Nem hagyományos modellek a formális nyelvek és automaták elméletében

Egyetemi doktori (PhD) értekezés

Szerző: Hegedüs László

Témavezető: Dr. Nagy Benedek

Debreceni Egyetem

Természettudományi Doktori Tanács

Informatikai Tudományok Doktori Iskola

Debrecen, 2016

Nem hagyományos modellek a formális nyelvek és automaták elméletében

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében az

Informatika tudományágban

Írta: **Hegedüs László** okleveles **Programtervező Informatikus**

Készült a Debreceni Egyetem **Informatikai Tudományok** doktori iskolája (**Elméleti számtástudomány, adatvédelem és kriptográfia** programja) keretében

Témavezető: Dr. Nagy Benedek

A doktori szigorlati bizottság:

elnök:	Dr.	....
tagok:	Dr.	....
	Dr.	....

A doktori szigorlat időpontja: 201 . . . .

Az értekezés bírálói:

Dr.	....
Dr.	....
Dr.	....

A bírálóbizottság:

elnök:	Dr.	....
tagok:	Dr.	....
	Dr.	....
	Dr.	....
	Dr.	....

Az értekezés védésének időpontja: 201 . . . .

Tartalomjegyzék

1 Bevezetés

Az utóbbi néhány évtizedben a számítástudomány területén egyre nagyobb figyelmet szentelnek a nem hagyományos modellek vizsgálatának. A fogalom igen tág, minden olyan modellt ide sorolhatunk, amely a klasszikus szakirodalomban (például [23, 40, 41, 42, 31, 25]) tárgyaltaktól valamilyen formában eltér. A felmerülő ötletek egyik fő forrása a biológia. Ezen terület központi eleme a bioinspirált számítástudomány (bioinspired computing), mely az élőlényekben végbemenő biológiai és kémiai folyamatokon alapuló műveletek elméleti és gyakorlati alkalmazási lehetőségeit vizsgálja. A DNS kettős spirál szerkezetének felfedezése *James D. Watson* és *Francis H. C. Crick* nevéhez fűződik [49], rájuk utal a dolgozatban később tárgyalt Watson-Crick automata elnevezése is. Eredményük publikálása óta a tudomány gyors fejlődésének köszönhetően már tetszőleges nukleotid láncokat hozhatunk létre, melyekkel akár matematikai problémák is megoldhatók. *Leonard Adleman* mutatta meg, hogyan lehet a DNS láncokat egy Hamilton út probléma megoldására használni [13]. Módszerének érdekessége, hogy abban az adat és a program nem különül el egymástól. Bár gyakorlati szempontból a DNS láncokkal végzett műveletek időigényesek, az általuk inspirált matematikai modellek vizsgálata új szemléletmódot alakított ki. A természetben gyűrű alakú DNS láncok is gyakran előfordulnak, ilyen például a mitokondriális DNS [29]. Ez tekinthető a dolgozat első felében tárgyalt körszavak biológiai motivációjának. Ugyancsak az élő sejtek működése ösztönözte a membránokon alapuló számításokat. Egy membrán rendszer (P-rendszer [43]) különböző régiókból áll, melyek objektumokat, molekulákat tartalmaznak. Ezek a molekulák meghatározott szabályok szerint lépnek át a membránok falán, és vándorolnak egyikből a másikba, továbbá a legkülső membrán a környezetből is vehet fel, illetve eraszthet ki molekulákat. Így ez a modell egy párhuzamos rendszert ír le, melynek kifejezőereje megegyezik a Turing gépekével.

A számítástudományban – főleg az informatikusok szemében – a legtöbb adat bizonyos karakterekből álló diszkrét (és esetünkben véges) szimbólumsorozatnak, azaz egy szónak felel meg. Szónak tekinthető például egy olyan diszkrét függvény is, amelynek értékkészlete egy megszámlálható halmaz. Érthető, hogy a szavak bizonyos tulajdonságainak vizsgálata egy meglehetősen érdekes és fontos feladat. A felderített tulajdonságok alapján különböző csoportokat alkothatunk, melyeket tovább finomíthatunk akár egészen addig, amíg egyetlen szó válik az elemzés tárgyává, majd a megfigyelések alapján kialakított hipotézist – ha van értelme az általánosításnak – a nagyobb osztályok felé haladva teszteljük és újabb csoportokat alakítunk ki, egyre többet megtudva a felfedezett tulajdonságról. Ez a módszer nem csak a szavak kombinatorikus tulajdonságainak felderítéséhez alkalmazható, a tudomány valamennyi területén megállja a helyét.

Mivel kettő modellel foglalkoztam, a dolgozat két nagyobb egységre bontható. Egyikben a körszavak periodikus tulajdonságait mutatom be, a másikban pedig az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákat tárgyalom. A két téma nem teljesen független egymástól, de mindkettő a formális nyelvek és automaták két különböző területéhez kapcsolódik. Emiatt mindkét fejezethez tartozik egy-egy rövid bevezető rész, melyben az adott témához tartozó alapvető fogalmak, definíciók, tételek is ismertetésre kerülnek.

Dolgozatom első fele körszavak kombinatorikájával foglalkozik. A szakirodalom jelentős része a hagyományos (lineáris) szavak köré épül, de az utóbbi években a körszavak iránti érdeklődés is megnőtt [17, 18, 45]. Ennek egyik lehetséges oka az, hogy az osztály definíciója egyszerű, de a hagyományos szavakra érvényes tételek körszavakra nem mindig teljesülnek, így érdekes kihívást nyújt azok általánosítása. Fontos tulajdonsága a hagyományos szavaknak a lokális és globális periodicitás. A 2. fejezetben ennek a tulajdonságnak egy általánosítását dolgoztam ki.

A dolgozat második részében az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákkal foglalkoztam. Néhány modell különlegessége az, hogy állapotok helyett más, esetleg időben nem változó tulajdonságokkal jellemzik őket. Lehet, hogy állapotaik egyáltalán nincsenek is, vagy nincs jelentős befolyásuk a számítás során. Ilyen modellre példa a P-rendszer [43], ahol a számítás az élő sejthez hasonló membránokkal és bennük raktározható különböző objektumokkal zajlik. Egy P-rendszert tehát a membránok struktúrája és a hozzájuk rendelt szabályok egyértelműen meghatározzák. A rendszer „állapota" (ha állapot alatt a struktúra és a szabályok összességét értjük) pedig nem változik, kivéve, ha megengedünk olyan szabályokat, melyekkel membránok hozhatók létre, illetve semmisíthetők meg. A különböző membránok tartalmának felsorolása pedig inkább konfiguráció-leírásnak felel meg. A kiszámíthatóság ezen modelljének kifejezőereje is ekvivalens a Turing-gépekével. Bár kifejezőerő tekintetében a Turing-gépektől elmaradnak, megemlíthetjük még az egyállapotú veremautomatákat is, melyek pontosan azokat a nyelveket fogadják el (a környezetfüggetlen nyelvek osztályát), mint az állapotokkal rendelkező változataik. A számlálóautomaták a veremautomaták olyan speciális változatának tekinthetők, melyek vermeibe csak egyféle szimbólum tehető, leszámítva a speciális verem alja jelet, amit sem eltávolítani, sem a verembe helyezni nem lehet. Továbbá, minden egyes lépésnél legfeljebb egy szimbólumnyival változhat egy-egy verem tartalma. *Marvin L. Minsky* megmutatta, hogy egy Gödel-kódolást alkalmazva, kettő vagy több számlálóval (és állapotokkal) már Turing-ekvivalens automatát kapunk [35]. A bizonyítás főbb lépéseit később megemlítjük az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákkal kapcsolatban. A 3.2. fejezetben számlálóautomaták olyan speciális egyállapotú változatait vizsgálom, melyeknek két olvasófeje van és ezek a számítás elején az input szó két ellentétes végéről indulnak.

Itt ejtenék néhány szót a dolgozatban használt elnevezésekről. Az általam vizsgált automataosztály elemeit egyállapotú automatáknak nevezem, ezzel valamelyest eltérve az angol stateless automata (állapot-nélküli automata) kifejezéstől. Véleményem szerint az általam használt név jobban tükrözi az automata felépítését, működését. Legalábbis a magyar nyelvben. A szavak kombinatorikája területén vizsgált modell tagjait pedig az egyszerűség kedvéért körszavaknak hívom. Az idegen nyelvű szakirodalomban általában a cyclic word (ciklikus szó), necklace (nyaklánc) vagy circular word (cirkuláris szó) elnevezést használják.

A dolgozat a következőképpen épül fel. A 2.1. fejezetben a szavak kombinatorikájának alapvető fogalmait ismertetem néhány fontosabb tétellel egyetemben. Ezután a 2.2. fejezetben bevezetem a körszó fogalmát és a periódus általánosítása után vizsgálom a körszavak periodicitási tulajdonságait, különös tekintettel az úgynevezett gyenge periódusokra. Ehhez bizonyos mértékben kapcsolódik a 2.3. fejezet, melyben a körszavak szemléltetéséről esik szó. Az első rész lezárásaként a 2.4. fejezetben röviden összefoglalom a körszavakkal kapcsolatos eredményeimet és következtetéseimet. A dolgozat második fele a számlálóautomaták és a több olvasófejjel rendelkező automaták elméletébe való bevezetéssel kezdődik a 3.1. fejezetben. Ezt követi a 3.2. fejezet az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták tárgyalásával. Különböző nyelveket ismertetek és megmutatom, hogy azok milyen paraméterek mellett fogadhatók el a modellben, ezzel a tárgyalt automaták hierarchiába rendezhetők. Majd röviden összefoglalom eredményeimet a 3.3. fejezetben, ahol a hierarchiával kapcsolatos fontosabb összefüggéseket ábrázolom is.

Az egyes részekhez tartozó rövid összefoglalásokban említésre kerül, hogy a dolgozatban tárgyalt eredmények hol és milyen formában jelentek meg. A dolgozat végén pedig egy átfogó összefoglalás szerepel.

Megjegyezném, hogy a dolgozat nagy része a tudományos művekhez hasonlóan, többes szám első személyben íródott, leszámítva a bevezetést és az összefoglalásokat, amelyekben kiemelem azokat az eredményeket, melyeknek létrejöttéhez nagy részben hozzájárultam.

Arra törekedtem, hogy a dolgozatban előforduló különböző típusú objektumok a folyó szövegben is könnyen megkülönböztethetőek legyenek egymástól. Az 1.1 táblázat rendszerezi ezeket a jelöléseket. Továbbá, az olyan – általában más szerzőktől származó – eredmények (lemmák) végét, melyek bizonyítását nem közlöm, a $\square$ szimbólummal jelölöm. Egyébként pedig a kimondott tétel, lemma, állítás után közvetlenül következik a bizonyítás, melynek végét $\blacksquare$ zárja.

$\mathbb{N}$	a természetes számok halmaza, $\{1, 2, \dots\}$
$\mathbb{N}_0$	a nemnegatív egész számok halmaza, $\{0, 1, 2, \dots\}$
$\text{lnko}(k, \ell)$	k és ℓ egész számok legnagyobb közös osztója
$(p \bmod q)$	a p $q-$val való osztásának maradéka
$\lfloor \frac{k}{\ell} \rfloor$	$\frac{k}{\ell}$ alsó egész része
$A, B, X, Y, \dots$	halmazok
$\|A\|$	az A halmaz számossága
$\mathcal{P}(A)$	az A halmaz összes részhalmazának halmaza
Σ	ábécé, véges, nem üres halmaz
$a, b, c, \dots$	egy ábécé betűi
$\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{w}, \dots$	egy ábécé feletti szavak
$\|\boldsymbol{w}\|$	a $\boldsymbol{w}$ szó hossza
$\|\boldsymbol{w}\|_a$	a $\boldsymbol{w}$ szóban lévő a betűk darabszáma
λ	üresszó
$\sigma^k(\boldsymbol{w})$	a $\boldsymbol{w}$ szó k hosszú kezdőszeletének mozgatása a szó végére
$\boldsymbol{w}^R$	a $\boldsymbol{w}$ szó tükörképe
$\boldsymbol{w}_\circ$	$\boldsymbol{w}$ szóból képzett körszó
$\text{Pal}(\Sigma^*)$	Σ feletti palindrom szavak halmaza
$\text{Pal}^\circ(\Sigma^*)$	Σ feletti palindrom körszavak halmaza
$\boldsymbol{f}_n$	$n-$edik Fibonacci szó
$\boldsymbol{f}$	végtelen Fibonacci szó
$\tau_{\boldsymbol{w}_\circ}$	$\boldsymbol{w}_\circ$ körszó fája
$\mathcal{A}$	automata
δ	automata átmenetfüggvénye
$\mathcal{L}(\mathcal{A})$	az $\mathcal{A}$ automata által elfogadott nyelv
¢	szó eleje jel
\$	szó vége jel
$\text{sign}(k)$	egész számokon értelmezett előjelfüggvény

1.1. táblázat. A dolgozatban használt jelölések.

2 Körszavakkal kapcsolatos vizsgálódások

2.1. Szavak kombinatorikája

A szavaknak, más néven sztringeknek igen fontos szerepe van a számítástudományban és a matematikában egyaránt. Adatot, programot, tulajdonképpen mindenféle információt reprezentálhatnak. A matematikában különböző sorozatokként kezelhetjük őket. A szavak kombinatorikájának egyik fő feladata többek között ezen sorozatok egyes strukturális tulajdonságainak vizsgálata. Az elért eredmények hasznosíthatók az algoritmuselméletben, szövegfeldolgozásban, adatsűrítésben és a matematika egyes területein is (például az algebrában).

A szavak kombinatorikájával kapcsolatos első eredmények megjelenése az 1900-as évek elejére tehető, mikor is *Axel Thue* publikálta a négyzetmentes szavakkal kapcsolatos műveit [47, 48]. Továbbá *Percy A. MacMahon* egy könyve is ehhez a területhez köthető [34]. Az első áttekintő, összefoglaló művet *M. Lothaire* írói álnév alatt publikáló matematikusok csoportja írta [31]. Később még két hasonló jelentőségű könyvet készítettek el [32, 33]. Legtöbbjük *Marcel P. Schützenberger* tanítványa volt, akinek a nevéhez több fontos tétel is kapcsolódik, melyek egy részével a következő alfejezetben találkozhatunk. Előtte bevezetünk néhány alapvető fogalmat és definíciót, miközben ismertetjük a jelöléseket.

Feltételezzük az elemi halmazelméleti alapfogalmak és műveletek ismeretét, ezek ismertetésére nem térünk ki. A természetes számok halmazát, azaz az $\{1, 2, \ldots\}$ halmazt az $\mathbb{N}$ szimbólummal jelöljük, valamint használjuk majd az $\mathbb{N}_0$ jelölést is $\mathbb{N} \cup \{0\}$ helyett.

Egyik legfontosabb alapfogalmunk az ***ábécé***, mely egy nem üres véges halmaz. Jelölésére általában a Σ szimbólumot használjuk, elemeit pedig ***betűknek*** vagy ***szimbólumoknak*** nevezzük. A dolgozatban az ábécék elemeit és az azokat jelölő változókat dőlt karakterekkel ($a, b, w_1, \ldots$ stb.) írjuk és sokszor indexekkel látjuk el őket. Legyen $n \in \mathbb{N}$ egy tetszőleges pozitív egész szám. Minden $\boldsymbol{w} : \{1, \ldots, n\} \to \Sigma$ függvény definiál egy ***szót***, melyet a könnyebb olvashatóság kedvéért $\boldsymbol{w} = w_1 \cdots w_n$−ként írunk, ahol $w_i = \boldsymbol{w}(i)$ minden $i = 1, \ldots, n$−re. Ahogy látható, a szavakat – kivéve akkor, amikor biztosak vagyunk abban, hogy pontosan egy betűből állnak – félkövéren szedjük ($\boldsymbol{w}, \boldsymbol{u}, \ldots$ stb.). Azt mondjuk, hogy a $\boldsymbol{w} \in \Sigma^*$ szó ***unáris***, ha $|\Sigma| = 1$. Két szó ***konkatenációja*** alatt azok egymás után írását értjük, azaz adott $\boldsymbol{u} = u_1 \cdots u_n$ és $\boldsymbol{v} = v_1 \cdots v_m$ konkatenációjával az $n+m$ hosszú $\boldsymbol{uv} = u_1 \cdots u_n v_1 \cdots v_m$ szót kapjuk. Mivel algebrai megközelítésben ez a művelet felel meg a szorzásnak, ezért $\boldsymbol{uv}$ helyett olykor $\boldsymbol{u} \cdot \boldsymbol{v}$−t írunk. Legyenek $w_1, \ldots, w_n \in \Sigma$ ($n \in \mathbb{N}$) nem feltétlenül különböző betűk. A $\boldsymbol{w} = w_1 \cdots w_n$ szó ***hossza*** a benne lévő összes betű száma, ebben az esetben n. Azt is mondjuk, hogy a $\boldsymbol{w}$ szóban n darab ***pozíció*** van. Ezekre a sorszámukkal hivatkozunk és az $i = 1, \ldots, n$−edik pozícióban lévő betűt rendszerint w_i−vel jelöljük akkor is, ha explicit módon nem adtuk meg, hogy $\boldsymbol{w} = w_1 \cdots w_n$. Azt az egyértelműen meghatározott szót, melynek hossza nulla, ***üresszónak*** nevezzük és λ−val jelöljük. Adott Σ ábécé feletti szavak a konkatenáció művelettel ***félcsoportot*** alkotnak. A félcsoport elemeit Σ^+−szal jelöljük. Továbbá, $\lambda\boldsymbol{w} = \boldsymbol{w}\lambda$ tetszőleges $\boldsymbol{w} \in \Sigma^+$ szóra, tehát $\Sigma^+ \cup \{\lambda\}$ a konkatenáció műveletével ***monoidot*** alkot, melynek elemeit Σ^*–gal jelöljük. A konkatenáció műveletét felhasználva a matematikában megszokott módon definiálhatjuk a $\boldsymbol{w} \in \Sigma^*$ szó $n \in \mathbb{N}_0$–edik ***egész hatványát***:

$$\boldsymbol{w}^n = \begin{cases} \lambda\,, & \text{ha } n = 0, \\ \boldsymbol{w} \cdot \boldsymbol{w}^{n-1}\,, & \text{egyébként.} \end{cases}$$

A $\boldsymbol{w} \in \Sigma^*$ szó ***tört hatványát*** csak bizonyos esetekben értelmezzük. Legyen $p \in \mathbb{N}$ és jelölje q a $\boldsymbol{w}$ szó hosszát, azaz $q = |\boldsymbol{w}|$ és $\boldsymbol{w} = w_1 \cdots w_q$. Ekkor létezik a $\boldsymbol{w}^{\frac{p}{q}} = \boldsymbol{w}^k w_1 \cdots w_r$ szó, ahol $p = k \cdot q + r$.

Vegyük az $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \Sigma^*$ tetszőleges szavak szorzatát, azaz $\boldsymbol{w} = \boldsymbol{xyz}$–t. Ekkor azt mondjuk, hogy $\boldsymbol{y}$ ***faktora*** a $\boldsymbol{w}$ szónak. Ha ezen felül $\boldsymbol{x} = \lambda$, akkor ***kezdőszeletről***, $\boldsymbol{z} = \lambda$ esetén pedig ***zárószeletről*** beszélünk. Ez az $\boldsymbol{y}$ faktor valódi, ha $0 < |\boldsymbol{y}| < |\boldsymbol{w}|$. Az egyenlőtlenség fennállása esetén beszélünk valódi kezdőszeletről, illetve valódi zárószeletről is.

Rendezett ábécének nevezzük $\Sigma = \{a_1, \ldots, a_n\}$–t, ha adott mellé egy olyan $<$ reláció, melyre $a_1 < a_2 < \cdots < a_n$. Ez a reláció kiterjeszthető Σ^*–beli szavakra úgy, hogy $\boldsymbol{u} < \boldsymbol{v}$ pontosan akkor, ha

- $\boldsymbol{u}$ a $\boldsymbol{v}$ valódi kezdőszelete, vagy
- létezik olyan $i \leq \min\{|\boldsymbol{u}|, |\boldsymbol{v}|\}$ index, melyre $u_i < v_i$ és bármely $j = 1, \ldots, i-1$ esetén $u_j = v_j$.

A Σ^* szavak egy L részhalmazának $<$ reláció szerinti rendezett sorozatát nevezzük L egy ***lexikografikus rendezésének***.

Legyen adott egy $\boldsymbol{w} = w_1 \cdots w_n \in \Sigma^*$ szó. Azt mondjuk, hogy $\boldsymbol{x} \in \Sigma^*$ a $\boldsymbol{w}$ ***keretszava***, ha annak kezdőszelete és egyben zárószelete is. Az $\boldsymbol{x}$ keretszó nem triviális, ha $0 < |\boldsymbol{x}| < |\boldsymbol{w}|$. A $\boldsymbol{w}$ szó ***periódusa*** $p \in \mathbb{N}_0$, ha

$w_i = w_{i+p}$ minden $i = 1, \ldots, |\boldsymbol{w}| - p$ esetén. Periódusok és keretszavak kapcsolatáról jól ismert tényt fogalmaz meg a következő lemma.

2.1. Lemma. *Adott $\boldsymbol{w} \in \Sigma^*$ szónak az $\boldsymbol{x} \in \Sigma^*$ szó pontosan akkor keretszava, ha $p = |\boldsymbol{w}| - |\boldsymbol{x}|$ a $\boldsymbol{w}$ periódusa.* □

Azt mondjuk, hogy az $\boldsymbol{u} \in \Sigma^*$ és $\boldsymbol{v} \in \Sigma^*$ szavak egymás ***konjugáltjai*** (jelöléssel $\boldsymbol{u} \sim \boldsymbol{v}$), ha léteznek olyan $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$ szavak, hogy $\boldsymbol{u} = \boldsymbol{xy}$ és $\boldsymbol{v} = \boldsymbol{yx}$. A $\sim$ reláció egy ekvivalencia reláció, amely a Σ ábécé feletti szavak halmazát ekvivalencia osztályokra bontja. Ehhez a fogalomhoz kapcsolódik a szavak ***balra történő ciklikus elforgatása***, melyet a $\sigma(.)$ operátorral fogunk végezni. Ezt szokás ciklikus permutálásnak is nevezni. Legyen $\boldsymbol{w} = w_1 w_2 \cdots w_n$ egy tetszőleges szó. Ekkor $\sigma(\boldsymbol{w}) = w_2 \cdots w_n w_1$. A $\sigma(.)$ operátor $\boldsymbol{w}$ szóra történő $k \in \mathbb{N}_0$−szori alkalmazását pedig $\sigma^k(\boldsymbol{w})$−val jelöljük és a $\sigma(\sigma^{k-1}(\boldsymbol{w}))$ szót értjük alatta, valamint megállapodunk abban, hogy $\sigma^n(\boldsymbol{w}) = \sigma^0(\boldsymbol{w}) = \boldsymbol{w}$. Így értelmezhetjük a ciklikus elforgatást negatív irányban is. Azaz, ha $\ell \in \mathbb{N}$, akkor $\sigma^{-\ell}(\boldsymbol{w}) = \sigma^{n-\ell}(\boldsymbol{w})$.

Egy $\boldsymbol{w}$ szó ***primitív***, ha nem áll elő egyetlen nála rövidebb szó egész hatványaként sem, azaz, ha nincs olyan p periódusa, amely maradék nélkül osztja $|\boldsymbol{w}|$−t. Primitív szavak egy kisebb csoportját alkotják a Lyndon szavak. Egy rendezett ábécé feletti $\boldsymbol{w}$ szó ***Lyndon szó***, ha primitív és nem létezik olyan konjugáltja, amely lexikografikus rendezés szerint kisebb nála.

A fentebb tárgyalt szavak mellett beszélhetünk ***végtelen szavakról*** is. A $\boldsymbol{w} = w_1 \cdots w_j \cdots$ szó végtelen szó, ha bármely $j \in \mathbb{N}$ esetén w_j egyértelműen meghatározott. Végtelen szavakat megadhatunk például diszkrét függvényekkel, melyek értelmezési tartománya $\mathbb{N}$, értékkészletük pedig egy ábécé.

A következő alfejezetben láthatjuk, hogy a periódus és a konjugált fogalmak hogyan kapcsolódnak egymáshoz.

Szavak periodikus tulajdonságai

Ebben a részben a szavak kombinatorikájával kapcsolatos legfontosabb eredményeket adjuk meg, melyek a periódusok és konjugáltak közötti kapcsolatot írják le. Ezek olyan jól ismert tények, melyek *Nathan J. Fine* és *Herbert S. Wilf*, valamint *Marcel P. Schützengerber* és tanítványai nevéhez kötődnek. Ismertetésükre azért térünk ki, mert a dolgozat további része is hasonló összefüggéseken alapul. Forrásul szolgáltak a [31, 44] művek, melyekben az alább említésre kerülő lemmák bizonyításai is megtalálhatók, ezért azokat nem is részletezzük.

Először megadjuk *Nathan J. Fine* és *Herbert S. Wilf* ismert eredményének szavak periódusaira általánosított változatát. Jelölje $\mathrm{lnko}(k, \ell)$ a $k, \ell \in \mathbb{N}_0$ számok legnagyobb közös osztóját.

2.2. Lemma (Fine és Wilf periodicitási lemmája [31])**.** *Legyen* $\boldsymbol{w} \in \Sigma^*$ *tetszőleges. Ha* $\boldsymbol{w}$*−nek létezik két olyan különböző* p *és* q *periódusa, hogy* $p + q - \mathrm{lnko}(p, q) \leq |\boldsymbol{w}|$*, akkor* $\mathrm{lnko}(p, q)$ *is periódusa* $\boldsymbol{w}$*−nek.* □

Megjegyezzük, hogy a lemma egyik következménye az, hogy ha a $\boldsymbol{w}$ nem unáris szónak a p és q relatív prímek periódusai, akkor $|\boldsymbol{w}| < p + q - 1$.

A következő tény *Friedrich W. Levi* lemmájaként ismeretes [28].

2.3. Lemma. *Minden* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$ *esetén, ha* $\boldsymbol{uv} = \boldsymbol{xy}$*, akkor létezik olyan* $\boldsymbol{t} \in \Sigma^*$*, hogy*

$$\boldsymbol{u} = \boldsymbol{xt} \quad \text{és} \quad \boldsymbol{tv} = \boldsymbol{y}, \qquad \text{vagy} \qquad \boldsymbol{x} = \boldsymbol{ut} \quad \text{és} \quad \boldsymbol{v} = \boldsymbol{ty}. \qquad \square$$

Egy szó és adott keretszava közötti kapcsolatot a következő lemma írja le, melyet *Roger C. Lyndon* és *Marcel P. Schützenberger* bizonyított.

2.4. Lemma. *Legyenek* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{s} \in \Sigma^+$ *tetszőleges szavak. Ekkor* $\boldsymbol{us} = \boldsymbol{sv}$ *pontosan akkor teljesül, ha léteznek olyan* $\boldsymbol{y} \in \Sigma^+$*,* $\boldsymbol{z} \in \Sigma^*$ *szavak és egy* $k \in \mathbb{N}_0$ *egész, hogy* $\boldsymbol{u} = \boldsymbol{yz}$*,* $\boldsymbol{v} = \boldsymbol{zy}$*, valamint* $\boldsymbol{s} = (\boldsymbol{yz})^k\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{zy})^k$*.* □

Azaz egy szónak $\boldsymbol{s}$ pontosan akkor nem triviális keretszava, ha teljesülnek a fenti lemmában megfogalmazott követelmények.

A következő lemma – amely ugyancsak *Roger C. Lyndon* és *Marcel P. Schützenberger* nevéhez kötődik – a konkatenáció kommutativitásának feltételét definiálja.

2.5. Lemma. *Legyenek* $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^+$ *tetszőleges szavak. Ekkor a következő állítások ekvivalensek egymással:*

1. $\boldsymbol{uv} = \boldsymbol{vu}$*,*
2. $\boldsymbol{u}$ *és* $\boldsymbol{v}$ *egy* $\boldsymbol{z} \in \Sigma^+$ *szónak egész hatványai,*
3. *léteznek olyan* k *és* ℓ *egészek, hogy* $\boldsymbol{u}^k = \boldsymbol{v}^\ell$*.* □

2.2. Körszavak kombinatorikája

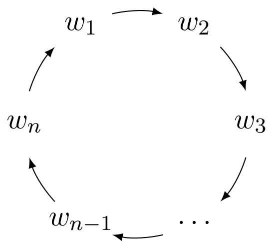
A fejezet további részében ismertetjük a vizsgált modellt és az azzal kapcsolatos új eredményeket. Ezzel a témával kapcsolatban fő feladat a körszavak periodicitási tulajdonságainak vizsgálata volt.

Az előbbiekben tárgyalt úgynevezett lineáris szavakat egy-egy láncolt listaként képzelhetjük el (2.1. ábra), melynek csomópontjai a szó betűit tartalmazzák, a mutatók pedig a következő betűt tartalmazó csomópontba erednek.

Ezzel a megközelítéssel lineáris szóból körszót úgy kapunk, hogy felveszünk egy mutatót az utolsó csomópontból az elsőbe, ahogy a 2.2. ábrán

$$w_1 \longrightarrow w_2 \longrightarrow w_3 \longrightarrow \cdots \longrightarrow w_{n-1} \longrightarrow w_n$$

2.1. ábra. Lineáris szó



2.2. ábra. Körszó előállítása

is láthatjuk. Megfigyelhetjük, hogy egy körszónak nem beszélhetünk elejéről, illetve végéről, így bármelyik pozíciótól indulva meglátogathatjuk az összes csomópontot. A mi értelmezésünkben pontosan addig lépünk egyik csomópontról a másikra, amíg meg nem teszünk egy teljes kört. A $\boldsymbol{w}$ szóból képzett körszót $\boldsymbol{w}_\circ$ jelöli és $\boldsymbol{w}$ összes konjugáltjának halmazát értjük alatta. Formálisan fogalmazva:

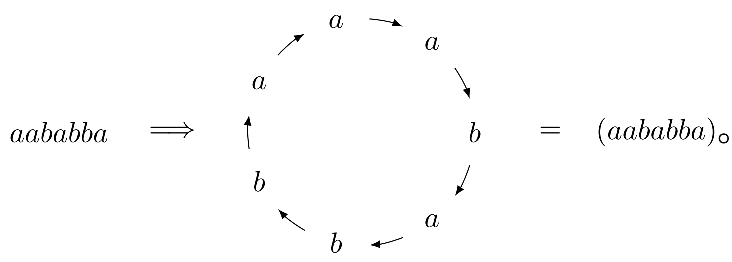
2.1. Definíció. *Legyen $\boldsymbol{w} \in \Sigma^*$. A $\boldsymbol{w}_\circ \subseteq \Sigma^*$ véges szóhalmazt a $\boldsymbol{w}$−ből képzett körszónak nevezzük, ha $\boldsymbol{w}_\circ = \{\boldsymbol{v} \in \Sigma^* \mid \boldsymbol{v} \sim \boldsymbol{w}\}$.*

Megjegyezzük, hogy egy körszót bármelyik eleme egyértelműen meghatároz, azaz tetszőleges $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{w}_\circ$ esetén $\boldsymbol{u}_\circ = \boldsymbol{v}_\circ$.

2.1. *Példa.* Vegyük az *aababba* szót. A belőle képzett körszót a 2.3. ábrán látható módon kapjuk.

Az ábráról leolvashatjuk, hogy

$$\begin{aligned} (aababba)_\circ &= \{aababba, ababbaa, babbaaa, abbaaab, \\ &\qquad bbaaaba, baaabab, aaababb\}. \end{aligned}$$



2.3. ábra. Az $(aababba)_{\boldsymbol{o}}$ körszó.

Észrevehetjük, hogy $\boldsymbol{w}_\circ$ pontosan a $\boldsymbol{ww}$ szó $|\boldsymbol{w}|$ hosszúságú faktorainak halmaza. Körszavak esetén is értelmezzük a ***faktor*** fogalmát és azt mondjuk, hogy az $\boldsymbol{y}$ szó a $\boldsymbol{w}_\circ$ körszó faktora, ha létezik olyan $\boldsymbol{v} \in \boldsymbol{w}_\circ$ szó, melynek faktora $\boldsymbol{y}$.

Hagyományos, lineáris szavak esetén érdekes alosztályt alkotnak a palindrom szavak. A $\boldsymbol{w} = w_1 \cdots w_n$ szó palindrom, ha $\boldsymbol{w} = \boldsymbol{w}^R$, ahol $\boldsymbol{w}^R = w_n \cdots w_1$. Ennek a tulajdonságnak köszönhetően a belőlük képzett körszavak is különlegesek, ahogy azt a következő állítások is alátámasztják.

2.1. Állítás (Palindrom körszavak elemei [2])**.** *Legyen $\boldsymbol{w} \in \Sigma^*$ egy tetszőleges szó. Ha létezik olyan $\boldsymbol{v} \in \boldsymbol{w}_\circ$, melyre $\boldsymbol{v} = \boldsymbol{v}^R$, akkor bármely $\boldsymbol{u} \in \boldsymbol{w}_\circ$ esetén $\boldsymbol{u}^R \in \boldsymbol{w}_\circ$ is teljesül.*

Bizonyítás. Tegyük fel, hogy létezik olyan $\boldsymbol{v} \in \boldsymbol{w}_\circ$ szó, melyre $\boldsymbol{v} = \boldsymbol{v}^R$ teljesül, azaz $\boldsymbol{v}$ palindrom szó. Ekkor $\boldsymbol{v} = x_1 \cdots x_m y x_m \cdots x_1$, ahol x_1, $\ldots$, $x_m \in \Sigma$, $y \in \Sigma \cup \{\lambda\}$ nem feltétlenül különböző betűk. Azt kell megmutatnunk, hogy bármely $\ell = 1, \ldots, 2m$ esetén $\sigma^\ell(v)^R \in \boldsymbol{w}_\circ$. Ha $\ell \leq m$, akkor $\sigma^\ell(\boldsymbol{v}) = x_{\ell+1} \cdots x_m y x_m \cdots x_1 x_1 \cdots x_\ell$. Ennek a szónak a tükörképe pontosan az $\boldsymbol{u} = \sigma^{-\ell}(\boldsymbol{v}) = x_\ell \cdots x_1 x_1 \cdots x_m y x_m \cdots x_{\ell+1}$ szó. Végül, mivel $\sigma^{-\ell}(\boldsymbol{v}) = \sigma^{|\boldsymbol{v}|-\ell}(\boldsymbol{v})$, az $\ell > m$ pozícióval való elforgatásokat is kezeltük. ■

2.2. Állítás (Páratlan hosszú palindrom körszavakról [2]). *Legyen* $\boldsymbol{w} \in \Sigma^*$ *egy tetszőleges szó, melynek hossza páratlan. Ha mind* $\boldsymbol{u}$*, mind* $\boldsymbol{u}^R$ *eleme* $\boldsymbol{w}_\circ$*−nek, akkor* $\boldsymbol{w}_\circ$ *tartalmaz palindrom szót.*

Bizonyítás. Tegyük fel, hogy $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{w}_\circ$ úgy, hogy $\boldsymbol{u} = \boldsymbol{v}^R$. Legyen $\boldsymbol{u} = x_1 \cdots x_n$ valamely $x_1, \ldots, x_n \in \Sigma$ nem feltétlenül különböző betűkre. Ha $\boldsymbol{u} = \boldsymbol{v}$, akkor az állítás triviális, ezért tegyük fel, hogy $\boldsymbol{u} \neq \boldsymbol{v}$. Ekkor létezik $\ell \in \mathbb{N}$, úgy, hogy $\boldsymbol{v} = x_{\ell+1} \cdots x_n x_1 \cdots x_\ell$. Azaz, $\boldsymbol{u} = x_1 \cdots x_n = x_\ell \cdots x_1 x_n \cdots x_{\ell+1}$ Megfigyelhetjük, hogy

$$x_1 \cdots x_\ell = x_\ell \cdots x_1 \quad \text{és} \quad x_{\ell+1} \cdots x_n = x_n \cdots x_{\ell+1}.$$

Mivel $|\boldsymbol{w}|$ páratlan, az $x_{\ell+1} \cdots x_n$ és az $x_1 \cdots x_\ell$ faktorok közül az egyik hossza páros. Tegyük fel, hogy $|x_1 \cdots x_\ell|$ az, azaz ℓ páros. Ekkor a fentiekből következik, hogy a $\sigma^{\frac{\ell}{2}}(u) = x_{\frac{\ell}{2}+1} \cdots x_\ell x_{\ell+1} \cdots x_n x_1 \cdots x_{\frac{\ell}{2}}$ szó palindrom. Egyébként, ha $|x_{\ell+1} \cdots x_n|$ páros, akkor $n - \ell$ lesz páros. Az előbbi esethez hasonlóan, a $\sigma^{-\frac{n-\ell}{2}}(u) = x_{\frac{n-\ell}{2}+1} \cdots x_n x_1 \cdots x_\ell x_{\ell+1} \cdots x_{\frac{n-\ell}{2}}$ szó palindrom. ■

Ha a 2.1. állításban csak a páratlan hosszú körszavakkal foglalkozunk, akkor a 2.2. állítás pontosan az ellenkező irányú állítást fejezi ki, így együtt egy elégséges és szükséges feltételt képezve. Páros hosszú körszavak esetén nem fogalmazható meg ilyen feltétel. Ugyanis, ha veszünk egy $((ab)^k)_\circ$ alakú körszót, ahol $k \in \mathbb{N}$, akkor mind $(ab)^k$, mind $((ab)^k)^R = (ba)^k$ eleme $((ab)^k)_\circ$−nek, viszont nem tartalmaz egyetlen palindrom szót sem.

Azokat a körszavakat, melyek tartalmaznak palindrom szavakat, ***palindrom körszavaknak*** nevezzük. Legyen Σ egy tetszőleges ábécé. Jelöljük $\mathrm{Pal}^\circ(\Sigma^*)$−gal a Σ ábécé feletti palindrom körszavak halmazát, azaz a $\mathrm{Pal}^\circ(\Sigma^*) = \{\boldsymbol{w}_\circ \mid \boldsymbol{w} \in \Sigma^*$ és létezik $\boldsymbol{u} \in \boldsymbol{w}_\circ$ úgy, hogy $\boldsymbol{u}$ palindrom szó$\}$ halmazt. Látható, hogy $\mathrm{Pal}^\circ(\Sigma^*)$ szavak halmazainak halmaza, melyeknek az unióját véve megkapjuk a következő nyelvet:

$$\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*)) = \bigcup \mathrm{Pal}^\circ(\Sigma^*) =$$

$$\bigcup \{\boldsymbol{w}_\circ \mid \boldsymbol{w} \in \Sigma^* \text{ és } \boldsymbol{w}_\circ \text{ egy palindrom körszó}\},$$

amely a Σ ábécé feletti palindrom szavak összes konjugáltjainak nyelve, azaz

$$\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*)) = \{\boldsymbol{w} \in \Sigma^* \mid \sigma^k(\boldsymbol{w}) \in \mathcal{L}(\mathrm{Pal}(\Sigma^*)) \text{ valamely } k \in \mathbb{N}_0 \text{ esetén}\},$$

ahol $\mathcal{L}(\mathrm{Pal}(\Sigma^*))$ jelöli a Σ ábécé feletti összes palindrom szó által alkotott nyelvet.

Felmerül a kérdés, hogy milyen bonyolult (a szó hosszának függvényében), eldönteni egy tetszőleges szóról, hogy az $\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*))$ nyelvbe tartozik-e vagy sem. Kiindulhatunk abból a tényből, hogy bármely $\boldsymbol{v} \in \boldsymbol{w}_\circ$ esetén, $\boldsymbol{v}$ a $\boldsymbol{ww}$ szó egy $|\boldsymbol{w}|$ hosszú faktora. Ha $\boldsymbol{u} \in \mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*))$, akkor létezik $\boldsymbol{uu}-$nak olyan $\boldsymbol{v}$ faktora, amely palindrom szó és $|\boldsymbol{v}| = |\boldsymbol{u}|$. Azaz,

$$\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*)) = \{\boldsymbol{u} \in \Sigma^* \mid \text{létezik } \boldsymbol{v} \in \mathcal{L}(\mathrm{Pal}(\Sigma^*)) \text{ úgy, hogy}$$

$$|\boldsymbol{v}| = |\boldsymbol{u}| \text{ és } \boldsymbol{v} \text{ faktora } \boldsymbol{uu} - \text{nak}\}.$$

Tehát egy $\boldsymbol{u}$ szó pontosan akkor eleme $\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*))-$nak, ha az $\boldsymbol{uu}-$ban lévő leghosszabb palindrom faktor legalább olyan hosszú, mint $\boldsymbol{u}$. Mivel egy adott szóban a leghosszabb palindrom faktort megtaláló algoritmus bonyolultsága lineáris [22, 27], ezért az $\mathcal{L}(\mathrm{Pal}^\circ(\Sigma^*))$ halmazba való tartozás is eldönthető lineáris időben.

A továbbiakban általános körszavak tárgyalásával foglalkozunk, különös tekintettel periodikus tulajdonságaikra.

Körszavak periódusainak tulajdonságai

A szavak kombinatorikájának egyik alapvető fogalma a periódus. Körszavakra történő általánosításakor különböző eseteket különböztethetünk meg attól függően, hogy feltételünk a körszó valamennyi elemére teljesül, vagy elég, ha csak az egyikre. A [6] műben foglalkoztunk először ezzel a kérdéskörrel. A megengedő eset formális definíciója a következő.

2.2. Definíció (Körszó gyenge periódusa [6]). *A p pozitív egész szám a $\boldsymbol{w}_\circ$ körszó* ***gyenge periódusa****, ha p legalább egy $\boldsymbol{v} \in \boldsymbol{w}_\circ$ szónak a periódusa.*

A szigorúbb megkötést – melyben mindegyik konjugáltnak rendelkeznie kell a periódussal – teljesítő pozitív egészt nevezhetjük ***erős periódusnak***. Nem triviális erős periódusai csak bizonyos speciális osztályba tartozó körszavaknak lehetnek. Ezek azok a körszavak, amelyek valamennyi eleme periodikus. Ebben a dolgozatban ezzel a fogalommal nem foglalkozunk, mivel kevésbé segít a körszavak jellemzésében, mint a gyenge periódus. Ez viszont nem jelenti azt, hogy szavaknak más relációkkal definiált ekvivalencia-osztályai esetén is elhanyagolható lenne az erős periódusok vizsgálata. Érdekes lehet a jövőben azt is megvizsgálni, hogy ezen periódusfogalom továbbra is ilyen szigorú marad-e, ha megengedjük, hogy körszavakat parciális szavakból (lásd [14]) kiindulva is képezhessünk, hiszen azok tartalmazhatnak olyan speciális joker betűket, melyek bármelyik másik betűvel azonosak. Ebben a dolgozatban erre nem kerül sor, de a jövőben felmerülhet, mint lehetséges kutatási irány.

Az erős periódusokkal ellentétben, a gyenge periódusok egymással való összevetése érdekes eredményekre vezet, melyeket az alábbi fejezetben tekintjük át.

Körszavak gyenge periódusai

A továbbiakban azokkal a periódusokkal foglalkozunk, melyek adott $\boldsymbol{w}_\circ$ körszó legalább egy elemének periódusai, más szóval, $\boldsymbol{w}_\circ$ gyenge periódusai. Ahogy a következő példa is mutatja, Fine és Wilf periodicitási lemmája, azaz a 2.2. lemma nem teljesül a gyenge periódusokra.

2.2. *Példa.* Vegyük az *aabab* szót. A 2.1. táblázatban láthatjuk ezen szó konjugáltjait és periódusaikat. Leolvashatjuk, hogy a 2 és a 3 is gyenge

$aabab$	5
$ababa$	$2, 4, 5$
$babaa$	5
$abaab$	$3, 5$
$baaba$	$3, 5$

2.1. táblázat. Az *aabab* szó konjugáltjai és periódusaik.

periódusa az $(aabab)_\circ$ körszónak, továbbá $2+3-1 \leq 5$, de az 1 nem gyenge periódusa.

Tulajdonképpen bármely $p > 2$ esetén konstruálható olyan $2p-1$ hosszú $\boldsymbol{w}$ szó, hogy $\boldsymbol{w}_\circ$−nek p és $p-1$ is gyenge periódusa, viszont az 1 már nem az. Legyen például $\boldsymbol{w} = ab^{p-1}ab^{p-2}$. Ekkor p nyilvánvalóan periódusa $\boldsymbol{w}$−nek és $p-1$ periódusa $\sigma^2(\boldsymbol{w}) = b^{p-2}ab^{p-2}ab$−nek. Tehát $\boldsymbol{w}_\circ$−nek mind p, mind $p-1$ gyenge periódusa.

A következő állítás arra ad elégséges és szükséges feltételt, hogy bináris $\{a, b\}$ ábécé felett egy adott $\boldsymbol{w}_\circ$ körszónak mikor lehet két egymás melletti egész szám is a periódusa.

2.3. Állítás (Bináris ábécé feletti körszavak szomszédos periódusairól [6])**.** *Legyen $\boldsymbol{w} \in \{a, b\}^+$ egy olyan szó, melyre $|\boldsymbol{w}| \geq 2$. Pontosan akkor létezik* *olyan $p > 1$ egész, hogy p és $p-1$ is gyenge periódusa $\boldsymbol{w}_\circ$−nek, ha az tartalmaz aa vagy bb faktort.*

Bizonyítás. Ha aa faktora $\boldsymbol{w}_\circ$−nek, akkor létezik egy $a\boldsymbol{u}a$ alakú $\boldsymbol{v} \in \boldsymbol{w}_\circ$ melynek $|\boldsymbol{w}|-1$ periódusa. Továbbá, $|\boldsymbol{w}|$ minden $\boldsymbol{w}_\circ$−beli elemnek periódusa, tehát $|\boldsymbol{w}|-1$ és $|\boldsymbol{w}|$ is gyenge periódusai $\boldsymbol{w}_\circ$−nek.

Az állítás szükségességének bizonyításához legyen $\boldsymbol{w} \in \{a,b\}^+$ egy olyan szó, melynek hosszára teljesül, hogy $|\boldsymbol{w}| \geq 2$. Továbbá tegyük fel, hogy $\boldsymbol{w}_\circ$ nem tartalmaz sem aa, sem bb faktort. Ekkor $|\boldsymbol{w}|$ páros, különben vagy $\boldsymbol{w}$, vagy $\sigma^1(\boldsymbol{w})$ tartalmazna aa vagy bb faktort, amely ellentmond az előző feltevésünknek. Tehát $\boldsymbol{w}_\circ = \{(ab)^k, (ba)^k\}$, ahol $k = \frac{|\boldsymbol{w}|}{2}$. Azaz $\boldsymbol{w}_\circ$ valamennyi gyenge periódusa $2m$ alakú, ahol $m \leq \frac{|\boldsymbol{w}|}{2}$. Ebben az esetben bármely két különböző gyenge periódus különbsége legalább kettő. ■

A 2.3. állítás nem feltétlenül áll fenn kettőnél több különböző szimbólumot tartalmazó szavak esetén, ahogy a következő példa is mutatja.

2.3. *Példa.* Vegyük a $\boldsymbol{w}_\circ = (abacbacb)_\circ$ körszót, melynek nem faktora sem aa, sem bb, sem cc. A 2.2. táblázatban láthatjuk $\boldsymbol{w}$ valamennyi konjugáltját, azaz $\boldsymbol{w}_\circ$ elemeit a periódusaikkal együtt.

$abacbacb - 8$	$bacbabac - 5,8$
$bacbacba - 3,6,8$	$acbabacb - 5,8$
$acbacbab - 8$	$cbabacba - 5,8$
$cbacbaba - 8$	$babacbac - 8$

2.2. táblázat. Az *abacbacb* szó konjugáltjai és periódusaik.

Ebből kiderül, hogy mind 5, mind 6 gyenge periódusa $\boldsymbol{w}_\circ$−nek. Tehát legalább hárombetűs ábécé esetén létezik olyan szó, amely nem teljesíti a 2.3. állítást.

Lineáris szavaknál szót ejtettünk a periódusok és keretszavak kapcsolatáról. A két fogalom között fennálló kapcsolat alapján megfelelően kis értékű gyenge periódusokból következtethetünk további gyenge periódusok létezésére, ahogy a következő állítás is mutatja.

2.4. Állítás (Körszavak kis periódusairól [6])**.** *Ha $\boldsymbol{w}_\circ$ gyenge periódusa $p \leq \frac{|\boldsymbol{w}|}{2}$, akkor $|\boldsymbol{w}| - p$ is gyenge periódusa $\boldsymbol{w}_\circ-$nek.*

Bizonyítás. Tegyük fel, hogy $p \leq \frac{|\boldsymbol{w}|}{2}$ periódusa valamely $\sigma^k(\boldsymbol{w}) \in \boldsymbol{w}_\circ$ szónak. Ekkor $\sigma^p(\sigma^k(\boldsymbol{w}))-$nek van egy p hosszú keretszava, azaz periódusa $|\boldsymbol{w}| - p$, amely egyben gyenge periódusa is $\boldsymbol{w}_\circ-$nek. ■

Hagyományos, lineáris szavak esetén minden $\boldsymbol{w} \in \Sigma^*$ szóhoz létezik legalább egy olyan $\boldsymbol{r} \in \Sigma^*$ szó amelyre $\boldsymbol{w} = \boldsymbol{r}^{\frac{n}{\ell}}$, ahol $n = |\boldsymbol{w}|$ és $\ell = |\boldsymbol{r}|$. A legrövidebb ilyen $\boldsymbol{r}$ szót nevezzük a $\boldsymbol{w}$ szó ***gyökének***, vagy generátorának. A lineáris szavak gyöke valamennyi esetben egyértelműen meghatározott [46, 12–13. oldal]. Az $\boldsymbol{r}^{\frac{n}{\ell}}$ alakú felírást szokás ***normál formának*** is nevezni.

A körszavakat bármelyik elemük egyértelműen meghatározza. Ezt kihasználva definiálhatunk egy, a fentihez hasonló fogalmat.

2.3. Definíció. *Legyen $\boldsymbol{w} \in \Sigma^*$ egy tetszőleges szó. Azt mondjuk, hogy $\boldsymbol{w}_\circ-$nek az $\boldsymbol{r} \in \Sigma^*$ szó* ***gyöke****, ha $|\boldsymbol{w}| = n$, $|\boldsymbol{r}| = \ell$ és $\boldsymbol{r}^{\frac{n}{\ell}} \in \boldsymbol{w}_\circ$, valamint $\boldsymbol{w}_\circ-$nek nincs $\ell-$nél kisebb gyenge periódusa.*

Tehát a legkisebb gyenge periódussal rendelkező elemből indulunk ki. Itt a körszó gyöke nem mindig egyértelműen meghatározott, ez látszik a 2.4. példából is.

2.4. *Példa.* Az $(abbabbb)_\circ$ körszó elemei periódusaikkal együtt a 2.3. táblázatban láthatók.

$abbabbb$ – 7,	$bbabbba$ – 4, 7,
$babbbab$ – 4, 6, 7,	$abbbabb$ – 4, 7,
$bbbabba$ – 7,	$bbabbab$ – 3, 6, 7,
$babbabb$ – 3, 6, 7.	–

2.3. táblázat. Az $(abbabbb)_\circ$ körszó elemei periódusaikkal.

Látható, hogy mind $bbabbab$–nek, mind $babbabb$–nek periódusa 3, így a körszó gyöke bba és bab is.

A körszó gyökeinek megtalálása egyike azon feladatoknak, melynek megoldásához a gyenge periódusok kiszámolása, egymáshoz való viszonyuk vizsgálata szükséges. A $\boldsymbol{w}_\circ$ körszó gyenge periódusainak felsorolása megoldható úgy, hogy megkeressük valamennyi elemének az összes periódusát, majd vesszük ezek unióját. Ezzel az eljárással az a probléma, hogy a körszó belső szerkezetét teljesen figyelmen kívül hagyja. Ezzel szemben mi arra törekszünk, hogy a körszó struktúrája és a lehetséges periódusok közötti összefüggéseket feltárjuk.

A következő tételben megfogalmazunk elégséges és szükséges feltételeket ahhoz, hogy egy adott körszó rendelkezzen adott gyenge periódussal.

2.1. Tétel (Gyenge periódus létezésének szükséges és elégséges feltétele [2])**.** *Legyen $\boldsymbol{w} \in \Sigma^*$ egy tetszőleges szó. Ha p a $\boldsymbol{w}_\circ$ egy gyenge periódusa, akkor létezik olyan $|\boldsymbol{w}| - p$ hosszú $\boldsymbol{x} \in \Sigma^*$ szó és $\ell \in \mathbb{N}_0$, hogy $\sigma^\ell(\boldsymbol{x})\boldsymbol{x}$ a $\boldsymbol{ww}$ egy faktora.*

A következők is teljesülnek:

I. Ha $\boldsymbol{ww}$ tartalmaz egy olyan $\boldsymbol{xx}$ ($\boldsymbol{x} \in \Sigma^+$) faktort, melyre $|\boldsymbol{x}| < |\boldsymbol{w}|$, *akkor* $p = |\boldsymbol{w}| - |\boldsymbol{x}|$ *a* $\boldsymbol{w}_\circ$ *egy gyenge periódusa.*

II. Ha létezik olyan $p > 0$ *egész, hogy a következő feltételek fennállnak:*

- $|\boldsymbol{w}| = k \cdot p + r$ *valamely* $k > 0$, $0 \leq r < p$ *esetén, valamint*
- *létezik olyan* p *hosszú* $\boldsymbol{v} \in \Sigma^*$ *szó és annak egy* $\boldsymbol{v'}$ *kezdőszelete, hogy valamely* $0 < s < k$ *esetén a* $\boldsymbol{ww}$ *szó egy faktora*

$$\sigma^{s\cdot p}(\boldsymbol{v}^{k-1}\boldsymbol{v'})\sigma^{(s-1)\cdot p}(\boldsymbol{v}^{k-1}\boldsymbol{v'}),$$

akkor p *a* $\boldsymbol{w}_\circ$ *körszó gyenge periódusa.*

Bizonyítás. Ha $\boldsymbol{w}$ periódusa p, akkor létezik egy $|\boldsymbol{w}|-p$ hosszú $\boldsymbol{x}$ keretszava is. Ekkor a $\boldsymbol{ww}$ szó nyilvánvalóan tartalmaz egy $2 \cdot (|\boldsymbol{w}| - p)$ hosszú $\boldsymbol{xx}$ négyzetszót.

Természetesen a $\boldsymbol{w}_\circ$ egy gyenge periódusa nem feltétlenül lesz a w egy periódusa. Tegyük fel, hogy p valamely $\boldsymbol{u} \in \boldsymbol{w}_\circ$ szó periódusa. Ha $p > \left\lfloor \frac{|\boldsymbol{w}|}{2} \right\rfloor$, akkor $\boldsymbol{uu}$ bármely ciklikus elforgatása (beleértve $\boldsymbol{ww}$–t is) tartalmaz egy $2 \cdot (|\boldsymbol{w}| - p)$ hosszú négyzetszót.

Abban az esetben, ha $p \leq \left\lfloor \frac{|\boldsymbol{w}|}{2} \right\rfloor$, az $\boldsymbol{u}$ szó felírható $\boldsymbol{u} = \boldsymbol{v}^k\boldsymbol{v'}$ alakban, ahol $\boldsymbol{v}, \boldsymbol{v'} \in \Sigma^*$ és k egy pozitív egész, továbbá $|\boldsymbol{v}| = p$ és $\boldsymbol{v'}$ a $\boldsymbol{v}$ kezdőszelete. Különböző eseteket kell megvizsgálnunk attól függően, hogy $\boldsymbol{uu}$–ból milyen mértékű ciklikus forgatással kapható meg $\boldsymbol{ww}$. A bizonyítás további részében jelölje j azt a nemnegatív egész számot, melyre $\sigma^j(\boldsymbol{uu}) = \boldsymbol{ww}$ teljesül.

1. Ha $j \leq p$, akkor a $\sigma^j(\boldsymbol{uu})$ szó tartalmaz egy $\boldsymbol{xx} = (\boldsymbol{v}^{k-1}\boldsymbol{v'})^2 = \sigma^0(\boldsymbol{x})\boldsymbol{x}$ négyzetszót. Itt $\ell = 0$.
2. Ha $p < j \leq kp$, ahol $k > 1$, akkor legyen $s = \left\lfloor \frac{j}{p} \right\rfloor$. Ekkor a $\sigma^j(\boldsymbol{uu})$ szónak létezik egy $\boldsymbol{v}^{k-s-1}\boldsymbol{v'}\boldsymbol{v}^s\boldsymbol{v}^{k-s}\boldsymbol{v'}\boldsymbol{v}^{s-1}$ faktora, azaz $\boldsymbol{x}$–et vá-

lasszuk $\boldsymbol{v}^{k-s}\boldsymbol{v'}\boldsymbol{v}^{s-1}$–nek. Ebben az esetben ℓ pontosan $|\boldsymbol{v}| = p$–vel egyezik meg.

3. Abban az esetben, ha $kp \leq j \leq |\boldsymbol{u}|$, a $\sigma^j(\boldsymbol{uu})$ szó $\boldsymbol{z}\boldsymbol{v}^k\boldsymbol{v'}\boldsymbol{v}^k\boldsymbol{z'}$ alakú, ahol $\boldsymbol{z'z} = \boldsymbol{v'}$. Látható, hogy $(\boldsymbol{v}^{k-1}\boldsymbol{v'})^2$ faktora az $\sigma^j(\boldsymbol{uu})$ szónak. Azaz $\ell = 0$ és $\boldsymbol{x} = \boldsymbol{v}^{k-1}\boldsymbol{v'}$.

Most lássuk be a fordított esetek helyességét. Az I. pont bizonyításához tegyük fel, hogy $\boldsymbol{ww} \in \Sigma^+$ egy olyan szó, melynek faktora $\boldsymbol{xx} \in \Sigma^*$. Abban az esetben, ha $\boldsymbol{x}$ a $\boldsymbol{w}$ keretszava, vagy $\boldsymbol{xx}$ már $\boldsymbol{w}$–ben is előfordul, a tételben megfogalmazott tulajdonság triviális. Ezért tegyük fel, hogy $\boldsymbol{ww} = \boldsymbol{uxxv}$, valamely $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$ szavakra, melyek hossza egymástól különböző. Ha $|\boldsymbol{u}| < |\boldsymbol{v}|$, akkor $\boldsymbol{w} = \boldsymbol{uxx'} = \boldsymbol{x''v}$, ahol $\boldsymbol{x'x''} = \boldsymbol{x}$ és $|\boldsymbol{x'}| = |\boldsymbol{v}| - |\boldsymbol{u}|$. Vegyük észre, hogy $\boldsymbol{x}$ keretszava a $\sigma^{-|\boldsymbol{x'}|}(\boldsymbol{w}) = \boldsymbol{x'ux} = \boldsymbol{x'x''v'}$ szónak (ahol $\boldsymbol{v'}$ a $\boldsymbol{v}$ szó megfelelő kezdőszelete). Azaz $p = |\boldsymbol{w}| - |\boldsymbol{x}|$ a $\sigma^{-|\boldsymbol{x'}|}(\boldsymbol{w})$ szó egy periódusa (így $\boldsymbol{w}_\circ$ egy gyenge periódusa). A másik esetet, mikor is $|\boldsymbol{u}| > |\boldsymbol{v}|$, hasonló módon bizonyíthatjuk. Ekkor $\boldsymbol{w} = \boldsymbol{ux'} = \boldsymbol{x''xv}$, ahol $\boldsymbol{x'x''} = \boldsymbol{x}$ és $|\boldsymbol{x'}| = |\boldsymbol{u}| - |\boldsymbol{v}|$. Ezúttal is találhatunk a $\boldsymbol{w}$ szónak olyan elforgatását, név szerint a $\sigma^{-|\boldsymbol{x''}|}(\boldsymbol{w})$ szót, amelynek keretszava $\boldsymbol{x}$, így periódusa p.

A II. pontban leírt feltétel akkor állhat elő, ha $\boldsymbol{w}_\circ$ gyenge periódusa $p \leq \left\lfloor \frac{|w|}{2} \right\rfloor$, de $\boldsymbol{ww}$ nem tartalmaz $2\cdot(|w| - p)$ hosszú négyzetfaktort. Tegyük fel, hogy $k, p, \boldsymbol{v}, \boldsymbol{v'}, \boldsymbol{u}, s$ a feltételeket teljesítik. Ekkor

$$\boldsymbol{ww} = \boldsymbol{y}\sigma^{s\cdot p}(\boldsymbol{v}^{k-1}\boldsymbol{v'})\sigma^{(s-1)\cdot p}(\boldsymbol{v}^{k-1}\boldsymbol{v'})\boldsymbol{y'} =$$

$$\boldsymbol{y}\boldsymbol{v}^{k-s-1}\boldsymbol{v'}\boldsymbol{v}^{s}\boldsymbol{v}^{k-s}\boldsymbol{v'}\boldsymbol{v}^{s-1}\boldsymbol{y'} =$$

$$\boldsymbol{y}\boldsymbol{v}^{k-s-1}\boldsymbol{v'}\boldsymbol{v}^{k}\boldsymbol{v'}\boldsymbol{v}^{s-1}\boldsymbol{y'}$$

teljesül valamilyen $\boldsymbol{y}, \boldsymbol{y'} \in \Sigma^*$ szavakra. Azaz

$$\sigma^{|\boldsymbol{y}|+(k-s-1)\cdot p+|\boldsymbol{v'}|}(\boldsymbol{ww}) = \boldsymbol{v}^k\boldsymbol{v'}\boldsymbol{v}^{s-1}\boldsymbol{y'}\boldsymbol{y}\boldsymbol{v}^{k-s-1}\boldsymbol{v'}.$$

Ennek a szónak pedig létezik egy $|\boldsymbol{w}|$ hosszú kezdőszelete, pontosan az $\boldsymbol{u} \in \boldsymbol{w}_\circ$ szó, melynek periódusa p. Tehát p a $\boldsymbol{w}_\circ$ gyenge periódusa. ■

Nem triviális, hogy léteznek a II. pont bizonyításában tárgyalt szavak, ezért most adunk egy példát. Vegyük az $(abbbaabaabbba)_\circ$ körszót, melynek periódusa 5. Azonban, ahogy a 2.4. táblázatban megfigyelhető, az $abbbaabaabbbaabbbaabaabbba$ szóban nem található 16 betűből álló négyzetszó. Hárombetűs ábécé feletti példaként említhetnénk a $(bcababca)_\circ$ körszót, melynek a 3 gyenge periódusa, de a $bcababcabcababca$ szónak nincs 10 hosszúságú négyzetfaktora.

$abbbaaba \cdot abbbaabb$	$baabbbaa \cdot bbbaabaa$
$bbbaabaa \cdot bbbaabbb$	$aabbbaab \cdot bbaabaab$
$bbaabaab \cdot bbaabbba$	$abbbaabb \cdot baabaabb$
$baabaabb \cdot baabbbaa$	$bbbaabbb \cdot aabaabbb$
$aabaabbb \cdot aabbbaab$	$bbaabbba \cdot abaabbba$
$abaabbba \cdot abbbaaba$	

2.4. táblázat. Az $abbbaabaabbbaabbbaabaabbba$ szó 16 hosszúságú faktorai.

A továbbiakban a különböző gyenge periódusok egymáshoz való viszonyát vizsgáljuk. Tegyük fel, hogy algoritmussal el szeretnénk dönteni adott $n, q, p \in \mathbb{N}$ értékek esetén, hogy, ha létezik, akkor melyik az a lehető legnagyobb Σ ábécé, hogy létezik olyan $\boldsymbol{w} \in \Sigma^*$ szó, melynek hossza n és q, p a $\boldsymbol{w}_\circ$ gyenge periódusai. A következő mohó algoritmus segítségével megkonstruálható valamennyi ilyen lehetséges $\boldsymbol{w}$ szó. Az algoritmus feltételezi, hogy $q < p < n$, továbbá, az általánosság csorbítása nélkül feltesszük, hogy q a $\boldsymbol{w}$ szó, p pedig $\boldsymbol{w}$ valamely konjugáltjának periódusa.

1: **function** KORSZOKERESO(n, p, q)
2:     **for** $i \leftarrow 0$ **to** $n-1$ **do**
3:         $w[i+1] \leftarrow (i \bmod q) + 1$
4:     $\ell \leftarrow 1$
5:     $azonosBetuk \leftarrow [I_{q\times q}]^n$
6:     **while** $\ell \le n$ **do**
7:         **for** $i \leftarrow 0$ **to** $n-p$ **do**
8:             $a \leftarrow w[((i+\ell-1) \bmod n)+1]$
9:             $b \leftarrow w[((i+\ell+p-1) \bmod n)+1]$
10:             **if** $azonosBetuk[\ell][a][b] = 0$ **then**
11:                 $azonosBetuk[\ell][a][b] \leftarrow 1$
12:                 $azonosBetuk[\ell][b][a] \leftarrow 1$
13:         **for** $i \leftarrow 1$ **to** q **do**
14:             **for** $j \leftarrow 1$ **to** q **do**
15:                 **if** $azonosBetuk[\ell][i][j] = 1$ **then**
16:                     **for** $k \leftarrow 1$ **to** q **do**
17:                         **if** $azonosBetuk[\ell][i][k] = 1$ **then**
18:                             $azonosBetuk[\ell][j][k] \leftarrow 1$
19:         $\ell \leftarrow \ell + 1$
20:     **return** $azonosBetuk$

Az algoritmus a 2.–3. sorokban feltölt egy n méretű tömböt az 1, …, q értékekkel. Ez reprezentálja azt, hogy a $\boldsymbol{w}$ szónak q a periódusa, azaz $\boldsymbol{w} = (w_1 \cdots w_q)^{\frac{n}{q}}$. Ezután az összes lehetséges elforgatáshoz rendel egy $q \times q$ dimenziós tömböt, amely azt jelöli, hogy adott elforgatás esetén a w_1, …, w_q betűk közül melyek szükségszerűen azonosak. A kezdeti feltételezésünk az, hogy a w_1, …, w_q betűk páronként különböznek, ezért valamennyi $\ell = 1$, …, n esetén az $azonosBetuk[\ell]$ tömb a $q \times q$ méretű egységmátrixszal egyezik meg. A **while** cikluson belül egyenként frissíti a tömböt feltételezve azt, hogy $\sigma^{\ell}(\boldsymbol{w})$ periódusa p (7.–12. sor). A ciklus végén lévő 13.–18. sorok az egyenlőség tranzitivitása miatt szükségesek. Az algoritmus végül visszatér az $azonosBetuk$ tömbbel, ami az egyenlőségi mátrixokat tartalmazza.

Egy-egy $azonosBetuk-$beli mátrix egy betűk feletti ekvivalenciarelációt reprezentál. Ha azt a mátrixot választjuk, melyben a lehető legtöbb a különböző ekvivalenciaosztályok száma, akkor megkapjuk, hogy melyik az a legnagyobb ábécé, amelyik felett létezik a keresett tulajdonságokkal rendelkező szó.

2.5. *Példa.* Az algoritmust futtatva az $n = 7$, $p = 4$, $q = 3$ paraméterekkel a következő $azonosBetuk-$beli mátrixokat kapjuk.

$$azonosBetuk[1] = azonosBetuk[6] =$$

$$azonosBetuk[7] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$azonosBetuk[2] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$azonosBetuk[3] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$azonosBetuk[4] = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$azonosBetuk[5] = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Az $azonosBetuk[1]$, $azonosBetuk[6]$, vagy $azonosBetuk[7]$ mátrixok bármelyikét választva a körszó valamennyi betűje megegyezik, azaz egybetűs ábécé feletti szót kapnánk. Érdemes tehát valamelyik másik mátrixszal dolgozni, ha olyan szót szeretnénk konstruálni, melyben nem minden betű azonos. Például az $azonosBetuk[4]$ mátrixot választva láthatjuk, hogy egyetlen megszorításunk az, hogy $w_1 = w_4$. Tehát w_2 és w_3 tetszőleges betűk lehetnek. Válasszuk például a $w_1 = w_4 = a$, $w_2 = b$ és $w_3 = c$ értékeket. Ekkor a $\boldsymbol{w} = abcaabc$ szót kapjuk, melynek periódusa 4, ahogy a kiindulásnál is feltettük, valamint a $\sigma^4(\boldsymbol{w}) = abcabca$ szó periódusa 3. Tehát $\boldsymbol{w}_\circ$ gyenge periódusa 3 és 4 is.

Nem véletlenül választottuk a 3 és 4 értékeket. Az egymással relatív prím periódusok nagy hatással vannak a szavak szerkezetére. A hagyományos, lineáris szavak esetében a 2.2. lemma igen erős megszorítást mond ki adott nem unáris $\boldsymbol{w}$ szó hosszára, ha két olyan periódusa van, melyek relatív prímek. Pontosabban, ha p és q relatív prímek (azaz $\text{lnko}(p, q) = 1$) a $\boldsymbol{w}$ szó periódusai, akkor a 2.2. lemma szerint $|\boldsymbol{w}| < p + q - 1$ vagy $\boldsymbol{w}$ unáris szó.

Felmerül a kérdés, hogy tudunk-e hasonló feltételt mondani körszavak relatív prím gyenge periódusairól. Ezzel jellemezhetjük, hogy egy adott szó elforgatása esetén mennyire maradnak meg a periodikus tulajdonságok. A következő tétel írja le a relatív prím gyenge periódusok kapcsolatát.

2.2. Tétel (Relatív prím gyenge periódusok [2]). *Legyenek $p, q \in \mathbb{N}$ olyanok, hogy $2 \leq q < p$. Továbbá, legyen $\boldsymbol{w} \in \Sigma^+$ egy olyan szó, hogy $\boldsymbol{w}_\circ$ gyenge periódusa p és q. Jelölje m a (p mod q) értéket. Ha* $\operatorname{lnko}(p, q) = 1$ *és $\boldsymbol{w}$ nem unáris szó, akkor $|\boldsymbol{w}|$−ra teljesül a következő egyenlőtlenség:*

$$|\boldsymbol{w}| \leq \begin{cases} 2p - m\,, & \text{ha } m = 1, \\ 2p + q - m\,, & \text{ha } m > 1. \end{cases}$$

Bizonyítás. A két esettel külön foglalkozunk. Tegyük fel, hogy p és q relatív prímek, valamint $m = (p \bmod q) = 1$. Először megmutatjuk, hogy lehet olyan $2p - m$ hosszú $\boldsymbol{w}$ szót konstruálni, hogy $\boldsymbol{w}_\circ$ gyenge periódusai között van p és q is, majd belátjuk, hogy minden $\boldsymbol{w}$−nél hosszabb szó csakis unáris lehet, ha a belőle képzett körszónak gyenge periódusa p és q.

Megjegyezzük, hogy, ha $q < p$ és $(p \bmod q) = m$, akkor létezik olyan $k \in \mathbb{N}$, hogy $p = kq + m$. Tehát a szó hosszának felső korlátjait $2p - m = 2kq + m$, valamint $2p + q - m = (2k + 1)q + m$ formában is megadhatjuk.

Legyen $\boldsymbol{v} \in \Sigma^*$ egy q hosszú nem unáris szó és $\boldsymbol{u} = \boldsymbol{v}^{\frac{p}{q}} = \boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor} v_1$. Könnyen látható, hogy a $\boldsymbol{w} = \boldsymbol{u}\boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor}$ szó periódusa p, mivel $|\boldsymbol{u}| = p$ és $\boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor}$ az $\boldsymbol{u}$ kezdőszelete. Továbbá a $\sigma^p(\boldsymbol{w}) = \boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor}\boldsymbol{u} = \boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor}\boldsymbol{v}^{\left\lfloor \frac{p}{q} \right\rfloor} v_1$ szó periódusa q. Tehát $\boldsymbol{w}_\circ$ gyenge periódusa p és q, hossza pedig $q \cdot \left\lfloor \frac{p}{q} \right\rfloor + q \cdot \left\lfloor \frac{p}{q} \right\rfloor + 1 = 2 \cdot (p - 1) + 1 = 2p - 1$. Be kell még látnunk, hogy nem létezik olyan nem unáris $\boldsymbol{z}$ szó, mely hosszabb, mint $\boldsymbol{w}$ és $\boldsymbol{z}_\circ$ gyenge periódusa p, valamint q is. Legyen $\boldsymbol{z} \in \Sigma^*$ úgy, hogy $|\boldsymbol{z}| = |\boldsymbol{w}_\circ| + r$, ahol $r \in \mathbb{N}$. Azt kell megmutatnunk, hogy tetszőleges $r \in \mathbb{N}$ esetén $\boldsymbol{z}$ unáris szó. Feltehetjük, hogy q periódusa $\boldsymbol{z}$−nek, azaz $\boldsymbol{z} = (z_1 \cdots z_q)^{\frac{2kq+m+r}{q}} = (z_1 \cdots z_q)^{2k + \frac{1+r}{q}}$.

Induljunk ki a

$$\boldsymbol{z}' = (z_1 \cdots z_q)^k z_1 \cdots z_{r+1} (z_1 \cdots z_q)^k$$

szóból, amely a $\boldsymbol{z}-$nek egy elforgatása, pontosabban $\boldsymbol{z}' = \sigma^{kq}(\boldsymbol{z})$. Ha $\boldsymbol{z}'-$t legalább q pozícióval jobbra vagy balra forgatjuk, akkor mindkét esetben tartalmaz a szó egy $(z_1 \cdots z_q)^{k+1}$ faktort, melynek periódusa q. Ha ennek a faktornak p is periódusa lenne, akkor a 2.2. lemma értelmében $\boldsymbol{z}$ unáris. Tehát csakis kisebb mértékű forgatásokat vehetünk számításba. Továbbá azt is láthatjuk, hogy $r < q - 1$ ugyancsak a fenti gondolatmenet alapján. Indirekt feltevésünk szerint $p = kq + 1$ periódusa a $\boldsymbol{z}'$ egy elforgatásának. Legyen $\ell < q$ és vegyük a $\sigma^{-\ell}(\boldsymbol{z}')$ jobbra történő elforgatást.

$$z_{q-\ell+1} \cdots z_q (z_1 \cdots z_q)^k z_1 \cdots z_{r+1} (z_1 \cdots z_q)^{k-1} z_1 \cdots z_{q-\ell}$$

Ekkor a következő összefüggéseket figyelhetjük meg:

- Ha $k \geq 2$, akkor

  $$z_1 \cdots z_r = z_2 \cdots z_{r+1},$$

  $$z_{r+1} \cdots z_q = z_1 \cdots z_{q-r},$$

  $$z_1 \cdots z_r = z_{q-r+1} \cdots z_q,$$

  melyekből következik, hogy $\boldsymbol{z}$ periódusa 1, azaz unáris szó.

- Ha $k = 1$, akkor a $z_{q-\ell+1} \cdots z_q z_1 \cdots z_q z_1 \cdots z_{r+1} z_1 \cdots z_{q-\ell}$ szóról megállapíthatjuk a következőket:

  $$z_{q-\ell+1} \cdots z_{q-1} z_q = z_{q-\ell+2} \cdots z_q z_1,$$

  $$z_1 \cdots z_r = z_2 \cdots z_{r+1},$$

  $$z_{r+1} \cdots z_q = z_1 \cdots z_{q-r}, \text{ ha } r > \ell,$$

  $$z_{r+1} \cdots z_{q-\ell+r} = z_1 \cdots z_{q-\ell}, \text{ ha } r \leq \ell.$$

Ezekből az összefüggésekből is az következik, hogy $\boldsymbol{z}$ valamennyi betűje azonos.

A balra való forgatás esete ezzel analóg módon belátható. Ezzel a tétel teljesülését az $m = 1$ esetben igazoltuk.

Most legyen $m = (p \bmod q) > 1$ és vegyünk olyan $\boldsymbol{v} \in \Sigma^*$ szót, melynek hossza q és létezik egy $q - m$ hosszú keretszava, melyet $\boldsymbol{s}$−sel jelölünk. Legyen $\boldsymbol{u} = \boldsymbol{v}^k\boldsymbol{v'}$, úgy, hogy $|\boldsymbol{u}| = p$ és $\boldsymbol{v'} = v_1 \cdots v_m$. Megkonstruáljuk a $\boldsymbol{w} = \boldsymbol{u}\boldsymbol{v}^{\lceil \frac{p}{q} \rceil} = \boldsymbol{v}^k\boldsymbol{v'}\boldsymbol{v}^{k+1} = \boldsymbol{u}^2\boldsymbol{s}$ szót. Mivel $\boldsymbol{s}$ a $\boldsymbol{v}$ keretszava, ezért az $\boldsymbol{u}$ kezdőszelete is. Ebből az következik, hogy $\boldsymbol{w}$ periódusa $|\boldsymbol{u}| = p$. A $\boldsymbol{w}$ szó p pozícióval történő balra forgatásával megkapjuk a $\sigma^p(\boldsymbol{w}) = \boldsymbol{v}^{k+1}\boldsymbol{v}^k\boldsymbol{v'}$ szót, melynek $|\boldsymbol{v}| = q$ nyilvánvalóan periódusa. Tehát konstruáltunk olyan $\boldsymbol{w}$ szót, hogy a $\boldsymbol{w}_\circ$ körszó gyenge periódusa p és q is, továbbá, $\boldsymbol{w}$ hossza pontosan $2p + q - m$. Itt is kifejezhetjük p−t a q−ból, azaz $p = kq + m$, így a $\boldsymbol{w}$ szó hossza $(2k + 1)q + m$. Hasonlóan az előző gondolatmenethez, tegyük fel, hogy létezik olyan $\boldsymbol{z}$ szó, melynek hossza $|\boldsymbol{z}| = |\boldsymbol{w}| + r$ és $\boldsymbol{z}_\circ$ gyenge periódusai között szerepel p, q. Ebben az esetben is létezik egy

$$\boldsymbol{z'} = (z_1 \cdots z_q)^k z_1 \cdots z_{m+r}(z_1 \cdots z_q)^{k+1}$$

szó. Most is feltehetjük, hogy $r < q - 1$, hiszen ellenkező esetben minden elforgatás tartalmazna egy olyan $(k+1)q + m - 1$ hosszú faktort, melynek periódusa p és q. Feltesszük, hogy valamely $\ell < q$ esetén $\sigma^\ell(\boldsymbol{z'})$ vagy $\sigma^{-\ell}(\boldsymbol{z'})$ periódusa p. Vizsgáljuk most is a jobbra forgatást, azaz a $\sigma^{-\ell}(\boldsymbol{z'})$ szót:

$$z_{q-\ell+1} \cdots z_q(z_1 \cdots z_q)^k z_1 \cdots z_{r+m}(z_1 \cdots z_q)^k z_1 \cdots z_{q-\ell}$$

melynek az indirekt feltevésünk szerint periódusa $p = kq + m$. Az $m = 1$ esethez hasonló összefüggések írhatók fel.

- Ha $k \geq 2$, akkor

$$z_1 \cdots z_r = z_{1+m} \cdots z_{r+m},$$

$$z_{r+1}\cdots z_q = z_1\cdots z_{q-r},$$

$$z_1\cdots z_r = z_{q-r+1}\cdots z_q,$$

$$z_{r+1}\cdots z_{r+m} = z_1\cdots z_m,$$

melyekből következik, hogy $\boldsymbol{z}$ periódusa 1, azaz unáris szó.

- Ha $k = 1$, akkor a

$$z_{q-\ell+1}\cdots z_q z_1\cdots z_q z_1\cdots z_{r+m} z_1\cdots z_q z_1\cdots z_{q-\ell}$$

szóban teljesül a $k \geq 2$ eset első három összefüggése, valamint, ha $m \geq \ell$, akkor

$$z_{q-\ell+1}\cdots z_q = z_{m-\ell+1}\cdots z_m,$$

$m < \ell$ esetén pedig

$$z_{q-\ell+1}\cdots z_{q-m} = z_{q+m-\ell}\cdots z_q \quad \text{és} \quad z_{q-m+1}\cdots z_q = z_1\cdots z_m.$$

Tehát $\boldsymbol{z}$ valamennyi betűje azonos.

Azaz nem konstruálható a megadott korlátoknál hosszabb olyan nem unáris szó, amelyből p, q periódusokkal rendelkező körszó képezhető. ∎

A következő tétel pedig kétbetűs ábécé feletti nem unáris körszavak különböző gyenge periódusainak maximális számára ad pontos felső korlátot.

2.3. Tétel (Különböző gyenge periódusok maximális száma [2])**.** *Legyen $n \geq 3$. Egy $\{a, b\}$ ábécé feletti n hosszú nem unáris szóból képzett körszó különböző gyenge periódusainak száma legfeljebb*

$$\begin{cases} \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{4} \right\rfloor + 1, & \text{ha } n \text{ páratlan,} \\ \frac{n}{2} + 1, & \text{ha } n \text{ páros és } (n \bmod 3) = 2, \\ \frac{n}{2}, & \text{egyébként.} \end{cases}$$

Bizonyítás. Legyen $\boldsymbol{w} \in \Sigma^*$ egy $n \geq 3$ hosszú nem unáris szó és jelöljük P_w−vel a $\boldsymbol{w}_\circ$ gyenge periódusainak halmazát. Tehát, P_w az $\{1, \ldots, n\}$ halmaz egy részhalmaza. Mivel csakis nem unáris szavakból képzett körszavakkal foglalkozunk, használhatjuk a 2.2 tételt a különböző periódusok összeférhetőségnek ellenőrzésére. A P_w lehetséges elemeit pedig növekvő sorrendben vizsgáljuk. Tegyük fel, hogy P_w legkisebb eleme $k \in \{1, \ldots, n\}$. Ekkor $t \cdot k \in P_w$ is teljesül minden $t = 1, \ldots, \left\lfloor \frac{n}{k} \right\rfloor$ esetén. Így pontosan $\left\lfloor \frac{n}{k} \right\rfloor$ darab olyan elemet kapunk, amely biztosan eleme P_w−nek. Vegyük észre, hogy bármely $\ell \in P_w$ esetén, ha $\ell \leq \left\lfloor \frac{n}{2} \right\rfloor$, akkor ℓ a k többszöröse, különben a 2.2. tétel miatt $\boldsymbol{w}_\circ$ unáris lenne. Tehát a P_w-beli különböző elemek száma legfeljebb $\left\lfloor \frac{n}{2k} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + 1$. Ez az érték $k = 2$ esetén maximális, azaz $|P_w| \leq \left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + 1$. Továbbá, ha k osztja n−t, akkor a 2.4. állítás szerint kikövetkeztethető gyenge periódusok, valamint n, már szerepelnek a k többszörösei között, ezért nem számolhatjuk őket újra. Tehát a $|P_w|$ értékre ekkor teljesül, hogy $|P_w| \leq \left\lfloor \frac{n}{4k} \right\rfloor + \frac{n}{2} + 1 - \left\lfloor \frac{n}{4k} \right\rfloor - 1 = \left\lfloor \frac{n}{2} \right\rfloor$.

A határérték további finomítását különböző esetekre bontva tárgyaljuk.

1. Ha n páratlan, akkor $|P_w| \leq \left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + 1$ pontos felső korlát, mivel az $((ab)^{\left\lfloor \frac{n}{2} \right\rfloor} a)_\circ$ alakú körszavaknak pontosan $\left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + 1$ darab különböző gyenge periódusa van. Ezek fordított sorrendben n, a kettő többszörösei és azok, melyeket a 2.4. állítás segítségével kaphatunk az előbbiekből.

2. Tegyük fel, hogy n páros és $(n \bmod 3) = 2$. Ha n páros, észrevehetjük, hogy $P_w = \{2, 4, \ldots, n\}$−ből az következik, hogy $\boldsymbol{w}_\circ$−nek egyetlen páratlan gyenge periódusa sincs. Természetesen egy másik páros számot választva legkisebb gyenge periódusnak csak tovább csökkentjük a lehetséges gyenge periódusok számát. Tehát meg kell vizsgálnunk, hogy páratlan k választása esetén juthatunk-e több

gyenge periódushoz. Ahogy fentebb említettük, ha k osztja n−t, akkor $|P_w| \leq \left\lfloor \frac{n}{2} \right\rfloor$, tehát csak olyan k értékekkel kell foglalkoznunk, melyek nem osztói n−nek. Ekkor bármely $p > \left\lfloor \frac{n}{2} \right\rfloor$ gyenge periódus esetén vagy $p \in \{n-1,\, n-2,\, \ldots,\, n-(n \bmod k)+1\}$, vagy $p = n - t \cdot k$ valamely $t \in \{1,\, \ldots,\, \left\lfloor \frac{n}{k} \right\rfloor\}$ esetén. Tehát $|P_w| \leq \left\lfloor \frac{n}{2k} \right\rfloor + \left\lfloor \frac{n}{k} \right\rfloor + (n \bmod k) - 1 \leq \left\lfloor \frac{3n}{2k} \right\rfloor + (n \bmod k) - 1$. Ekkor $k = 3$ választásával maximalizálhatjuk az értéket, azaz $|P_w| \leq \left\lfloor \frac{3n}{6} \right\rfloor + 1 = \frac{n}{2} + 1$.

Ez a korlát pontos, mivel az $((aba)^{\left\lfloor \frac{n}{3} \right\rfloor} ab)_\circ$ alakú körszavak gyenge periódusai: n, $n-1$, $t \cdot 3$ minden $t = 1,\, \ldots,\, \left\lfloor \frac{n}{3} \right\rfloor$ −ra, valamint $n - t \cdot 3$ minden $t = 1,\, \ldots,\, \left\lfloor \frac{n}{6} \right\rfloor$ esetén.

3. A második ponthoz hasonlóan, k−t választhatjuk kettőnek, vagy egy olyan egész számnak, amely nem osztja n−t. Ha $k = 2$, akkor $P_w = \{2,\, 4,\, \ldots,\, n\}$, és $|P_w| \leq \left\lfloor \frac{3n}{2k} \right\rfloor + (n \bmod k) - 1$ minden más lehetséges k érték esetén. Vegyük észre, hogy, ha $k > 2$, akkor $\left\lfloor \frac{3n}{2k} \right\rfloor + (n \bmod k) - 1 \leq \frac{n}{2}$. A $\boldsymbol{w}_\circ$ legkisebb gyenge periódusa ebben az esetben 2 (vagy választható a 3, ha $(n \bmod 3) = 1$). Tehát $|P_w| \leq \frac{n}{2}$.

   A korlát ebben az esetben is pontos, mivel az $((ab)^{\frac{n}{2}})_\circ$ körszavaknak pontosan ennyi különböző gyenge periódusuk van.

Ezzel megmutattuk, hogy a tétel állításában megadott korlát valamennyi esetben pontos. ∎

2.3. Körszavak ábrázolása hierarchikus adatszerkezettel

Ebben a fejezetben különböző körszavak vizuális ábrázolásaival foglalkozunk. Ezeket a módszereket alkalmazva szemléltethetünk egyes tulajdonságokat, valamint foglalkozhatunk olyan kérdésekkel, melyek a leírási bonyolultságra irányulnak.

Szavak ábrázolása hierarchikus adatszerkezetekkel

Számos szövegeken operáló algoritmus esetén a műveletek gyorsabb végrehajtása miatt szokás a nem változó szöveg vagy minta hierarchikus adatszerkezetben való tárolása. Ilyen adatszerkezet például a zárószeletfa (suffix trie) [15].

2.4. Definíció. ***Fa*** *alatt olyan* $\tau = (V, E)$ *hierarchikus adatszerkezetet értünk, melyben*

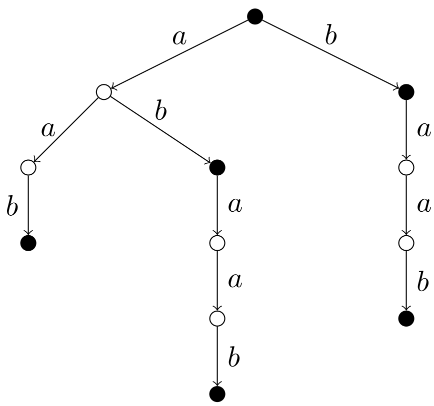
- V *a fa* ***csomópontjainak*** *halmaza,*
- $E \subset V \times V$ *pedig a fa* ***éleinek*** *halmaza.*

Továbbá teljesülnek a következők:

- *létezik egy* $r \in V$ *csomópont, a fa* ***gyökere****,*
- $(v, r) \notin E$, $(v, v) \notin E$ *bármely* $v \in V$*−re teljesül,*
- *tetszőleges* $u \in V$ *csomópont esetén létezik olyan* $u_1, \ldots, u_n \in V$ *csomópontok sorozata, hogy* $u_1 = r$, $u_n = u$ *és* $(u_i, u_{i+1}) \in E$ $(i = 1, \ldots, n-1)$*, valamint,*
- *ha* $(v, u) \in E$*, akkor nem léteznek olyan* $u_1, \ldots, u_n \in V$ *csomópontok, hogy* $u_1 = u, u_n = v$ *és bármely* $i = 1, \ldots, n-1$ *esetén*

$(u_i, u_{i+1}) \in E$.

A $\tau = (V, E)$ fa ***kitüntetett csomópontjait*** egy $K \subseteq V$ halmazzal adhatjuk meg. A $\boldsymbol{w} \in \Sigma^*$ szó ***zárószelet fája*** egy olyan $\tau_{\boldsymbol{w}}$ fa, melyben a gyökérből a kitüntetett csomópontokba vezető utak címkéiből előálló szavak pontosan $\boldsymbol{w}$ zárószeleteit adják. Egy ilyen fa látható a 2.4 ábrán. Megjegyezzük, hogy minden levélelem és a gyökér is kitüntetett csomópont.



2.4. ábra. A τ_{abaab} zárószelet fa.

Ilyen fák használatával igen egyszerű eldönteni azt, hogy adott $\boldsymbol{v}$ szó a $\boldsymbol{w}$ faktora-e, hiszen a megfelelő éleken lépkedve ezt karakterenként ellenőrizhetjük. A szakirodalomban számos olyan algoritmus található, amelyek zárószeletfák előállításával foglalkoznak, illetve azokat használják [15].

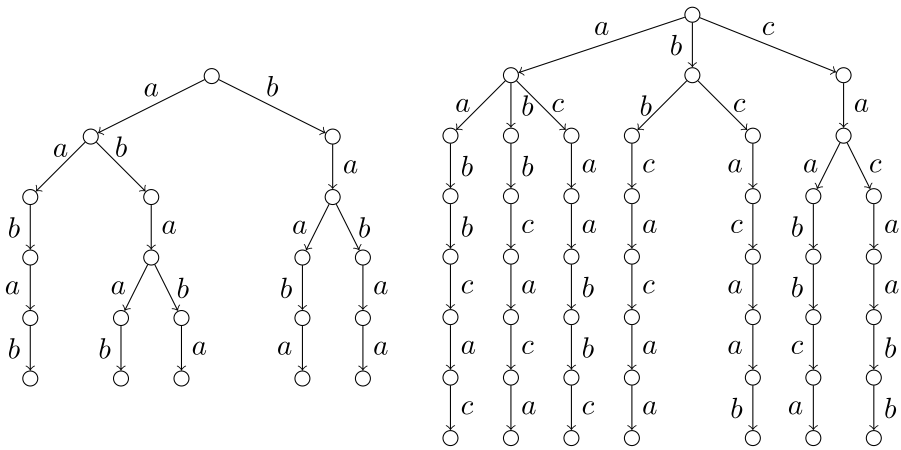
Körszavak ábrázolása fákkal

A fejezet további részében körszavak fákkal való reprezentálással foglalkozunk, mellyel egy adott körszó különböző elemeinek közös kezdőszeleteinek vizsgálatát alapozzuk meg. Körszavaknak az itt tárgyalt ábrázolási módja a [7] műben jelent meg először. A körszóhoz tartozó fa definíciója a következő.

2.5. Definíció. *Azt mondjuk, hogy a $\tau_{\boldsymbol{w}_\circ}$ a $\boldsymbol{w}_\circ$ körszó fája, ha $v = v_1 \cdots v_n \in \boldsymbol{w}_\circ$ pontosan akkor, ha léteznek $\tau_{\boldsymbol{w}_\circ}$−ben olyan $t_0, \ldots, t_n$ csomópontok, hogy*

- *t_0 a $\tau_{\boldsymbol{w}_\circ}$ fa gyökere, t_n levélelem, és*
- *minden $i = 1, \ldots, n$ esetén létezik a t_{i-1} csomópontból t_i csomópontba vezető él, melynek címkéje v_i.*

Ez az ábrázolási módszer a fentebb tárgyalt zárószeletfa adatszerkezethez erősen kapcsolódik, mint az a 2.5. ábrán is látható. Mivel a körszavak fáiban tetszőleges úton előálló szó zárószelete a körszó valamelyik elemének, ezért nem jelöljük külön a kitüntetett csomópontokat.



2.5. ábra. Az $(aabab)_\circ$ és $(aabbcac)_\circ$ körszavak fái.

A fának beszélhetünk különböző szintjeiről. Az $r \in V$ csomópont a nulladik szinten helyezkedik el, azaz $h(r) = 0$, ha r a fa gyökere, továbbá, ha u és v a fa csomópontjai és létezik u−ból v−be vezető közvetlen él, akkor $h(v) = h(u) + 1$.

Az, hogy a $\tau_{\boldsymbol{w}_\circ}$ fában mely szinten fordulnak elő elágazások, $\boldsymbol{w}_\circ$ elemeinek közös kezdőszeleteiből kikövetkeztethető. Pontosabban, a $k \in \mathbb{N}_0$ szinten pontosan akkor van elágazás, ha léteznek olyan $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{w}_\circ$ szavak, melyek leghosszabb közös kezdőszeletének hossza k. Ez teljesül a nulladik szintre is, mivel, ha $\boldsymbol{w}_\circ$ tartalmaz két különböző betűt, akkor létezik két olyan eleme, melyek már a legelső betűjükben különböznek, így leghosszabb közös kezdőszeletük az üresszó.

Továbbá azt is kimondhatjuk, hogy, ha a $\tau_{\boldsymbol{w}_\circ}$ fa a $k \in \mathbb{N}$ szintjén elágazik, akkor tartalmaz elágazást a $k-1$–edik szinten is. Ehhez kapcsolódó állítás, hogy, ha $\boldsymbol{w} \in \{a, b\}^*$ és a $\tau_{\boldsymbol{w}_\circ}$ fában van egy elágazás $|\boldsymbol{w}|-2$ magasságban, akkor pontosan egy elágazást tartalmaz minden $m = 0, \ldots, |\boldsymbol{w}|-2$ szinten. Ugyanis, tegyük fel, hogy a szinteket egyesével sorra vesszük a gyökérből indulva. Ha a gyökérnél van elágazás, akkor a lehetséges konjugáltak (körszó elemeinek) száma legalább kettő. Valamint minden további szinten lévő elágazás eggyel növeli a konjugáltak különböző kezdőszeleteinek számát. Ha feltételezzük, hogy minden további szinten egy elágazás van, akkor a $|\boldsymbol{w}| - 2$ magasságban lévő elágazást elérve a különböző kezdőszeletek száma $|\boldsymbol{w}|$, ami egy $|\boldsymbol{w}|$ hosszú körszó esetén maximális, tehát a fa sem a $|\boldsymbol{w}| - 1$, sem a $|\boldsymbol{w}|$ szinteken nem tartalmazhat elágazást, de minden fentebb lévő szinten pontosan egyet tartalmaz.

Ilyen fákat kapunk, ha úgynevezett ***Fibonacci szavakból*** [46] képzett körszavakat ábrázolunk (például a 2.5. ábra bal oldala).

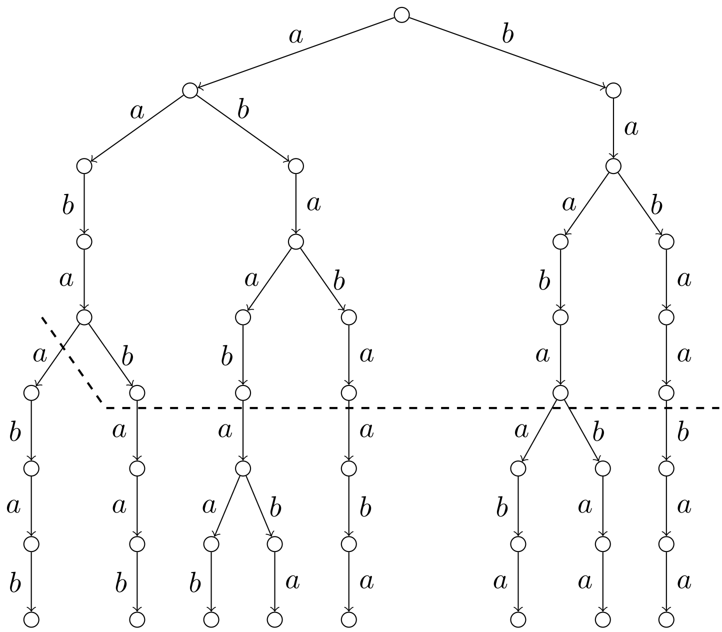
2.6. Definíció. *Legyen $\boldsymbol{f}_0 = b$, $\boldsymbol{f}_1 = a$ és tetszőleges $n \geq 2$ esetén $\boldsymbol{f}_n = \boldsymbol{f}_{n-1}\boldsymbol{f}_{n-2}$. A $\boldsymbol{f}_i$ szót ($i \in \mathbb{N}_0$) az i–edik **Fibonacci szónak,** a belőle képzett körszót pedig i–edik **Fibonacci körszónak** nevezzük.*

Ezen szavak hossza pontosan a Fibonacci számok sorozatát követi. Az

$\{\boldsymbol{f}_i\}_{i=0}^{\infty}$ rekurzív sorozat határértékét szokás nevezni ***végtelen Fibonacci szónak***. Megjegyezzük, hogy végtelen szavakból nem képzünk körszavakat, mivel nem beszélhetünk utolsó betűjükről, de a fával való ábrázolás lehetővé teszi, hogy a végtelen Fibonacci szót a véges Fibonacci szavakból kiindulva kezeljük.

Először vegyük az alábbi példát, amely szemlélteti a $\tau_{(\boldsymbol{f}_5)_\circ}$ és $\tau_{(\boldsymbol{f}_6)_\circ}$ fák kapcsolatát.

2.6. *Példa.* A 2.6. ábrán látható, hogy $\tau_{(\boldsymbol{f}_5)_\circ}$ gyökérből kiinduló részfája a $\tau_{(\boldsymbol{f}_6)_\circ}$ fának.



2.6. ábra. A $\tau_{(\boldsymbol{f}_5)_\circ}$ fa előfordulása $\tau_{(\boldsymbol{f}_6)_\circ}$ fában. Szaggatott vonal jelzi a $\tau_{(\boldsymbol{f}_5)_\circ}$ fa határát.

Belátjuk a következő állítást, amely a különböző Fibonacci körszavakhoz tartozó fák közötti kapcsolatot írja le.

2.5. Állítás (Fibonacci körszavakhoz tartozó fák hierarchiája [7]). *Bármely $2 < i < j$ esetén, $\tau_{(\boldsymbol{f}_i)_\circ}$ gyökérből kiinduló részfája a $\tau_{(\boldsymbol{f}_j)_\circ}$ fának.*

Bizonyítás. A definíció szerint $\boldsymbol{f}_n = \boldsymbol{f}_{n-1}\boldsymbol{f}_{n-2}$ tetszőleges $n \geq 2$ esetén. Elég belátni, hogy $(\boldsymbol{f}_j)_\circ$−ben $(\boldsymbol{f}_{j-1})_\circ$ minden eleme előfordul valamelyik szó kezdőszeleteként. Tudjuk, hogy

$$\boldsymbol{f}_{j-1} = \boldsymbol{f}_{j-2}\boldsymbol{f}_{j-3}, \text{ és}$$

$$\boldsymbol{f}_j = \boldsymbol{f}_{j-1}\boldsymbol{f}_{j-2} = \boldsymbol{f}_{j-2}\boldsymbol{f}_{j-3}\boldsymbol{f}_{j-2},$$

azaz $\sigma^k(\boldsymbol{f}_j)$ kezdőszelete $\sigma^k(\boldsymbol{f}_{j-1})$ minden $k = 0, \ldots, |\boldsymbol{f}_{j-2}|$ esetén. Továbbá $\boldsymbol{f}_{j-2} = \boldsymbol{f}_{j-3}\boldsymbol{f}_{j-4}$ és $\sigma^{|\boldsymbol{f}_{j-2}|}(\boldsymbol{f}_j) = \boldsymbol{f}_{j-3}\boldsymbol{f}_{j-2}\boldsymbol{f}_{j-3}\boldsymbol{f}_{j-4}$, melyből láthatjuk, hogy bármely $\ell = 1, \ldots, |\boldsymbol{f}_{j-3}| - 1$ esetén $\sigma^{\ell+|\boldsymbol{f}_{j-2}|}(\boldsymbol{f}_j)$ kezdőszelete $\sigma^{\ell+|\boldsymbol{f}_{j-2}|}(\boldsymbol{f}_{j-1})$. Tehát $\boldsymbol{f}_{j-1}$ valamennyi ciklikus elforgatása kezdőszelete $(\boldsymbol{f}_j)_\circ$ valamely elemének. Ebből következik, hogy $\tau_{(\boldsymbol{f}_{j-1})_\circ}$ gyökérből kiinduló részfája a $\tau_{(\boldsymbol{f}_j)_\circ}$ fának, ami maga után vonja az állítást. ■

A fenti állítás azt is jelenti, hogy a véges Fibonacci körszavak fái olyan sorozatot alkotnak, melynek beszélhetünk fixpontjáról, vagy határértékéről, amit jelöljünk $\tau_{\boldsymbol{f}_\circ}$−rel, mivel tekinthetünk rá úgy, mintha a végtelen Fibonacci „körszóhoz” tartozó fa lenne. Ebben a fában a gyökérben kezdődő valamennyi út az $\boldsymbol{f}$ szó egy faktorát adja. Ez a faktor természetesen lehet $\boldsymbol{f}$ egy végtelen zárószelete is, ha véges sok lépés után nem állunk meg. Tehát levonhatjuk az alábbi következtetést.

2.1. Következmény (A végtelen Fibonacci szó bizonyos faktorairól [7]). *A végtelen Fibonacci szónak valamennyi $(\boldsymbol{f}_i)_\circ$−beli szó faktora bármely $i \in \mathbb{N}_0$ esetén.* □

2.4. Összefoglalás

A dolgozat első részét ezzel a rövid összefoglalással zárjuk. Betekintést nyerhettünk a szavak kombinatorikájának egy kis szeletébe és megismerhettünk egy olyan modellt, melyben elméleti és remélhetőleg gyakorlati szempontból is nagy lehetőségek rejlenek. A körszavak kombinatorikája nem túl sűrűn kutatott terület, de, ahogy a [17, 18, 45] művekből is kiderül, a szakirodalomban nem hagyják teljesen figyelmen kívül.

Ahogy a 2.2. fejezetből is látszik, saját munkámmal a körszavak periodikus tulajdonságainak meghatározásához, annak egy új megközelítéséhez járultam hozzá. A fejezet teljes mértékben saját eredményeket tartalmaz, melyeket témavezetőm irányítása mellett számoltam ki. Eredményeim eddig két konferenciakiadványban, valamint egy folyóiratcikkben jelentek meg [6, 7, 2]. A dolgozatban a legtöbb bizonyítást átdolgoztam, bizonyos részeket jobban kifejtettem. A 2.3. és a 2.4. állítások, valamint a 2.2. és a 2.3. tételek a [6] cikkben kerültek közlésre. Ezen közleményre építve dolgoztam ki a 2.1. tételt, amely a 2.1. és a 2.2. állításokkal a [2] folyóiratcikkben kapott helyet. A 2.3. fejezet pedig a [7] eredményein alapul.

Tulajdonképpen a 2.2. és a 2.3. tételek egy 2.2. lemmához hasonló eredmény felé vezető út alapkövei. Az előbbi azt adja meg, hogy milyen feltételek mellett lehet egy nem unáris körszónak két különböző relatív prím szám is periódusa, míg az utóbbi egy nem unáris körszó különböző lehetséges periódusainak maximális számát adja meg a körszó hosszának függvényében. Eddigi rövid munkálkodásom során nem sikerült a 2.2. lemmával analóg állítást bebizonyítani, de a kutatás egyik szála abba az irányba mutat.

Egy másik kutatási irányt jelent a fákkal való ábrázolásból kiindulva a körszavakat elfogadó automaták vizsgálata. Ezen belül leginkább az automaták bonyolultságára vonatkozó kérdések érdekesek. Például: Adott $\boldsymbol{w}_\circ$ körszó gyenge periódusai és a $\boldsymbol{w}_\circ$ nyelvet elfogadó véges automata állapotainak száma között milyen összefüggés van? Ehhez kapcsolódóan a palindrom körszavak esetén kétfejű automatákkal is érdemes foglalkozni.

Jelen dolgozatban közölt eredmények és a körszavak periódusainak új megközelítése reményeink szerint későbbi eredményes kutatásokat alapozhatnak meg.

3 Automataelméleti vizsgálódások

A dolgozat további részében egyállapotú számlálóautomaták egy részosztályáról lesz szó, melyet egy rövid bevezetés után fogunk tárgyalni. Ezen a területen a hangsúlyt a különböző automaták által elfogadott nyelvek közötti hierarchia vizsgálatára helyeztük. Ez az itt vizsgált modell esetében példanyelvek (ellenpéldák) szemléltetésével tűnt a leghatékonyabbnak.

Kezdetben a szakirodalom kapcsolódó eredményeit, fogalmait tekintjük át miközben egységesített jelölést vezetünk be. A bevezetést a számlálóautomaták tárgyalásával kezdjük a 3.1. alfejezetben, majd egy általánosításukat, az egyállapotú számlálóautomatákat mutatjuk be. A több olvasófejjel rendelkező automaták ismertetésével zárjuk az előzmények áttekintését. A 3.2. fejezetben vizsgáljuk az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákat. Eredményeinket a lehetséges további kutatási irányok megemlítésével a 3.3. fejezetben foglaljuk össze. Feltételezzük a formális nyelvekkel kapcsolatos alapvető fogalmak, a Chomsky hierarchia és különböző nyelvosztályok definícióinak ismeretét. Ezek mind megtalálhatók a [19] műben.

3.1. Egyállapotú számlálóautomaták és több olvasófejjel rendelkező automaták

Számlálóautomaták

Az automataelmélet egyik legegyszerűbb modellje a véges automata (finite automaton), melynek véges sok belső állapota van és működése az automata aktuális belső állapotától, valamint az olvasott inputtól függ. Ha az automatát nem algebrai modellként kezeljük, hanem felismerő, elfogadó eszközként, akkor egyes állapotokat kitüntetünk. Ekkor az automatát iniciális véges elfogadóautomatának nevezzük. Az automata működése diszkrét időskálán történik, a számítás megszámlálható sok lépésre bontható. A számítás elején az automata kezdőállapotban van és az olvasófeje az input első betűjén áll. Az automata számítása pontosan akkor sikeres (azaz elfogadó), ha az input minden betűje feldolgozásra került és az automata egy elfogadó állapotban van. Egy automatának kezdőállapotaiból és elfogadó állapotaiból is lehet több, ugyanakkor, bármely több kezdőállapottal rendelkező automatához megadható azzal ekvivalens (ugyanazt a nyelvet elfogadó) egyetlen kezdőállapottal rendelkező iniciális véges elfogadóautomata. A továbbiakban jelölje $\mathcal{P}(X)$ az X halmaz összes részhalmazának halmazát, azaz az X hatványhalmazát. Az iniciális véges elfogadóautomata definíciója ekkor a következő.

3.1. Definíció. *Az $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ ötöst* ***iniciális véges elfogadóautomatának*** *nevezzük, ahol*

- *Q egy véges, nem üres halmaz, az* ***állapotok halmaza,***
- *Σ egy* ***ábécé***
- *$\delta : \Sigma \times Q \to \mathcal{P}(Q)$ az* ***átmenetfüggvény,***

- $q_0 \in Q$ *a* ***kezdőállapot*** *és*
- $F \subseteq Q$ *az* ***elfogadó állapotok halmaza****.*

A véges automaták részletesebb tárgyalása megtalálható például a [19] jegyzetben. Leginkább olyan esetekben használhatók, amikor a számítás során csak véges sok különböző esetet (állapotot) szükséges megkülönböztetni. Nem képes például arra, hogy az inputban bizonyos betűk előfordulásainak számát összehasonlítsa. Ilyen esetekben használhatók a számlálóautomaták (counter machine), azaz olyan véges sok állapottal rendelkező automaták, melyek fel vannak szerelve véges sok számlálóval. Egy-egy számlálóban pontosan egy nem negatív egész értéket lehet tárolni és az eggyel való növelés, illetve csökkentés műveletek értelmezettek rajta. Utóbbi csak akkor, ha a számláló értéke nem nulla. A számlálón értelmezett egy $\mathrm{sign} : \mathbb{N} \to \{0, 1\}$ függvény a következőképpen:

$$\mathrm{sign}(j) = \begin{cases} 0 & \text{ha } j = 0, \\ 1 & \text{egyébként.} \end{cases}$$

A sign függvényt kiterjesztjük érték $n-$esekre, azaz $\mathrm{sign}(s_1, \ldots, s_n) = (\mathrm{sign}(s_1), \ldots, \mathrm{sign}(s_n))$, hogy $n-$számlálós automatákat tudjunk kezelni.

A számlálóautomata a számítás egyes lépéseiben az olvasott input szimbólum, az aktuális állapot és a számlálók értékéből a sign függvény segítségével származtatott vektor alapján dönt a végrehajtandó utasításokról, melyek lehetnek:

1. belső állapot megváltoztatása,
2. olvasófej jobbra léptetése,
3. a számlálók értékeinek megengedett módon történő megváltoztatása,

illetve ezek tetszőleges kombinációi.

A számlálóautomata formális definíciója a következő.

3.2. Definíció. *Az $\mathcal{A} = (Q, \Sigma, m, \delta, q_0, F)$ hatost* ***iniciális számlálóautomatának*** *nevezzük, ahol*

- *Q egy véges halmaz, az automata* ***belső állapotainak halmaza***
- *Σ egy véges* ***ábécé****,*
- *m az automata számlálóinak darabszáma*
- *$\delta : \Sigma \times Q \times \{0,1\}^m \to \mathcal{P}(\{0,1\} \times Q \times \{0,+,-\}^m)$ az* ***átmenetfüggvény****, továbbá megköveteljük, hogy tetszőleges $a \in \Sigma$, $q \in Q$ valamint $s = (s_1, \ldots, s_m) \in \{0,1\}^m$ esetén, ha $\delta(a,q,s)$ értelmezett (azaz nem az üres halmazt adja eredményül), akkor nem létezik olyan $i \in \{1, \ldots, m\}$, hogy $s_i = 0$ és $t_i = -$ valamely $(d, q', t_1, \ldots, t_m) \in \delta(a,q,s)$−re, azaz nulla értékű számláló csökkentése nem megengedett,*
- *$q_0 \in Q$ a* ***kezdőállapot****,*
- *$F \subseteq Q$ az* ***elfogadó állapotok halmaza****.*

Az $\mathcal{A} = (Q, \Sigma, m, \delta, q_0, F)$ iniciális számlálóautomata ***konfigurációit*** $C = (\boldsymbol{w}, q, s)$ alakú párokkal írjuk le, ahol $\boldsymbol{w} \in \Sigma^*$, $q \in Q$, valamint $s = (s_1, \ldots, s_m) \in \mathbb{N}_0^m$.

Azt mondjuk, hogy a $C_1 = (\boldsymbol{w}, q, s)$ konfigurációból a $C_2 = (\boldsymbol{w'}, q', s')$ (ahol $s = (s_1, \ldots, s_m), s' = (s'_1, \ldots, s'_m) \in \mathbb{N}_0^m$) konfiguráció közvetlenül elérhető (jelöléssel $C_1 \vdash_{\mathcal{A}} C_2$), ha teljesül a következő két feltétel:

1. az alábbiak közül az egyik fennáll:
   - $\boldsymbol{w} = a\boldsymbol{w'}$, $a \in \Sigma$ és $(1, q', t_1, \ldots, t_m) \in \delta(a, q, \mathrm{sign}(s))$, vagy

- $\boldsymbol{w} = \boldsymbol{w'} = a\boldsymbol{u} \in \Sigma^*$, ahol $a \in \Sigma \cup \{\lambda\}$ (mind $\boldsymbol{u}$, mind a lehet az üresszó is) és $(0, q', t_1, \ldots, t_m) \in \delta(a, q, \mathrm{sign}(s))$,

2. valamint, minden $i = 1, \ldots, m$–re teljesül, hogy

$$s_i' = \begin{cases} s_i + 1\,, & \text{ha } \ t_i = +, \\ s_i - 1\,, & \text{ha } \ s_i > 0 \text{ és } t_i = -, \\ s_i\,, & \text{egyébként.} \end{cases}$$

Más szóval a $(d, q', t) \in \delta(a, q, \mathrm{sign}(s))$ párok d tagja jelöli, hogy az olvasófej elmozdul-e jobbra az adott átmenetben ($d = 1$ esetén), vagy helyben marad (ha $d = 0$); q' lesz az új belső állapot ($q' = q$ is megengedett); a t műveletvektor pedig megadja a számlálókon elvégzendő műveleteket (+: növelés, −: csökkentés, 0: változatlanul hagyás).

Az így definiált számlálóautomata hasonló a *Marvin L. Minsky* által regiszter gépnek (program machine, register machine) nevezett modellhez [35]. A regiszterek működése a számlálókéval azonos, a programot pedig az állapotátmenet függvény írja le. *Marvin L. Minsky* megmutatta, hogy egy két számlálóval rendelkező regiszter gép (így számlálóautomata) kifejezőereje megegyezik a Turing gépekével.

3.1. Lemma (Számlálóautomaták kifejezőerejéről [35])**.** *Bármely T Turing géphez létezik olyan M_T regiszter gép mindössze két regiszterrel, amely T működését szimulálja.* □

A bizonyítás főbb lépései a következők:

1. Bármely T Turing gép esetén megadható olyan automata, amelynek két verme van és T működését szimulálja. A vermek a $\{0, 1\}$ bináris ábécé betűit tartalmazhatják.

2. Egy bináris számot, azaz bináris szó betűit tartalmazó verem szimulálható két számlálóval. Feltesszük, hogy a verem tetején a leg-

kisebb helyiértékű bit helyezkedik el. Ekkor, ha a verem tetejére 0 szimbólumot szúrunk, az a kettővel való szorzásnak felel meg, a verem tetejére szúrt 1 pedig kettővel való szorzást és eggyel való növelést jelöli. A legfelső szimbólum eltávolítása pedig pontosan a fenti két művelet inverze. Ezen műveletek két számlálóval végrehajthatók, ugyanis az egyik számláló értéke a veremben lévő szám, a másik számlálót pedig ideiglenes tárhelynek használhatjuk.

3. Négy számláló működése szimulálható két számlálóval. Az egyik számláló ebben az esetben is ideiglenes tárhelyként szolgál, míg a másik számláló a $2^i 3^j 5^k 7^\ell$ értéket tartalmazza, ahol $i, j, k, \ell \in \mathbb{N}_0$ a szimulálandó négy számláló értéke. A számláló értékének $2-$vel, $3-$mal, $5-$tel, illetve $7-$tel való szorzása rendre az i, j, k, ℓ számlálók eggyel való növelését reprezentálja, míg az osztás az eggyel való csökkentést.

Tehát a két számlálóval rendelkező iniciális számlálóautomaták kifejezőereje megegyezik a Turing gépekével. A továbbiakban olyan számlálóautomatákkal foglalkozunk, melyeknek pontosan egy állapota van.

Egyállapotú számlálóautomaták

Automaták ezen részosztályát *Ömer Eğecioğlu* és *Oscar H. Ibarra* vizsgálta [20, 21]. Közös műveikre épül a dolgozat további részében tárgyalt modell.

Az előbbiekre építve az egyállapotú számlálóautomata (stateless multicounter machine) formális definíciója a következő:

3.3. Definíció. *Az $\mathcal{A} = (\Sigma, \cent, \$, m, \delta)$ ötöst **egyállapotú számlálóautomatának** nevezzük, ahol*

- *Σ egy véges **ábécé**,*
- *¢, $\$ \notin \Sigma$ rendre a **kezdő** és **záró jelek**,*
- *$m \in \mathbb{N}_0$ a **számlálók darabszáma** és*
- *$\delta : (\Sigma \cup \{¢, \$\}) \times \{0,1\}^m \to \mathcal{P}(\{0,1\} \times \{0,+,-\}^m)$ az **átmenet-függvény**, továbbá megköveteljük, hogy bármely $a \in (\Sigma \cup \{¢, \$\})$, valamint $s = (s_1, \ldots, s_m) \in \{0,1\}^m$ esetén, ha $\delta(a, s)$ értelmezett (azaz nem az üres halmazt adja eredményül), akkor nem létezik olyan $i \in \{1, \ldots, m\}$, hogy $s_i = 0$ és $t_i = -$, ahol $(t_1, \ldots, t_m) \in \delta(a, s)$, azaz nulla értékű számláló csökkentése nem megengedett.*

A 3.3. definícióban megadott automatában az egyállapotú jelző a belső állapot állandó voltát jelzi. Ezért azt az átmeneteknél nem is kell figyelembe vennünk. A szakirodalomban szokás az ilyen automatákat állapotnélküli automatáknak (stateless automata) is nevezni. Az iniciális jelzőt is elhagytuk, mivel ebben a modellben nem beszélhetünk szigorú értelemben vett kezdőállapotról.

A továbbiakban, ha Σ egy ábécé és $\boldsymbol{w} \in \Sigma^*$, akkor $\boldsymbol{w}\Sigma^*$ jelöli azon szavak halmazát, melyek kezdőszelete $\boldsymbol{w}$. A $\Sigma^*\boldsymbol{w}$ halmaz pedig azokból a szavakból áll, melyeknek zárószelete $\boldsymbol{w}$. Ha pedig A és B szóhalmazok, akkor

$$AB = \{xy \mid x \in A, y \in B\}.$$

Az $\mathcal{A} = (\Sigma, ¢, \$, m, \delta)$ egyállapotú számlálóautomata ***konfigurációit*** $C = (\boldsymbol{w}, s)$ alakú párokkal jelöljük, ahol $\boldsymbol{w} \in \{¢, \lambda\}\Sigma^*\$$, valamint $s = (s_1, \ldots, s_m) \in \mathbb{N}_0^m$.

Azt mondjuk, hogy a $C_1 = (\boldsymbol{w}, s)$ konfigurációból a $C_2 = (\boldsymbol{w'}, s')$

(ahol $s = (s_1, \ldots, s_m), s' = (s'_1, \ldots, s'_m) \in \mathbb{N}_0^m$) konfiguráció közvetlenül elérhető (jelöléssel $C_1 \vdash_{\mathcal{A}} C_2$), ha teljesül a következő két feltétel:

1. az alábbiak közül az egyik fennáll:
   - $\boldsymbol{w} = a\boldsymbol{w'}$ és $(1, t_1, \ldots, t_m) \in \delta(a, \mathrm{sign}(s))$, vagy
   - $\boldsymbol{w} = \boldsymbol{w'} = a\boldsymbol{u} \in \Sigma^*\$$ (ahol $\boldsymbol{u}$ lehet üresszó is, ekkor $a = \$$) és $(0, t_1, \ldots, t_m) \in \delta(a, \mathrm{sign}(s))$,

   ahol $a \in \Sigma \cup \{¢, \$\}$,
2. valamint, minden $i = 1, \ldots, m$−re teljesül, hogy

$$s'_i = \begin{cases} s_i + 1\,, & \text{ha } \; t_i = +, \\ s_i - 1\,, & \text{ha } \; s_i > 0 \text{ és } t_i = -, \\ s_i\,, & \text{egyébként.} \end{cases}$$

Más szóval a $(d, t) \in \delta(a, \mathrm{sign}(s))$ párok d tagja jelöli, hogy az olvasófej elmozdul-e jobbra az adott átmenetben ($d = 1$ esetén), vagy helyben marad (ha $d = 0$); a t műveletvektor pedig megadja a számlálókon elvégzendő műveleteket (+: növelés, −: csökkentés, 0: változatlanul hagyás).

Az átmenetfüggvény egy alkalmazását sokszor az automata egy ***lépésének*** nevezzük.

Ha a szövegkörnyezetből kiderül, hogy melyik automatáról van szó, akkor $\vdash_{\mathcal{A}}$ helyett egyszerűen csak a $\vdash$ jelölést használjuk. Továbbá legyen $\vdash^*$ a $\vdash$ reflexív és tranzitív lezártja.

Az $\mathcal{A} = (\Sigma, ¢, \$, m, \delta)$ automata kezdőkonfigurációja a $\boldsymbol{w} \in \Sigma^*$ inputon $(¢\boldsymbol{w}\$, 0^m)$, elfogadó konfigurációja pedig $(\$, 0^m)$. Az $\mathcal{A}$ automata elfogadja a $\boldsymbol{w} \in \Sigma^*$ szót, ha a kezdőkonfigurációból kiindulva véges sok lépés után az elfogadó konfigurációhoz jutunk, azaz, ha $(¢\boldsymbol{w}\$, 0^m) \vdash^* (\$, 0^m)$. Az $\mathcal{A}$

automata által elfogadott nyelv pedig

$$\mathcal{L}(\mathcal{A}) = \{\boldsymbol{w} \in \Sigma^* \mid (¢\boldsymbol{w}\$, 0^m) \vdash_{\mathcal{A}}^* (\$, 0^m)\}.$$

Azokat a nem elfogadó konfigurációkat, melyekben az átmenetfüggvény nem alkalmazható, azaz melyekből nem lehet további konfigurációkba lépni, elutasító konfigurációknak nevezzük.

Legyen r egy számláló és vizsgáljuk a $C_1 \vdash \ldots \vdash C_\ell$ konfigurációk sorozatát, amelyet ***számításnak*** is nevezünk. Minden számítás során figyelemmel lehet követni, hogy az egyes számlálók hányszor kerülnek nemcsökkenő (növelések és változatlanul hagyások sorozata) állapotból csökkenőbe (legalább egy csökkentés művelet után). Ha ez adott r számlálóval tetszőleges számítás során legfeljebb $k-$szor történik, akkor azt mondjuk, hogy r $k-$szor fordul, vagy r egy ***$k-$forduló számláló*** ($k-$reversal counter). Ha nem létezik ilyen k egész, akkor r fordulásainak száma korlátlan és ***korlátlanul fordulónak*** nevezzük. Megjegyezzük, hogy az elfogadó és nem elfogadó számításokat is figyelembe vesszük r fordulásainak meghatározásánál. Sokszor egyes számlálók csak arra szolgálnak, hogy számon tartsák más számlálók fordulásait. Így oldható meg az, hogy, ha például az r számláló fordulásainak száma elérte a k határértéket, akkor az automata elutasító konfigurációba kerüljön.

A számlálókhoz hasonlóan, ha létezik egy olyan k egész, hogy adott $\mathcal{A}$ automatának minden számlálója (legfeljebb) $k-$forduló, akkor az $\mathcal{A}$ automatát is ***$k-$forduló automatának*** nevezzük. Ellenkező esetben $\mathcal{A}$ ***korlátlanul forduló automata***.

3.1. *Megjegyzés.* A fordulásokba néha a nemnövekvő állapotból növekvőbe való váltást is beleszámolják, viszont ekkor minden elfogadó számításban nulla vagy páratlan fordulás történik, mivel csökkenő fázisba páratlan fordulás után kerül a számláló és csak ekkor nullázódhat le. Az elfogadott nyelvek hierarchiáját tekintve viszont ez és a fentebb definiált megközelítés lényegét tekintve azonos.

Azt mondjuk, hogy az $\mathcal{A}$ automata ***valós-idejű*** (realtime), ha

$$\delta(a, \mathrm{sign}(s)) \subseteq (\{1\} \times \{0, +, -\}^m)$$

minden $a \in \Sigma \cup \{¢, \$\}$ és $s \in \mathbb{N}_0^m$ esetén. Azaz a valós-idejű automaták olvasófeje minden lépésben elmozdul, kivéve ha az átmenetfüggvény az üres halmazt adja eredményül. Ezen korlátozás nélkül az automata ***nem valós-idejű***. Utóbbi jelzőt megengedő esetben fogjuk használni, azaz a nem valós-idejű automaták halmaza tartalmazza a valós-idejű automatákat is.

Megjegyezzük, hogy valós-idejű automaták esetén valamennyi számítás elfogadó vagy elutasító konfigurációban ér véget, hiszen tetszőleges $\boldsymbol{w}$ input esetén az automata számítása véges sok konfigurációból álló sorozattal leírható, mivel az input feldolgozásra váró része minden lépésben legalább egy karakterrel csökken. Azaz a valós-idejű automaták véges sok lépés után megállnak tetszőleges inputon. Ez nem feltétlenül teljesül nem valós-idejű automaták esetén, ugyanis a számlálók értéke változtatható az olvasófejek elmozdítása nélkül. Jelenlegi ismereteink szerint nyitott kérdés, hogy milyen feltételek mellett dönthető el, hogy adott nem valós-idejű automata adott inputon véges sok lépés után megáll-e vagy sem.

Ha $|\delta(a, \mathrm{sign}(s))| \leq 1$ valamennyi $a \in \Sigma \cup \{¢, \$\}$ és $s \in \mathbb{N}_0^m$ esetén, akkor az automata ***determinisztikus***. Azaz egy számlálóautomata pontosan akkor determinisztikus, ha minden konfiguráció esetén, amennyiben létezik lehetséges lépés, akkor az egyértelműen meghatározott.

3.2. *Megjegyzés.* Megjegyezzük, hogy szokás a nemdeterminisztikus jelzőt is használni automaták esetében, viszont általános esetben a definíció szerint ilyen automatákról beszélünk, ezért ezt a jelzőt legtöbbször elhagyjuk. Csak akkor használjuk, amikor a két típust szembeállítjuk egymással.

A következőkben megadunk néhány fontosabb lemmát, melyek *Ömer Eğecioğlu* és *Oscar H. Ibarra* nevéhez fűződnek. Valamennyi megtalálható a [20, 21] művekben. Jelölje $|\boldsymbol{w}|_a$ a $\boldsymbol{w} \in \Sigma^*$ szóban lévő $a \in \Sigma$ betű előfordulásainak számát. Megjegyezzük, hogy a könnyebb olvashatóság kedvéért az m számlálóval rendelkező automatákat $m-$***számlálós*** automatáknak hívjuk.

Egyállapotú számlálóautomaták tulajdonságai

3.2. Lemma. *Legyen $L = \{\boldsymbol{w} \in \{a,b\}^* \mid |\boldsymbol{w}|_a = |\boldsymbol{w}|_b\}$. Ekkor létezik olyan nem valós-idejű $1-$forduló $2-$számlálós egyállapotú számlálóautomata, amely elfogadja $L-$t, de bármely $k, m \in \mathbb{N}$ értéket is véve, nem létezik olyan valós-idejű $k-$forduló $m-$számlálós automata, amely elfogadná.* □

3.3. Lemma. *Ha egy L nyelvet egy nem valós-idejű $k-$forduló $m-$számlálós egyállapotú számlálóautomata elfogad, akkor azt elfogadja egy nem valós-idejű $1-$forduló $(2k-1)m-$számlálós egyállapotú számlálóautomata is.* □

3.4. Lemma. *Ha $\mathcal{A} = (\{a\}, ¢, \$, m, \delta)$, akkor $\mathcal{L}(\mathcal{A}) = \{a^{n+\ell \cdot i} \mid i \in \mathbb{N}_0\}$ valamely $n, \ell \in \mathbb{N}_0$ egészekre.* □

3.5. Lemma. *Ha $\mathcal{A} = (\{a\}, ¢, \$, m, \delta)$, akkor $\mathcal{L}(\mathcal{A})$ vagy végtelen, vagy egyelemű nyelv.* □

3.6. Lemma. *Ha az $\mathcal{A} = (\{a\}, ¢, \$, m, \delta)$ egyállapotú számlálóautomata $1-$forduló, valamint az általa elfogadott nyelv egyelemű, azaz $\mathcal{L}(\mathcal{A}) = \{a^n\}$ valamely $n \in \mathbb{N}_0$ esetén, akkor*

$$n \leq (m-1)2^m + m.$$

□

Azaz egy 1−forduló m−számlálós egyállapotú számlálóautomata minden számítás során legfeljebb $(m-1)2^m + m$ értékig tud "elszámolni". A 3.6. lemma k−forduló egyállapotú számlálóautomatákra általánosított változata a következő:

3.7. Lemma. *Ha az $\mathcal{A} = (\{a\}, ¢, \$, m, \delta)$ egyállapotú számlálóautomata k−forduló, valamint az általa elfogadott nyelv egyelemű, azaz $\mathcal{L}(\mathcal{A}) = \{a^n\}$, akkor*

$$n \leq ((2k-1)m-1)2^{(2k-1)m} + (2k-1)m.$$

□

3.8. Lemma. *Legyen $m \in \mathbb{N}$. Ekkor bármely $k < \frac{2^{m-1}}{m}$ esetén létezik olyan L nyelv, melyet elfogad egy $(k+1)$−forduló m−számlálós egyállapotú számlálóautomata, ellenben nem létezik olyan k−forduló m−számlálós egyállapotú számlálóautomata, amely azt elfogadná.* □

A k-ra vonatkozó korlát az előző tételben arra utal, hogy az automatának bizonyos mennyiségű számlálóra van szüksége ahhoz, hogy a fordulások számát követni tudja, mivel az automatának implicit tulajdonsága az, hogy számlálói hányszor fordulnak. Tehát $k > \frac{2^{m-1}}{m}$ esetén nem lehet garantálni, hogy nincs olyan számítás, mely során az automata nem fordul k−nál többet. Ezért a dolgozat további részében, ha általánosságban beszélünk k−forduló m−számlálós egyállapotú számlálóautomatákról, akkor feltesszük, hogy $k \leq \frac{2^{m-1}}{m}$.

Ahhoz, hogy rátérhessünk a vizsgált modell ismertetésére, előbb a több olvasófejjel rendelkező automatákkal kell foglalkoznunk.

Több olvasófejjel rendelkező automaták

Az előző alfejezetben láthattuk, hogyan lehet a véges elfogadóautomatákból kiindulva egy egészen különböző tulajdonságokkal bíró automataosztályhoz jutni. A számlálóautomaták mellett igen elterjedt a több olvasófejjel rendelkező automaták vizsgálata is [24, 26]. Minket a modellünk szempontjából a két olvasófejes elfogadóautomaták érdekelnek, melyeket a következő általános definíció speciális eseteként tudunk értelmezni.

3.4. Definíció. *Legyen* $\ell \in \mathbb{N}$. *Az* $\mathcal{A} = (Q, \Sigma, \text{¢}, \$, \ell, \delta, q_0, F)$ *nyolcast* $\ell-$***olvasófejes automatának*** *nevezzük, ahol*

- Q ***állapotok*** *egy nem üres, véges halmaza,*
- Σ *egy nem üres, véges* ***input ábécé****,*
- $\text{¢}, \$ \notin \Sigma$ *a* ***kezdő-*** *és* ***záró jelek****,*
- ℓ *az* ***olvasófejek száma****,*
- $\delta : Q \times (\Sigma \cup \{\text{¢}, \$\})^\ell \to \mathcal{P}(Q \times \{0, 1\}^\ell)$ *az* ***átmenetfüggvény****,*
- $q_0 \in Q$ *a* ***kezdőállapot****,*
- $F \subseteq Q$ *az* ***elfogadó állapotok halmaza****.*

Egy ilyen automata működése vázlatosan a következőképpen zajlik a $\boldsymbol{w} \in \Sigma^*$ input feldolgozása esetén, melyet a szalagon a $\text{¢}\boldsymbol{w}\$$ karaktersorozat reprezentál:

- A számítás elején valamennyi olvasófej a ¢ szimbólumon áll és az automata a q_0 kezdőállapotban van.
- A δ átmenetfüggvény segítségével, az aktuális állapot és az olvasófejek által látott szimbólumok alapján eldönti az automata, hogy

melyik állapotba menjen át és mely olvasófejeket léptesse. Egy olvasófej vagy helyben marad (0), vagy jobbra lép (1), amennyiben lehetséges (ha a szalagon van még nem olvasott szimbólum a $-on kívül).

- Az automata állapotot vált és az olvasófejeket ezzel párhuzamosan lépteti az előző pontban leírtaknak megfelelően.
- Ha az automata valamennyi olvasófeje a $ szimbólumon áll és az aktuális állapot elfogadó állapot, akkor az input elfogadásra kerül.
- Ha az automata nem elfogadó állapotban van és az átmenetfüggvény az üres halmazt adja, azaz nem lehetséges a továbblépés, akkor az inputot elutasítja.

A következőkben tárgyaljuk a Watson-Crick automatákat [16], melyek a 2$-$olvasófejes automaták egy változatának felelnek meg.

Watson-Crick automaták

Ahogy a dolgozat bevezetésében említettük, a Watson-Crick automaták a bioinspirált számítási modellek közé tartoznak és nevük *James D. Watson* és *Francis H. C. Crick* kutatók DNS láncok szerkezetével kapcsolatos munkásságára utal.

A Watson-Crick automaták inputként DNS láncokat, illetve azokat reprezentáló szópárokat várnak. Ennek megfelelően a matematikai modelljükben definiálva van egy úgynevezett ***Watson-Crick komplementaritási reláció***, melyet $\varrho-$val jelölünk és a természetes megközelítésben a

$$\varrho_{DNS} = \{(A,T),(T,A),(G,C),(C,G)\}$$

szimmetrikus relációt értjük alatta, amely pontosan a biológiai motivációból származó DNS láncokat alkotó molekulák közötti relációt írja le. A matematikai modellben megengedünk tetszőleges szimmetrikus relációt. A Watson-Crick komplementaritási reláció segítségével definiáljuk a megengedett szópárokat. Legyen Σ egy tetszőleges ábécé és $\varrho : \Sigma \to \Sigma$ egy rajta értelmezett Watson-Crick komplementaritási reláció. Ekkor a $(\boldsymbol{u}, \boldsymbol{v}) \in \Sigma^* \times \Sigma^*$ pontosan akkor ***megengedett szópár***, ha $|\boldsymbol{u}| = |\boldsymbol{v}|$ és valamennyi $i \in \{1, \ldots, |\boldsymbol{u}|\}$ esetén $(u_i, v_i) \in \varrho$. Szokás az ilyen szópárokat $\left[\begin{smallmatrix}\boldsymbol{u}\\ \boldsymbol{v}\end{smallmatrix}\right]$ alakban is írni, ezzel is szemléltetve, hogy a Watson-Crick automaták DNS láncokat dolgoznak fel.

A Watson-Crick automaták formális definíciója tehát a következő.

3.5. Definíció. *Az $\mathcal{A} = (Q, \Sigma, \varrho, \delta, q_0, F)$ hatost* ***Watson-Crick automatának*** *nevezzük, ahol*

- *Q* ***állapotok*** *egy nem üres, véges halmaza,*
- *Σ egy nem üres, véges* ***input ábécé,***
- *$\varrho \subseteq \Sigma \times \Sigma$ egy* ***Watson-Crick komplementaritási reláció***
- *$\delta : Q \times \Sigma^* \times \Sigma^* \to \mathcal{P}(Q)$ az* ***átmenetfüggvény****, amely csak véges sok $Q \times \Sigma^* \times \Sigma^*-$beli hármas esetén nem az üres halmaz,*
- *$q_0 \in Q$ a* ***kezdőállapot****,*
- *$F \subseteq Q$ az* ***elfogadó állapotok halmaza****.*

Vegyük észre, hogy megengedjük, hogy az olvasófejek egy-egy betű helyett véges hosszú szavakat olvassanak, ezzel is kiemelve, hogy egyes enzimek a DNS láncok egy-egy több molekulából álló részét érzékelik. Tehát a számítás során az automata következő lépése az aktuális belső állapotból, valamint az olvasófejek által olvasott szavakból határozható meg az átmenetfüggvény segítségével.

Megjegyezzük, hogy a megengedett szópárokat szokás csak az egyik taggal reprezentálni, mivel bármely Watson-Crick automatához létezik vele ekvivalens olyan Watson-Crick automata, amelyben a ϱ reláció az identitás [30]. Ezért a kifejezőerő csorbulása nélkül az egyszerűség kedvéért feltesszük, hogy $\varrho \subseteq \Sigma \times \Sigma$ csak (a, a) alakú párokat tartalmaz, ahol $a \in \Sigma$.

A 3.4. definícióhoz képest – azon túl, hogy az olvasófejek szavakat olvashatnak – elhagytuk az átmenetfüggvény képéből az olvasófejek léptetésére vonatkozó információt, mivel úgy vesszük, hogy az adott olvasófej át is lépi az inputnak azt a faktorát, amit elolvas. Ez a faktor természetesen lehet az üresszó is az egyik, vagy akár mindkét fej esetén.

Az átmenetfüggvény szerint a két olvasófej az input valamely véges hosszú részét látja. Az ilyen $(\boldsymbol{u}, \boldsymbol{v}) \in \Sigma^* \times \Sigma^*$ párok és az állapotok különböző tulajdonságai alapján a Watson-Crick automatákat csoportosíthatjuk. Az $\mathcal{A} = (Q, \Sigma, \varrho, \delta, q_0, F)$ Watson-Crick automata

- ***általános***, ha nincs rá semmilyen megszorítás,
- ***egyállapotú***, ha $|Q| = |F| = 1$, mely esetben ez az egy állapot a kezdőállapot is,
- ***teljes-elfogadó***, ha $Q = F$,
- ***egyszerű***, ha minden $\delta(q, \boldsymbol{u}, \boldsymbol{v})$ $(\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*, q \in Q)$ átmenet esetén vagy $\boldsymbol{u} = \lambda$, vagy $\boldsymbol{v} = \lambda$,
- 1-***korlátozott***, ha $|\boldsymbol{uv}| = 1$ minden $\delta(q, (\boldsymbol{u}, \boldsymbol{v}))$ átmenet esetén.

Adott Watson-Crick automata működését konfigurációk sorozatával írhatjuk le. Mivel az automatának két olvasófeje van, ezért mindkét fej esetében számon kell tartani, hogy az input mely részét kell még feldolgoznia. Tehát egy általános konfigurációt $(\boldsymbol{x}, \boldsymbol{y}, q) \in \Sigma^* \times \Sigma^* \times Q$ alakban adhatunk meg, ami azt írja le, hogy a két olvasófej által feldolgozásra váró részszavak az $\boldsymbol{x}$, és $\boldsymbol{y}$, valamint az automata a q állapotban van. Következésképpen az automata kezdőkonfigurációja a $\boldsymbol{w} \in \Sigma^*$ input esetén $(\boldsymbol{w}, \boldsymbol{w}, q_0)$, az automata elfogadó befejező konfigurációi pedig (λ, λ, q_v) alakúak, ahol $q_v \in F$ elfogadó állapot.

Legyen $\mathcal{A}$ egy tetszőleges Watson-Crick automata és vegyük annak kettő lehetséges konfigurációját. Ezeket jelölje $C_1 = (\boldsymbol{x_1}, \boldsymbol{y_1}, q_1)$ és $C_2 = (\boldsymbol{x_2}, \boldsymbol{y_2}, q_2)$. Azt mondjuk, hogy a C_1 konfigurációból közvetlenül elérhető a C_2 konfiguráció (jelöléssel $C_1 \vdash_{\mathcal{A}} C_2$), ha léteznek olyan $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$ szavak, hogy $\boldsymbol{x_1} = \boldsymbol{u}\boldsymbol{x_2}$, $\boldsymbol{y_1} = \boldsymbol{v}\boldsymbol{y_2}$ és $q_2 \in \delta(q_1, \boldsymbol{u}, \boldsymbol{v})$. Az így definiált $\vdash_{\mathcal{A}}$ reláció reflexív, tranzitív lezártját $\vdash_{\mathcal{A}}^*$ $-$gal jelöljük. Ezen reláció segítségével definiálhatjuk az $\mathcal{A}$ automata által elfogadott nyelvet a következőképpen:

$$\mathcal{L}(\mathcal{A}) = \{\boldsymbol{w} \in \Sigma^* \mid (\boldsymbol{w}, \boldsymbol{w}, q_0) \vdash_{\mathcal{A}}^* (\lambda, \lambda, q_v),\ \text{ahol } q_v \in F\}.$$

Az általános, egyállapotú, teljes-elfogadó, egyszerű, és 1-korlátozott Watson-Crick automaták által elfogadott nyelvek osztályait rendre általános, egyállapotú, teljes-elfogadó, egyszerű, valamint 1-korlátozott nyelvosztályoknak nevezzük.

A Watson-Crick automaták egy érdekes osztályát képezik az $5' \to 3'$ ***Watson-Crick automaták*** [36, 37], melyek esetében a két olvasófej a számítás elején az input két végén helyezkedik el, szemben a hagyományos Watson-Crick automatákkal, melyeknél mindkét fej az input elejéről indul. Az általam vizsgált automatamodell alapját olyan $5' \to 3'$ Watson-Crick automaták képezik, melyeknél a számítás addig tart, amíg a két olvasófej nem találkozik (ekkor már a teljes input feldolgozásra került). Ezeket az $5' \to 3'$ Watson-Crick automatákat szokás érzékelő (sensing) Watson-Crick automatáknak nevezni.

Ezen automatákkal kapcsolatban az alábbi lemmákat emelnénk ki. A modell mélyreható tárgyalása megtalálható a [38] műben.

3.9. Lemma. *Az érzékelő* $5' \to 3'$ *Watson-Crick automaták által elfogadott nyelvek osztálya megegyezik a lineáris nyelvek osztályával.* □

Az $\mathcal{A} = (Q, \Sigma, \varrho, \delta, q_0, F)$ $5' \to 3'$ Watson-Crick automata determinisztikus, ha tetszőleges $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$ és $q \in Q$ esetén $|\delta(q, \boldsymbol{u}, \boldsymbol{v})| \leq 1$, azaz, ha a δ átmenetfüggvény egy helyen értelmezett, akkor ott egyelemű halmazt ad eredményül.

3.10. Lemma. *A determinisztikus érzékelő* $5' \to 3'$ *Watson-Crick automaták által elfogadott nyelvek osztálya valódi részhalmaza a lineáris nyelvek osztályának.* □

A továbbiakban a dolgozat másik fő vizsgálati tárgyát képező egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták tárgyalására térünk át. Hivatkozni fogunk az úgynevezett reguláris, környezetfüggetlen, illetve környezetfüggő nyelvek osztályára, melyek számos, a témával foglalkozó szakirodalomban ismertetésre kerülnek (például [19]), így ezek definiálására nem térünk ki.

3.2. Egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták

A fenti bevezetés és az alapfogalmak lefektetése után ebben a fejezetben elkezdhetjük az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták vizsgálatát.

3.6. Definíció. ***Egyállapotú*** $5' \to 3'$ ***Watson-Crick számlálóautomatának*** *nevezzük az* $\mathcal{A} = (\Sigma, ¢, \$, m, \delta)$ *ötöst, ahol*

- *Σ egy nem üres, véges **ábécé**,*
- *$¢, \$ \notin \Sigma$ rendre **kezdő** és **záró jelek**,*
- *$m \in \mathbb{N}_0$ az automata számlálóinak darabszáma,*
- *$\delta = \delta_1 \cup \delta_2$ az **átmenetfüggvény**, ahol*

$$\delta_1 : (\Sigma \cup \{¢, \$\})^2 \times \{0,1\}^m \to \mathcal{P}(\{0,1\}^2 \times \{0,+,-\}^m)$$

$$\delta_2 : \{\lambda\}^2 \times \{0,1\}^m \to \mathcal{P}(\{0\}^2 \times \{0,+,-\}^m),$$

továbbá megköveteljük, hogy tetszőleges $a, b \in (\Sigma \cup \{¢, \$\})$, valamint $s = (s_1, \ldots, s_m) \in \{0,1\}^m$ esetén, ha $\delta(a,b,s)$ értelmezett (nem az üres halmaz), akkor nem létezik olyan $i \in \{1, \ldots, m\}$, hogy $s_i = 0$ és $t_i = -$, ahol $(t_1, \ldots, t_m) \in \delta(a,b,s)$, azaz nulla értékű számláló csökkentése nem megengedett.

Ez a modell az egyállapotú számlálóautomatáktól csak az olvasófejek számában különbözik. A számítás elején a két olvasófej az input két végéről indul, ahogy az $5' \to 3'$ Watson-Crick automaták esetében is. Ezért a két olvasófejet hívhatjuk ***jobb*** és ***bal*** olvasófejnek is. Általános esetben „érzékelik" egymást, azaz nem lehetséges, hogy az olvasás során keresztezzék egymás útját. Mindkét olvasófej pontosan egy-egy szimbólumot olvashat, kivéve, ha a teljes input feldolgozásra került, mikor is mindkét olvasófej λ–t olvas.

A dolgozat további részében kizárólag egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákról lesz szó, ezért az egyszerűbb olvashatóság kedvéért legtöbbször a rövidebb *automata* megnevezést fogjuk használni, ha nem vezet félreértésre.

Megjegyezzük, hogy az átmenetfüggvény δ_2 része megengedi, hogy a számlálók értékét megváltoztassuk miután a teljes input feldolgozásra került. Ebben az esetben az olvasófejek nem mozdulhatnak, mivel már nincs feldolgozandó szimbólum. Azt is megköveteljük, hogy az átmenetfüggvény δ_2 része csak akkor kerülhessen alkalmazásra, ha az input „elfogyott". Később ezt formálisan is leírjuk. Az átmenetfüggvény ilyen felbontására azért van szükség, hogy a hagyományos számlálóautomatáknál ismertetett valós-idejű és nem valós-idejű működési módok itt is megvalósíthatók legyenek.

A $C = (\boldsymbol{w}, s) \in \{¢, \lambda\}\Sigma^*\{\$, \lambda\} \times \mathbb{N}_0^m$ párt az $\mathcal{A}$ automata egy konfigurációjának nevezzük. Vegyük észre, hogy a jobb olvasófej is mozoghat, ezért $\boldsymbol{w}$ nem feltétlenül végződik \$ szimbólumra.

Az egy olvasófejjel rendelkező automatákhoz hasonlóan a

$$C_2 = (\boldsymbol{w'}, s') = (\boldsymbol{w'}, s'_1, \ldots, s'_m)$$

konfiguráció közvetlenül elérhető a

$$C_1 = (\boldsymbol{w}, s) = (\boldsymbol{w}, s_1, \ldots, s_m)$$

konfigurációból (jelöléssel: $C_1 \vdash_{\mathcal{A}} C_2$), ha az alábbiak közül az egyik teljesül:

- $\boldsymbol{w} = a\boldsymbol{w'}b$, $|\boldsymbol{w}| \geq 2$ és $(1, 1, t_1, \ldots, t_m) \in \delta(a, b, \mathrm{sign}(s))$,
- $\boldsymbol{w} = a\boldsymbol{w'} = \boldsymbol{u}b$ valamely $\boldsymbol{u} \in \Sigma^*$ esetén, $|\boldsymbol{w}| \geq 1$ és $(1, 0, t_1, \ldots, t_m) \in \delta(a, b, \mathrm{sign}(s))$,
- $\boldsymbol{w} = \boldsymbol{w'}b = a\boldsymbol{u}$ valamely $\boldsymbol{u} \in \Sigma^*$ esetén, $|\boldsymbol{w}| \geq 1$ és $(0, 1, t_1, \ldots, t_m) \in \delta(a, b, \mathrm{sign}(s))$,
- $\boldsymbol{w} = a\boldsymbol{u} = \boldsymbol{v}b = \boldsymbol{w'}$ valamely $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$ szavak esetén, $|\boldsymbol{w}| \geq 1$ és $(0, 0, t_1, \ldots, t_m) \in \delta(a, b, \mathrm{sign}(s))$,
- $\boldsymbol{w} = \lambda = \boldsymbol{w'}$ és $(0, 0, t_1, \ldots, t_m) \in \delta(\lambda, \lambda, \mathrm{sign}(s))$,

továbbá

$$s_i' = \begin{cases} s_i + 1\,, & \text{ha } \; t_i = +, \\ s_i - 1\,, & \text{ha } \; s_i > 0 \text{ és } t_i = -, \\ s_i\,, & \text{egyébként (azaz, ha } \; t_i = 0), \end{cases}$$

minden $i = 1, \ldots, m$–re.

Tehát az olvasófejek látják az input feldolgozatlan részének első és utolsó szimbólumát. Előfordulhat, hogy mindkét fej ugyanazt a szimbólumot olvassa, vagy akár az üresszót is, ha a teljes input feldolgozásra került. Az átmenetfüggvény alapján az automata eldönti, hogy melyik olvasófej lép (az is lehetséges, hogy mindkettő, vagy egyik sem lép). Az olvasófejek mozgása az input feldolgozatlan részének hosszától is függ. A definíció alapján a két olvasófej egyszerre nem lépheti át ugyanazt a szimbólumot. A számlálók aktualizálása az egy olvasófejjel rendelkező automaták esetéhez hasonlóan zajlik.

Megjegyezzük, hogy, ha a szövegkörnyezetből kiderül, hogy melyik automatáról van szó, akkor $\vdash_{\mathcal{A}}$ helyett az egyszerűbb $\vdash$ jelölést használjuk és annak reflexív, tranzitív lezártját, $\vdash^*$ –ot. Legyen $\mathcal{A} = (\Sigma, ¢, \$, m, \delta)$ egy tetszőleges $5' \rightarrow 3'$ Watson-Crick számlálóautomata, valamint $\boldsymbol{w} \in \Sigma^*$ egy lehetséges input szó. Ekkor a $C = (¢\boldsymbol{w}\$, 0^m)$ konfigurációt az $\mathcal{A}$ automata $\boldsymbol{w}$ inputon történő számításához tartozó kezdőkonfigurációnak (röviden csak kezdőkonfigurációnak) nevezzük. A $C_v = (\lambda, 0^m)$ konfigurációt pedig elfogadó konfigurációnak hívjuk. Az $\mathcal{A}$ automata által elfogadott nyelv pedig

$$\mathcal{L}(\mathcal{A}) = \{\boldsymbol{w} \in \Sigma^* \mid (¢\boldsymbol{w}\$, 0^m) \vdash^* (\lambda, 0^m)\}.$$

Azokat a nem elfogadó konfigurációkat, melyekben az átmenetfüggvény nem alkalmazható, azaz melyekből nem lehet további konfigurációkba lépni, elutasító konfigurációknak nevezzük.

Watson-Crick számlálóautomaták között is megkülönböztetünk valós-idejű és nem valós-idejű automatákat az alábbi definíció szerint.

3.7. Definíció. *Egy egyállapotú $5' \to 3'$ Watson-Crick számlálóautomata **valós-idejű**, ha minden a, $b \in (\Sigma \cup \{¢, \$\})$, $s \in \{0,1\}^m$ esetén*

$$\delta(a, b, \mathrm{sign}(s)) \subseteq \left\{(d_1, d_2, t) \in \{0,1\}^2 \times \{0, +, -\}^m \mid d_1 + d_2 \geq 1\right\}.$$

*Azaz, ha minden értelmezett átmenet esetén legalább az egyik olvasófej elmozdul. Ha ez a korlátozás nincs előírva, akkor az automatát **nem valós-idejűnek** nevezzük.*

3.3. *Megjegyzés.* Megjegyezzük, hogy valós-idejű automaták esetén a δ átmenetfüggvény 3.6. definíció szerinti felbontásában δ_2 üres. Továbbá minden valós-idejű automata egyben nem valós-idejű is.

Egy egyállapotú $5' \to 3'$ Watson-Crick számlálóautomata ***determinisztikus***, ha valamennyi a, $b \in (\Sigma \cup \{¢, \$\})$ és $s \in \{0,1\}^m$ esetén $|\delta(a, b, \mathrm{sign}(s))| \leq 1$. Azaz egy egyállapotú $5' \to 3'$ Watson-Crick számlálóautomata pontosan akkor determinisztikus, ha minden konfiguráció esetén, amennyiben létezik lehetséges lépés (azaz az átmenetfüggvény nem az üres halmazt adja eredményül), akkor az egyértelműen meghatározott. Itt is megjegyezzük, hogy a nemdeterminisztikus jelzőt általában elhagyjuk.

A könnyebb olvashatóság érdekében, ha egy példában az automata átmenetfüggvénye csak kevés paraméter esetén nem az üres halmaz, akkor egyszerűbb jelölésmódot használunk annak definiálására. Ekkor bármely

$$(d_1, d_2, t_1, \ldots, t_m) \in \delta(a, b, e_1, \ldots, e_m)$$

lehetséges átmenetet egy szabályként adunk meg, melynek alakja

$$(a, b, [e_1, \ldots, e_m]) \to (d_1, d_2, [t_1, \ldots, t_m]).$$

Ez nem vezet félreértésre sem a determinisztikus, sem a nemdeterminisztikus automaták esetén.

A következő példában az előbb bevezetett jelölést felhasználva megadunk egy olyan automatát, amely az

$$L = \{a^n b^n c^n d^n \mid n \geq 1\}$$

nyelvet fogadja el.

3.1. *Példa.* Legyen $\mathcal{A}_{abcd} = (\{a, b, c, d\}, ¢, \$, 2, \delta)$ a következő átmenetfüggvénnyel

$$\begin{aligned}
&(¢, \$, [0,0]) \to (1,1,[0,+]), &\quad &(b,c,[1,1]) \to (1,1,[-,-]),\\
&(a,d,[0,1]) \to (1,1,[+,0]), &\quad &(b,c,[1,0]) \to (1,1,[-,0]),\\
&(a,d,[1,1]) \to (1,1,[+,0]).
\end{aligned}$$

Látható, hogy ennek az automatának kettő számlálója van. Nevezzük ezeket r_1−nek és r_2−nek. A számítás elején, azaz, amikor az olvasófejek a $(¢, \$)$ párt látják, az automata r_2−t növeli, amellyel jelzi, hogy (a, d) párokat kell olvasnia. Minden elolvasott (a, d) pár esetén az r_1 számlálót eggyel növeli, így n−ig számol. Egyértelmű, hogy az a és d szimbólumok száma megegyezik, mivel ezeket csak egyszerre tudja elolvasni az automata. Ellenkező esetben az automata „zsákutcába” jutna, egyetlen szabályt sem lehetne alkalmazni. Az első (b, c) pár után az r_2 számláló értéke újra nullára csökken, ezzel jelezve, hogy most már csak (b, c) párok olvasására van lehetőség. Ezzel egy időben az r_1 számláló értéke is csökken eggyel, mivel egy (b, c) pár elolvasásra került. Valamint a további (b, c) párok elolvasása esetén is csökken r_1 értéke. Ha az összes (b, c) pár száma pontosan n, akkor mindkét számláló értéke 0 lesz, miután a teljes input feldolgozásra került. Ha a (b, c) párok száma nem n, akkor vagy az r_1 értéke lesz pozitív a számítás végén, vagy a számláló kiürül mielőtt az összes (b, c) pár feldolgozásra kerülne. Ekkor az input nyilvánvalóan nem $a^n b^n c^n d^n$ alakú, tehát az automata nem fogadja el, mert nem alkalmazható az átmenetfüggvény. Ezekből következik, hogy az automata pontosan az $a^n b^n c^n d^n$ $(n \geq 1)$ alakú szavakat fogadja el. Azaz $\mathcal{L}(\mathcal{A}_{abcd}) = \{a^n b^n c^n d^n \mid n \geq 1\}$. ◀

A 3.1. példában bemutatott automata 1−forduló, mivel minden számlálója 1−forduló. Azt is elmondhatjuk róla, hogy valós-idejű, mivel minden átmenetnél elmozdul legalább az egyik olvasófej. Valójában mindkét olvasófej lép valamennyi átmenet esetén.

Az $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$ nyelv is elfogadható 1−forduló automatával, ahogy a következő példa is mutatja:

3.2. *Példa.* Legyen $\mathcal{A}_{abcd_2} = (\{a, b, c, d\}, ¢, \$, 2, \delta)$ az alábbi átmenetfüggvénnyel.

$$\begin{array}{ll}
(¢, \$, [0,0]) \to (1, 1, [0, +]) & (b, d, [1,1]) \to (1, 1, [0, -]) \\
(a, d, [0,1]) \to (1, 0, [+, 0]) & (b, d, [1,0]) \to (1, 1, [0, 0]) \\
(a, d, [1,1]) \to (1, 0, [+, 0]) & (c, c, [1,0]) \to (1, 0, [-, 0])
\end{array}$$

Ekkor $\mathcal{A}_{abcd_2}$ pontosan az $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$ nyelvet fogadja el. Hasonlóan az előző példához, r_2 itt is csak bináris számláló, azt jelzi, hogy az a szimbólumok megszámlálása folyik. Minden egyes elolvasott a esetén az r_1 számláló értéke eggyel nő. Az első (b, d) pár megjelenésekor az r_2 értéke nullára állítódik, ami azt jelzi, hogy a b és d szimbólumok számának ellenőrzése következik és az input többi része nem tartalmazhat a−t. Ha a b−k és d−k száma megegyezik, akkor az automatának már csak azt kell ellenőriznie, hogy n darab c maradt-e az inputból. Ez az r_1 számláló segítségével könnyedén elvégezhető, mivel r_1 értéke pontosan n. ◀

A fenti automata is valós-idejű, de a 3.1. példabelivel szemben van olyan átmenete, amelyben az egyik olvasófej helyben marad.

Determinisztikus egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták

A következőkben a tárgyalt automaták determinisztikus változatával foglalkozunk, különös tekintettel az általuk elfogadott nyelvekre. Megjegyezzük, hogy már a 3.1. és a 3.2. példákban szereplő automaták is determinisztikusak.

3.1. Tétel (Determinisztikus számlálóautomaták korlátai [3])**.** *Az $L_{ab} = \{(ab)^n \mid n \in \mathbb{N}\}$ nyelvet egyetlen determinisztikus valós-idejű egyállapotú $k-$forduló $m-$számlálós automata sem fogadja el ($k, m \geq 1$).*

Bizonyítás. Indirekt tegyük fel, hogy létezik olyan determinisztikus valós-idejű $k-$forduló $m-$számlálós automata, amely az L_{ab} nyelv elemeit elfogadja. Csak valós idejű automatákat tekintünk, tehát a nyelv szavainak feldolgozása közben minden lépésben legalább az egyik olvasófej elmozdul. Tudjuk, hogy valamennyi számítás végén minden számláló értéke nulla és az utolsó lépés előtt egy-egy olvasófej az $a, b, ¢, \$$ szimbólumok valamelyikét látja. Jelöljük az utolsó előtti konfigurációban a bal, illetve jobb olvasófej által látott szimbólumot $x-$szel, illetve $y-$nal. Mivel a számítás során legalább az egyik olvasófej lép, nem teljesülhet egyszerre, hogy $x = ¢$ és $y = \$$, valamint az $x = y = ¢$ és $x = y = \$$ esetek azonosnak tekinthetők (legalábbis az L_{ab} nyelv feldolgozásakor), abból a szempontból, hogy ekkor csak az egyik olvasófej mozdul el a számítás során. Most tegyük fel, hogy csakis a bal olvasófej lép, azaz a számítás utolsó lépése előtt $x = y = \$$. Mivel valós-idejű automatáról beszélünk, képesnek kell lennie valamennyi számlálót kinullázni, ha a bal olvasófej $\$-$t lát. Tehát egy-egy számláló értéke legfeljebb 1 lehet és legfeljebb $k-$szor váltakozhat 0 és 1 között. Továbbá minden lépés során legalább az egyik számláló értékét kell változtatni (0-ról 1-re, vagy fordítva), különben a szóban az adott lépésben elolvasott szimbólum megkettőzésével kapnánk egy olyan szót, amely nem eleme $L_{ab}-$nek, de az automata mégis elfogadja. Ezen megszorítások mellett a számlálók legfeljebb 2^m különböző számítási állapotot jelölhetnek. Eltekintve attól a ténytől, hogy a fordulások nyomon követésére is számlálókat kell igénybe venni, feltehetjük, hogy a 2^m darab számítási állapot legfeljebb $k-$szor fordulhat elő (mivel $k-$forduló az automata). Azaz $n > k \cdot 2^m$ esetén $(ab)^n$ (mivel csak egy fej lép, akár az $(ab)^{\frac{n}{2}}$ szót is elég venni) feldolgozása során legalább egy további fordulás, vagy számláló szükséges. Azaz nem létezik olyan determinisztikus valós-idejű $k-$forduló $m-$számlálós automata, amely az L_{ab} nyelvet elfogadná úgy, hogy csakis az egyik olvasófejet használja.

Tegyük fel, hogy a számítás során mindkét olvasófej lép (nem feltétlenül mindig egyszerre). Ebben az esetben is igaz, hogy az automatának képesnek kell lennie arra, hogy minden számláló értékét lenullázza a számítás utolsó lépésében. Ha mindkét fej elmozdul valamennyi lépésben, akkor a fejek által olvasott szimbólumpárok váltakozva (a, b) és (b, a). A fenti gondolatmenetet követve újra arra a következtetésre jutunk, hogy $n > k \cdot 2^m$ esetén $(ab)^n$ feldolgozása során nem elégséges az m darab $k-$szor forduló számláló. Ezért tegyük fel, hogy léteznek olyan lépések, amikor valamelyik olvasófej nem mozdul el. Ekkor sem léphet egyik fej sem úgy, hogy közben egyetlen nem nulla számláló értéke sem változik, hiszen az általa átlépett betű megsokszorozásával olyan szót is elfogadhatna az automata, amely nem eleme az L nyelvnek. Egy számláló értéke pedig nem lehet több, mint kettő, mivel elfogadás esetén az utolsó lépésnél valamennyi számlálót le kell nullázni. A fenti esettel ellentétben azért lehet egy számláló értéke kettő is, mivel előfordulhat, hogy két egymás utáni lépésben a bal, majd a jobb olvasófej azonos betűt lép át (például a–t) és így megjegyezheti, hogy a soron következő lépés semmiképpen sem lehet az utolsó. Hasonlóan az előző esetekhez, $n > k \cdot 3^m$ bő felső korlátot használva, $(ab)^n$ elfogadása nem lehetséges m darab k–forduló számlálóval.

Tehát beláttuk, hogy nem létezik olyan determinisztikus valós-idejű k–forduló m–számlálós automata, amely az L_{ab} nyelvet elfogadná. ■

3.1. Következmény. *Létezik olyan reguláris nyelv, amelyet egyetlen determinisztikus valós-idejű egyállapotú k–forduló m–számlálós automata sem fogad el, függetlenül a k, m értékektől.*

Bizonyítás. A 3.1. tételben említett $L_{ab} = \{(ab)^n \mid n \in \mathbb{N}\}$ nyelv pontosan egy ilyen reguláris nyelv. ■

Láthattuk, hogy a valós-idejű automaták esetén a számlálókban tárolható értékek az input hosszától függnek, melyeknek a számítás végére ki kell ürülniük. Ez megnehezíti (sok esetben lehetetlenné teszi) két számláló tartalmának összehasonlítását, vagy, ami még fontosabb, a számlálókban lévő értékekkel történő műveletek elvégzését.

A 3.2 tételben megmutatjuk, hogy a valós-idejű megkötés elhagyásával, kevés korlátlanul forduló számlálóval mire képesek ezek az automaták. Előtte definiáljuk a Parikh vektor és a szemilineáris nyelv fogalmát [39]. Legyen $\Sigma = \{a_1, \ldots, a_n\}$ egy rendezett n betűből álló ábécé és $\boldsymbol{w} \in \Sigma^*$. Ekkor a $\boldsymbol{w}$ szó ***Parikh vektora*** $\Phi(\boldsymbol{w}) = (|\boldsymbol{w}|_{a_1}, \ldots, |\boldsymbol{w}|_{a_n})$ egy olyan n dimenziós vektor, amely azt mutatja, hogy a $\boldsymbol{w}$ szóban az ábécé adott betűi hányszor szerepelnek. Adott L nyelv ***Parikh halmazát*** pedig a benne lévő szavak Parikh vektorai alkotják, azaz

$$\Phi(L) = \{\Phi(\boldsymbol{w}) \mid \boldsymbol{w} \in L\}\,.$$

Ha léteznek olyan $v_0, \ldots, v_k \in \mathbb{N}_0^n$ vektorok, hogy bármely $u \in \Phi(L)$ esetén léteznek olyan $\ell_1, \ldots, \ell_n \in \mathbb{N}$ egészek, hogy $u = v_0 + \ell_1 v_1 + \ldots + \ell_n v_n$, akkor azt mondjuk, hogy az L nyelv (Parikh értelemben) lineáris. Ha $\Phi(L)$ véges sok lineáris halmaz uniója, akkor azt mondjuk, hogy L (Parikh értelemben) szemilineáris. Valamennyi környezetfüggetlen nyelv szemilineáris [39]. Egy jól ismert környezetfüggő, nem szemilineáris nyelv $L = \{a^{n^2} \mid n \geq 0\}$, amelyről a következő tételben megmutatjuk, hogy viszonylag kevés számlálóval elfogadható.

3.2. Tétel (Egy szemilineáris, nem környezetfüggő nyelvet elfogadó automata [4])**.** *Létezik olyan determinisztikus nem valós-idejű egyállapotú $5' \to 3'$ Watson-Crick számlálóautomata, amely az $L = \{a^{n^2} \mid n \geq 0\}$ nyelvet fogadja el.*

Bizonyítás. A bizonyítás konstruktív és azon a tényen alapul, hogy az $n-$edik négyzetszám pontosan az első n darab páratlan szám összegeként áll elő, a páratlan számok pedig pontosan $2k + 1$ alakúak, ahol $k \in \mathbb{N}_0$. Olyan számlálóautomatát konstruálunk, amely működése során egy bizonyos műveletsorozatot ismétel. Egy teljes műveletsorozatot nevezzünk iterációnak. A legelső lépésben a ¢ és \$ szimbólumok átlépésre kerülnek, ez nem tartozik bele egyik iterációba sem. Az első iterációban pontosan egy darab a szimbólumot olvasunk. Ha már az $i-$edik ($i \in \mathbb{N}$) iterációban k darab $a-$t olvastunk, akkor az $i + 1-$edikben $k + 2$ darabot fogunk. Az értékek számon tartásához négy számlálóra lesz szükségünk, ezeket C_1, C_2, C_m és C_p-vel jelöljük. A C_p számláló azért felelős, hogy az aktuális iterációban páratlan darabszámú $a-$t olvassunk. Az aktuálisat megelőző iterációban elolvasott $a-$k számának tárolására pedig a C_1 és C_2 számlálókat fogjuk használni. Most nézzük az automata részletes működését.

Az első iterációban a bal olvasófej elolvas egyetlen $a-$t és az automata a C_p számlálót növeli eggyel. Ekkor, ha már nincs több a az inputban, a számlálók (azaz csak C_p) lenullázhatók, mivel 1 négyzetszám. Az összes többi iterációban két eset lehetséges:

- $C_m = 0$: Ekkor az automata C_p-t csökkenti, C_1-et növeli, miközben mindkét olvasófejjel átlép egy-egy szimbólumot. Ha $C_p = 0$, akkor az automata a következő ciklust hajtja végre: amíg $C_2 \neq 0$, az olvasófejek egy-egy $a-$t olvasnak, C_2 értéke csökken, C_1 értéke pedig nő (azaz C_2 tartalma is C_1-be kerül). Ezután C_1 tartalma pontosan eggyel nagyobb, mint C_2-é volt, C_2 pedig üres.

- $C_m = 1$: Hasonlóan az előző esethez, viszont itt C_2-be történik másolás, azaz C_2 értéke lesz eggyel nagyobb, mint C_1-é volt.

Az iteráció végén pedig, mivel $C_p = 0$, azaz páros számú $a-$t olvastunk, az automata a jobb oldali olvasófejjel elolvas egy a szimbólumot miközben C_p értékét növeli és C_m-et növeli vagy csökkenti, attól függően, hogy $C_m = 0$ vagy $C_m = 1$.

Ha pedig nem maradt több a az inputban, akkor a nem valós-idejű szabályok szerint a számlálók lenullázhatók.

Az így megkonstruált automata a következő:

$$\mathcal{A} = (\{a\}, ¢, \$, 4, \delta),$$

ahol δ a következő:

$$(¢, \$, [0,0,0,0]) \to (1,1,[0,0,0,0])$$

$$(a, a, [0,0,0,0]) \to (1,0,[0,0,0,+])$$

$$\begin{aligned}
&(a, a, [0,0,0,1]) \to (1,1,[+,0,0,-])\\
&(a, a, [0,1,0,1]) \to (1,1,[+,0,0,-])\\
&(a, a, [1,1,0,0]) \to (1,1,[+,-,0,0])\\
&(a, a, [1,0,0,0]) \to (0,1,[0,0,+,+])
\end{aligned}$$

$$\begin{aligned}
&(a, a, [1,0,1,1]) \to (1,1,[0,+,0,-])\\
&(a, a, [1,1,1,0]) \to (1,1,[-,+,0,0])\\
&(a, a, [0,1,1,0]) \to (0,1,[0,0,-,+])
\end{aligned}$$

$$\begin{aligned}
&(\lambda, \lambda, [0,1,0,1]) \to (0,0,[0,-,0,-])\\
&(\lambda, \lambda, [0,1,0,0]) \to (0,0,[0,-,0,0])\\
&(\lambda, \lambda, [1,0,1,1]) \to (0,0,[-,0,-,-])\\
&(\lambda, \lambda, [1,0,0,0]) \to (0,0,[-,0,0,0])
\end{aligned}$$

Látható, hogy öt részre osztva adtuk meg a szabályokat. Ezek rendre a legelső lépést, első iterációt, $C_m = 0$, $C_m = 1$ esetet, valamint a befejező nem valós-idejű szabályokat tartalmazzák. ■

3.3. *Példa.* Az előző automata működése az $a^{16} = aaaaaaaaaaaaaaaa$ inputon:

$$
\begin{array}{lcl}
(¢aaaaaaaaaaaaaaaa\$, & & [0,0,0,0]) \vdash \\
(\ \ aaaaaaaaaaaaaaaa & , & [0,0,0,0]) \vdash \\
(\quad aaaaaaaaaaaaaaa & , & [0,0,0,1]) \vdash \\
(\quad aaaaaaaaaaaaa & , & [1,0,0,0]) \vdash \\
(\quad aaaaaaaaaaaa & , & [1,0,1,1]) \vdash \\
(\qquad aaaaaaaaaa & , & [1,1,1,0]) \vdash \\
(\qquad aaaaaaaa & , & [0,2,1,0]) \vdash \\
(\qquad aaaaaaa & , & [0,2,0,1]) \vdash \\
(\qquad\ aaaaa & , & [1,2,0,0]) \vdash \\
(\qquad\quad aaa & , & [2,1,0,0]) \vdash \\
(\qquad\quad\ a & , & [3,0,0,0]) \vdash \\
(\qquad\quad\ | & , & [3,0,1,1]) \vdash \\
(\qquad\quad\ | & , & [2,0,0,0]) \vdash \\
(\qquad\quad\ | & , & [1,0,0,0]) \vdash \\
(\qquad\quad\ | & , & [0,0,0,0])
\end{array}
$$

Az egyes lépésekben látható az input feldolgozásra váró része, amely az olvasófejek mozgását is mutatja. A | szimbólumot csak annak a pontnak a szemléltetésre használjuk, ahol a két olvasófej találkozott. Ez pontosan az utoljára feldolgozott a előtt történik, mivel azt a jobb olvasófej lépi át. Onnantól kezdve mindkét olvasófej csak λ−t lát. ◀

3.2. Következmény. *A determinisztikus nem valós-idejű egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták által elfogadott nyelvek osztálya tartalmaz nem szemilineáris, így nem környezetfüggetlen nyelvet.*

Nemdeterminisztikus egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták

Eddigi eredményeink determinisztikus automatákra vonatkoztak. Megmutatjuk, hogy a nemdeterminisztikus automaták kifejezőereje nagyobb, mint a megfelelő determinisztikus változatoké. Legyen $n \in \mathbb{N}$ és definiáljuk a következő paraméterezett nyelvet:

$$L_n = \big\{ a^i c b^{j_1} c b^{j_2} \cdots c b^{j_n} \mid i \in \mathbb{N}, j_1, \ldots, j_n \in \mathbb{N}_0 \text{ és}$$
$$i = j_\ell \text{ valamely } \ell \in \{1, \ldots, n\} \text{ esetén} \big\}\,.$$

Tehát L_n olyan szavakat tartalmaz, melyek valahány darab $a-$val kezdődnek, ami után n darab, $b-$kből álló blokk következik egy-egy $c-$vel elválasztva. Megengedjük a nulla hosszú blokkokat, azaz akár több c is lehet egymás mellett. Az egyik ilyen blokk mérete pontosan akkora, mint a szó elején lévő $a-$kból álló blokk.

Vegyük valamennyi L_n nyelv unióját, azaz az

$$L' = \bigcup_{n \in \mathbb{N}} L_n$$

nyelvet, melyet a következő tétel bizonyításában fogunk használni.

3.3. Tétel (Nemdeterminisztikus automaták kifejezőereje [1])**.** *Létezik olyan nyelv, melyet elfogad egy nemdeterminisztikus valós-idejű egyállapotú $1-$forduló $1-$számlálós automata, de egyetlen $k, m \in \mathbb{N}$ esetén sem fogadja el egy determinisztikus valós-idejű egyállapotú $k-$forduló $m-$számlálós automata.*

Bizonyítás. Vegyük azt a nemdeterminisztikus automatát, melyet az alábbi szabályokkal adunk meg.

$$\begin{array}{ll}
(¢, \$, [0]) \to (0, 1, [0]) & (a, b, [1]) \to (1, 1, [0]) \\
(¢, \$, [0]) \to (1, 1, [+]) & (c, c, [1]) \to (0, 1, [-]) \\
(¢, b, [0]) \to (0, 1, [0]) & (c, b, [0]) \to (0, 1, [0]) \\
(¢, c, [0]) \to (0, 1, [0]) & (c, c, [0]) \to (0, 1, [0]) \\
(¢, c, [0]) \to (1, 1, [+]) &
\end{array}$$

Ez az automata pontosan a fentebb említett L' nyelvet fogadja el. A szabályokat két nagy csoportra bonthatjuk. A baloldalon lévők segítségével az automata a jobb olvasófejet mozgatja és nemdeterminisztikusan választ egy $b-$kből álló blokkot, amit vagy egy c „előz” meg jobbról, vagy a \$ szimbólum (mikor is a szó végén lévő blokkra esik a választás). A számláló bináris, az értéke pontosan akkor egy, ha megtörtént a választás. A jobboldali szabályok alapján pedig az automata addig olvassa az (a, b) párokat, amíg azok mind el nem fogynak. Ekkor mindkét olvasófej szükségszerűen $c-$t lát (lehet, hogy ugyanazt a betűt, ha például L_1-beli szót dolgoz fel az automata). Ezt a $c-$t a jobb olvasófej átlépi és innentől kezdve további (a, b) párok olvasása nem lehetséges, mivel a számláló értéke is nullázódik. A szó további részében csak b és c betűk szerepelhetnek, ezeket a jobb olvasófej átlépi, amíg az input el nem fogy.

Adott számítás során egy valós idejű számlálóautomata véges sok értéket tud tárolni a számlálóiban, mivel a számítás végére ki kell ürítenie azokat. Determinisztikus automaták esetén a szó elején lévő $a-$k számát pedig valamennyi $b-$kből álló blokkal össze kell tudni hasonlítani. Ehhez kettő vagy több blokk esetén mindenképpen szükséges az $a-$k számának letárolása. Mivel két olvasófej áll rendelkezésre, megoldható, hogy az $a-$k megszámolása közben elvégezzük az egyik blokkal való összehasonlítást. Továbbá, ha az $a-$k száma már ismert, egyszerre két blokkot lehet vizsgálni (szintén a két olvasófejnek köszönhetően). Azaz $n \in \mathbb{N}$ darab blokk esetén $\lceil \frac{n-1}{2} \rceil$ alkalommal lesz szükség az $a-$k számára. Igen optimális működést feltételezve egy $k-$forduló automata valamennyi forduláskor mind az m darab számlálóban le tudja tárolni ezt az értéket, azaz összesen $k \cdot m$ összehasonlítást tud végezni. Tehát egy $k-$forduló $m-$számlálós automata legfeljebb $n = 2(k \cdot m) + 1$ blokkot tartalmazó szavak feldolgozására képes.

Így bármely $k, m \in \mathbb{N}$ esetén létezik olyan $n \in \mathbb{N}$, hogy az L_n nyelvet egyetlen determinisztikus valós idejű egyállapotú $k-$forduló $m-$számlálós automata sem fogadja el. Tehát nem létezik az L' nyelvet elfogadó determinisztikus automata. ∎

Különböző paraméterekre vonatkozó hierarchiák

Az előbbiekben példanyelvekkel megmutattuk, hogy az automaták bizonyos paraméterei közötti különbségek hogyan mutatkoznak meg az elfogadott nyelvekben. Automata alatt továbbra is egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatát értünk. Először a különböző paraméterekkel rendelkező automaták által elfogadott nyelvek osztályainak jelölését ismertetjük.

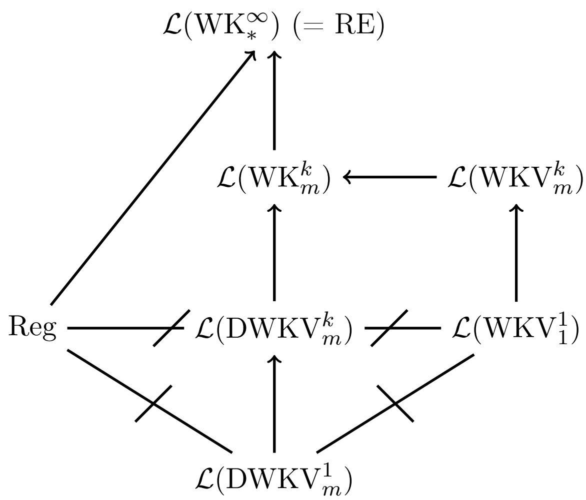
Legyenek $k, m \in \mathbb{N}$ és $k < \frac{2^{m-1}}{m}$ a 3.8. lemma teljesülése végett, ugyanis az automata implicite oldja meg a számlálók fordulásainak korlátozását. Jelölje WK_m^k a (nemdeterminisztikus) nem valós-idejű egyállapotú $k-$forduló $m-$számlálós automaták osztályát, DWK_m^k pedig az azonos paraméterekkel rendelkező determinisztikus automatákból álló osztályt. Hasonlóan, rendre WKV_m^k és DWKV_m^k jelöli a valós-idejű, valamint a determinisztikus valós-idejű részhalmazokat. Továbbá bevezetjük a WK_m^∞ jelölést, amely a korlátozás nélkül forduló $m-$számlálós automaták halmazát jelöli.

Ha K egy automataosztály, akkor $\mathcal{L}(K)-$val jelöljük az osztályba tartozó automaták által elfogadott nyelveket, azaz

$$\mathcal{L}(K) = \{\mathcal{L}(\mathcal{A}) \mid \mathcal{A} \in K\}.$$

Eredményeink a 3.1. ábrán látható hierarchiát támasztják alá, melyen a nyilak valódi tartalmazást jelentenek, az áthúzott vonalak pedig azt jelzik, hogy a két nyelvosztály halmazelméleti tartalmazást tekintve nem összehasonlítható, azaz mindkét osztályban vannak olyan nyelvek, amelyek a másikban nem szerepelnek. Ha két osztály között az ábrán nem jelöltünk semmilyen relációt, akkor annak teljes felderítése még megoldatlan feladat. Továbbá a tartalmazási reláció tranzitivitása miatt a redundáns nyilakat elhagytuk.

3.4. Tétel (Egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták által elfogadott nyelvek)**.** *Legyenek* $k, m \in \mathbb{N}$ *úgy, hogy* $k < \frac{2^{m-1}}{m}$ *és* $m \geq 2$. *Ekkor az egyállapotú* $5' \to 3'$ *Watson-Crick számlálóautomaták által elfogadott nyelvek osztályai között az alábbi hierarchia áll fenn.*



3.1. ábra. Az ábrán Reg jelöli a reguláris, RE pedig a rekurzív felsorolható nyelvek osztályát.

1615

Bizonyítás. Az $\mathcal{L}(\mathrm{DWKV}_m^1) \subset \mathcal{L}(\mathrm{DWKV}_m^k)$ és a $\mathcal{L}(\mathrm{WKV}_1^1) \subset \mathcal{L}(\mathrm{WKV}_m^k)$ relációk a 3.7. lemma következményei. A 3.3. tétel alapján $\mathcal{L}(\mathrm{WKV}_1^1)$ tartalmaz olyan nyelvet, melyet sem $\mathcal{L}(\mathrm{DWKV}_m^1)$, sem $\mathcal{L}(\mathrm{DWKV}_m^k)$, viszont a 3.6. és 3.7. lemmák miatt

$$\{a^n \mid n = ((2k-1)m - 1)2^{(2k-1)m} + (2k-1)m\} \in \mathcal{L}(\mathrm{DWKV}_m^k)$$

és

$$\{a^n \mid n = (m-1)2^m + m\} \in \mathcal{L}(\mathrm{DWKV}^1_m),$$

mely nyelvek $m \geq 2$ esetén nem elemei $\mathcal{L}(\mathrm{WKV}^1_1)$−nek. A 3.1. tétel és annak következménye alapján fennáll, hogy a reguláris nyelvek osztályával halmazelméleti tartalmazás szerint nem hasonlítható össze sem $\mathcal{L}(\mathrm{DWKV}^1_m)$, sem $\mathcal{L}(\mathrm{DWKV}^k_m)$.

Ahogy a 3.1. lemmában kimondtuk, *Marvin L. Minsky* megmutatta, hogy a két számlálóval rendelkező véges automaták által elfogadott nyelvek osztálya megegyezik a rekurzív felsorolható nyelvekkel [35]. Mivel állapotokat lehet szimulálni számlálókkal, ezért $\mathcal{L}(\mathrm{WK}^\infty_*) = \mathrm{RE}$, ahol $\mathrm{WK}^\infty_* = \bigcup_{m\in\mathbb{N}} \mathrm{WK}^\infty_m$. ∎

3.3. Összefoglalás

A társszerzőimmel publikált cikkekben [1, 3, 8, 4] máshogy definiáltuk az egyállapotú $5' \rightarrow 3'$ számlálóautomaták átmenetfüggvényét, de úgy vélem, hogy ezzel az újabb, ekvivalens definícióval természetesebben sikerült általánosítani a hagyományos számlálóautomaták működését. Valamint az átmenetek definiálása is egyszerűbb. Az új definíció használata és a részletek kifejtése miatt jelentősen átdolgoztam a tételek, állítások bizonyításait.

A 3.2. alfejezetben tárgyalt eredmények a [3] cikkben kerültek közlésre, mely bővített változata a [4] műnek. A 3.2. alfejezet paraméterezett nyelve és a rá vonatkozó tétel hasonló nyelvhierarchiára vonatkozó eredményekkel együtt az [1, 8] cikkekben jelent meg.

Nyitott kérdés, hogy a 3.4. tételben található $\mathcal{L}(\mathrm{WK}^\infty_*) = \mathrm{RE}$ reláció bal oldala finomítható-e, azaz létezik-e olyan m, melyre $\mathcal{L}(\mathrm{WK}^\infty_m) = \mathrm{RE}$.

A kiszámolt eredményekre építve meghatároztunk bizonyos paraméterekre vonatkozó hierarchiákat. További kutatásokkal lehetséges ezek bővítése, finomítása. Egy megkezdett, de kevésbé kidolgozott irány az olyan egyállapotú $5' \to 3'$ Watson-Crick számlálóautomaták vizsgálata, melyek olvasófejei nem csak egy szimbólumot tudnak elolvasni, hanem korlátos hosszú szavakat, hasonlóan a 3.1. fejezetben ismertetett általános Watson-Crick automatákhoz [5]. Ekkor egy újabb paraméter is megjelenik, az átmenetek sugara, ami adott átmenetben a fejek által látott szavak hosszainak összege. Ebből származtatható az automata sugara.

A valós-idejű és nem valós-idejű automaták definiálásakor egy jelenleg még megoldatlan problémára hívtuk fel az olvasó figyelmét. Még nem ismert, hogy milyen feltételek mellett dönthető el, hogy adott nem valós-idejű számlálóautomata adott inputon véges sok lépés után megáll-e vagy sem.

Tehát ezen a területen is vannak még olyan nyitott kérdések, melyek megválaszolásra várnak.

4 Összefoglalás

A dolgozatban két új modell került ismertetésre, melyek a formális nyelvek és automaták tudományterület alapvető eszközeinek különböző, jól meghatározott módosításaival állnak elő. Ennek megfelelően két egységre bontottuk a tárgyalást. Az első rész a körszavak kombinatorikájával foglalkozott, ezen belül is a körszavak periodikus tulajdonságaival, a második rész pedig az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákkal és az általuk elfogadott nyelvek közötti hierarchiával.

A 2.1. fejezet bevezető jellegű, benne a szavak kombinatorikája témakör alapfogalmait ismertettük, végül a hagyományos (lineárisan reprezentált) szavak periódusaival kapcsolatos néhány fontos eredmény felelevenítésével zártuk. Ezután két fejezetben tárgyaltuk a dolgozat egyik fő modelljét, a körszavakat. A 2.2. fejezet elején megadtuk a körszó definícióját, amely a hagyományos szavak egy általánosításának felel meg. A 2.2 alfejezet elején általánosítottuk a periódus fogalmát körszavakra, ezzel megadva a gyenge periódus definícióját. A fejezet további részében a hagyományos szavak esetén ismert legfontosabb eredményekkel analóg állításokat bizonyítottunk be körszavakra. Megadtuk annak szükséges és elégséges feltételét, hogy adott körszó mikor rendelkezik adott gyenge periódussal (2.1. tétel), majd az egy körszóhoz tartozó különböző gyenge periódusok egymáshoz való viszonyát vizsgáltuk. Itt először megadtunk egy mohó algoritmust, mely alapján előállítható olyan adott hosszúságú körszó, amelynek gyenge periódusai között szerepel két adott egész szám. Egy példával szemléltettük ezen algoritmus alkalmazását. Ezután a 2.2. tételben kimondtuk, hogy egy nem unáris körszó milyen feltételek teljesülése mellett rendelkezhet két olyan adott periódussal, melyek relatív prímek. Ez a 2.2. lemma egy speciális esetének felel meg. Itt megjegyezzük, hogy *Nathan J. Fine* és *Herbert S. Wilf* teljes lemmájának körszavak gyenge periódusaira való általánosítása még nyitott probléma, mivel az eredeti lemma két tetszőleges (nem feltétlenül relatív prím) periódus esetére fogalmaz meg feltételt. Ennek ellenére úgy véljük, hogy eredményünk egy nagy lépést jelent a probléma megoldásának irányába. A fejezetben megmutattuk azt is, hogy adott hossz esetén legfeljebb hány különböző gyenge periódussal rendelkezhet egy nem unáris körszó (2.3. tétel). Körszavak fákkal történő ábrázolásával foglalkoztunk a 2.3. fejezetben, melyben leginkább Fibonacci szavakból képzett körszavakra helyeztük a hangsúlyt, ugyanis a hozzájuk tartozó fák szerkezete igen érdekes. Az ismertetett megközelítés alkalmazható például a végtelen Fibonacci szó egyes tulajdonságainak vizualizálásához is. További kutatási irány lehet, hogy a fejezetben említett tulajdonságokat fel lehet-e használni más, a Fibonacci szóhoz hasonló végtelen szavak jellemzéséhez.

A dolgozat második része a számlálóautomaták egy speciális osztályával foglalkozik, az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákkal. Ennek a résznek sajátossága, hogy több különböző modell jelölésrendszerét is egységesíti, amely megkönnyíti a tárgyalást. Bevezetésként a 3.1. fejezetben ismertetésre kerültek az egyállapotú számlálóautomaták és néhány velük kapcsolatos fontosabb eredmény. A több olvasófejjel rendelkező automaták, különös tekintettel a Watson-Crick automaták is itt kerülnek definiálásra. A két modell kombinálásával jutunk el az egyállapotú $5' \to 3'$ Watson-Crick számlálóautomatákhoz, amelynek vizsgálatával a dolgozat további része foglalkozik. Először is a 3.2. fejezetben megadtuk a modell általános definícióját és a számítás leírására használható konfigurációkat, amely a megjelent cikkekben tárgyalt modellel ekvivalens, de annál könnyebben kezelhető. Az ilyen automaták kifejezőerejét már a fejezet elején két nem környezetfüggetlen, környezetfüggő nyelvet elfogadó egyszerű példával szemléltettük. A 3.2. alfejezetben a modell determinisztikus változatainak vizsgálatára került a hangsúly. Mutattunk olyan környezetfüggetlen és reguláris nyelvet, amelyek elfogadása véges sok számlálóval sem lehetséges, ha megköveteljük, hogy az automata olvasófejei közül legalább az egyik elmozdul minden lépésben. Ezután a megengedőbb modell kifejezőerejét szemléltettük egy ismert környezetfüggő, nem szemilineáris nyelvet elfogadó szabályokkal. A 3.2. alfejezetben pedig megmutattuk, hogy a determinisztikusság elhagyásával a modell kifejezőereje tovább növelhető. A dolgozat ezen részét a bebizonyított tételekre és szemléltetett példákra alapozott nyelvhierarchiák szemléltetésével zártuk.

Mindkét rész esetében ismertettünk lehetséges további kutatási irányokat, melyeket az adott részhez tartozó összefoglalásokban külön kiemeltünk. Véleményünk szerint a két vizsgált modellel kapcsolatos szakmai érdeklődés sem elhanyagolható, amelyet a megjelent publikációk is alátámasztanak.

5 Summary

During the last decades we can observe a growing interest in unconventional models of computation. The field is vast, since it consists of all models that are different in one way or another from the ones described in classical technical literature [25]. Biology proved to be a major source of ideas. The central element of the field is bioinspired computing which deals with the practical and theoretical applications of biological and chemical processes observed inside living things. My research is based on two models.

In nature we can find rings formed from DNA strands (e.g., mitochondrial DNA) that can be considered as a motivation for analyzing the properties of circular words. Nevertheless, my approach to circular words is purely mathematical and independent of the biological motivation. The notion of circular word is not new, other researchers investigated certain properties of them [17, 18, 45], but I introduced a new notion of periodicity which lead to the discovery of brand new results. The analysis of periodic properties of words goes back to the beginning of the 1900s when *Axel Thue* published his articles about square-free words [47, 48]. The first compilation of results in this area was written by a group of people under the pen name *M. Lothaire* [31]. Most of them were students of *Marcel P. Schützenberger* who is well known for his contributions to this field. All of the work mentioned above deals with ordinary (linear) words and rarely mention circular words, even though the field has great potential and many interesting open questions.

I assume that the reader is familiar with the basic notions and notation of the theory of formal languages and automata [25]. The results of my work on periodic properties of circular words can be summarized as follows. We consider a circular word as a set that contains exactly those words that are conjugates (cyclic shifts) of each-other. Using the notion of period for ordinary words I defined the ***weak period*** which is a period of at least one element of a given circular word. That is, the integer $p > 0$ is a weak period of the circular word $\boldsymbol{w}_\circ$ if and only if p is a period of at least one $\boldsymbol{v} \in \boldsymbol{w}_\circ$ [6]. I found and proved a necessary and sufficient condition for a given integer p to be a weak period of a given circular word $\boldsymbol{w}_\circ$ [2]. For weak periods of circular words I proved a generalization of the case of the Periodicity Lemma of Fine and Wilf where p and q are relatively prime. The complete generalization of the Lemma remains an open problem, but I believe that my result covers a significant part of the solution. In connection to this issue I stated and proved a strict upper limit on the number of distinct weak periods of a non-unary circular word of given length. Regarding the representation of circular words I analyzed the potential in the tree data structure [2, 7]. Similarly to suffix tries I defined the notion tree of a circular word. A tree of a circular word is a hierarchical data structure such that there is a one-to-one correspondence between the routes from the root to the leaf nodes and the elements of the circular word. This representation proved to be useful when I considered circular words obtained from Fibonacci words. I observed a recurring pattern in the trees of circular Fibonacci words and discovered that the tree of a circular Fibonacci word contains all trees of shorter circular Fibonacci words as rooted subtrees. From this I concluded that all elements of all Fibonacci words are factors of the infinite Fibonacci word. This confirms that the analysis of this model should not be neglected even from the point of view of classical models. Supplementing the results above, my dissertation contains the necessary lemmas with their proofs that build the foundation on which these results were based.

The other topic of my research is the stateless multicounter $5' \to 3'$ Watson-Crick automaton which is a special variation of the well known Watson-Crick automaton of DNA computing. The model is new and it has several parameters that affect the expressing power significantly. Therefore my research targeted the hierarchy of accepted languages based on these parameters.

The counter machine as a basis of computation can be found in classical literature. One of the most influential works in this area were written by *Marvin L. Minsky* who proved that the solution of any algorithmic problem can be simulated by counter machines that have only two counters [35]. Some articles of *Ömer Eğecioğlu* and *Oscar H. Ibarra* deal with a special case of these automata, namely stateless multicounter machines [20, 21]. My advisor, Benedek Nagy have done research on $5' \to 3'$ Watson-Crick automata which is a bioinspired computational model. Based on the previous research we started working together with *Ömer Eğecioğlu* on stateless multicounter $5' \to 3'$ Watson-Crick automata. After unifying the notation of stateless multicounter machines and mutlihead automata I gave the formal definition of stateless multicounter $5' \to 3'$ Watson-Crick automata [4]. I defined the configurations and a relation over configurations that describe the computation of an automaton in a way that prohibits the reading heads to cross each-other. I proved that there exists a regular language that cannot be accepted by any finite number of counters by a deterministic stateless multicounter $5' \to 3'$ Watson-Crick automaton [3]. I also defined a deterministic stateless multicounter $5' \to 3'$ Watson-Crick automata with four counters that accepts a non-context-free semilinear language [3]. Thus I showed that the class of languages accepted by deterministic non-realtime stateless multicounter $5' \to 3'$ Watson-Crick automata contains a semilinear non-context-free language. I proved that the expressive power of nondeterministic stateless multicounter $5' \to 3'$ Watson-Crick automata is greater than that of their deterministic counterparts[8].

I provided constructive proofs for my claims wherever it was possible. There remain several open questions regarding the two models that I have investigated, thus I can continue working on this area in the future.

6 Köszönetnyilvánítás

Hálás köszönettel tartozom édesanyámnak, aki felnevelt és minden döntésemben támogatott. Nélküle nem jöhetett volna létre ez a dolgozat.

Köszönet illeti témavezetőmet, Dr. Nagy Benedeket, akivel még mesterszakos hallgatóként kezdtem dolgozni és azóta támogatja tudományos fejlődésemet, mely véleményem szerint csak végtelen türelemmel lehetséges. Iránymutatásával, tanácsaival minden lehetőséget megteremtett ahhoz, hogy eredményes kutatómunkát végezhessek a választott területen.

Szakmai fejlődésemhez hozzájárult a Debreceni Egyetem Informatikai Karának szinte valamennyi volt és jelenlegi oktatója. Köszönöm Dr. Dömösi Pálnak, hogy érdekes előadásaival felkeltette figyelmem a formális nyelvek és automaták iránt; Dr. Pethő Attilának, hogy már mesterszakos hallgatóként számítástudománnyal foglalkozhattam és mindig támogatta a tudományos munkámat, rengeteget segített a munkahelyi vitám megszervezésében; Dr. Várterész Magdának, hogy tanácsokkal látott el tanulmányaim során.

Köszönöm Dr. Vaszil Györgynek, hogy a munkahelyi vitára bocsátott dolgozatom bírálatát rövid határidőn belül elkészítette és megjegyzéseivel, javaslataival hozzájárult dolgozatom színvonalának emeléséhez.

Köszönettel tartozom továbbá mindazoknak, akik példaként és ellenpéldaként szolgáltak eddigi életem során.

Saját közlemények

Nemzetközi folyóiratokban megjelent művek

[1] Ömer Eğecioğlu – László Hegedüs – Benedek Nagy: Hierarchies of stateless multicounter $5' \to 3'$ Watson–Crick automata languages. Fundamenta Informaticae vol. 110(1–4) 2011, 111–123. p.

[2] László Hegedüs – Benedek Nagy: On Periodic Properties of Circular Words. Discrete Mathematics vol. 339(3). 2016, Elsevier, 1189–1197. p.

[3] László Hegedüs – Benedek Nagy – Ömer Eğecioğlu: Stateless multicounter $5' \to 3'$ Watson–Crick automata: the deterministic case. Natural Computing vol. 3. 2012, Springer Berlin Heidelberg, 361–368. p.

Nemzetközi konferenciakötetekben megjelent művek

[4] Ömer Eğecioğlu – László Hegedüs – Benedek Nagy: Stateless multicounter $5' \to 3'$ Watson–Crick automata. In *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications*, Bio-Inspired Computing: Theories and Applications. 2010, 1599–1606. p.

[5] László Hegedüs – Benedek Nagy: On string reading stateless multicounter $5' \to 3'$ Watson-Crick automata In Giancarlo Mauri et al. (eds.): *UCNC, Unconventional Computation and Natural Computation*, LNCS, Lecture Notes in Computer Science vol. 7956. 2013, 257–258. p.

[6] László Hegedüs – Benedek Nagy: Periodicity of circular words. In Juhani Karhumäki et al. (eds.): *Local Proceedings of WORDS 2013*, TUCS Lecture Notes vol. 20. 2013, Turku Centre for Computer Science, 45–56. p.

[7] László Hegedüs – Benedek Nagy: Representations of circular words. In *AFL, Automata and Formal Languages 2014*, EPTCS, Electronic Proceedings in Theoretical Computer Science vol. 151. 2014, 261–270. p.

[8] Benedek Nagy – László Hegedüs – Ömer Eğecioğlu: Hierarchy results on stateless multicounter $5' \to 3'$ Watson–Crick automata. In *Advances in Computational Intelligence*, International Work Conference on Artificial Neural Networks, LNCS, Lecture Notes in Computer Science vol. 6691. 2011, 465–472. p.

A dolgozatban nem hivatkozott saját közlemények

[9] Veronika Halász, László Hegedüs, István Hornyák, Benedek Nagy: Solving application oriented graph theoretical problems with DNA computing. In J. C. Bansal et al. (eds.): *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)* Advances in Intelligent Systems and Computing

(AISC) vol. 201, Springer. 2012, 75–85. p.

Egyéb konferenciaelőadások, absztraktok

[10] László Hegedüs: On Some Subsystems of Interval-Valued Logic. *CSCS - The Eighth Conference of PhD Students in Computer Science*, Szeged, Hungary, 2012, 21. p.

[11] László Hegedüs, Benedek Nagy: Periodicity of Circular Words. *CSCS - The Eighth Conference of PhD Students in Computer Science*, Szeged, Hungary, 2012, 22. p.

[12] Veronika Halász, László Hegedüs, István Hornyák, Benedek Nagy: Solving application oriented graph theoretical problems with DNA computing. *Turing Centenary Conference, Computability in Europe (CiE) - How the World Computes*, Cambridge, UK, 2012.

Irodalomjegyzék

[13] Leonard M. Adleman: Molecular computation of solutions to combinatorial problems. Science vol. 266(5187). 1021–1024. p.

[14] Francine Blanchet-Sadri: *Algorithmic Combinatorics on Partial Words*. Discrete Mathematics and Its Applications. 2008, Chapman & Hall/CRC, Taylor & Francis Group.

[15] Maxime Crochemore – Wojciech Rytter: *Jewels of Stringology: Text Algorithms*. Hackensack, NJ, USA, 2002, World Scientific.

[16] Elena Czeizler – Eugen Czeizler: A short survey on Watson–Crick automata. *Bulletin of the European Association for Theoretical Computer Science* 88., 2006, 104–119. p.

[17] James D. Currie – Daniel S. Fitzpatrick: Circular words avoiding patterns. In Proceedings of the 6th International Conference on Developments in Language Theory. LNCS 2450. Springer-Verlag, 2002, pp. 319–325.

[18] Volker Diekert – Tero Harju – Dirk Nowotka: Factorizations of cyclic words. In Workshop on Words and Automata at CSR, 7. 2006.

[19] Dömösi Pál – Falucskai János – Horváth Géza – Mecsei Zoltán – Nagy Benedek: *Formális nyelvek és automaták*. 2011, Debreceni Egyetem, jegyzet.

[20] Ömer Eğecioğlu – Oscar H. Ibarra: On stateless multicounter machi-

nes. In *Proceedings of 5th Conference on Computability in Europe 2009*, Lecture Notes in Computer Science vol. 5635. 2009, Springer Berlin Heidelberg, 178–187. p.

[21] Oscar H. Ibarra – Ömer Eğecioğlu: Hierarchies and characterizations of stateless multicounter machines. In *Proceedings of 15th International Computing and Combinatorics Conference*, International Computing and Combinatorics Conference, Lecture Notes in Computer Science vol. 5609. 2009, Springer Berlin Heidelberg, 408–417. p.

[22] Zvi Galil – Joel Seiferas: Recognizing certain repetitions and reversals within strings. In *17th Annual Symposium on Foundations of Computer Science*. 1976, IEEE, 236–252. p.

[23] Ferenc Gécseg – István Peák: *Algebraic theory of automata.* Disquisitiones Mathematicae Hungaricae. 1972, Akadémiai Kiadó.

[24] Markus Holzer – Martin Kutrib – Andreas Malcher: Multi-head finite automata: Characterizations, concepts and open problems. In *Electronic Proceedings in Theoretical Computer Science*, International Workshop on The Complexity of Simple Programs. 2009, 93–107. p.

[25] John E. Hopcroft – Rajeev Motwani – Jeffrey D. Ullman: Introduction to Automata Theory, Languages, and Computation, 3rd ed. 2006, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[26] Oscar H. Ibarra: On two-way multihead automata. *Journal of Computer and System Sciences* 7., 1973, 28–36. p.

[27] Johan Jeuring: The derivation of on-line algorithms, with an application to finding palindromes. *Algorithmica* 11., 1994, 146–184. p.

[28] Friedrich Wilhelm Levi: On semigroups. *Bulletin of the Calcutta Mathematical Society* 36., 1944, 141–146. p.

[29] Kiss János: Biológiai kislexikon. Typotex Elektronikus Kiadó Kft. 2007.

[30] Dietrich Kuske – Peter Weigel: The role of the complementarity relation in Watson-Crick automata and sticker systems. In Developments in Language Theory, DLT 2004 LNCS, Lecture Notes in Computer Science vol. 3340, 2004, 272–283. p.

[31] M. Lothaire: *Combinatorics on Words.* Cambridge Mathematical Library. 1997, Cambridge University Press (1983, Addison-Wesley).

[32] M. Lothaire: *Algebraic combinatorics on Words.* Encyclopedia of mathematics and its Applications vol. 90. 2002, Cambridge University Press.

[33] M. Lothaire: *Applied combinatorics on Words.* Encyclopedia of Mathematics and its Applications vol. 105. 2005, Cambridge University Press.

[34] Percy A. MacMahon: *Combinatory Analysis.* 1915, Cambridge University Press.

[35] Marvin L. Minsky: *Computation: Finite and Infinite Machines.* Upper Saddle River, NJ, USA, 1967, Prentice-Hall, Inc.

[36] Benedek Nagy: On $5' \to 3'$ sensing Watson–Crick finite automata. In *13th International Meeting on DNA Computing*, International Meeting on DNA Computing, Lecture Notes in Computer Science vol. 4848. 2008, Springer Berlin Heidelberg, 256–262. p.

[37] Benedek Nagy: On a hierarchy of $5' \to 3'$ sensing Watson-Crick finite automata languages. In *JLC, Journal of Logic and Computation* vol. 23. 2013, Oxford University Press, 855–872. p.

[38] Benedek Nagy: *DNS számítógépek és formális modelljeik.* 2014, Typo-

tex kiadó

[39] Rohit J. Parikh: On Context-Free Languages. JACM, Journal of the ACM vol. 13(4) New York, NY, USA, 1966, ACM, 570–581. p.

[40] Peák István: *Bevezetés az automaták elméletébe I.* ELTE tankönyv. 1977, Tankönyvkiadó Vállalat.

[41] Peák István: *Bevezetés az automaták elméletébe II.* ELTE tankönyv. 1978, Tankönyvkiadó Vállalat.

[42] Peák István: *Bevezetés az automaták elméletébe III.* ELTE tankönyv. 1980, Tankönyvkiadó Vállalat.

[43] Gheorghe Păun: From cells to computers: Computing with membranes (P systems). *Biosystems* 59., 2001, 139–158. p.

[44] Jeffrey Shallit: *A Second Course in Formal Languages and Automata Theory.* New York, NY, USA, 2008, Cambridge University Press.

[45] Arseny M. Shur: On ternary square-free circular words. The Electronic Journal of Combinatorics vol. 17, 2010.

[46] William F. Smyth: *Computing Patterns in Strings.* ACM Press. 2003, Pearson Addison-Wesley (UK).

[47] Axel Thue: Über unendliche Zeichenreihen. *Norske Vid. Skrifter I Mat.-Nat. Kl., Christiania*, 1906, 1–22. p.

[48] Axel Thue: Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Skrifter I Mat.-Nat. Kl., Christiania*, 1912, 1–67. p.

[49] James D. Watson – Francis H. C. Crick: A Structure for Deoxyribose Nucleic Acid. Nature vol. 171. 737–738. p.