

- $S \rightarrow a \in H'$, ha van olyan $A \in V'_N$, hogy $A \rightarrow a \in H_1$ és $A \in U(S)$.
- $X \rightarrow YZ \in H'$, ha van olyan $A, B, C \in V'_N$, melyre teljesülnek a következők: $A \rightarrow BC \in H_1$, $A \in U(X)$, $Y \in U(B)$, $Z \in U(C)$. Továbbá a H_1 szabályai közül kerüljenek H' -be az $A \rightarrow a$ alakúak, ahol $A \in V'_N$, $a \in V_T$.

Az így kapott $G' = \langle V'_N, V_T, S, H' \rangle$ nyelvtanra $L(G) = L(G')$, mivel az $X \rightarrow Y$ szabályok alkalmazását az előbbi két csoportba tartozó szabályok alkalmazásával biztosítjuk és fordítva. $\square$

5.2 Levezetési gráf

Környezetfüggetlen nyelvtannál minden levezetés nagyon egyszerűen ábrázolható egy irányított gráf segítségével. Tekintsünk egy $S \Rightarrow^* P$ levezetést, ahol $P \in V_T^*$. A gráf csúcsainak nemterminális, illetve terminális szimbólumokat feleltethetünk meg: a fa gyökeréhez S -t, leveleihez rendre terminális szimbólumokat, a közbülső csúcsaihoz pedig nemterminális jeleket rendelhetünk. Az egy csúcsból kiinduló élek annak a helyettesítési szabálynak az alkalmazását jelölik, amelynek bal oldalán az élek közös kiindulási pontjában található nemterminális jel áll, a jobb oldalán pedig az élek végpontjaiban található nemterminális, illetve terminális jelek sorozata (az egyes éleket balról jobbra véve sorra). Azt, hogy egyes szabályokat milyen sorrendben alkalmazunk, nem mindig tudjuk egyértelműen rekonstruálni. Egy levezetési gráf mélységén a benne az S gyökértől induló és levélelemhez tartó irányított utak hosszának maximumát értjük.

A levezetési gráf fogalma lehetővé teszi, hogy választ keressünk a formális nyelvek elméletének egyik fontos problémájára, az úgynevezett szóproblémára. Ezt a problémát a következő formában szokás megfogalmazni: egy adott L nyelv és egy adott P szó esetén milyen feltételek mellett dönthető el algoritmikusan az, hogy $P \in L$ reláció teljesül-e vagy sem. Könnyű észrevenni, hogy végtelen nyelvek esetén végtelen sok levezetési gráf van, sőt azok mélysége sem korlátos. A szóprobléma eldöntése ezeknek a gráfoknak az egyenkénti megvizsgálását jelentené, ami általában algoritmikusan nem megoldható. Azonban speciális esetekben, például szó hosszát nem csökkentő nyelvtanok (lásd 6. fejezet is) esetén a levezetési gráfoknak csak egy véges részhalmazát kell csak a vizsgálatunkba bevonni, ami garantálja a probléma algoritmikus eldönthetőségét.

Példa 5.1 Legyen $G = \langle \{S, A, B\}, \{a, b, c\}, S, H \rangle$, ahol $H = \{S \rightarrow ABc, A \rightarrow aB, A \rightarrow Bc, B \rightarrow aAc, B \rightarrow bc\}$. Ebben a nyelvtanban megadható a következő levezetés: $S \Rightarrow ABc \Rightarrow AaAcc \Rightarrow BcaAcc \Rightarrow BcaaBcc \Rightarrow bccaaBcc \Rightarrow bccaabccc$.

Igaz a következő pumpálós lemma:

Tétel 5.2 (Bar-Hillel lemma) Bármely L környezetfüggetlen nyelvhez létezik $p, q \in \mathbb{N}$ úgy, hogy $\forall P \in L$ esetén $|P| > p$ szóra $P = UXWYZ$ alakban írható, ahol $|XWY| \leq q$, $XY \neq \lambda$ és $UX^iWY^iZ \in L$ minden $i \geq 0$ egész számra.

Bizonyítás: Röviden ez a lemma azt mondja ki, hogy a nyelvben minden elég hosszú szóhoz végtelen sok (rokon szerkezetű) további szó található. A bizonyításnál elég λ -mentes nyelvtanra szorítkoznunk. Feltételezzük továbbá, hogy a nyelvtanunk Chomsky-féle normál alakban adott. Ha egy $P \in L(G)$ szónak a levezetése olyan fával ábrázolható, amelyben a leghosszabb út k hosszúságú, akkor a Chomsky-féle normál alak miatt $|P| \leq 2^k$. Tegyük fel, hogy a V_N elemeinek száma n és legyen $p = 2^n, q = 2^{n+1}$. Ha most $P \in L$ és $|P| > p$, akkor az $S \Rightarrow^* P$ levezetésfájában a leghosszabb útnak n -nél hosszabbnak kell lennie. Vegyük ennek az útnak az utolsó $n+1$ hosszúságú szakaszát. Lesz olyan $A \in V_N$ változó, amely ezen a szakaszon legalább kétszer előfordul. Vegyük ennek a változónak két ilyen előfordulását. Ezek közül az elsőhöz (az S -hez közelebb fekvőhöz) tartozó részfa végpontjainak megfelelő szó legyen Q , a másik részfa végpontjainak megfelelő szó legyen W . Ezekre nyilván $A \Rightarrow^* Q$ és $A \Rightarrow^* W$, továbbá a W részszoja Q -nak, tehát $Q = XWY$ valamely $X, Y \in V_T^*$ szavakra. Emellett nyilván $P = UQZ$ is teljesül valamilyen $U, Z \in V_T^*$ szavakra. Az A változó szóban forgó előfordulásainak a megválasztása miatt $|Q| \leq 2^{n+1}$. Másrészt $S \Rightarrow^* UAZ$ és $A \Rightarrow^* XAY$ is fennáll, tehát tetszőleges $i \geq 0$ egész számra $S \Rightarrow^* UX^iWY^iZ$. Itt nem lehet X és Y mindkettő λ , mert az $A \Rightarrow^* XAY$ levezetés legalább egy lépést tartalmaz, tekintve, hogy az A -nak két különböző előfordulásáról van szó. Akkor pedig ebben a levezetésben az első lépés csak egy $A \rightarrow BC$ alakú szabály alkalmazása lehet, s ezért $|XY| \geq 1$, miután a nyelvtanunk λ -mentes. Ezzel befejeztük a lemma bizonyítását. $\square$

Következmény 5.2 Az $\{a^i b^i c^i \mid i \geq 1\}$ nyelv nem környezetfüggetlen.

Bizonyítás: Ha volna olyan környezetfüggetlen nyelvtan, amely generálja ezt a nyelvet, akkor a lemma szerint volna olyan $P = UXWYZ$ szó, hogy $XY \neq \lambda$, és minden $i \geq 0$ -ra UX^iWY^iZ ehhez a nyelvhez tartozik. Az $UX^iWY^iZ = a^j b^j c^j$ összefüggést azonban az U, X, W, Y, Z szavak semmilyen konkrét választása mellett sem lehet végtelen sok i -re és j -re kielégíteni. Ugyanis X és Y közül egyik sem tartalmazhat többféle betűt. Ha viszont csak egyfélét tartalmaznak, akkor az a, b, c közül legalább az egyiknek a kitevője i -től független lesz. $\square$

5.3 Lineáris nyelvtanok

Most a környezetfüggetlen nyelvek (illetve nyelvtanok) speciális részhalmazait vizsgáljuk. Emlékeztetőül, egy környezetfüggetlen nyelvtant *lineárisnak* mondunk, ha minden szabálya $P \rightarrow a$, illetve $P \rightarrow aQb$ alakú, ahol $P, Q \in V_N$, $a, b \in V_T^*$. Továbbá *bal-lineárisnak* (*jobb-lineárisnak*) nevezzük, ha minden a (minden b) azonos λ -val. A definíció szerint az, hogy egy nyelvtan jobb-lineáris, ugyanazt jelenti, hogy 3-típusú.

Tétel 5.3 Minden bal-lineáris nyelvtan által generált nyelv 3-típusú.

Bizonyítás: Legyen $G = \langle V_N, V_T, S, H \rangle$ bal-lineáris nyelvtan, legyen $V_N = \{S, A_1, A_2, \dots, A_n\}$. Az általánosság megszorítása nélkül feltehetjük, hogy S nem szerepel egyetlen

szabály jobb oldalán sem. (Ellenkező esetben ugyanis egy új S_0 nemterminális és $S_0 \rightarrow S$ szabály bevezetésével ez elérhető.) Szerkesszük meg a $G' = \langle V'_N, V_T, S', H' \rangle$ nyelvtant úgy, hogy $V'_N = \{S', B_1, \dots, B_n\}$ és a H' -beli szabályok a következők:

- $S' \rightarrow P \in H'$, ha $S \rightarrow P \in H$ és $P \in V_T^*$
- $S' \rightarrow PB_k \in H'$, ha $A_k \rightarrow P \in H$ és $P \in V_T^*$
- $B_l \rightarrow QB_k \in H'$, ha $A_k \rightarrow A_lQ \in H$ és $Q \in V_T^*$
- $B_l \rightarrow Q \in H'$, ha $S \rightarrow A_lQ \in H$ és $Q \in V_T^*$

Az így kapott G' nyelvtan nyilván 3-típusú, és belátható, hogy $L(G) = L(G')$. Legyen $W \in L(G)$. Ha $S \rightarrow W \in H$, akkor $S' \rightarrow W \in H$. Ha pedig létezik egy

$$S \Rightarrow A_{i_1}P_1 \Rightarrow A_{i_2}P_2P_1 \Rightarrow \dots \Rightarrow A_{i_{m-1}}P_{m-1} \dots P_1 \Rightarrow P_mP_{m-1} \dots P_1$$

alakú levezetés G -ben, ahol $P_1, P_2, \dots, P_m \in V_T^*$, akkor G' -ben megadható egy

$$S \Rightarrow P_mB_{i_{m-1}} \Rightarrow P_mP_{m-1}B_{i_{m-2}} \Rightarrow \dots \Rightarrow P_m \dots P_2B_1 \Rightarrow P_mP_{m-1} \dots P_2P_1$$

alakú levezetés, tehát $W \in L(G')$. Megfordítva ugyanígy belátható, hogy $L(G') \subseteq L(G)$. $\square$

A $P = a_1a_2 \dots a_n$ szó tükörképén a $P^{-1} = a_n \dots a_2a_1$ szót értjük. Nyilván $(P^{-1})^{-1} = P$.

Egy L nyelv tükörképén az

$$L^{-1} = \{P \mid \exists Q \in L, P = Q^{-1}\}$$

nyelvet értjük.

Következmény 5.3 A $\mathcal{L}_3$ nyelv osztály a tükrözésre nézve zárt.

Bizonyítás: Ha G egy 3-típusú nyelvtan, és $P \in L(G)$, akkor P^{-1} tükörszó levezethető lesz abban a nyelvtanban, amelyet G -ből úgy nyerünk, hogy minden szabály jobb oldalát a tükörképével helyettesítjük. Az így nyert nyelvtan bal-lineáris lesz. $\square$

6. Környezetfüggő nyelvek

Egy G generatív nyelvtant *hosszúságot nem csökkentőnek* mondunk, ha minden $P \rightarrow Q \in H$ szabályra igaz, hogy $|P| \leq |Q|$.

Nyilván minden λ -mentes környezetfüggő nyelvtan hosszúságot nem csökkentő. Másrészt teljesül a következő tétel:

Tétel 6.1 *Minden hosszúságot nem csökkentő nyelvtanhoz megadható egy vele ekvivalens környezetfüggő nyelvtan.*

Bizonyítás: Legyen $G = \langle V_N, V_T, S, H \rangle$ egy hosszúságot nem csökkentő nyelvtan. Feltehetjük, hogy terminális jelek csak az $A \rightarrow a$ alakú szabályokban fordulnak elő. Legyen $P \rightarrow Q \in H$ egy tetszőleges szabálya G -nek. Ha $|P| = 1$, akkor a szabályunk a kívánt alakú. Ha $|P| > 1$, akkor P és Q a következő alakban írhatók fel:

$$P = X_1 X_2 \dots X_m, \quad Q = Y_1 Y_2 \dots Y_n$$

Ekkor $2 \leq m \leq n$, és $X_i, Y_j \in V_N$. Vezessünk be m darab új változót e szabály helyettesítéséhez ($Z_1, \dots, Z_m \notin (V_N \cup V_T)$), és vegyük az alábbi szabályokat:

$$X_1 X_2 \dots X_m \rightarrow Z_1 X_2 \dots X_m$$

$$Z_1 X_2 \dots X_m \rightarrow Z_1 Z_2 \dots X_m$$

...

$$Z_1 Z_2 \dots Z_{m-1} X_m \rightarrow Z_1 Z_2 \dots Z_{m-1} Z_m Y_{m+1} \dots Y_n$$

$$Z_1 Z_2 \dots Z_{m-1} Z_m Y_{m+1} \dots Y_n \rightarrow Y_1 Z_2 \dots Z_{m-1} Z_m Y_{m+1} \dots Y_n$$

...

$$Y_1 Y_2 \dots Y_{m-1} Z_m Y_{m+1} \dots Y_n \rightarrow Y_1 Y_2 \dots Y_{m-1} Y_m Y_{m+1} \dots Y_n.$$

Belátható, hogy $P \rightarrow Q$ szabályt ezekkel az új szabályokkal helyettesítve egy, az eredetivel ekvivalens nyelvtant kapunk. Ezek az új szabályok pedig megfelelnek az 1-típusú nyelvtan előírásainak. Ezt a helyettesítést minden nem-környezetfüggő szabályra külön-külön elvégezhetjük. $\square$

Következmény 6.1 *Az $\{a^i b^i c^i \mid i = 1, 2, \dots\}$ nyelv környezetfüggő, tehát $\mathcal{L}_2 \subset \mathcal{L}_1$.*

Bizonyítás: A nyelvet a 3.1. példában egy hosszúságot nem csökkentő nyelvtan segítségével adtuk meg, így az környezetfüggő. Másrészt az 5.2. következmény alapján nyilvánvaló, hogy a nyelv nem környezetfüggetlen. $\square$

Egy nyelvtan *Kuroda-féle normál alakban* van, ha szabályai a következő alakban vannak: $P \rightarrow a$, $P \rightarrow Q$, $P \rightarrow QR$, $PQ \rightarrow RS$, ahol $a \in V_T$ és $P, Q, R, S \in V_N$.

Tétel 6.2 *Minden hosszúságot nem csökkentő nyelvtanhoz létezik vele ekvivalens, Kuroda-féle normál alakban lévő nyelvtan.*

Bizonyítás: Legyen $G = \langle V_N, V_T, S, H \rangle$ egy hosszúságot nem csökkentő nyelvtan. Feltehetjük, hogy a nyelvtan normális alakban van. Legyen $P \rightarrow Q \in H$ $P = X_1 \dots X_m$ $Q = Y_1 \dots Y_n$. Világos, hogy a nyelvtan tulajdonságai miatt $m \leq n$.

- $m = 1, n = 1, 2$ esetén a szabályunk megfelelő alakban van.
- $m = 2, n = 2$ esetben úgyszintén.
- $m = 1, n > 2$ esetben járjunk el úgy a $P \rightarrow QR$ szabály segítségével, ahogy a Chomsky-féle normál alaknál tettük.

- $m \geq 2, n > 2$ esetben vezessünk be az adott szabály helyettesítésére az új $Z_1, Z_2, \dots, Z_{n-1}$ nemterminális jeleket és vegyük a következő szabályokat:

$$X_1 X_2 \rightarrow Y_1 Z_2$$

$$Z_2 X_3 \rightarrow Y_2 Z_3$$

...

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

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

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

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

Ezekkel a szabályokkal helyettesítve az eredeti szabályokat, a kiinduló nyelvtanokkal ekvivalens nyelvtant kapunk. $\square$

Következmény 6.2 Minden λ -mentes környezetfüggetlen nyelvtanhoz megadható vele ekvivalens nyelvtan, amely Kuroda-féle normál alakban van. $\square$

A következő észrevétel Révész Györgytől származik:

A Kuroda-féle normál alakban az $AB \rightarrow CD$ alakú szabályok természetesen helyettesíthetők környezetfüggő szabályokkal, éspedig a következőkkel: $AB \rightarrow AB'$, $AB' \rightarrow A'B'$, $A'B' \rightarrow A'D$, $A'D \rightarrow CD$. Itt A' , B' újonnan bevezetett nemterminálisok.

7. Eldönthetőség

A formális nyelvek elméletének egyik jellegzetes problémája annak eldöntése, hogy egy adott P szó beletartozik-e egy adott nyelvbe. Az a megfigyelés, hogy a környezetfüggő nyelvtanok a szó hosszát nem csökkentik, azt eredményezik, hogy elegendő az adott nyelvtanhoz tartozó levezetési gráfok halmazának azon véges részhalmazát megvizsgálni, amely gráfokra a levélelemekhez tartozó szavak hossza nem nagyobb, mint $|P|+1$. Ilyen fa (effektíven) véges sok van, így ez a probléma eldönthető környezetfüggő nyelvek esetén. A 0-típusú nyelvek esetén azonban a $P \in L$ reláció általában algoritmikusan nem eldönthető. Ez a fejezet ezt a problémakört vizsgálja meg részletesebben. (Lásd a 4.5. fejezetet is.)

Egy L nyelvet *rekurzív*nek nevezünk, ha a $P \in L$ tartalmazási probléma algoritmikusan eldönthető.

Egy L nyelvet *rekurzíve felsorolható*nak nevezünk, ha van olyan eljárás, amely az összes $P \in L$ szót valamilyen sorrendben (esetleg ismétlésekkel) felsorolja.

Megjegyzés 7.1 Minden rekurzív nyelv nyilván rekurzíve felsorolható. Nem kell ugyanis mást tennünk, mint rendre előállítani az összes $P \in V_T^*$ szót, miközben minden egyes új szó előállítása után alkalmazzuk rá az eldöntési algoritmust, és bele vesszük a felsorolásba, ha igen választ kapunk, egyébként elhagyjuk.

Tétel 7.1 Egy L nyelv akkor és csak akkor rekurzív, ha mind az L mind az $\bar{L}$ rekurzíve felsorolható.

Bizonyítás: Ha L rekurzív, akkor a $P \in L$ probléma algoritmikusan eldönthető, akkor ugyanez áll az $\bar{L}$ nyelvre is, hiszen $P \in L$ akkor és csak akkor teljesül, ha $P \notin \bar{L}$. Eszerint ugyanazt az eldöntési algoritmust használhatjuk az $\bar{L}$ -re, azzal a különbséggel, hogy amit L esetén elfogadtunk azt most nem, és fordítva.

Másik irány. Tegyük fel, hogy mind az L mind az $\bar{L}$ rekurzíve felsorolható. Kombináljuk az L és az $\bar{L}$ felsorolását biztosító eljárásokat úgy, hogy váltakozva hol az egyikkel, hol a másikkal állítunk elő egy-egy szót, miáltal egy olyan $P_0, P_1, \dots$ felsorolást kapunk, ahol $P_{2i} \in L$, $P_{2i+1} \in \bar{L}$ minden $i = 0, 1, 2, \dots$ értékre. Mivel a felsorolás teljes, ezért a P keresett szónak valahol elő kell fordulnia, így csak azt kell eldöntenünk, hogy páros vagy páratlan pozícióban fordul-e elő, így tulajdonképpen egy döntési algoritmust adtunk meg az L -re. $\square$

Tétel 7.2 Minden 1-típusú nyelv rekurzív, és minden 0-típusú nyelv rekurzíve felsorolható. $\square$

Tétel 7.3 Van olyan rekurzív nyelv, amely nem 1-típusú.

Bizonyítás: Minden 1-típusú nyelvtant megadhatunk úgy, hogy felsoroljuk a szabályait. Feltehetjük, hogy a nyelvtan normális alakú. A terminális ábécét a szabályok implicit módon definiálják. Ezek után tekintsük azt az 1-típusú nyelvtant, amelynek terminális ábécéje $V_T = \{0, 1\}$. Kódoljuk a nemterminális jeleket a $01, 011, 0111, \dots$

jelsorozatokkal, ahol az S kezdőszimbólumnak mindig a 01 kód feleljen meg. Kódoljuk továbbá a 0 és az 1 terminális jeleket 00 és 001 szavakkal, a $\rightarrow$ és a $\#$ (elválasztó-) jeleket 0011, illetve 00111 szavakkal. Ezek alapján minden nyelvtant kifejezhetjük egy $\{0,1\}^*$ -beli szóval. Rendezzük az összes $\{0,1\}^*$ -beli szót valamilyen módon (ezzel a nyelvtanokat is sorbarendeztük).

Legyen W_i a $\{0,1\}^*$ megadott rendezés értelmében az i -edik eleme a $\{0,1\}^*$ -nak és definiáljuk az L nyelvet úgy, hogy

$$L = \{W_i \mid W_i \notin L(G_i)\}$$

ahol G_i az i -edik nyelvtant jelenti. Az L nyelv rekurzív, mert a $P \in L$ véges sok lépésben eldönthető. Ugyanis egy adott P -ről eldönthetjük, hogy hányadik eleme a rendezett halmaznak. Legyen ez i , akkor meghatározzuk a G_i -t, ami úgy történik, hogy rendre előállítjuk a $\{0,1\}^*$ -beli szavakat (célszerűen ugyanabban a rendezési sorrendben, mint amit az előbb használtunk), és mindegyikről egyenként eldöntjük, hogy ez a kódolás megfelel-e egy 1-típusú nyelvtannak vagy sem. Ez szintén megtehető véges lépésben. Miután meghatároztuk a G_i -t, eldönthetjük, hogy $P \in L(G_i)$ teljesül-e, ami szintén véges számú lépésben megtehető.

Most belátjuk, hogy L nem 1-típusú. Ha ugyanis az volna, akkor volna egy olyan G_j a fenti felsorolásban, amelyre $L = L(G_j)$. Tekintsük a $\{0,1\}^*$ j -edik elemét, amit W_j -vel jelöltünk. Ha $W_j \in L(G_j)$, akkor az L definíciója értelmében $W_j \notin L = L(G_j)$ ami ellentmondás. Ha $W_j \notin L(G_j)$, akkor $W_j \in L = L(G_j)$ szintén ellentmondás, tehát L nem 1-típusú. $\square$

Tétel 7.4 *A rekurzíve felsorolható nyelvek osztálya megegyezik a 0-típusú nyelvek osztályával.* $\square$

8. Szintaktikai elemzés

A gyakorlati alkalmazásoknál általában nem elég egy formális nyelv felismerési problémáját csupán eldöntési problémának tekinteni, hanem egy $P \in L$ szóról azt is meg kell állapítanunk, hogy az hogyan vezethető le az adott nyelvtanban. A szintaktikai elemzés tehát az adott nyelv szavainak a nyelvtani szerkezetét van hivatott feltárni. Programozási nyelvek esetén is lényeges szerepe van a szintaktikai elemzésnek, mert a programok értelmezése a nyelvtani szerkezet ismeretére épül. A következőkben a környezetfüggetlen nyelvek esetén vizsgáljuk meg a szintaktikai elemzés lehetőségét.

Tegyük fel, hogy az adott környezetfüggetlen nyelvtan Chomsky-féle normál alakra van hozva. Példaként tekintsük a következő nyelvtant:

$$G = \langle \{S, A, B, C, D\}, \{a, b\}, S, H \rangle,$$

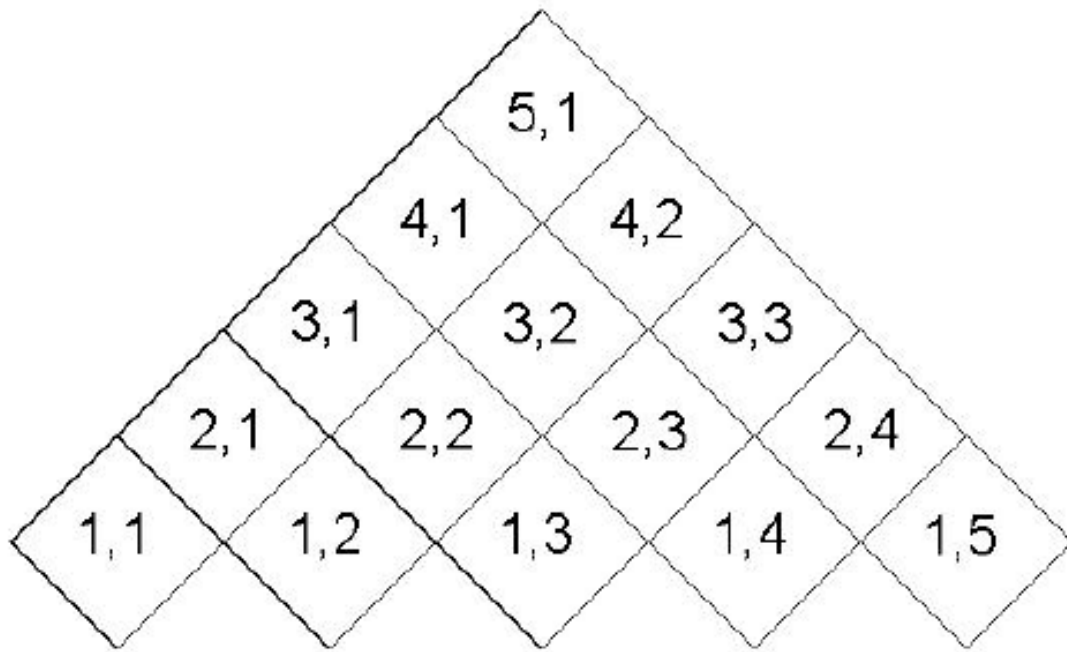
ahol a H -beli szabályok a következők: $S \rightarrow AB$, $S \rightarrow CD$, $S \rightarrow CB$, $S \rightarrow SS$, $A \rightarrow BC$, $C \rightarrow DD$, $B \rightarrow SC$, $C \rightarrow b$, $A \rightarrow a$, $D \rightarrow BA$, $B \rightarrow b$.

A Cocke-Younger-Kasami-féle (, vagy röviden CYK-) felismerő algoritmus ezeknek a szabályoknak az alapján egy tetszőleges bemenő szóhoz igyekszik megkonstruálni a megfelelő levezetési gráfot. A Chomsky-féle normál alak miatt ez — a végpontokba befutó élektől eltekintve — egy bináris fa lesz, amelynek a méretét a bemenő szó hosszúsága fogja meghatározni. Ha például a bemenő szó mindössze négybetűs, akkor az összes lehetséges levezetési fának a mélysége maximum 5.

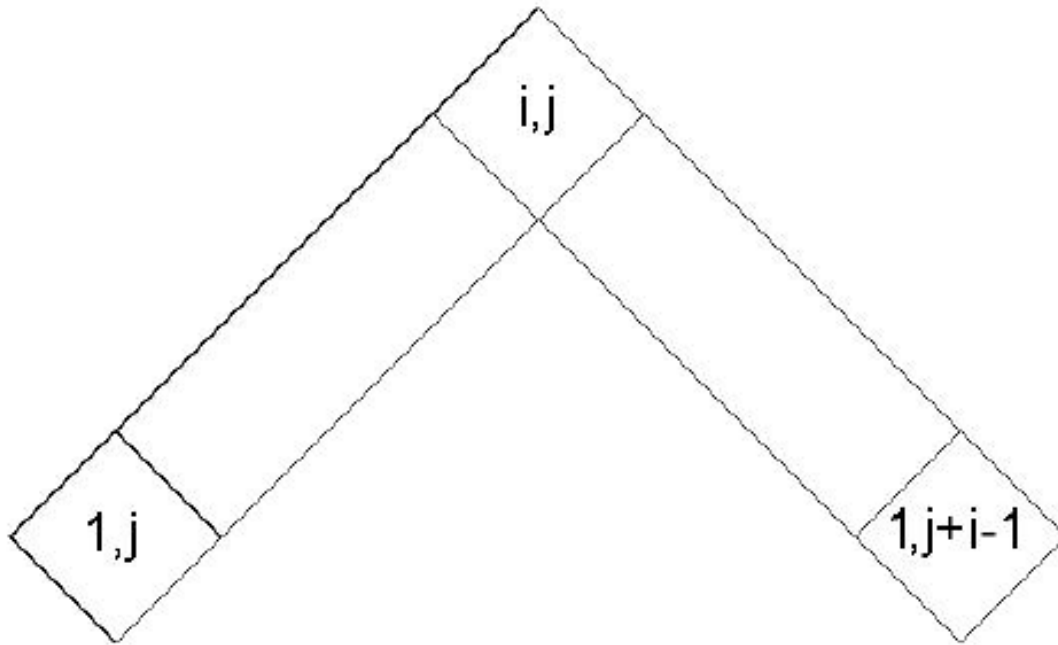
Egy n hosszúságú bemenő szó esetében a levezetési fa összes lehetséges csúcsai egy olyan háromszög mátrixba rendezhetők, amelynek a legalsó sora n -elemű, és a többi sora eggyel kevesebb elemű, mint a közvetlen alatta levő sor. Egy adott fa természetesen nem használhatja fel az összes mezőjét ennek a mátrixnak, csak az $n = 2$ esetben.

A felismerési algoritmus ennek a háromszög mátrixnak az elemeit határozza meg úgy, hogy minden szögponthoz beírja azokat a nemterminális jeleket, amelyekből a kérdéses szögponthoz tartozó részháromszög végpontjainak megfelelő bemenő szórészlet levezethető. Ha a mátrix kitöltését befejeztük, akkor már csak azt kell megnéznünk, hogy a háromszög legfelső csúcsához be van-e írva az S . Ha igen, akkor az adott bemenő szó levezethető az S -ből, tehát hozzátartozik az adott nyelvtan által generált nyelvhez. Ellenkező esetben viszont biztos, hogy nem tartozik hozzá, mert a kitöltés során minden lehetőséget kimerítettünk.

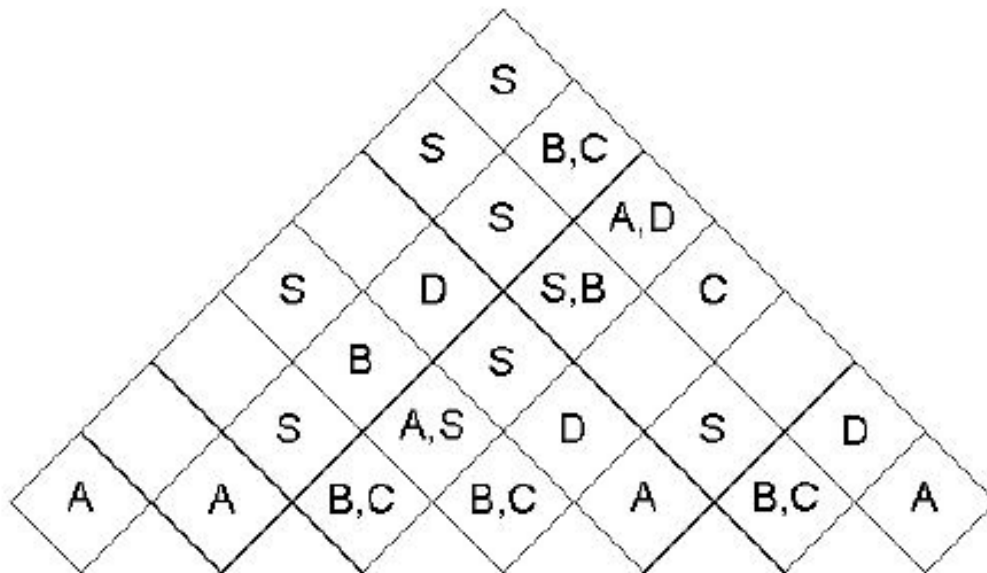
A szóban forgó háromszög mátrixot felismerési mátrixnak nevezzük, és minden elemének tulajdonképpen egy rekeszt feleltetünk meg, ahová a változókat beírjuk. (Egy-egy rekeszbe a nemterminálisok egy-egy halmaza fog kerülni, mégpedig pontosan azok, amelyekből a mező alatti háromszögnek megfelelő terminális szó levezethető.) Ezeket a rekeszeket az 1. ábrán megadott módon indexeljük. (Az ábrán a felismerési mátrix indexelését $n = 5$ esetre láthatjuk.) A kitöltést az alsó (= első) sornál kezdve soronként balról-jobbra végezzük, a sorokat pedig alulról felfelé történő sorrendben töltjük ki.



Legyen a megvizsgálandó bemenő szó $w = a_1 a_2 \dots a_n$. Az első sor i -edik rekeszébe beírjuk az X -et, ha szerepel egy $X \rightarrow a_i$ szabály a nyelvtanban. (Ugyanabba a rekeszbe általában több változót is beírhatunk.) A mátrix további sorainak a kitöltésekor, minden egyes rekesz tartalmának a meghatározása az adott rekesz alatti részháromszög befogói mentén található rekeszek tartalmának a felhasználásával történik. Pontosabban az (i, j) rekesz kitöltéséhez az $(1, j)$ rekeszben található változókat egyenként párosítjuk az $(i - 1, j + 1)$ rekeszben találhatókval, majd ugyanezt tesszük a $(2, j)$ és $(i - 2, j + 2), \dots$, és végül az $(i - 1, j)$ és $(1, j + i - 1)$ rekesz-párokkal. Ha egy ilyen párosítás alkalmával olyan párt találunk, amely szerepel egy szabály jobb oldalán (a megfelelő sorrendben), akkor a szabály bal oldalán álló változót felvesszük az (i, j) rekeszbe. (A feldolgozás menetét a következő ábra szemlélteti.)



A példánkban az *aabbaba* bemenő szóhoz így módon megkonstruált felismerési mátrixot az alábbi ábra szemlélteti.



$(U \in (V_N \cup V_T)^*)$. Az előbbiek értelmében ugyanis ez azt jelenti, hogy $S \rightarrow U \in H$, és $U \Rightarrow^* a_1 \dots a_n$.

Lássuk most az algoritmus pontos leírását. Az egyszerűség kedvéért szorítkozzunk először λ -mentes környezetfüggetlen nyelvtanokra. Jelöljük a bemenő szó i -edik betűjét a_i -vel, tehát legyen $P = a_1 \dots a_n$. Legyen továbbá $G = \langle V_N, V_T, S, H \rangle$ egy λ -mentes környezetfüggetlen nyelvtan, és jelöljük a 4. ábra szerint indexelt felismerési mátrix i -edik sorának j -edik elemét $F_{i,j}$ -vel. Kezdetben a felismerési mátrix minden eleme üres, és az algoritmus mindegyikbe egy vagy több pontozott szabályt írhat be, vagy üresen is hagyhatja. A felismerési mátrix kitöltését oszloponként alulról fölfelé végezzük, kivéve a fődiagonálisban lévő elemeket, amelyeket az adott oszlopban utoljára töltünk ki. Az egyes oszlopokat balról jobbra vesszük sorra. (A mondottakból következik, hogy $F_{n,n}$ kitöltésére tulajdonképpen nincs szükség.) Az Early-féle algoritmus lépései a következők.

1. Minden $S \rightarrow W \in H$ szabályra legyen $S \rightarrow .W \in F_{0,0}$, és legyen $j = 0$.
2. Ha $B \rightarrow U \in H$ és van olyan $k \leq j$, hogy $A \rightarrow X.BY \in F_{k,j}$ valamely $X, Y \in (V_N \cup V_T)^*$ -ra, akkor legyen $B \rightarrow .U \in F_{j,j}$. Ekkor az újonnan beírt szabályokra is megvizsgálni az adott feltételek fennállását.
3. Ha $j < n$, akkor növeljük j értékét eggyel, és legyen $i = j - 1$. Ha $j = n$, akkor készen vagyunk a mátrix kitöltésével.
4. Legyen $A \rightarrow Xa_j.Y \in F_{i,j}$, ha $A \rightarrow X.a_jY \in F_{i,j-1}$.
5. Legyen $A \rightarrow XB.Y \in F_{i,j}$, ha van olyan k , melyre $i \leq k < j$, és $A \rightarrow X.BY \in F_{i,k}$, $B \rightarrow U \in F_{k,j}$ valamely $U \in (V_N \cup V_T)^*$ -ra.
6. Ha $i > 0$, akkor csökkentsük az i értékét eggyel, és térjünk vissza a negyedik lépésre. Ha $i = 0$, akkor pedig térjünk vissza a második lépésre.

Ezzel az algoritmussal tehát megadtuk a felismerési mátrixot, amelyből a P szó levezetése viszonylag egyszerűen rekonstruálható, amennyiben $P \in L(G)$. Az algoritmus helyességének a bizonyításához be kell látnunk a következő tételt.

Tétel 8.1 *Az Early-féle algoritmussal nyert felismerési mátrixban egy $A \rightarrow X.Y$ pontozott szabály ($X, Y \in (V_N \cup V_T)^*$) pontosan akkor szerepel az $F_{i,j}$ -ben, ha $A \rightarrow XY \in H$, $X \Rightarrow^* a_{i+1} \dots a_j$, és van olyan $Z \in (V_N \cup V_T)^*$ szó, amelyre $S \Rightarrow^* a_1 \dots a_i AZ$.*

Bizonyítás: Először a feltétel szükségességét látjuk be. A bizonyítást a mátrix elemeire vonatkozó teljes indukcióval végezzük.

Az $F_{0,0}$ esetében az 1. és a 2. lépések szerint csak olyan pontozott szabályok jöhetnek szóba, ahol a jobb oldalon a pont előtti rész üres. Ha tehát $A \rightarrow .Y$, akkor vagy $A = S$ és $S \rightarrow Y \in H$, vagy $A \rightarrow Y \in H$ és $S \Rightarrow^* AZ$ valamely Z szóra.

Az indukciót ezek után az (i, j) indexpároknak egy olyan elrendezésével végezzük, amely pontosan megfelel a mátrix kitöltési sorrendjének. A fődiagonális elemeit külön kell kezelnünk. Tegyük fel tehát, hogy a bizonyítandó állítás minden $F_{r,s}$ -re teljesül, hacsak $s < j$, vagy $s = j$ és $i < r < j$.

Legyen most $A \rightarrow X.Y \in F_{i,j}$. Ha $i \neq j$, akkor ezt a szabályt csak az algoritmus 4. vagy 5. lépésében írhattuk be ide. Az első esetben $X = X'a_j$ és $A \rightarrow X'.a_jY \in F_{i,j-1}$. De akkor az indukciós feltevésből $S \Rightarrow^* a_1 \dots a_i AZ$ következik, valamilyen Z szóra, és

$X' \Rightarrow^* a_{i+1} \dots a_{j-1}$, amiből az állításunk közvetlenül adódik. Ha viszont az 5. lépésben írtuk be a szabályt, akkor $X = X'B$ valamely X' szóra és B változóra úgy, hogy van egy olyan k index, melyre $A \rightarrow X'.BY \in F_{i,k}$, és $B \rightarrow U. \in F_{k,j}$ valamely U szóra.

Az indukciós feltevésből most $S \Rightarrow^* a_1 \dots a_i AZ$, valamint $X' \Rightarrow^* a_{i+1} \dots a_k$, illetve $S \Rightarrow^* a_1 \dots a_k BZ$, és $U \Rightarrow^* a_{k+1} \dots a_j$ adódik. Ezekből a $B \rightarrow U. \in H$ figyelembevételével nyilván

$$X'B \Rightarrow^* a_{i+1} \dots a_k a_{k+1} \dots a_j$$

amivel az állításunkat bebizonyítottuk.

Hátra van még az $i = j$ eset, vagyis a fődiagonálisban levő elemek esete. Az indukciós feltevés ebben az esetben azokra az $F_{r,s}$ elemekre vonatkozik, ahol $s < j$, vagy $s = j$ és $0 < r < j$. A fődiagonális elemeibe csak az algoritmus 2. lépésében írhattunk be szabályokat. De itt is csak olyan $B \rightarrow .U$ alakú szabályok jöhetnek szóba, amelyekre van olyan $i \leq j$, hogy $A \rightarrow X.BY \in F_{i,j}$. Itt az $i = j$ eset is előfordulhat. Tekintsük azonban azt a szabályt, amelyet az algoritmus először helyez el az $F_{j,j}$ -ben. Ebben az esetben az $i < j$ kell, hogy fennálljon. Az indukciós feltevésből tehát $S \Rightarrow^* a_1 \dots a_i AZ$ és $X \Rightarrow^* a_{i+1} \dots a_j$ következik. Ugyanakkor nyilván $A \Rightarrow^* XBY$ is fennáll, amiből $S \Rightarrow^* a_1 \dots a_j BYZ$ következik, s ezzel állításunkat igazoltuk. Az indukciós feltevést most már minden új szabály elhelyezésénél alkalmazhatjuk az $F_{j,j}$ -be beírt szabályokra is. Ha tehát egy $B \rightarrow .U$ szabályt az algoritmus azért vesz fel az $F_{j,j}$ -be, mert $B \rightarrow U. \in H$, és $A \rightarrow X.BY \in F_{j,j}$, akkor itt nyilván $X = \lambda$ és $S \Rightarrow^* a_1 \dots a_j AZ$, és ezért $S \Rightarrow^* a_1 \dots a_j BYZ$, amivel a bizonyítást befejeztük.

A feltétel elégségességének bizonyítását ugyancsak teljes indukcióval végezzük. Először nézzük az $F_{0,0}$ -t. Tegyük fel, hogy $S \Rightarrow^* BZ$ teljesül valamely B változóra és Z szóra, és $B \rightarrow U. \in H$. Az $S \Rightarrow^* BZ$ levezetés lépésszáma vonatkozóan egy külön teljes indukciót végzünk. Ha a lépésszám nulla, akkor $B = S$ és $Z = \lambda$, és ezért az algoritmus a $B \rightarrow .U$ szabályt már az első lépésben felveszi az $F_{0,0}$ -ba. Tegyük fel, hogy az állítás k lépésre igaz, és legyen az $S \Rightarrow^* BZ$ levezetés lépésszáma $k + 1$. Ekkor nyilván van olyan A nemterminális és Z' szó, hogy $S \Rightarrow^* AZ'$, $A \Rightarrow^* BZ''$, ahol $Z = Z''Z'$ és az $S \Rightarrow^* AZ'$ levezetés lépésszáma k . Ezért az indukciós feltevésből $A \rightarrow .BZ'' \in F_{0,0}$, tehát az algoritmus 2. lépése szerint $B \rightarrow .U \in F_{0,0}$.

Most tegyük fel, hogy az algoritmus elégséges volta igaz minden olyan $F_{r,s}$ -re, ahol $s < j$, vagy $s = j$ és $i < r < j$. Legyen akkor $A \rightarrow XY \in H$, $X \Rightarrow^* a_{i+1} \dots a_j$, és $S \Rightarrow^* a_1 \dots a_i AZ$ valamely Z szóra. Ha most $X = X'a_j$, akkor az indukciós feltevés szerint $A \rightarrow X'.a_j Y \in F_{i,j-1}$, és az algoritmus 4. lépése alapján $A \rightarrow X'a_j.Y \in F_{i,j}$. Egyébként $X = X'B$, ahol $B \Rightarrow^* a_{k+1} \dots a_j$ és $X' \Rightarrow^* a_{i+1} \dots a_k$ valamilyen k -ra. Az indukciós feltevésből most $A \rightarrow X'.BY \in F_{i,k}$ és $B \rightarrow U. \in F_{k,j}$ adódik valamilyen U szóra, amelyre $U \Rightarrow^* a_{k+1} \dots a_j$ és $B \rightarrow U. \in H$. De akkor az algoritmus 5. lépése szerint $A \rightarrow X'B.Y \in F_{i,j}$.

Végül vegyük a fődiagonális elemeit. Az indukciós feltevést most azokra az $F_{r,s}$ elemekre tesszük, amelyekre $s \leq j$ és $0 < r < j$. Legyen tehát $A \rightarrow Y \in H$ és $S \Rightarrow^* a_1 \dots a_j AZ$. Ez a levezetés nyilván átrendezhető úgy, hogy a változók közül a legbaloldalibbat helyettesítjük be először mindaddig, amíg az a_j -t meg nem kapjuk. Eszerint van egy olyan A' változó, hogy $S \Rightarrow^* a_1 \dots a_k A'Z' \Rightarrow^* a_1 \dots a_k a_{k+1} \dots a_j AZ''Z'$, ahol $Z''Z' \Rightarrow^* Z$. Ha itt $k < j$, akkor az indukciós feltevés szerint $A' \rightarrow a_{k+1} \dots a_j.AZ'' \in F_{k,j}$, és így az algoritmus 2. lépése szerint

$A \rightarrow .Y \in F_{j,j}$. A $k = j$ esetben viszont az indukciós feltevést már alkalmazhatjuk az $A' \rightarrow AZ''$ szabályra és az $S \Rightarrow^* a_1 \dots a_j A' Z'$ levezetésre, hiszen ez a levezetés véges számú lépésből áll, tehát véges számú lépésben el kell jutnunk egy olyan esethez, ahol $k < j$. Az $A' \rightarrow .AZ'' \in F_{j,j}$ -ből pedig az algoritmus 2. lépése szerint $A \rightarrow .Y \in F_{j,j}$ következik, és ezzel a tétel bizonyítását befejeztük. $\square$

Az Early-féle algoritmus működését a következő egyszerű kis példán szemléltetjük. A G generatív nyelvtan legyen a következő formában definiált:

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

ahol H szabályhalmaz a következő: $S \rightarrow S + A, A \rightarrow A \times B, B \rightarrow (S), S \rightarrow A, A \rightarrow B, B \rightarrow a$.

Tekintsük most az $a \times a + a$ bemenő szó szintaktikai elemzését. Az algoritmus az első lépésben az $S \rightarrow .S + A, S \rightarrow .A$ szabályokat veszi fel az $F_{0,0}$ -ba. Ezután a 2. lépés alapján kiegészíti az $F_{0,0}$ -at az $A \rightarrow .A \times B, A \rightarrow .B, B \rightarrow .(S), B \rightarrow a$ szabályokkal. Utána j értéke 1 lesz, és az algoritmus a 4. lépésben az $F_{0,1}$ -be felveszi a $B \rightarrow a$ pontozott szabályt, mivel a bemenő szó első betűje a . Az 5. lépésben az $F_{0,1}$ kiegészül még az alábbi szabályokkal: $A \rightarrow B., A \rightarrow A. \times B, S \rightarrow A., S \rightarrow S. + A$.

Ezután a 2. lépés következik, ahol most $F_{1,1}$ -et kell kitölteni. De az $F_{0,1}$ -ben szereplő szabályokhoz nem található egyetlen olyan szabály sem, amely a feltételeket kielégíti, hiszen az $F_{0,1}$ -beli szabályokban a pont után közvetlenül nem is fordul elő változó. Ezért az $F_{1,1}$ üres marad. Az algoritmus ezután a $j = 2$ értékre hajtja végre a 4. lépést, melynek során az $F_{1,2}$ nyilván szintén üres marad, az $F_{0,2}$ -be pedig az $A \rightarrow A \times .B$ szabály fog bekerülni. Az 5. lépésben mind az $F_{1,2}$, mind az $F_{0,2}$ változatlan marad. Az $F_{2,2}$ -be ezután a 2. lépés során bekerülnek a $B \rightarrow .(S)$ és $B \rightarrow .a$ szabályok, mivel $A \rightarrow A \times .B \in F_{0,2}$.

A felismerési mátrix kitöltésének további menetét nem részletezzük, de a következő ábrán megadjuk a teljes mátrixot. Mivel ebben a mátrixban $S \rightarrow S + A. \in F_{0,5}$, a kérdéses bemenő szó benne van az $L(G)$ -ben. A tételünk szerint ugyanis ez pontosan akkor áll fenn, ha $S \rightarrow S + A \in H$ és $S + A \Rightarrow^* a \times a + a$, továbbá van olyan Z szó, amelyre $S \Rightarrow^* SZ$, mely utóbbi összefüggés már nem is lényeges.

$S \rightarrow \cdot S + A$ $S \rightarrow \cdot A$ $A \rightarrow \cdot A X B$ $A \rightarrow \cdot B$ $B \rightarrow \cdot (S)$ $B \rightarrow \cdot a$	$B \rightarrow a \cdot$ $A \rightarrow B \cdot$ $A \rightarrow A \cdot X B$ $S \rightarrow A \cdot$ $S \rightarrow S \cdot + A$	$A \rightarrow A X \cdot B$	$A \rightarrow A X B \cdot$ $S \rightarrow A \cdot$ $A \rightarrow A \cdot X B$ $S \rightarrow S \cdot + A$	$S \rightarrow S + \cdot A$	$S \rightarrow S + A \cdot$ $S \rightarrow S \cdot + A$
	---	---	---	---	---
		$B \rightarrow \cdot (S)$ $B \rightarrow \cdot a$	$B \rightarrow a \cdot$	---	---
			---	---	---
				$A \rightarrow \cdot A X B$ $A \rightarrow \cdot B$ $B \rightarrow \cdot (S)$ $B \rightarrow \cdot a$	$B \rightarrow a \cdot$ $A \rightarrow B \cdot$ $A \rightarrow A \cdot X B$

Itt jegyezzük meg, hogy létezik olyan változata az Early-algoritmusnak, amely nem csak λ -mentes nyelvtanokra működik.

9. Irodalomjegyzék

1. A. V. Aho: Indexed Grammars - An Extension of Context-Free Grammars, *Journ. of ACM*, **15** (1968), pp.647-671.
2. J. Dassow, M. Ito és Gh. Paun: On the Subword Density of Languages, *SEA Bull. Math.*, **18** (1994), pp.49-61.
3. Demetrovics János, Jordan Denev és Radiszlav Pavlov: *A számítástudomány matematikai alapjai*, Tankönyvkiadó, Budapest, 1989.
4. T. Hayashi: On Derivation Trees of Indexed Grammars - An Extension of the uvwxy - Theorem, *Publ. RIMS, Kyoto Univ.*, **9** (1973), pp.61-92.
5. J. E. Hopcroft & J. D. Ullmann: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Massachusetts, Menlo Park, California, London, Amsterdam, Don Mills, Ontario, Sidney, 1979.
6. Peák István: *Bevezetés az automaták elméletébe I.*, Tankönyvkiadó, Budapest, 1988.
7. Peák István: *Bevezetés az automaták elméletébe II.*, Tankönyvkiadó, Budapest, 1989.
8. Peák István: *Bevezetés az automaták elméletébe III.*, Tankönyvkiadó, Budapest, 1990.
9. Gy. E. Révész: *Introduction to Formal Languages*, McGraw-Hill, New York, St Louis, San Francisco, Auckland, Bogota, Hamburg, Johannesburg, London, Madrid, Mexico, Montreal, New Delhi, Panama, Paris, Sao Paulo, Singapore, Sydney, Tokyo, Toronto, 1983.
10. Révész György: *Bevezetés a formális nyelvek elméletébe I-II.*, Tankönyvkiadó, Budapest, 1989.
11. A. Rosenfeld: *Picture Languages, Formal Models for Picture Recognition*, Academic Press, New York, 1979.
12. A. Salomaa, *Formal Languages*, Academic Press, New York, London, 1973.
13. H.J. Shyr, *Free Monoids and Languages*, National Chung-Hsing University, Taichung, Taiwan, R.O.C., Ho Min Book Company, 1991.