UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
DEPARTMAN ZA
MATEMATIKU I INFORMATIKU

Marijana Čotrić

ĆELIJSKI AUTOMATI I L-SISTEMI

- master teza -

Mentor:
dr Rozália Sz. Madarász

Novi Sad, 2016.

Sadržaj

Predgovor

Glava 1

Ćelijski automati

Neformalno, ***ćelijski automat*** je diskretan dinamički sistem u obliku pravilne n-dimenzionalne rešetke ćelija, od kojih je svaka u tačno jednom od konačno mnogo stanja. Ćelije menjaju svoja stanja u zavisnosti od stanja određenih susednih ćelija i u skladu sa lokalnim pravilom ažuriranja. Sve ćelije menjaju svoja stanja istovremeno, koristeći isto pravilo ažuriranja. Postupak se ponavlja u diskretnim vremenskim koracima. Ispostavlja se da neverovatno jednostavna pravila ažuriranja mogu proizvesti ekstremno složene dinamike kada se primenjuju na ovakav način. Ćelijski automati su:

- diskretni u vremenu i prostoru,
- homogeni u vremenu i prostoru (isto pravilo ažuriranja za sve ćelije u svakom vremenskom trenutku),
- lokalni u svojim interakcijama.

Mnogi procesi u prirodi se odvijaju pod lokalnim i homogenim pravilima, što ih čini primamljivim za modeliranje i simuliranje pomoću ćelijskih automata. Ćelijski automati su takođe i matematički modeli za masivno paralelno računanje. Jednostavna pravila ažuriranja mogu dati ćelijski automat koji je računski univerzalan, odnosno sposoban za obavljanje proizvoljnih računskih zadataka.

1.1 Osnovne definicije

U ovom odeljku dajemo najosnovnije definicije i oznake teorije ćelijskih automata. Ćelijski automat ćemo označavati skraćeno sa CA.

Definicija 1.1 *Neka je $d \in \mathbb{N}$, d**-dimenzionalan ćelijski automat** (CA) je uređena četvorka $A = (\mathbb{Z}^d, S, N, f)$ gde je*

> $\mathbb{Z}^d$ *– d-dimenzionalan prostor koji nazivamo **ćelijski prostor**, a njegove elemente nazivamo **ćelije**,*
>
> S *– konačan skup koji nazivamo **skup stanja**,*
>
> N *– uređena m-torka*
> $$N = (\vec{n}_1, \vec{n}_2, \dots, \vec{n}_m)$$

kod koje je $\vec{n}_i \in \mathbb{Z}^d$ i $\vec{n}_i \neq \vec{n}_j$ za sve $i \neq j$. Nazivamo je d***-dimenzionalan vektor susedstva*** *(veličine m). Elementi $\vec{n}_i$ određuju lokacije suseda svake ćelije: ćelija $\vec{n} \in \mathbb{Z}^d$ ima m suseda $\vec{n} + \vec{n}_i$, za $i = 1, 2, \ldots, m$.*

f – funkcija $f : S^m \longrightarrow S$ koja se istovremeno pimenjuje na sve ćelije $\vec{n} \in \mathbb{Z}^d$ i svakoj od njih dodeljuje novo stanje na osnovu stanja njenih suseda. Ako susedi ćelije imaju stanja $s_1, s_2, \ldots, s_m$ onda će novo stanje ćelije biti $f(s_1, s_2, \ldots, s_m)$. Nazivamo je ***lokalno pravilo ažuriranja****.*

Definicija 1.2 *Neka je dat d-dimenzionalan CA $A = (\mathbb{Z}^d, S, N, f)$. Tada se funkcija*

$$c : \mathbb{Z}^d \longrightarrow S$$

koja svakoj ćeliji dodeljuje stanje, naziva ***konfiguracija*** d***-dimenzionalnog CA****. Stanje ćelije $\vec{n} \in \mathbb{Z}^d$ je $c(\vec{n})$.*

Konfiguracija c nam zapravo daje stanja svih ćelija prostora $\mathbb{Z}^d$ u nekom vremenskom trenutku. Skup svih funkcija f koje elementima skupa A dodeljuju elemente skupa B označavamo sa B^A, stoga je $S^{\mathbb{Z}^d}$ skup svih konfiguracija. Sada je npr. u jednodimenzionalnom slučaju, $d = 1$, skup svih konfiguracija $S^{\mathbb{Z}} = \{f | f : \mathbb{Z} \longrightarrow S\}$. Kako se na sve ćelije CA primenjuje isto pravilo ažuriranja istovremeno, dolazi do globalne promene konfiguracije. Konfiguracija c se menja u konfiguraciju c' gde za sve $\vec{n} \in \mathbb{Z}^d$

$$c'(\vec{n}) = f[c(\vec{n} + \vec{n}_1), c(\vec{n} + \vec{n}_2), \ldots, c(\vec{n} + \vec{n}_m)].$$

Definicija 1.3 *Neka je $A = (\mathbb{Z}^d, S, N, f)$ d-dimenzionalan CA. Transformacija $c \mapsto c'$ naziva se* ***globalna funkcija prelaska*** *d-dimenzionalnog CA. To je funkcija*

$$G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$$

koja je iterirana, odnosno primenjuje se iznova i iznova. Stoga primenom funkcije G dobijamo ***vremensku evoluciju sistema***

$$c_0 \mapsto G(c_0) \mapsto G^2(c_0) \mapsto G^3(c_0) \mapsto \ldots,$$

gde je c_0 ***početna konfiguracija****. Niz $orb(c_0) = c_0, G(c_0), G^2(c_0), \ldots$ naziva se* ***orbita*** *početne konfiguracije c_0.*

Ovde se vreme odnosi na broj primena funkcije G, odnosno svaka primena funkcije G traje jedan vremenski trenutak, pa je $G^t(c_0)$ konfiguracija u trenutku t, za sve $t = 0, 1, 2 \ldots$. Postoje još i ***dvosmerne beskonačne orbite***, tj. nizovi konfiguracija oblika

$$\ldots, c_{-2}, c_{-1}, c_0, c_1, c_2, \ldots,$$

gde je $G(c_i) = c_{i+1}$ za sve $i \in \mathbb{Z}$. Kod njih vreme t „teče" kroz čitav skup celih brojeva i nema početne konfiguracije. Globalnu funkciju prelaska ćelijskog automata A označavamo sa $G[A]$, ili jednostavnije G kada je iz konteksta jasno da je to funkcija prelaska ćelijskog automata A. Svaka funkcija G koja je funkcija prelaska nekog CA naziva se ***CA funkcija***. Za CA funkcije važi sledeće tvrđenje koje navodimo bez dokaza.

Teorema 1.4 *Ako su G i H CA funkcije, onda je i njihova kompozicija $G \circ H$ CA funkcija.*

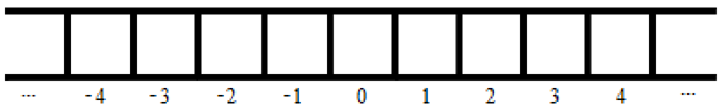
Definicija 1.5 *Za ćelijske automate A i B kažemo da su **ekvivalentni** ako imaju istu funkciju prelaska, odnosno ako važi G[A]=G[B].*

Jasno, ekvivalentni CA imaju isti ćelijski prostor $\mathbb{Z}^d$ i skup stanja S, ali mogu imati različite vektore susedstva, pa samim tim i različita lokalna pravila ažuriranja. U sledećem odeljku ćemo pokazati da među ekvivalentnim CA postoji jedinstven CA čiji je vektor susedstva minimalan u smislu da se susedstvo svake ćelije $\vec{n} \in \mathbb{Z}^d$ određeno minimalnim vektorom susedstva sadrži u susedstvima te ćelije, određenim vektorima susedstva svih ostalih ekvivalentnih CA. Ostali ekvivalentni CA imaju dodatne susede koji zapravo nemaju nikakav uticaj na sledeće stanje ćelije. Zbog toga te susede nazivamo „lažnim”.

Primer 1.6 ***(xor CA)*** *Dat je jednodimenzionalan CA čiji je skup stanja* $S = \{0, 1\}$*, vektor susedstva* $N = (0, 1)$ *i pravilo ažuriranja* $f : \{0, 1\}^2 \longrightarrow \{0, 1\}$ *zadato na sledeći način*

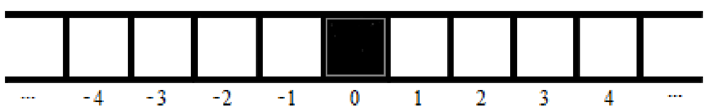
$$f(a, b) = a + b \ (mod\ 2).$$

Njegov ćelijski prostor $\mathbb{Z}$ *možemo slikovito predstaviti kao beskonačnu „**traku**” podeljenu na kvadrate i indeksiranu celim brojevima kao na slici 1.1, gde svaki kvadrat predstavlja jednu ćeliju. Novo stanje ćelije će biti ostatak pri deljenju zbira stanja ćelije i stanja njenog desnog suseda sa dva. Lokalno pravilo ažuriranja ovog CA je zapravo logička operacija „**izričita disjunkcija**” (*$\veebar$*).*



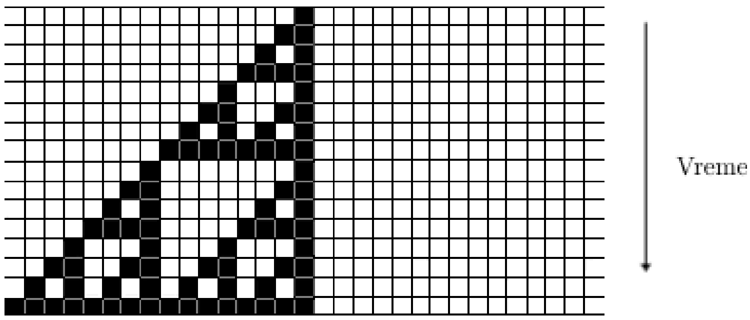
Slika 1.1: Ćelije jednodimenzionalnog CA.

Razmatraćemo, na primer, početnu konfiguraciju c_0 *kod koje je* $c_0(0) = 1$ *i* $c_0(i) = 0$ *za sve* $i \neq 0$*, odnosno samo ćelija sa indeksom 0 ima stanje 1. Stanju 0 pridružujemo **belu**, a stanju 1 **crnu boju**. Sada možemo dati slikoviti prikaz svake konfiguracije c ovog CA ako na traci koja predstavlja ćelijski prostor* $\mathbb{Z}$ *svaku ćeliju (kvadrat) čije je stanje 0 obojimo belom, a svaku ćeliju čije je stanje 1 obojimo crnom bojom. Ovaj prikaz nazivamo **traka boja konfiguracije c**. Na slici 1.2 predstavljena je početna konfiguracija* c_0*.*



Slika 1.2: Traka boja početne konfiguracije c_0.

U sledećem vremenskom trenutku imaćemo konfiguraciju $c_1 = G(c_0)$ *kod koje je* $c_1(0) = c_1(-1) = 1$ *i* $c_1(i) = 0$ *za sve* $i \neq -1, 0$*. Nastavljajući ovako, dobijamo* *vremensku evoluciju $c_2 = G(c_1)$, $c_3 = G(c_2)$ itd. Na slici 1.3 dat je dijagram u kojem je predstavljeno prvih 16 konfiguracija ove evolucije. Svaka od tih konfiguracije predstavljena je svojom trakom boja. Najviša traka predstavlja početnu konfiguraciju c_0, a sledeće predstavljaju uzastopne elemente orbite $orb(c_0)$ pa možemo reći da vreme na ovom dijagramu teče na dole.*



Slika 1.3: Prostorno-vremenski dijagram *xor* CA.

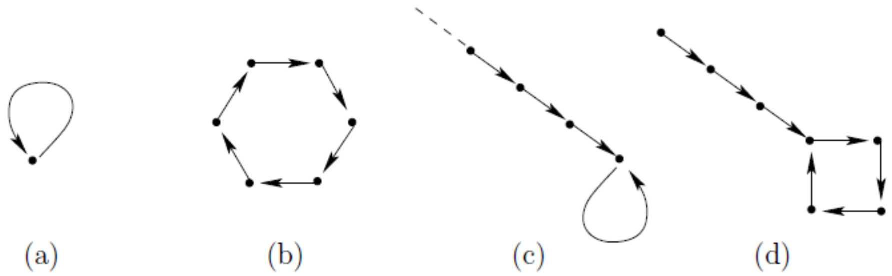
Prostorno-vremenski dijagram je slikoviti prikaz orbite. U slučaju jednodimenzionalnog CA, svakom stanju iz skupa S pridružujemo jednu boju, pri čemu nema ponavljanja boja, i konfiguracije predstavljamo odgovarajućim trakama boja slično kao u prethodnom primeru. Traka konfiguracije $G(c)$ crta se ispod trake konfiguracije c tako da vreme teče na dole. Najviša traka predstavlja početnu konfiguraciju. Stoga prostorno-vremenski dijagram orbite konfiguracije c ispunjava samo donju poluravan. Sa druge strane, prostorno-vremenski dijagram dvosmerne beskonačne orbite ispunjava čitavu ravan jer nema početne konfiguracije (početnog trenutka).

Generalno, prostorno-vremenski dijagram d-dimenzionalnog CA je $(d+1)$-dimenzionalni ,,crtež" gde d predstavlja dimenziju prostora, a dodatna dimenzija se koristi za vreme. Vreme uzima vrednosti iz $\mathbb{N}$ ili $\mathbb{Z}$ u zavisnosti od toga da li je u pitanju dijagram orbite početne konfiguracije ili dijagram dvosmerne beskonačne orbite. U prvom slučaju dijagram je element od $S^{\mathbb{Z}^d \times \mathbb{N}}$, a u drugom od $S^{\mathbb{Z}^d \times \mathbb{Z}}$.

Za konfiguraciju c kažemo da je

- ***fiksna tačka*** funkcije prelaska G ako je $G(c) = c$,
- ***periodična*** ako je $G^t(c) = c$ za neko $t \in \mathbb{Z}^+$. Svako t koje zadovoljava jednakost $G^t(c) = c$ naziva se period od c, a najmanje takvo t naziva se *najmanji period od* c,
- ***na kraju fiksna tačka*** ako postoji $n \in \mathbb{N}$ takvo da je $G^{n+1}(c) = G^n(c)$, tj. $G^n(c)$ je fiksna tačka za neko $n \in \mathbb{N}$,
- ***na kraju periodična*** ako postoje $n \in \mathbb{N}$ i $t \in \mathbb{Z}^+$ takvi da važi $G^{n+t}(c) = G^n(c)$, tj. $G^n(c)$ je periodična za to n.

Ista terminologija se koristi i za orbite. Za jednosmernu (dvosmernu) beskonačnu orbitu kazemo da je fiksna ako su sve konfiguracije koje sadrži fiksne tačke, periodična ako sadrži samo periodične konfiguracije, na kraju periodična ako sadrži neku konfiguraciju koja je fiksna tačka i na kraju periodična ako sadrzi neku periodičnu konfiguraciju. Vidi sliku 1.4 za ilustraciju ovih pojmova na dvosmernim beskonačnim orbitama. Slika prikazuje delove faznih prostora nekih CA. ***Fazni prostor*** je beskonačni usmereni graf čiji su čvorovi konfiguracije i za svaku konfiguraciju c postoji tačno jedna izlazna grana koja vodi ka $G(c)$. Treba imati na umu da fazni prostor ima beskonačno mnogo čvorova, pa se na slikama uvek može prikazati samo jedan njegov deo, npr. deo koji predstavlja neku orbitu, kao što je to slučaj na slici 1.4.



Slika 1.4: (a) fiksna orbita, (b) periodična orbita, (c) na kraju fiksna orbita, (d) na kraju periodična orbita.

1.2 Susedstva

Definicija 1.7 *Neka je* $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$ *d-dimenzionalan vektor susedstva CA* $A = (\mathbb{Z}^d, S, N, f)$.
(a) Funkcija $\widetilde{N} : \mathbb{Z}^d \longrightarrow \mathbb{Z}^{d^m}$ *definisana sa*

$$\widetilde{N}(\vec{n}) = (\vec{n} + \vec{n}_1, \vec{n} + \vec{n}_2, \ldots, \vec{n} + \vec{n}_m) \text{ za svako } \vec{n} \in \mathbb{Z}^d,$$

naziva se ***uređeno susedstvo CA****.*
(b) Neka je $B = \{\{\vec{n}\} \mid \vec{n} \in \mathbb{Z}^d\}$*. Preslikavanje* $\widehat{N} : B \longrightarrow \mathcal{P}(\mathbb{Z}^d)$ *definisano sa*

$$\widehat{N}(\{\vec{n}\}) = \{\vec{n} + \vec{n}_1, \vec{n} + \vec{n}_2, \ldots, \vec{n} + \vec{n}_m\}$$

naziva se ***neuređeno susedstvo CA****.*

Dakle, funkcija $\widetilde{N}$ ($\widehat{N}$) za svaku ćeliju $n \in \mathbb{Z}^d$ daje njeno uređeno (neuređeno) susedstvo.

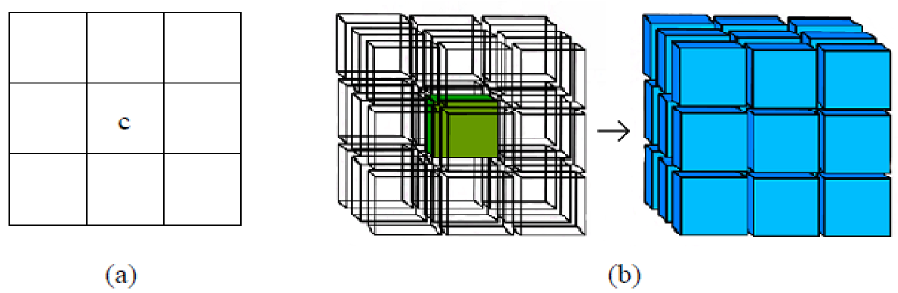
Primetimo da vektor susedstva svakog CA u potpunosti određuje njegovo uređeno i neuređeno susedstvo. Lako zaključujemo da je $N = \widetilde{N}(\vec{0})$ kao i da je $\widehat{N}(\{\vec{0}\})$ neuređen skup koji sadrži elemente vektora susedstva N. Redosled elemenata vektora susedstva N je u suštini nebitan, važan je samo kao poredak u kojem je dato m ulaznih vrednosti u lokalnom pravilu ažuriranja $f : S^m \longrightarrow S$. Stoga je kod navođenja CA dovoljno dati neuređenu verziju $\widehat{N}(\{\vec{0}\})$ vektora susedstva N, dok god je uloga različitih elemenata vektora susedstva jasna iz opisa lokalnog pravila ažuriranja, takodje je u tom slučaju dovoljno razmatrati samo neuređeno susedstvo CA.

U nastavku dajemo definicije najpoznatijih susedstava CA.

Definicija 1.8 *Neuređeno susedstvo d-dimenzionalnog CA, za čiji je vektor susedstva N ispunjeno*

$$\widehat{N}(\{\vec{0}\}) = \{(k_1, k_2 \ldots, k_d) \in \mathbb{Z}^d \mid |k_i| \le r \text{ za sve } i = 1, 2, \ldots, d\}$$

gde je $r \in \mathbb{N}_0$, naziva se ***Moore–susedstvo***[1] ***poluprečnika r*** *(ili samo susedstvo poluprečnika r) i označava sa M_r^d.*

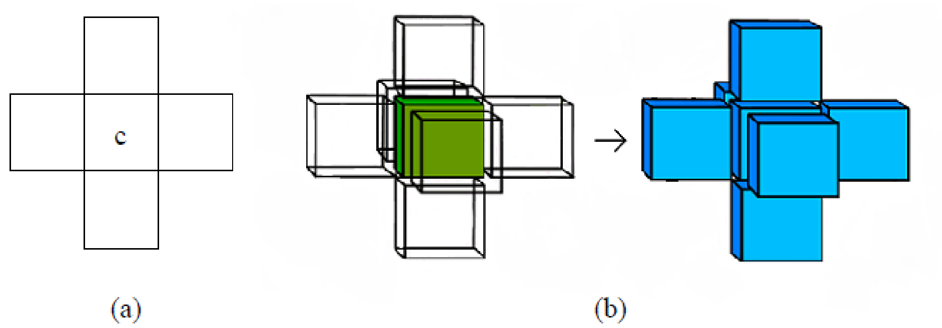


Slika 1.5: Moore–susedstvo poluprečnika $r = 1$: (a) ćelije c dvodimenzionalnog CA, (b) ćelije trodimenzionalnog CA. Ćelija je prikazana levo zelenom bojom, a njeno susedstvo desno, plavom bojom.

Definicija 1.9 *Neuređeno susedstvo d-dimenzionalnog CA za čiji je vektor susedstva N ispunjeno*

$$\widehat{N}(\{\vec{0}\}) = \{(k_1, k_2 \ldots, k_d) \in \mathbb{Z}^d \mid \sum_{i=1}^{d} |k_i| \le r\}$$

gde je $r \in \mathbb{N}_0$, naziva se ***von Neumann–susedstvo***[2] ***poluprečnika r*** *i označava sa N_r^d.*



Slika 1.6: von Neumann–susedstvo poluprečnika $r = 1$: (a) ćelije c dvodimenzionalnog CA, (b) ćelije trodimenzionalnog CA. Ćelija je prikazana levo zelenom bojom, a njeno susedstvo desno, plavom bojom.

[1] Osmislio ga je Edward Forrest Moore (1925-2003), američki profesor matematike i informatike.

[2] Susedstvo je osmislio John von Neumann (1903-1957), mađarsko-američki matematičar i naučnik.

Definicija 1.10 *Neuređeno susedstvo d-dimenzionalnog CA za čiji je vektor susedstva N ispunjeno*

$$\widehat{N}(\{\vec{0}\}) = \{(k_1, k_2 \ldots, k_d) \in \mathbb{Z}^d \mid \; k_i \in \{0,1\} \textit{ za sve } i = 1, 2, \ldots, d\}$$

*naziva se **susedstvo poluprečnika** $\frac{1}{2}$.*

Primetimo da CA xor iz primera 1.6 ima susedstvo poluprečnika $\frac{1}{2}$.

Definicija 1.11 *Neka je dat CA čiji je vektor susedstva $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$ i lokalno pravilo ažuriranja $f : S^m \longrightarrow S$. Element $\vec{n}_j$ vektora susedstva naziva se **lažni element** ako je $f(s_1, s_2, \ldots, s_m) = f(t_1, t_2, \ldots, t_m)$ kad god je $s_i = t_i$ za sve $i \neq j$.*

Ako je element $\vec{n}_j$ vektora susedstva N lažni onda iz ove definicije lako zaključujemo da za sve ćelije $\vec{n} \in \mathbb{Z}^d$ važi da stanje njihovog j-og suseda $\vec{n} + \vec{n}_j$ ne utiče na njihovo stanje u sledećem vremenskom trenutku. Stoga $\vec{n}_j$ možemo ukloniti iz vektora susedstva N. Dobijamo CA ekvivalentan polaznom, čiji vektor susedstva ima $m - 1$ elemenata.

Definicija 1.12 *Vektor susedstva $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$ koji nema lažnih suseda naziva se **minimalan vektor susedstva**.*

Uklanjanjem svih lažnih suseda iz vektora susedstva nekog CA dobijamo njemu ekvivalentan CA koji ima minimalan vektor susedstva. Minimalan vektor susedstva CA je jedinstven:

Teorema 1.13 *Ako su A i B ekvivalentni CA koji imaju minimalne vektore susedstva onda je $A = B$ (do na redosled elemenata u vektoru susedstva).*

Dokaz. Dovoljno je dokazati da vektori susedstva od A i B sadrže iste elemente. Neka je $\vec{n}$ proizvoljan element vektora susedstva od A, odnosno ćelija $\vec{n}$ je sused ćelije $\vec{0}$ u A. Pošto A nema lažnih suseda postoje dve konfiguracije c i e takve da važi $c(\vec{n}) \neq e(\vec{n})$, $c(\vec{k}) = e(\vec{k})$ za sve $\vec{k} \neq \vec{n}$ i $c'(\vec{0}) \neq e'(\vec{0})$ gde je $c' = G(c)$ i $e' = G(e)$ i G je globalna funkcija prelaska od A. Kako su A i B ekvivalentni, G je funkcija prelaska i od B. Odavde sledi da $\vec{n}$ mora biti sused od $\vec{0}$ i u B, jer bismo u suprotnom imali $c'(\vec{0}) = e'(\vec{0})$.

Analogno, svaki element vektora susedstva od B je element vektora susedstva od A. □

1.3 Elementarni CA

Definicija 1.14 *Jednodimenzionalan CA kod koga je $S = \{0,1\}$, $N = (-1, 0, 1)$ i lokalno pravilo ažuriranja $f : S^3 \longrightarrow S$ naziva se **elementarni CA**.*

Iz definicije zaključujemo da svi elementarni CA imaju isti skup stanja, isto Moore–susedstvo poluprečnika $r = 1$, kao i da se međusobno razlikuju samo po lokalnom pravilu ažuriranja f. Lokalna pravila ažuriranja elementarnih CA nazivamo još i ***elementarna pravila***. Postoji 256 različitih elementarnih CA, jer različitih elementarnih pravila $S^3 \longrightarrow S$ ima $2^8 = 256$. Osamdesetih godina prošlog veka naučnik ***Stephen Wolfram***[3] intenzivno je proučavao elementarne

[3]Britanski naučnik rođen 1959. godine. Poznat po brojnim rezultatima u informatici, matematici i teorijskoj fizici.

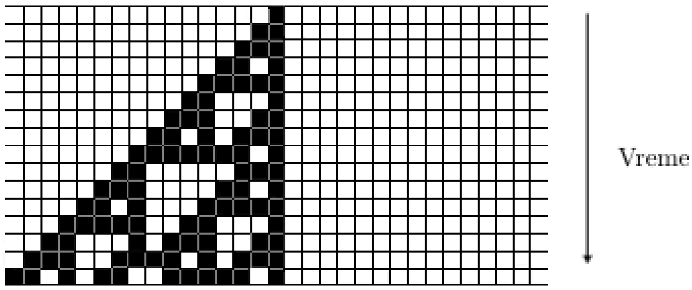
CA i njihova elementarna pravila. Jedan od najvećih rezultata tog njegovog rada je ***šema imenovanja elementarnih pravila***: Svako elementarno pravilo f je određeno osmobitnom sekvencom

$$f(111)f(110)f(101)f(100)f(011)f(010)f(001)f(000)$$

koja je uvek binarni zapis nekog celog broja iz intervala $[0, 255]$, pri čemu različitim elementarnim pravilima odgovaraju različiti celi brojevi iz ovog intervala. Ceo broj iz intervala $[0, 255]$ koji odgovara elementarnom pravilu datog elementarnog CA je njegov ***Wolfram-ov broj***.

Primer 1.15 ***(pravilo 110)*** *Osmobitni binarni zapis broja 110 je 01101110 pa elementarni CA sa Wolfram-ovim brojem 110 ima lokalno pravilo ažuriranja*

$$f(111) = 0, \quad f(110) = 1, \quad f(101) = 1, \quad f(100) = 0,$$
$$f(011) = 1, \quad f(010) = 1, \quad f(001) = 1, \quad f(000) = 0.$$



Slika 1.7: Prostorno-vremenski dijagram CA sa Wolfram-ovim brojem 110. Prvih 16 konfiguracija orbite početne konfiguracije c_0.

Stephen Wolfram i njegov asistent Matthew Cook[4] *su 2004. godine dokazali da je ovaj CA* ***računski univerzalan****. Na slici 1.7 dat je njegov prostorno-vremenski dijagram za početnu konfiguraciju c_0 kod koje je $c_0(0) = 1$ i $c_0(i) = 0$ za sve $i \neq 0$.*

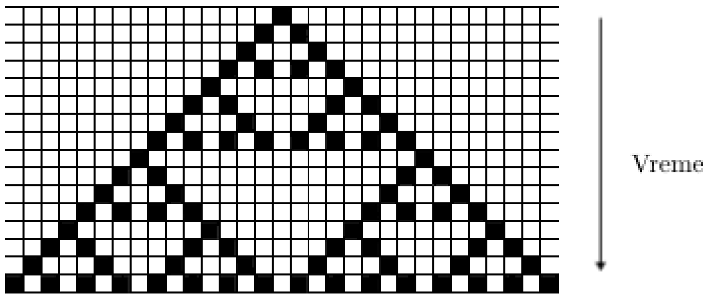
Primer 1.16 ***(pravilo 90)*** *Broj 90 ima osmobitni binarni zapis 01011010 pa elementarni CA čiji je Wolfram-ov broj 90 ima pravilo ažuriranja*

$$f(111) = 0, \quad f(110) = 1, \quad f(101) = 0, \quad f(100) = 1,$$
$$f(011) = 1, \quad f(010) = 0, \quad f(001) = 1, \quad f(000) = 0.$$

Ovim pravilom, za početnu konfiguraciju c_0 istu kao u prethodnom primeru, dobijamo ***trougao Sierpinski-og****, koji se pojavio u Italijanskoj umetnosti još u trinaestom veku, a opisao ga je Sierpinski*[5] *1915. godine. Ovaj fraktal se danas koristi u računarskoj grafici za grafičko predstavljanje planina i drugih terena. Na slici 1.8 dat je dijagram na kojem je predstavljeno prvih 16 konfiguracija orbite početne konfiguracije c_0, dobijene primenom ovog pravila.*

[4]Američki naučnik rođen 1970. godine.

[5]Waclaw Franciszek Sierpinski (1882-1969), poljski matematičar.



Slika 1.8: Prostorno-vremenski dijagram CA sa Wolfram-ovim brojem 90.

Neformalno, ***fraktal*** je geometrijski lik koji se može razložiti na manje delove tako da je svaki od njih, makar približno, umanjena kopija celine. Kaže se da je takav lik sam sebi sličan. Termin je izveo ***Benoit Mandelbrot***[6] 1975. godine i potiče od latinske reči ***fractus*** što znači slomljen. Pored toga što su izlomljeni, za fraktale je karakteristično da se isti oblik stalno ponavlja. Ako se neki deo fraktala uveća izgledaće kao ceo fraktal. Fraktal često ima sledeće osobine:

- finu strukturu na proizvoljno malom uvećanju,
- previše je nepravilan da bi mogao biti opisan klasičnim euklidskim jezikom,
- sam je sebi sličan,
- Hauzdorfovu dimenziju koja je veća od njegove topološke dimenzije,
- jednostavnu i rekurzivnu definiciju.

Fraktali su svuda oko nas. Ne samo u obliku i izgledu stvari koje nas okružuju, već i u samoj srži fenomena, u funkcijama koje opisuju jednostavnije i kompleksnije sisteme i procese. Naravno, umetnost ih iskorištava do krajnjih granica, a našli su primenu i u kinematografiji u izradi specijalnih efekata. Prirodni oblici koji aproksimiraju fraktale do izvesne granice su oblaci, planine, munje, morske obale, pahulje snega, ali i neke biljke i životinje.

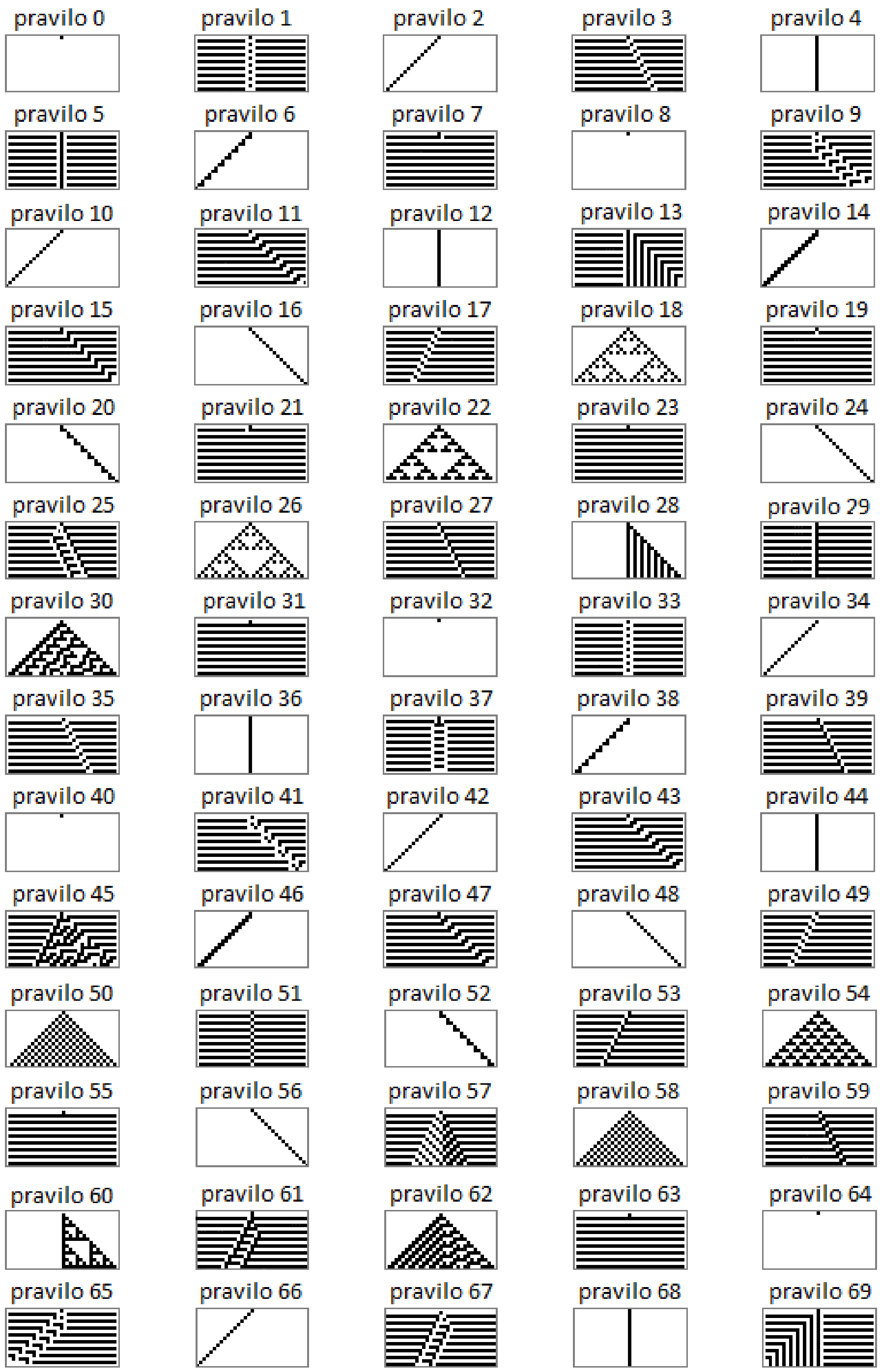
Fraktal trougao Sierpinski-og se dobija tako što se krene od punog trougla kome se ,,iseče" središnji trougao (trougao čija su temena središta stranica početnog trougla). Zatim se ova procedura nastavi sa novodobijena tri trougla, a zatim sa novodobijenih devet i tako dalje u beskonačnost, kao sto je prikazano na sledećoj slici.

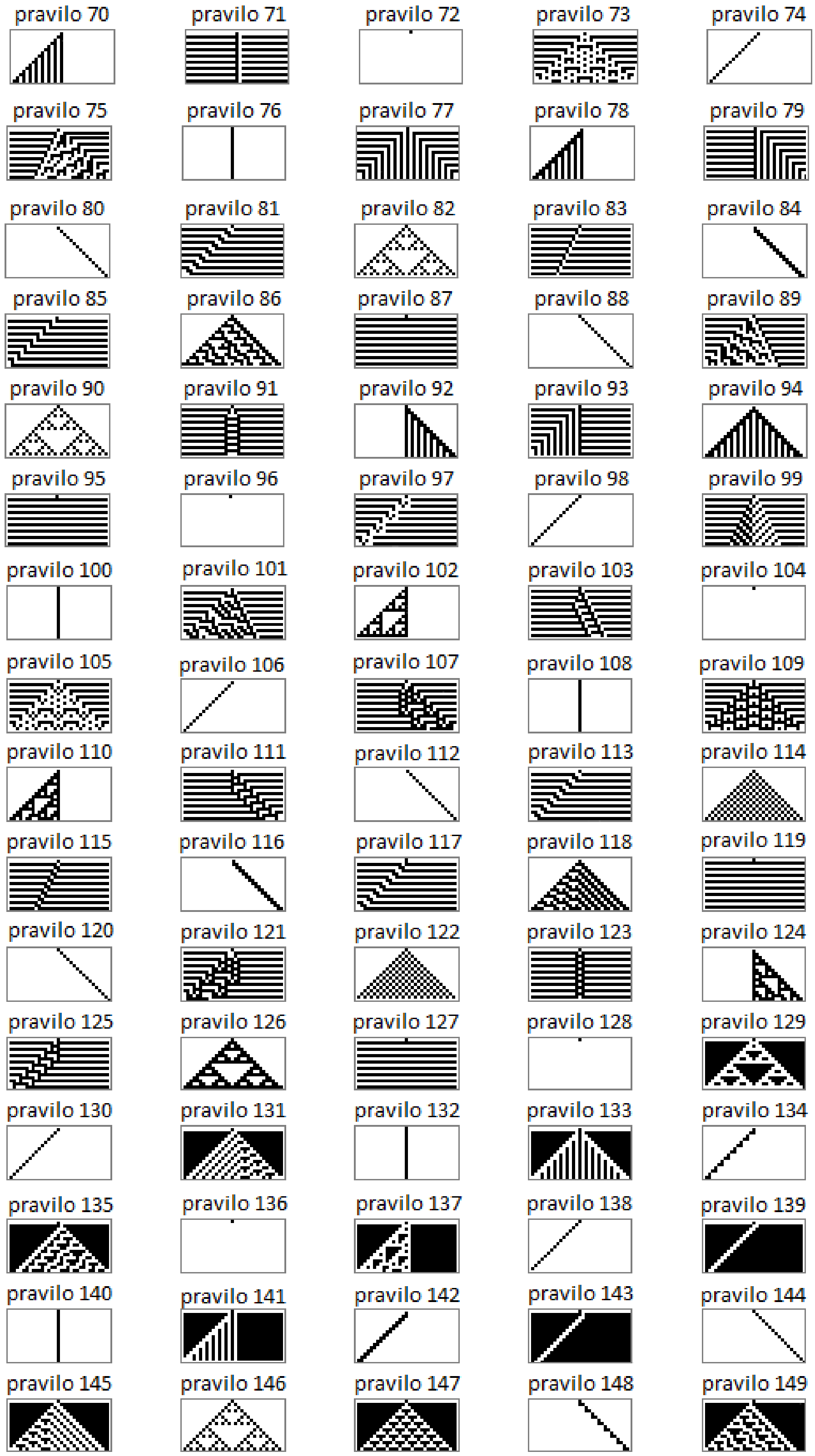
Slika 1.9: Prvih pet koraka evolucije trougla Sierpinski-og.

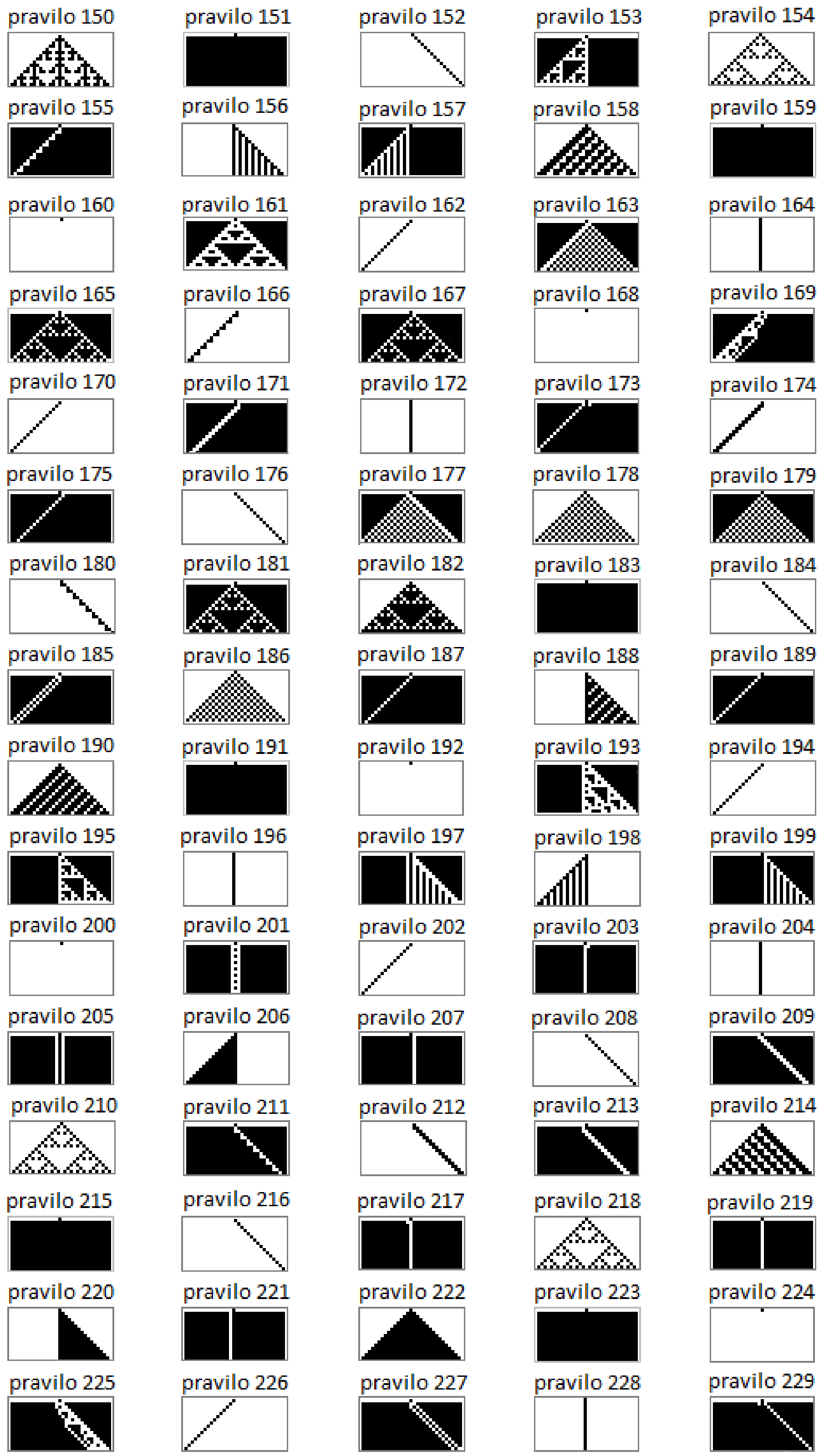
[6]Francuski matematičar (1924-2010). ,,Otac" fraktalne geometrije.

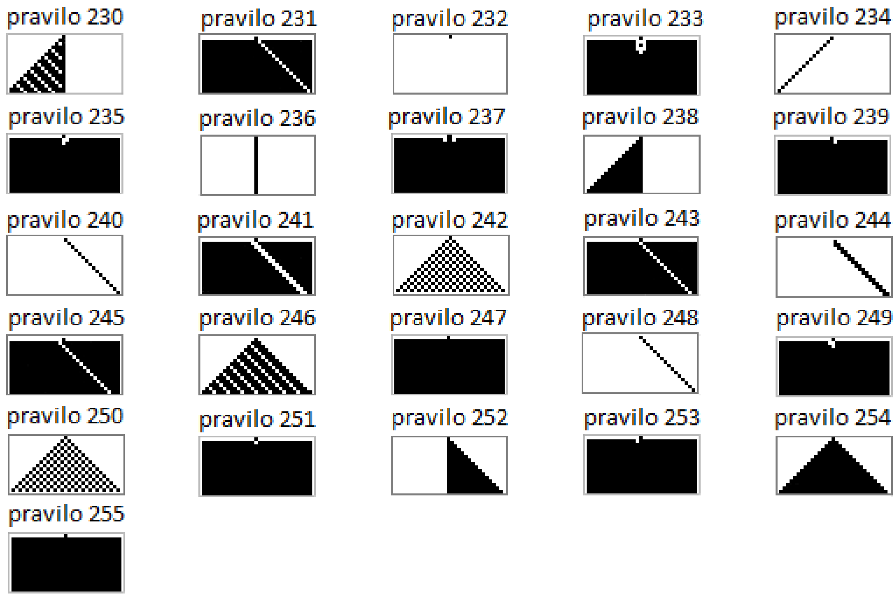
Radoznalom čitaocu koji želi detaljnije da se upozna sa fraktalnom geometrijom preporučujemo [15] i [16].

U nastavku dajemo prostorno-vremenske dijagrame svih elemetarnih CA za početnu konfiguraciju istu kao u prethodna dva primera.




pravilo 70
pravilo 71
pravilo 72
pravilo 73
pravilo 74
pravilo 75
pravilo 76
pravilo 77
pravilo 78
pravilo 79
pravilo 80
pravilo 81
pravilo 82
pravilo 83
pravilo 84
pravilo 85
pravilo 86
pravilo 87
pravilo 88
pravilo 89
pravilo 90
pravilo 91
pravilo 92
pravilo 93
pravilo 94
pravilo 95
pravilo 96
pravilo 97
pravilo 98
pravilo 99
pravilo 100
pravilo 101
pravilo 102
pravilo 103
pravilo 104
pravilo 105
pravilo 106
pravilo 107
pravilo 108
pravilo 109
pravilo 110
pravilo 111
pravilo 112
pravilo 113
pravilo 114
pravilo 115
pravilo 116
pravilo 117
pravilo 118
pravilo 119
pravilo 120
pravilo 121
pravilo 122
pravilo 123
pravilo 124
pravilo 125
pravilo 126
pravilo 127
pravilo 128
pravilo 129
pravilo 130
pravilo 131
pravilo 132
pravilo 133
pravilo 134
pravilo 135
pravilo 136
pravilo 137
pravilo 138
pravilo 139
pravilo 140
pravilo 141
pravilo 142
pravilo 143
pravilo 144
pravilo 145
pravilo 146
pravilo 147
pravilo 148
pravilo 149


pravilo 150
pravilo 151
pravilo 152
pravilo 153
pravilo 154
pravilo 155
pravilo 156
pravilo 157
pravilo 158
pravilo 159
pravilo 160
pravilo 161
pravilo 162
pravilo 163
pravilo 164
pravilo 165
pravilo 166
pravilo 167
pravilo 168
pravilo 169
pravilo 170
pravilo 171
pravilo 172
pravilo 173
pravilo 174
pravilo 175
pravilo 176
pravilo 177
pravilo 178
pravilo 179
pravilo 180
pravilo 181
pravilo 182
pravilo 183
pravilo 184
pravilo 185
pravilo 186
pravilo 187
pravilo 188
pravilo 189
pravilo 190
pravilo 191
pravilo 192
pravilo 193
pravilo 194
pravilo 195
pravilo 196
pravilo 197
pravilo 198
pravilo 199
pravilo 200
pravilo 201
pravilo 202
pravilo 203
pravilo 204
pravilo 205
pravilo 206
pravilo 207
pravilo 208
pravilo 209
pravilo 210
pravilo 211
pravilo 212
pravilo 213
pravilo 214
pravilo 215
pravilo 216
pravilo 217
pravilo 218
pravilo 219
pravilo 220
pravilo 221
pravilo 222
pravilo 223
pravilo 224
pravilo 225
pravilo 226
pravilo 227
pravilo 228
pravilo 229



Slika 1.10: Dijagrami elementarnih CA.

Treba imati na umu da su neka pravila ažuriranja elementarnih CA ekvivalentna do na neku od geometrijskih transformacija ili zamenu uloga stanja 0 i 1 u definiciji pravila ažuriranja, pa su ekvivalentna u računskom smislu. Od 256 pravila ažuriranja suštinski neekvivalentnih pravila ima ukupno 88. Tako je npr. elementarno pravilo 110 ekvivalentno do na osnu simetriju sa pravilom 124, dok je sa pravilom 137 ekvivalentno do na zamenu uloga stanja 0 i 1 u definiciji pravila ažuriranja (videti i prostorno-vremenske dijagrame ovih pravila na slici 1.10). Elementarna pravila ekvivalentna do na osnu simetriju nazivaju se ***osnosimetrična pravila***, a pravila ekvivalentna do na zamenu uloga stanja 0 i 1 ***komplementarna pravila***. Takođe, na neko elementarno pravilo mogu se sukcesivno primeniti prethodne dve transformacije čime ćemo dobiti njemu ***osmosimetrično komplementarno pravilo***. Osmosimetrično komplementarno pravilo za pravilo 110 je pravilo 193. Elementarnih pravila koja su sama sebi osnosimetrična ima 64 i to su: 0, 1, 4, 5, 18, 19, 22, 23, 32, 33, 36, 37, 50, 51, 54, 55, 72, 73, 76, 77, 90, 91, 94, 95, 104, 105, 108, 109, 122, 123, 126, 127, 128, 129, 132, 133, 146, 147, 150, 151, 160, 161, 164, 165, 178, 179, 182, 183, 200, 201, 204, 205, 218, 219, 222, 223, 232, 233, 236, 237, 250, 251, 254, i 255.

Wolfram je na osnovu empirijskog posmatranja ponašanja elementarnih CA za slučajne početne konfiguracije, izvršio njihovu podelu u četiri klase, koje nazivamo ***Wolframove klase CA***. On je klase definisao na sledeći način:

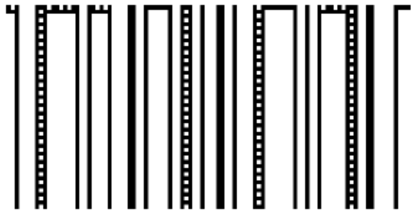
(W1) skoro sve početne konfiguracije vode do iste jedinstvene fiksna tačka konfiguracije,

(W2) skoro sve početne konfiguracije vode do konfiguracije koja se periodično ponavlja,

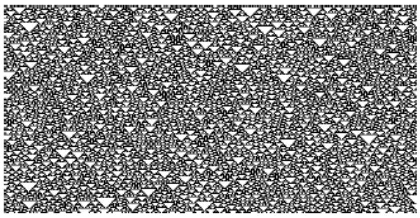
(W3) skoro sve početne konfiguracije vode do različitih ponašanja, *(W4)* skoro sve početne konfiguracije vode do složenih ponašanja, ponekad dugovečnih. Pojavljuju se strukture čije su međusobne interakcije jako složene.

Ova klasifikacija je kasnije proširena na sve jednodimenzionalne i dvodimenzionalne CA. Napominjemo da ove definicije klasa nisu matematički precizne i da postoje i druge klasifikacije elementarnih CA koje su preciznije. Na slici 1.11 dati su primeri tipičnih prostorno-vremenskih dijagrama CA iz svake od klasa. Wolfram pretpostavlja da su CA iz klase W4 računski univerzalni, mada je do danas računska univerzalnost dokazana samo za elementarni CA sa pravilom 110 iz ove klase.

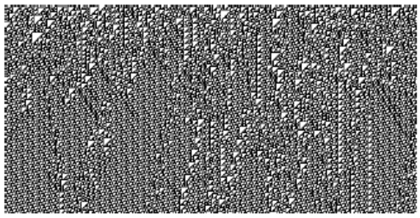
(klasa 1: pravilo 160)



(klasa 2: pravilo 108)



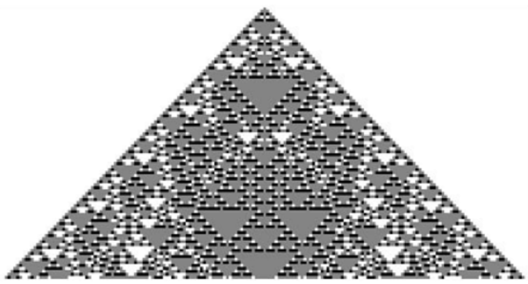
(klasa 3: pravilo 126)



(klasa 4: pravilo 110)

Slika 1.11: Prostorno-vremenski dijagrami elementarnih CA iz svake od četiri Wolframove klase CA.

Wolframova šema imenovanja se lako generalizuje na jednodimenzionalne CA sa većim susedstvom i skupom stanja. Jednodimenzionalan CA sa Moore-susedstvom poluprečnika r i k stanja identifikuje se brojem koji u sistemu sa osnovom k ima k^{2r+1} cifara.



Slika 1.12: CA sa Wolframovim brojem 1020 za početnu konfiguraciju c_0, takvu da $c_0(0) = 1$ i $c_0(i) = 0$ za sve $i \neq 0$.

Na slici 1.12 je dat prostorno-vremenski dijagram jednodimenzionalnog CA sa Wolframovim brojem 1020. Ovaj CA ima Moore-susedstvo poluprečnika $r = 1$, skup stanja $S = \{0, 1, 2\}$, a lokalno pravilo ažuriranja f je određeno dvadesetsedmocifrenim brojem u sistemu sa osnovom tri koji odgovara broju 1020. Stanju 0 pridružujemo belu, stanju 1 sivu i stanju 2 crnu boju.

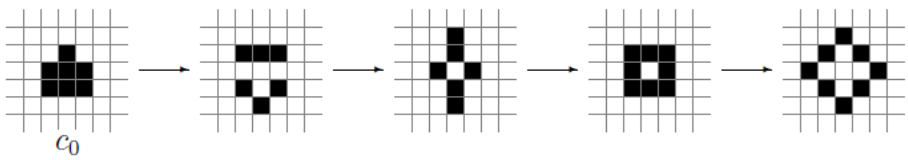
1.4 Igra života (Game-of-Life)

Najpoznatiji dvodimenzionalni CA je ***Igra života*** koju je izmislio John Conway[7] 1970. Njegov ćelijski prostor je Z^2, a skup stanja $S = \{0, 1\}$, pri čemu za ćeliju sa stanjem 1 kažemo da je živa i pridružujemo joj crnu boju, dok za ćeliju sa stanjem 0 kažemo da je mrtva i pridužujemo joj belu boju. Vektor susedstva je $N = \{(-1, -1),\ (-1, 0),\ (-1, 1),\ (0, -1),\ (0, 0),\ (0, 1),\ (1, -1),\ (1, 0),\ (1, 1)\}$, odnosno igra života ima Moore-susedsvo poluprečnika $r = 1$. Standardna oznaka Igre života je „***B3/S23***", jer je njeno lokalno pravilo ažuriranja definisano na sledeći način:

- ***Mrtva ćelija*** će oživeti (engl. „born") ako ima tačno tri živa suseda, inače ostaje mrtva.

- ***Živa ćelija*** preživljava (engl. „survives") ako i samo ako među njenih osam ćelija suseda ima tačno dve ili tri žive ćelije. Manje od dva živa suseda uzrokuje njenu smrt zbog izolacije, a više od tri živa suseda uzrokuje smrt zbog prenatrpanosti.

Na slici 1.13 dat je primer vremenske evolucije početne konfiguracije c_0 u Igri života. Igra života je izvanredna jer je njeno lokalno pravilo ažuriranja jednostavno, a ponašanje konfiguracija tokom vremena nepredvidivo.



Slika 1.13: Prvih pet koraka vremenske evolucije početne konfiguracije c_0 u Igri života.

Konfiguracije Igre života koje imaju konačan broj živih ćelija nazivamo ***uzorcima***. Tokom godina ljubitelji Igre života su napravili kategorizaciju uzoraka sa različitim ponašanjima tokom vremena. Za neke od kategorija uzoraka se koristi sledeća terminologija:

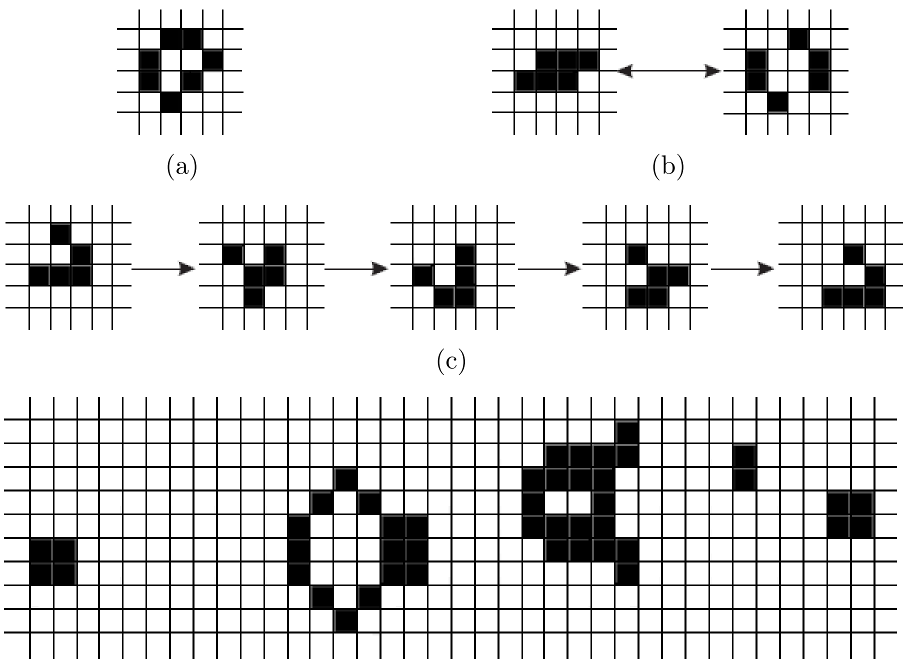
- ***mrtva priroda***: fiksna tačka uzorak. Pravilo ažuriranja ne menja stanja ćelija tokom vremena. Najmanja mrtva priroda naziva se ***blok***. Blok čine četiri žive ćelije koje obrazuju kvadrat, ostale ćelije su mrtve. Još jedan primer mrtve prirode dat je na slici 1.14(a). Broj mrtvih priroda koje imaju n živih ćelija je za $n = 1, 2, 3, \ldots$ redom 0, 0, 0, 2, 1, 5, 4, 9, 10, 25, 46, 121, 240, 619, 1353,. . . .

[7]Britanski matematičar rođen 1937. godine.

- ***oscilator***: periodičan uzorak. Pravilo ažuriranja može promeniti uzorak, ali nakon određenog broja koraka polazni uzorak se ponovo pojavljuje. Mrtva priroda je poseban tip oscilatora. Jedan oscilator sa periodom dva dat je na slici 1.14(b).

- ***svemirski brod***: uzorak koji se nakon određenog broja koraka pojavljuje ponovo, moguće na nekoj drugoj poziciji na mreži $\mathbb{Z}^2$. Oscilator je svemirski brod koji se ne pomera. Prvi otkriveni svemirski brod naziva se ***jedrenjak*** i on je dat na slici 1.14(c).

- ***puška***: konačan uzorak koji je kao i oscilator periodičan, ali pored toga periodično emituje (ispaljuje) svemirske brodove, pri čemu je period puške uvek umnožak perioda emitovanja. Na slici 1.14(d) dat je primer puške koja emituje jedrenjake, tzv. ***jedrenjak puška***.

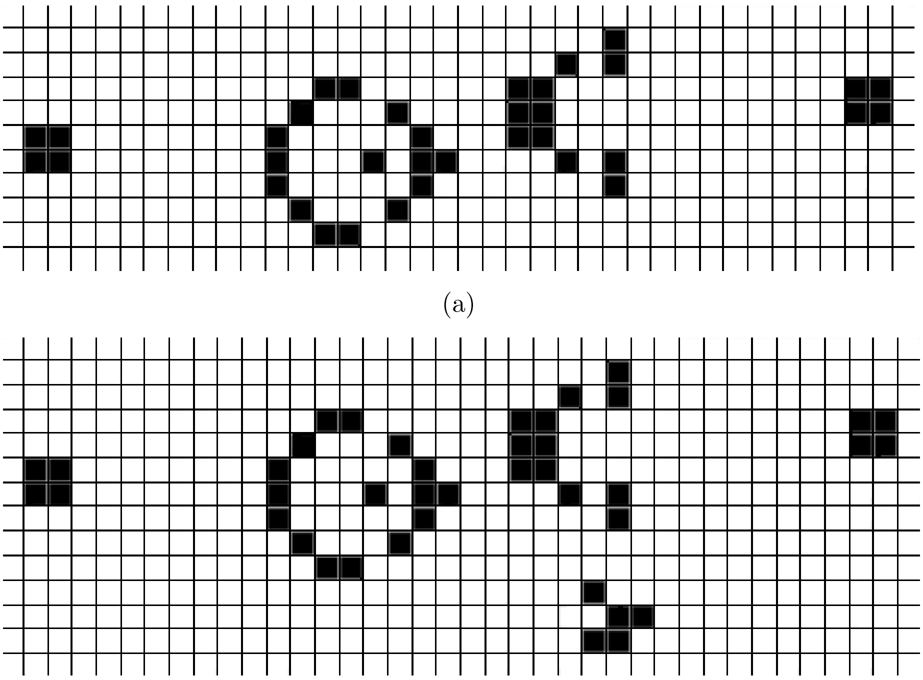


Slika 1.14: Primer objekata Igre života: (a) mrtva priroda, (b) oscilator, (c) jedrenjak, (d) jedrenjak puška.

Iste godine koje je igru i osmislio, Conway je pretpostavio da ne postoji uzorak koji može rasti neograničeno tokom vremena tj. da ni za jednu početnu konfiguraciju sa konačnim brojem živih ćelija, populacija ne može rasti iznad neke konačne gornje granice. U oktobru 1970. Igra života je prvi put publikovana u kolumni ,,Matematičke igre" Martin-a Gardner-a[8] i tom prilikom je Conway ponudio nagradu od 50 dolara prvom koji do kraja te godine dokaže ili

[8]Popularni američki matematičar (1914-2010). Oko četvrt veka je bio autor pomenute kolumne. Mnoge matematičke teme su upravo u ovoj kolumni po prvi put predstavljene široj naučnoj javnosti. Jedna od njih su i ranije spomenti fraktali.

opovrgne njegovu pretpostavku. Nagradu je već u novembru osvojio tim koji je predvodio Bill Gosper[9] sa Tehnološkog instituta u Masačusetsu. Oni su opovrgli pretpostavku Conway-a kontraprimerom koji je poznat pod nazivom ***Gosperova jedrenjak puška*** (vidi sliku 1.15). To je ujedno bila i prva jedrenjak puška otkrivena u Igri života. Ono što u ovom kontraprimeru obezbeđuje rast broja članova populacije tokom vremena jeste periodično emitovanje jedrenjaka. Sve do 2015. godine ovo je bila najmanja poznata jedrenjak puška.



(a)

(b)

Slika 1.15: (a) Gosperova jedrenjak puška, (b) jedna konfiguracija evolucije Gosperove jedrenjak puške u kojoj se vidi emitovani jedrenjak.

Igra života je, baš kao i elementarni CA sa pravilom ažuriranja 110, **računski univerzalna**, odnosno sposobna da simulira rad bilo kog drugog CA, Turing-ove mašine ili bilo kog drugog sistema za koji je poznato da može biti preveden u univerzalan sistem. Osnove dokaza univerzalnosti Igre života dali su, nezavisno jedan od drugog, Berlekamp[10] 1982. godine i Gosper 1983. Paul Rendell[11] je 2000. godine u Igri života implementirao Turing-ovu mašinu koja se može proširiti do univerzalne Turing-ove mašine. Za detalje o ovome i univerzalnosti Igre života videti [2].

Napomenimo da postoje brojne varijante igre života, koje imaju nešto drugačije pravilo ažuriranja od klasične igre ili drugačiji ćelijski prostor. Neke od najpoznatijih varijanti su ***HighLife*** („B36/S23”), ***3D Life***, ***The Immigration Game***, ***HexLife***, itd.

[9] Poznati američki programer i matematičar, rođen 1943. godine.
[10] Elwyn Berlekamp, američki matematičar rođen 1940. godine.
[11] Britanski informatičar. Docent Univerziteta Zapadne Engleske.

Glava 2

L-sistemi

L-sisteme je osmislio 1968. godine Aristid Lindenmayer, mađarski biolog i botaničar sa Univerziteta u Utrehtu, kao formalne matematičke modele za opisivanje ponašanja ćelija biljaka i modeliranje procesa rasta i razvoja biljaka. U prvim radovima ([8]) Lindenmayer-a i saradnika izvedeni su L-sistemi za formalno opisivanje razvoja jednostavnih višećelijskih organizama, kao što su alge, u smislu deobe, rasta i odumiranja pojedinačnih ćelija. Tako su među prvima objavljeni L-sistemi za modeliranje razvoja crvene alge ***Callithamnion roseum*** i modrozelene alge ***Anabaena catenula.*** Kasnije su ti L-sistemi prošireni kako bi opisivali više biljke i složene strukture grananja.

Drugačiji pristup interpretaciji L-sistema predložili su Szilard[1] i Quinton 1979. godine. Oni su pokazali da neverovatno jednostavni kontekstno slobodni L-sistemi mogu generisati fraktale. Njihov rezultat je kasnije proširen u više pravaca. Kao što smo ranije već rekli, fraktali se u kompjuterskoj grafici koriste za generisanje slika koje predstavljaju prirodne objekte: oblake, sneg, munje, morske obale, planinske vence...

Danas, L-sistemi imaju najveću primenu upravo u kompjuterskoj grafici, gde se koriste za generisanje fraktala i realistično modeliranje biljaka. Pored ovoga, koriste se u reprodukciji tradicionalne ***East Indian*** umetničke forme kao i kod grafički motivisanih algoritama za muzičke kompozicije.

U naredna dva odeljka ćemo se podsetiti osnovnih pojmova i definicija algebre jezika i teorije formalnih gramatika sa kojima smo se već susreli u [13]. Tu se takođe mogu naći i dokazi svih teorema koje ćemo u naredna dva odeljka navesti.

2.1 Algebre jezika

Neka je W neprazan, konačan skup simbola (slova). Takav skup W nazivamo ***azbuka.*** Svaki konačan niz simbola iz W nazivamo ***reč*** ili ***string*** nad W. Reč koja ne sadrži nijedan simbol je ***prazna reč*** i nju označavamo sa λ. Skup svih reči nad W označavamo sa W^*, a skup svih nepraznih reči nad W sa W^+. Na primer, $W^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$ za $W = \{a, b\}$. ***Dužina reči*** w, u oznaci $|w|$ jeste broj slova u w. Tako je $|\lambda| = 0$. Za reč $v \in W^*$ kažemo da je ***podreč*** od $w \in W^*$ ako postoje $u,\ z \in W^*$ tako da je $w = uvz$. Prazna reč je

[1] Andrew L. Szilard informatičar, profesor emeritus na Univerzitetu Zapadnog Ontaria.

podreč svake reči. Ako je $w = uv$, za u kažemo da je ***prefiks*** od w. Prazna reč i sama reč w su uvek prefiks od w. ***Pravi prefiks*** u od w je neprazan prefiks različit od samog w. ***Sufiks*** i ***pravi sufiks*** se definišu slično.

Definicija 2.1 *Neka je W neka azbuka. Svaki podskup L skupa svih reči W^* nad W zovemo **jezik** nad W. L je **konačan jezik** ako sadrži konačan broj reči. Na skupu $\mathcal{P}(W^*)$ svih jezika nad W definišemo operacije $+$ **(sabiranje)**, $\cdot$ **(konkatenacija)** i $*$ **(iteracija)** na sledeći način: ako su $R, S \in \mathcal{P}(W^*)$ onda*

$$R + S = R \cup S,$$

$$R \cdot S = \{w_1 \cdot w_2 \mid w_1 \in R \wedge w_2 \in S\},$$

$$R^* = \{w \mid (\exists n \in \mathbb{N}) w \in R^*\},$$

gde je $R^0 = \{\lambda\},\ R^{n+1} = R^n \cdot R.$

Algebru

$$\mathcal{L}_W = (\mathcal{P}(W^*),\ +\ ,\ \cdot\ ,\ \{\lambda\},\ *)$$

*zovemo **algebra jezika** nad W.*

Definicija 2.2 *Neka je W neka azbuka. Tada **regularne izraze** nad W dobijamo na sledeći način:*

(a) $\emptyset$ i λ su regularni izrazi nad W,

(b) ako je $a \in W$, onda je a regularni izraz nad W,

(c) Ako su r i s regularni izrazi nad W onda su to i $r + s$, $r \cdot s$ i r^*.

Skup svih regularnih izraza nad W označavamo sa Reg(W).

Definicija 2.3 *Neka je W neka azbuka. **Vrednost regularnog izraza** nad W jeste preslikavanje $\mathcal{L} : Reg(W) \to \mathcal{P}(W^*)$ definisano na sledeći način:*

(a) $\mathcal{L}(\emptyset) = \emptyset,\ \mathcal{L}(\lambda) = \{\lambda\}$,

(b) ako je $a \in W$, onda $\mathcal{L}(a) = \{a\}$,

(c)ako su $r,\ s \in Reg(W)$ onda

$$\mathcal{L}(r + s) = \mathcal{L}(r) + \mathcal{L}(s),$$

$$\mathcal{L}(r \cdot s) = \mathcal{L}(r) \cdot \mathcal{L}(s),$$

$$\mathcal{L}(r^*) = (\mathcal{L}(r))^*.$$

*Ako je $r \in Reg(W)$ i $\mathcal{L}(r) = A$, kažemo da **r predstavlja jezik A**.*

Napomenimo da u prethodnoj definiciji simbole $+$ i $*$ koristimo u dva različita značenja: sa leve strane jednakosti kao sintaktičke znakove u sastavu regularnih izraza, a sa desne strane jednakosti kao oznake operacija u algebri jezika.

Definicija 2.4 *Za jezik kažemo da je **regularan** ako se može predstaviti regularnim izrazom.*

Teorema 2.5 *Svaki konačan jezik nad nekom azbukom W je regularan.*

2.2 Formalne gramatike

Formalne gramatike su sistemi prerade stringova (engl. rewriting systems). Uveo ih je američki lingvista Noam Chomsky 1957. godine.

Definicija 2.6 ***Formalna ili generativna gramatika*** *G je uređena četvorka*

$$(V, T, S, P)$$

gde su V i T konačni, disjunktni skupovi, $S \in V$, a P konačan skup uređenih parova $(\alpha,\ \beta)$ gde su $\alpha, \beta \in (V \cup T)^*$ i reč α sadrži bar jedan simbol iz V.

Ako je $G = (V, T, S, P)$ generativna gramatika, onda kažemo da je:

V – skup ***neterminalnih simbola***,

T – skup ***terminalnih simbola***,

S – ***početni simbol***,

P – skup ***produkcija***.

Po dogovoru produkcije $(\alpha,\ \beta)$ obeležavamo sa $\alpha \to \beta$. Takođe, ako eksplicitno nije drugačije rečeno, važe sledeći dogovori:

(a) Velika slova A, B, C, D, E *i* S će označavati neterminalne simbole. S će biti početni simbol, ako se drugačije ne kaže.

(b) Mala slova a, b, c, d, e i cifre će označavati terminalne simbole,

(c) Velika slova X, Y, Z označavaju simbole koji su terminalni ili neterminalni,

(d) Mala slova u, v, w, x, y, z označavaju terminalne reči (sastavljene od terminalnih simbola),

(e) Mala grčka slova $\alpha, \beta, \gamma, \delta$ označavaju reči sastavljene od terminalnih i neterminalnih slova.

Definicija 2.7 *Neka je $G = (V, T, S, P)$ formalna gramatika. Definišimo binarnu relaciju $\Rightarrow$ na skupu $(V \cup T)^*$ na sledeči način: ako su $\alpha,\ \beta \in (V \cup T)^*$, tada*

$$\alpha \Rightarrow \beta \quad \text{akko} \quad \text{postoje } \alpha_1, \alpha_2 \in (V \cup T)^* \text{ i postoji produkcija } \gamma \to \delta \text{ iz } P \text{ tako da je } \alpha = \alpha_1 \gamma \alpha_2 \text{ i } \beta = \alpha_1 \delta \alpha_2.$$

Relaciju $\stackrel{*}{\Rightarrow}$ na skupu $(V \cup T)^*$ definišemo kao refleksivno - tranzitivno zatvorenje relacije $\Rightarrow$.

Uočimo da po definiciji refleksivno - tranzitivnog zatvorenja relacije $\Rightarrow$ imamo da

$$\alpha \stackrel{*}{\Rightarrow} \beta \quad \text{akko} \quad \alpha = \beta \text{ ili postoji konačan niz reči } \alpha_1, \alpha_2, \ldots, \alpha_n \text{ iz } (V \cup T)^* \text{ tako da je } \alpha = \alpha_1,\ \beta = \alpha_n \text{ i } \alpha_i \Rightarrow \alpha_{i+1} \text{ za sve } i,\ 1 \le i \le n.$$

Svaki konačan niz reči $\alpha_1, \ldots, \alpha_n$ iz $(V \cup T)^*$, takav da je $\alpha_1 = S$ i $\alpha_i \Rightarrow \alpha_{i+1}$, za sve i, $1 \leq i \leq n$, zovemo ***izvođenje*** reči α_n u gramatici G. Nije teško zaključiti da se u svakom koraku izvođenja primenjuje tačno jedna produkcija.

Definicija 2.8 *Neka je $G = (V,\ T,\ S,\ P)$ formalna gramatika. Jezik generisan ovom gramatikom je jezik $L(G)$ definisan sa $L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$.*

U sledećoj definiciji iznosimo najvažniju podelu formalnih gramatika prema obliku produkcija, koju je dao Noam Chomsky. Po njemu se ta podela zove ***Hijerarhija Chomsky***.

Definicija 2.9 *Neka je $G = (V, T, S, P)$ formalna gramatika. Za G kažemo da je **tipa i** ($1 \leq i \leq 3$) ako produkcije zadovoljavaju sledeće uslove:*

> $i = 3$*: Sve produkcije su oblika $A \to wB$ i $A \to w$ gde su $A, B \in V, w \in T^*$.*
>
> $i = 2$*: Sve produkcije su oblika $A \to \alpha$ gde je $A \in V,\ \alpha \in (V \cup T)^*$.*
>
> $i = 1$*: Sve produkcije su oblika*
>
> > *(1) $\alpha A \beta \to \alpha\gamma\beta$ gde je $A \in V,\ \alpha,\ \beta,\ \gamma \in (V \cup T)^*,\ \gamma \neq \lambda$,*
> >
> > *(2) $S \to \lambda$,*

pod uslovom da, ako gramatika sadrži produkciju $S \to \lambda$, onda se simbol S ne javlja sa desne strane ni jedne produkcije gramatike G.

Klasu svih formalnih gramatika (dakle, bez posebnih ograničenja na produkcije) zovemo i gramatikama tipa 0. Klasu jezika koji se mogu generisati gramatikama tipa i $(0 \leq i \leq 3)$ zovemo jezici tipa i i označavamo ih sa L_i. Jezike tipa 0 još nazivamo i ***rekurzivno nabrojivi (RE)***.

Kod gramatika tipa 1 produkcije omogućavaju da se u izvođenju neke reči umesto nekog neterminalnog simbola A zameni reč γ, ali samo u slučaju da se taj simbol A nalazi u tačno određenom kontekstu. Stoga ove gramatike, kao i njima odgovarajuću klasu jezika, nazivamo ***kontekstno osetljivim (CS)***.

U slučaju gramatika tipa 2, sve produkcije su oblika $A \to \alpha$, gde je A neki neterminalni simbol, a α reč sastavljena od terminalnih i neterminalnih simbola. U tim gramatikama se dakle u svakom koraku izvođenja zamenjuje tačno jedan neterminalni simbol i to bez obzira gde se on u datoj reči nalazi (tj. bez obzira na kontekst). Te gramatike i njima odgovarajuće jezike nazivamo ***kontekstno slobodnim (CF)***.

Teorema 2.10 *Klasa svih regularnih jezika se poklapa sa klasom svih jezika tipa 3.*

Na osnovu ovog tvrđenja zaključujemo da jezike tipa 3 možemo nazivati ***regulanim jezicima (REG)***, a da pri tom ne dođe do zabune, jer smo u definiciji 2.4 regularne jezike definisali kao jezike koji se mogu predstaviti regularnim izrazima.

Teorema 2.11 *$K \subset REG \subset CF \subset CS \subset RE$, gde K klasa svih konačnih jezika, a "$\subset$" označava striktnu inkluziju.*

Primer 2.12 *Gramatika G zadata produkcijama*
(a)

$$S \to aS,\ S \to \lambda$$

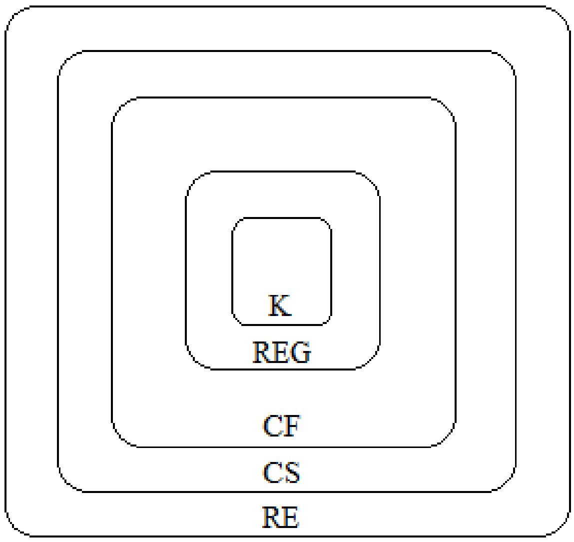
generiše regularan jezik $\{a^n \mid n \in \mathbb{N}\}$, *što zaključujemo iz oblika produkcija.*
(b)

$$S \to \lambda,\ S \to aSb$$

generiše kontekstno slobodan jezik $\{a^n b^n \mid n \geq 1\}$ *koji nije regularan.*
(c)

$$S \to abc,\ Ac \to Bbcc,\ Ab \to bA,\ aB \to aaA,\ bB \to Bb,\ aB \to aa,\ S \to aAbc$$

generiše kontekstno osetljiv jezik $\{a^n b^n c^n \mid n \geq 1\}$ *koji nije kontekstno slobodan.*



Slika 2.1: Odnos klasa jezika hijerarhije Chomsky.

2.3 L-sistemi

Kao što smo videli u prethodnom odeljku, formalne gramatike su sistemi prerade reči u kojima se u svakom koraku izvođenja primenjuje tačno jedna produkcija i deo reči dobijene u prethodnom koraku izvođenja biva zamenjen u skladu sa primenjenom produkcijom. L-sistemi su sistemi prerade reči kod kojih u svakom koraku izvođenja sva slova u reči bivaju zamenjena istovremeno u skladu sa produkcijama. Takođe kod L-sistema nema neterminalnih simbola, koje imamo kod formalnih gramatika.

Primer 2.13 *Neka je* $W = \{a\}$ *azbuka ,* $a \to a^2$ *jedina produkcija i* a^3 *početna reč. Ako bismo izvodili kao kod formalnih gramatika u jednom koraku izvođenja mogli bismo zameniti samo jedno a sa* a^2. *Uzastopnom primenom produkcije dobijamo niz reči* $a^3, a^4, a^5, a^6, \ldots$ *i generisani jezik je* $\{a^i \mid i \geq 3\}$. *Ako bi izvodili* *kao kod L- sistema, u jednom koraku izvođenja svako slovo a zamenili bismo sa a^2, pa uzastopnom primenom produkcije dobijamo niz reči $a^3, a^6, a^{12}, a^{24}, \ldots$, odnosno generisani jezik je $\{a^{3\cdot 2^i} \mid i \geq 0\}$.*

Definicija 2.14 *Neka su $m, n \in \mathbb{N}_0$. **(m, n) L-sistem** je uređena trojka*

$$G = (W, P, w_0)$$

*gde je W konačna neprazna **azbuka**, P konačan skup **produkcija**, odnosno uređenih četvorki (u, a, v, α) gde je $u \in \bigcup_{i=0}^{m} W^i$, $a \in W$, $v \in \bigcup_{i=0}^{n} W^i$ i $\alpha \in W^*$, i $w_0 \in W^+$ **početna reč**.*

Produkcije po dogovoru obeležavamo sa $(u, a, v) \to \alpha$.

Definicija 2.15 *Neka je $G = (W, P, w_0)$ (m, n) L-sistem, $a_1, a_2, \ldots, a_k \in W$ i $\alpha_1, \alpha_2, \ldots, \alpha_k \in W^*$. Binarnu relaciju $\underset{G}{\Rightarrow}$ na W^* definišemo po slučajevima, na sledeći način:*
(a) $m = 0 \wedge n = 0$, tada

$a_1 a_2 \ldots a_k \underset{G}{\Rightarrow} \alpha_1 \alpha_2 \ldots \alpha_k$ *ako* $(\lambda, a_i, \lambda) \to \alpha_i \in P$ *za sve i*, $1 \leq i \leq k$.

(b) $m = 0 \wedge n \neq 0$, tada

$a_1 a_2 \ldots a_k \underset{G}{\Rightarrow} \alpha_1 \alpha_2 \ldots \alpha_k$ *ako* $(\lambda, a_i, a_{i+1} \ldots a_{i+n}) \to \alpha_i \in P$ *za sve i*, $1 \leq i \leq k$, *gde uzimamo da je* $a_j = \lambda$ *kad god je* $j > k$.

(c) $m \neq 0 \wedge n = 0$, relacija se definiše slično kao u prethodnom slučaju.
(d) $m \neq 0 \wedge n \neq 0$, tada

$a_1 a_2 \ldots a_k \underset{G}{\Rightarrow} \alpha_1 \alpha_2 \ldots \alpha_k$ *ako* $(a_{i-m} \ldots a_{i-1}, a_i, a_{i+1} \ldots a_{i+n}) \to \alpha_i \in P$ *za sve i*, $1 \leq i \leq k$, *gde uzimamo da je* $a_j = \lambda$ *kad god je* $j < 1$ *ili* $j > k$.

Relaciju $\overset{*}{\underset{G}{\Rightarrow}}$ na skupu W^* definišemo kao refleksivno-tranzitivno zatvorenje relacije $\underset{G}{\Rightarrow}$.

Sada jednostavno zaključujemo da parametri m i n određuju kontekst u kom se nalazi svako slovo za zamenu. Iz definicije refleksivno-tranzitivnog zatvorenja relacije imamo da

$w \overset{*}{\underset{G}{\Rightarrow}} v$ akko $w = v$ ili postoji konačan niz reči $v_1, v_2, \ldots, v_l$ iz W^* tako da je $v_1 = w$, $v_l = v$ i $v_i \underset{G}{\Rightarrow} v_{i+1}$ za sve i, $1 \leq i \leq l$.

Svaki konačan niz reči $v_1, v_2, \ldots, v_l$ iz W^*, takav da je $v_1 = w_0$ i $v_i \underset{G}{\Rightarrow} v_{i+1}$, za sve i, $1 \leq i \leq l$, zovemo ***izvođenje reči*** v_l u (m, n) L-sistemu G. Iz definicije relacije $\underset{G}{\Rightarrow}$ lako se zaključuje da se u svakom koraku izvođenja na svako slovo u reči primenjuje tačno jedna produkcija.

Definicija 2.16 *Neka je $G = (W, P, w_0)$ (m, n) L-sistem. Jezik generisan sa G je*

$$\mathcal{L}(G) = \{v \in W^* \mid w_0 \overset{*}{\underset{G}{\Rightarrow}} v\}.$$

Ovako definisane (m, n) L-sisteme nazivamo ***osnovnim*** ili ***čistim*** L-sistemima, jer postoje njihova brojna proširenja, kao što su tabelarni (m, n) L-sistemi, (m, n) L-sistemi sa mehanizmima filtriranja itd., sa kojima ćemo se kasnije upoznati. Podelu čistih (m, n) L-sistema, isto kao i podelu formalnih gramatika, vršimo na osnovu oblika produkcija. U nastavku dajemo definiciju dve osnovne klase.

Definicija 2.17 *(a) Klasu* ***kontekstno osetljivih (m, n) L-sistema*** *čine svi (m, n) L-sistemi, bez ikakvih ograničenja na produkcije. Ovu klasu označavamo sa $\mathcal{IL}$.*

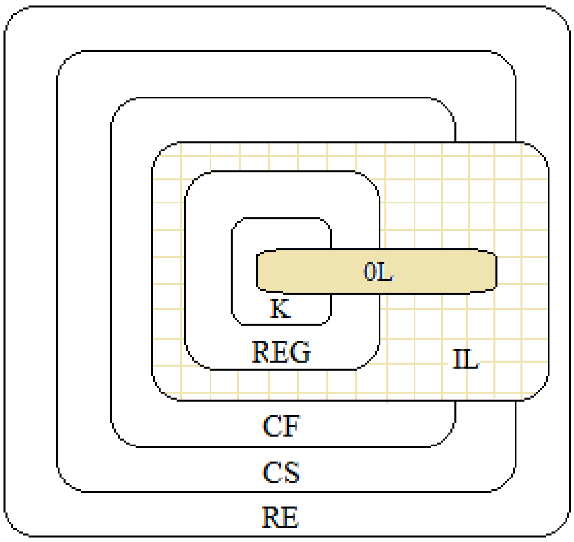
(b) Klasu ***kontekstno slobodnih (m, n) L-sistema*** *cine svi $(0, 0)$ L-sistemi, odnosno svi sistemi čije su sve produkcije oblika $(\lambda, a, \lambda) \to \alpha$ (jednostavnije $a \to \alpha$). Ovu klasu označavamo sa $0\mathcal{L}$.*

Kod (m, n) L-sistema iz klase $0\mathcal{L}$, zamena slova u reči ne zavisi od konteksta u kojem se to slovo nalazi. Po dogovoru, iste oznake koristimo i za odgovarajuće klase jezika. Iz definicija klasa $\mathcal{IL}$ i $0\mathcal{L}$ sistema lako zaključujemo da za odgovarajuće klase jezika važi $0\mathcal{L} \subseteq \mathcal{IL}$. Vrlo je važno napomenuti da se klasa jezika $\mathcal{IL}$ ($0\mathcal{L}$) razlikuje od klase *CS (CF)* jezika generisanih kontekstno osetljivim (kontekstno slobodnim) formalnim gramatikama. Na slici 2.2 prikazan je odnos klasa jezika hijerarhije Chomsky i klasa jezika $\mathcal{IL}$ i $0\mathcal{L}$. Sledeća notacija je uobičajena u literaturi, a delom proizilazi iz gornje definicije.

$(0, 0)$ L-sistemi $\equiv 0\mathcal{L}$ sistemi,

$(1, 0)$ L-sistemi ili $(0, 1)$ L- sistemi $\equiv 1\mathcal{L}$ sistemi,

$(1, 1)$ L-sistemi $\equiv 2\mathcal{L}$ sistemi.



Slika 2.2: Odnos klasa jezika hijerarhije Chomsky i klasa jezika $\mathcal{IL}$ i $0\mathcal{L}$.

Primer 2.18 *Razmatramo* $(1,0)$ *L-sistem* $G = (\{a,b,c,d\},\ P,\ ad)$, *gde je skup produkcija* P *zadat sledećom tabelom u kojoj se levi susedi nalaze u koloni, a slova za zamenu u redu:*

	a	b	c	d
λ	c	b	a	d
a	a	b	a	d
b	a	b	a	d
c	b	c	a	ad
d	a	b	a	d

Niz reči, generisan ovim sistemom je:

$$ad, cd, aad, cad, abd, cbd, acd, caad, abad, cbad, \ldots$$

Ovaj sistem je primer kontekstno osetljivog sistema koji nije kontekstno slobodan, jer zamena svakog slova u reči zavisi od njegovog levog suseda.

Definicija 2.19 (m,n) *L-sistem* $G = (W, P, w_0)$ *je:*

(a) ***determinstički****, ako poseduje svojstvo determinizma odnosno, ako za svako slovo azbuke* W *važi da* $(u,a,v) \to \alpha$, $(u,a,v) \to \beta \in P$ *implicira* $\alpha = \beta$.

*(b)****reprodukcijski****, ako u procesu izvođenja nema brisanja slova, odnosno ako njegov skup produkcija* P *ne sadrži produkcije oblika* $(u,a,v) \to \lambda$. *Kazemo još i da je sistem* ***λ-slobodan****.*

Nije teško zaključiti da neki (m,n) L-sistemi mogu posedovati oba svojstva iz prethodne definicije. Da neka klasa (m,n) L-sistema poseduje svojstvo determinizma naznačavamo prefiksom $\mathcal{D}$ u oznaci klase, da je λ-slobodna prefiksom $\mathcal{P}$, a da poseduje oba svojstva prefiksom $\mathcal{PD}$. Stoga pored klasa navedenih u definiciji 2.17 imamo još i klase:

$\mathcal{D}0\mathcal{L}$ – klasa svih determinističkih kontekstno slobodnih (m,n) L-sistema,

$\mathcal{DIL}$ – klasa svih determinističkih kontekstno osetljivih (m,n) L-sistema,

$\mathcal{P}0\mathcal{L}$ – klasa svih reprodukcijskih kontekstno slobodnih (m,n) L-sistema,

$\mathcal{PIL}$ – klasa svih reprodukcijskih kontekstno osetljivih (m,n) L-sistema,

$\mathcal{PD}0\mathcal{L}$ – klasa svih reprodukcijskih determinističkih kontekstno slobodnih (m,n) L-sistema,

$\mathcal{PDIL}$ – klasa svih reprodukcijskih determinističkih kontekstno osetljivih (m,n) L-sistema.

Iste oznake koristimo i za odgovarajuće klase jezika. Prilično je jasno da za klase jezika važi $\mathcal{PD}0\mathcal{L} \subseteq \mathcal{P}0\mathcal{L} \subseteq 0\mathcal{L}$ kao i $\mathcal{PD}0\mathcal{L} \subseteq \mathcal{D}0\mathcal{L} \subseteq 0\mathcal{L}$, međutim nije jasno da li svuda važe stroge inkluzije. Analogne inkluzije važe i za kontekstno osetljive klase itd.

Uobičajeno je da se umesto oznake G za (m,n) L-sistem koristi oznaka koja predstavlja neko svojstvo tog sistema, kao što ćemo videti u sledećem primeru.

Primer 2.20 *Razmatramo* $(0,0)$ *L-sistem*
(a)

$$MISS_3 = (\{a\},\ P,\ a),\ gde\ je\ P = \{a \to \lambda,\ a \to a^2,\ a \to a^5\}.$$

Indukcijom se može pokazati da ovaj sistem generiše jezik

$$L(MISS_3) = \{a^i \mid i \neq 3\},$$

pa otuda i oznaka $MISS_3$*. Trivijalno, na osnovu definicije 1.11, zaključujemo da ovaj sistem pripada klasama* $0\mathcal{L}$ *pa samim tim i klasi* $\mathcal{IL}$*, dok na osnovu definicije 2.19 imamo da nije deterministički, jer za slovo a ima tri različite produkcije, a ni reprodukcijski, jer skup produkcija sadrži produkciju* $a \to \lambda$*.*
(b)

$$DEATH_b = (\{a,b\},\ P,\ ab^2a),\ gde\ je\ P = \{a \to ab^2a,\ b \to \lambda\}.$$

Ovaj sistem isto kao i sistem $MISS_3$ *pripada klsama* $0\mathcal{L}$ *i* $\mathcal{IL}$*, ali je i deterministički, jer za svako slovo azbuke ima tačno jednu produkciju. Odavde direktno sledi da, za razliku od sistema* $MISS_3$*, pripada i klasama* $\mathcal{D}0\mathcal{L}$ *i* $\mathcal{DIL}$*. Nije reprodukcijski jer sadrži produkciju* $b \to \lambda$*. Upravo zbog ove produkcije sistem nosi oznaku* $DEATH_b$*, jer ona obezbeđuje da u svakom koraku izvođenja nestanu sva slova b dobijena u prethodnom koraku izvođenja. Generisani jezik je*

$$L(DEATH_b) = \{(ab^2a)^{2^i} \mid i \geq 0\}.$$

(c)

$$LIN = (\{a,\ b\},\ P,\ ab),\ gde\ je\ P = \{a \to a,\ b \to ab\}.$$

Kao i prethodna dva sistema i ovaj sistem pripada klasama $0\mathcal{L}$ *i* $\mathcal{IL}$ *, ali je za razliku od njih i deterministički i reprodukcijski pa pripada klasama* $\mathcal{D}0\mathcal{L}$*,* $\mathcal{P}0\mathcal{L}$*,* $\mathcal{PD}0\mathcal{L}$*,* $\mathcal{DIL}$*,* $\mathcal{PIL}$*,* $\mathcal{PDIL}$*. Ovaj sistem generiše jezik*

$$L(LIN) = \{a^i b \mid i \geq 1\}.$$

Oznaka LIN se odnosi na linearan rast dužine reči: $j-$*ta reč u nizu izvođenja je dužine* $j+2$*, pri čemu smatramo da je početna reč ab na nultoj poziciji u nizu izvođenja.*

(d) $(0,1)$ *L-sistem iz primera 2.18 je deterministički i reprodukcijski jer poseduje oba svojstva iz definicije 2.19, odnosno pripada klasi* $\mathcal{PDIL}$*.*

U nastavku, definišemo dva proširenja (m,n) L-sistema.

Definicija 2.21 ***Tabelarni (m, n) L- sistem*** *sa q tablica,* $T_q(m,n)$ *L-sistem, je uređena trojka*

$$\widetilde{G} = (W, \widetilde{P}, w_0)$$

gde je W ***azbuka****,* $\widetilde{P} = \{P_1, P_2, \ldots, P_q\}$ ***skup tabela****, pri čemu je* $P_i,\ 1 \leq i \leq q$ *definisano isto kao i P u definiciji 2.14, i* w_0 ***početna reč****. Dakle,* T_q *(m, n) L-sistem je uređena trojka sa skupom* $\mathcal{P} = \{P_1, P_2, \ldots, P_q\}$ *takvim da je za sve* $i,\ 1 \leq i \leq q$*,* $G_i = (W, P_i, w)$ (m,n) *L-sistem.*

U tabelarnim (m,n) L-sistemima reči se izvode na sledeći način:

$$a_1 a_2 \ldots a_k \underset{\widetilde{G}}{\Rightarrow} \alpha_1 \alpha_2 \ldots \alpha_k\ , \qquad a_1,\ a_2,\ \ldots,\ a_k \in W \text{ i } \alpha_1,\ \alpha_2,\ \ldots, \alpha_k \in W^*$$

ako postoji tabela P_i u skupu tabela $\widetilde{P}$ takva da

$$a_1 a_2 \dots a_k \underset{G_i}{\Rightarrow} \alpha_1 \alpha_2 \dots \alpha_k.$$

Odnosno u svakom koraku izvođenja se mogu primenjivati samo produkcije iz jednog od skupova P_i. Uobičajeno $\overset{*}{\underset{\widetilde{G}}{\Rightarrow}}$ je zatvorenje relacije $\underset{\widetilde{G}}{\Rightarrow}$. Jezik generisan sa $\widetilde{G}$ je $\mathcal{L}(\widetilde{G}) = \{v \in W^* \mid w_0 \overset{*}{\underset{\widetilde{G}}{\Rightarrow}} v\}$.

Iste tipove klasa koje smo imali kod čistih (m, n) L-sistema imamo i kod tabelarnih sistema. $\widetilde{G}$ je XYT_qZL sistem ako je za $\widetilde{P} = \{P_1, P_2, \dots, P_q\}$ svako G_i, $1 \leq i \leq q$, XYZL sistem. Tako npr. imamo klase $\mathcal{T}0\mathcal{L}$, $\mathcal{DT}0\mathcal{L}$, $\mathcal{PDT}_2\mathcal{IL}$,... Ako je u oznaci neke klase tabelarnih L-sistema izostavljen broj tabela q onda ta klasa sadrži sve tabelarne sisteme koji zadovoljavaju naznačena svojstva bez obzira na broj tabela. Tako je $\mathcal{T}0\mathcal{L}$ **klasa svih tabelarnih kontekstno slobodnih L-sistema**. Napomenimo da je za $q = 1$ tabelarni (m, n) L-sistem zapravo čist (m, n) L-sistem iz definicije 2.14.

Definicija 2.22 ***(m, n) L-sistem sa q početnih reči,*** *$F_q(m, n)$ L-sistem, je uređena trojka*

$$\widetilde{G} = (W, P, F),$$

gde je W ***azbuka****, P* ***skup produkcija*** *definisan isto kao u definiciji 2.14 i $F = \{w_0', \dots, w_0^{(q)}\}$ konačan* ***skup početnih reči****. (m, n) L-sistem sa q početnih reči je uređena trojka $\widetilde{G} = (W, P, F)$ sa skupom $F = \{w_0', w_0'', \dots, w_0^{(q)}\}$ takvim da je za sve i, $1 \leq i \leq q$, $G_i = (W, P, w_0^{(i)})$ (m, n) L-sistem.*

Reči F_qL-sistema se izvode slično kao kod T_qL-sistema, takođe G je XYF_qZL sistem ako je za $F = \{w_0', w_0'', \dots, w_0^{(q)}\}$ svako G_i, $1 \leq i \leq q$, XYZL sistem. Za $q = 1$ (m, n) L-sistemi sa konačnim skupom početnih reči se svodi na čist (m, n) L-sistem iz definicije 2.14. Napomenimo da postoje (m, n) L-sistemi koji istovremeno imaju i svojstvo $\mathcal{T}$ i svojstvo $\mathcal{F}$. Recimo, klasu kontekstno slobodnih L-sistema koji poseduju oba ova svojstva označavamo sa $\mathcal{TF}0\mathcal{L}$. Nije teško zaključiti da sistemi iz ove klase mogu biti λ-slobodni, posedovati svojstvo determinizma, ili i jedno i drugo, pa tako imamo podklase $\mathcal{PTF}0\mathcal{L}$, $\mathcal{DTF}0\mathcal{L}$, $\mathcal{PDTF}0\mathcal{L}$.

Dalje u radu ćemo se baviti isključivo klasama kontekstno slobodnih čistih L-sistema i njihovih gore definisanih proširenja. Upoznaćemo se sa njihovim osnovnim osobinama, mehanizmima filtriranja itd.

2.4 Klase $0\mathcal{L}$ i $\mathcal{T}0\mathcal{L}$ sistema

Ovde ćemo dati nešto drugačije definicije klasa $0\mathcal{L}$ i $\mathcal{T}0\mathcal{L}$ sistema od onih datih u prethodnom odeljku, ali suština je naravno ista. Upoznaćemo se sa osnovnim osobinama najpoznatijih podklasa klase $0\mathcal{L}$ sistema, a to su klase $\mathcal{D}0\mathcal{L}$ i $\mathcal{PD}0\mathcal{L}$.

Definicija 2.23 ***Konačna substitucija*** *σ nad azbukom W je preslikavanje*

$$\sigma : W^* \to 2^{W^*}$$

takvo da :
(a) za sva slova $a \in W$ važi da je $\sigma(a)$ konačan i neprazan,

(b) za sve reči $\alpha,\ \beta \in W^*$ važi $\sigma(\alpha\beta) = \sigma(\alpha)\sigma(\beta)$
*pri čemu uzimamo da je $\sigma(\lambda) = \{\lambda\}$. Ako nijedan od skupova $\sigma(a),\ a \in W$ ne sadrži praznu reč λ onda za σ kažemo da je **λ-slobodna**. Ako za svako $a \in W^*$ skup $\sigma(a)$ sadrži tačno jednu reč, σ se naziva **morfizam**. Morfizam σ je **kodiranje** ako za sve $a \in W$ važi $|\sigma(a)| = 1$, odnosno **slabo kodiranje** ako važi $|\sigma(a)| \leq 1$ za sve $a \in W$.*

Prema gornjoj definiciji konačna substitucija σ se primenjuje na reč w tako što se svako slovo a reči w zameni nekom rečju iz $\sigma(a)$. Različita pojavljivanja slova a mogu biti zamenjena različitim rečima iz skupa $\sigma(a)$. Ako je σ morfizam onda svaki $\sigma(a)$ sadrži samo jednu reč što primenu ove konačne substitucije na neku reč čini ***determinističkom***. Pogodno je da se konačna substitucija σ zada navođenjem svih produkcija (pravila zamene) za svako slovo azbuke W, na primer:

$$a \to \lambda,\ a \to a^2,\ a \to a^7,$$

je ekvivalentno sa

$$\sigma(a) = \{\lambda,\ a^2,\ a^7\}.$$

Sada je, npr.

$$\sigma(a^2) = \sigma(a)\sigma(a) = \{\lambda,\ a^2,\ a^4,\ a^7,\ a^9,\ a^{14}\}.$$

Slično, substitucija (u suštini morfizam) definisana sa

$$\sigma_1(a) = \{b\},\ \sigma_1(b) = \{ab\},\ W = \{a,\ b\},$$

može se zadati navođenjem produkcija

$$a \to b,\ b \to ab.$$

U ovom slučaju, za svaku reč w nad azbukom W, $\sigma_1(w)$ sadrži samo jednu reč, npr.

$$\sigma_1(a^2ba) = \{b^2ab^2\} = b^2ab^2.$$

Gornji rezultati za $\sigma(a^2)$ i $\sigma_1(a^2ba)$ dobijeni su istovremenom primenom produkcija: sva slova u reči na koju se σ primenjuje zamenjena su istovremeno u skladu sa produkcijama. Primena σ na reč w znači da se nešto dešava svuda u w. Nijedan deo w ne može ostati nepromenjen osim ako imamo produkciju $a \to a$.

Definicija 2.24 *$0\mathcal{L}$ sistem je uređena trojka $G = (W,\ \sigma,\ w_0)$, gde je W konačna neprazna azbuka, σ je konačna substitucija nad W i $w_0 \in W^*$ početna reč. $0\mathcal{L}$ je deterministički odnosno $\mathcal{D}0\mathcal{L}$ sistem ako je σ morfizam, a $\mathcal{P}0\mathcal{L}$ ako je σ $\lambda -$ slobodna. Jezik koji $0\mathcal{L}$ sistem generiše je:*

$$L(G) = w_0 \cup \sigma(w_0) \cup \sigma(\sigma(w_0)) \cup \ldots = \bigcup_{i \geq 0} \sigma^i(w_0).$$

Nadalje ćemo konačnu substituciju koja je morfizam, radi jednostavnosti, označavati sa h. Neka je $G = (W,\ h,\ w_0)$ $\mathcal{D}0\mathcal{L}$ sistem. Zbog svojstva determinizma sistem G generiše svoj jezik u određenom redosledu, kao ***niz reči***:

$$w_0 = w,\ w_1 = h(w_0),\ w_2 = h(w_1) = h^2(w_0),\ w_3, \ldots$$

Ovaj niz označavamo sa $S(G)$. Prema tome u vezi sa $\mathcal{D}0\mathcal{L}$ sistemom govorimo o njegovom jeziku $L(G)$ i nizu $S(G)$ kao o dva različita usko povezana pojma. U sledećem primeru razmatramo neke $\mathcal{D}0\mathcal{L}$ sisteme sa posebnim osvrtom na njihove nizove reči $S(G)$.

Primer 2.25 *Razmatramo $\mathcal{D}0\mathcal{L}$ sistem*
(a)

$$EXP_2 = (\{a\}, h, a), \text{ gde je } h(a) = \{a^2\}.$$

Niz reči $S(EXP_2)$ se sastoji od reči a^{2^i}, $i \geq 0$ poređanih u rastućem redosledu po dužini reči.
(b)

$$FIB = (\{a\}, h, a), \text{ gde je } h(a) = \{b\} \text{ i } h(b) = \{ab\}.$$

Prvih nekoliko reči niza S(FIB) su:

$$a, b, ab, bab, abbab, bababbab.$$

Tvrdimo da se počev od reči ab, svaka sledeča reč u nizu dobija konkatenacijom prethodne dve reči u nizu, odnosno da je

$$w_n = w_{n-1}w_{n-2}, \; n \geq 2.$$

Ovo lako dokazujemo, uzimajuči u obzir definiciju morfizma h, na sledeći način:

$$\begin{aligned} w_n &= h^n(a) = h^{n-1}(h(a)) = h^{n-1}(b) = h^{n-2}(h(b)) = \\ &= h^{n-2}(ab) = h^{n-2}(a)h^{n-2}(b) = h^{n-2}(a)h^{n-2}(h(a)) = \\ &= h^{n-2}(a)h^{n-1}(a) = w_{n-2}w_{n-1}. \end{aligned}$$

Kao posledicu ove tvrdnje imamo da za dužine svih reči počev od druge važi

$$|w_n| = |w_{n-2}| + |w_{n-1}|, \; n \geq 2.$$

Nije teško zaključiti da je niz dužina reči iz niza S(G) dobro poznati Fibonacci-jev niz 1, 1, 2, 3, 5, 8, 13, 21, 34,..., pa otuda i oznaka FIB.
(c)

$$SQUARES = (\{a, b, c\}, h, a), \text{ gde je } h(a) = \{abc^2\}, \; h(b) = \{bc^2\}, \; h(c) = c.$$

Niz S(SQUARES) počinje rečima

$$a, abc^2, abc^2bc^2c^2, abc^2bc^2c^2bc^2c^2c^2, \ldots$$

Označavajući ponovo sa w_i, $i \geq 0$ reči u nizu, lako dokazujemo da važi:

$$|w_{i+1}| = |w_i| + 2i + 3, \text{ za sve } i \geq 0.$$

Sada indukcijom možemo pokazati da je $|w_i| = (i+1)^2$, za sve $i \geq 0$, zbog čega ovaj sistem i nosi oznaku SQUARES.

Deterministička zamena slova dovodi do određenih periodičnosti kod niza reči $S(G)$ $\mathcal{D}0\mathcal{L}$ sistema. Pretpostavimo da se neka reč pojavljuje dva puta u nizu izvođenja: $w_i = w_{i+j}$, za neko $i \geq 0$ i $j \geq 1$. Zbog determinizma reči koje slede nakon w_{i+j} podudaraju se sa onima koje slede nakon w_i, posebno,

$$w_i = w_{i+j} = w_{i+2j} = w_{i+3j} = \ldots$$

Dakle nakon neke ,,početne zbrke", odnosno počev od reči w_i, reči u nizu $S(G)$ počinju periodično da se ponavljaju, odakle direktno sledi da je jezik $L(G)$ konačan. Obrnuto, ako je jezik $\mathcal{D}0\mathcal{L}$ sistema konačan onda njegov niz mora biti periodičan počev od nekog $i \in \mathbb{N}_0$, što znači da je periodičnost niza $S(G)$ potreban i dovoljan uslov da jezik $L(G)$ bude konačan. U sledećem tvrđenju dajemo, bez dokaza, još neke rezultate vezane za periodičnost niza $S(G)$ $\mathcal{D}0\mathcal{L}$ sistema. Sa $alph(w)$ označavamo minimalnu azbuku koja sadrži sva slova koja se pojavljuju u reči w, sa $pref_k(w)$ prefiks reči w dužine k ili samu reč w, ako je $|w| < k$, $suf_k(w)$ se definiše slično.

Teorema 2.26 *Neka je w_i, $i \geq 0$ niz $\mathcal{D}0\mathcal{L}$ sistema $G = (W,\ h,\ w)$. Tada skupovi $W_i = alph(w_i)$, $i \geq 0$, formiraju niz koji je periodičan počev od nekog $q \in \mathbb{N}_0$, odnosno postoji neko $p > 0$ takvo da važi $W_i = W_{i+p}$ za sve $i \geq q$. Svako slovo koje se pojavljuje u nekom W_i pojavljuje se i u nekom W_j za čiji indeks j važi $j \leq card(W) - 1$. Ako je $L(G)$ beskonačan onda postoji pozitivan ceo broj t takav da, za svako $k > 0$, postoji $n > 0$ takvo da, za sve $i \geq n$ i $m \geq 0$,*

$$pref_{k-1}(w_i) = pref_{k-1}(w_{i+mt}) \ i \ suf_{k-1}(w_i) = suf_{k-1}(w_{i+mt}).$$

Dakle, ako je jezik sistema beskonačan, prefiksi (sufiksi) bilo koje izabrane dužine formiraju niz koji je periodičan počev od nekog n-tog prefiksa (sufiksa) u nizu, pri čemu je period nezavisan od izabrane dužine. Od dužine prefiksa (sufiksa) zavisi samo "početna zbrka".

Definicija 2.27 *Beskonačan niz reči w_i, $i \geq 0$, je* ***lokalno konkatenativan*** *ako je za neke pozitivne cele brojeve k, i_1, $i_2, \ldots, i_k$ i $q \geq \max(i_1, i_2, \ldots, i_k)$*

$$w_n = w_{n-i_1} \ldots w_{n-i_k}, \ kad \ god \ je \ n \geq q.$$

$\mathcal{D}0\mathcal{L}$ sistem G je lokalno konkatenativan akko je niz $S(G)$ lokalno konkatenativan.

Lokalno konkatenativni $\mathcal{D}0\mathcal{L}$ sistemi su veoma bitni jer je njihovo proučavanje otvorilo nove grane teorije L-sistema. Tipičan primer lokalno konkatenativnog $\mathcal{D}0\mathcal{L}$ sistema je sistem FIB iz prethodnog primera. Sistem EXP_2 je takođe lokalno konkatenativan jer niz S(EXP_2) zadovoljava

$$w_n = w_{n-1}w_{n-1}, \ za \ sve \ n \geq 1.$$

Za sada nije poznato da li postoji algoritam koji za dati $\mathcal{D}0\mathcal{L}$ sistem daje odgovor na pitanje da li je on lokalno konkatenativan ili ne. Algoritam je nađen samo za neke specijalne slučajeve.

Definicija 2.28 *0$\mathcal{L}$ sistem $G = (W, \sigma, w_0)$ je $\mathcal{PD}0\mathcal{L}$ ako je σ morfizam i λ-slobodna.*

Primetimo da su svi $\mathcal{D}0\mathcal{L}$ sistemi iz prethodnog primera ujedno i $\mathcal{PD}0\mathcal{L}$. Sada ćemo dati neka jednostavna zapažanja koja se odnose na razliku između $\mathcal{D}0\mathcal{L}$ i $\mathcal{PD}0\mathcal{L}$ sistema.

Za reči niza $S(G)$ svakog $\mathcal{PD}0\mathcal{L}$ sistema G važi

$$|w_i| \leq |w_{i+1}|, \ za \ sve \ i \leq 0, \tag{2.1}$$

jer nema produkcija oblika $a \to \lambda$. Primetimo još da ovaj niz ne sadrži praznu reč. Niz $\mathcal{D}0\mathcal{L}$ sistema iz primera 2.20(b) $S(DEATH_b)$ sastoji se iz svih reči $(ab^2a)^{2^i}$, $i \geq 0$ poređanih u rastućem redosledu po dužini reči. Tvrdimo da se jezik $L(DEATH_b)$ ne može generisati nijednim $\mathcal{PD}0\mathcal{L}$ sistemom G. Zaista, ab^2a bi morala biti početna reč takvog G, a $ab^2a \Rightarrow ab^2aab^2a$ prvi korak u izvođenju, jer je niz $S(DEATH_b)$ strogo rastući po dužini reči, a zbog (2.1) $S(G)$ mora biti neopadajući. Zbog determinizma oba slova a u reči ab^2a moraju produkovati istu podreč u reči ab^2aab^2a. To će se dogoditi samo ako je produkcija za a jedna od sledećih: $a \to \lambda$, $a \to ab^2a$, $a \to a$. Prve dve opcije nam ne odgovaraju jer daju sistem koji nije $\mathcal{PD}0\mathcal{L}$, ne odgovara nam ni treća jer nijedno pravilo za slovo b ne čini korak $b \Rightarrow b^2aab^2$ mogućim. Zaključujemo da je nemoguće generisati jezik $L(DEATH_b)$ pomoću $\mathcal{PD}0\mathcal{L}$ sistema. Sada je jasno da imamo strogu inkluziju ove dve klase jezika, odnosno da važi $\mathcal{PD}0\mathcal{L} \subset \mathcal{D}0\mathcal{L}$.

$\mathcal{D}0\mathcal{L}$ sistem G_1 zadat aksiomom aba i produkcijama $a \to aba$, $b \to \lambda$ generise isti niz reči jezika kao i $\mathcal{PD}0\mathcal{L}$ sistem G_2 zadat aksiomom aba i produkcijama $a \to a$, $b \to baab$, tj. važi $S(G_1) = S(G_2)$.

Definicija 2.29 *$\mathcal{D}0\mathcal{L}$ sistemi G_1 i G_2 su:*
(a) ***nizovno ekvivalentni*** *akko $S(G_1) = S(G_2)$,*
(b) ***ekvivalentni*** *akko $L(G_1) = L(G_2)$.*

Ovi pojmovi se analogno definišu i za sve ostale klase L-sistema, stim što se nizovna ekvivalencija može definisati samo za sisteme čiji se jezik generiše u određenom redosledu, kao niz. Sistemi G_1 i G_2 iz prethodnog paragrafa su ekvivalentni i nizovno ekvivalentni. Jasno iz nizovne ekvivalencije sledi ekvivalencija, ali obrnuto ne mora da važi. Sledeća dva $\mathcal{D}0\mathcal{L}$ sistema su ekvivalentna ali nisu nizovno ekvivalentna:

$$(\{a,b\},\ a \to b^2,\ b \to a,\ b)\ i\ (\{a,b\},\ a \to b,\ b \to a^2,\ a).$$

Među najvećim matematičim problemima sedamdesetih godina prošloga veka, vezanim za L-sisteme, bio je ***problem ekvivalencije $\mathcal{D}0\mathcal{L}$ sistema***: konstruisati algoritam koji za dva data $\mathcal{D}0\mathcal{L}$ sistema G_1 i G_2 odlučuje da li su oni ekvivalentni ili ne. Dodatno, za $\mathcal{D}0\mathcal{L}$ sisteme imamo i ***problem nizovne ekvivalencije***: da li je $S(G_1) = S(G_2)$ za date sisteme G_1 i G_2? Nielsen[2] je 1974. godine dokazao da rešenje jednog problema vodi ka rešenju drugog i obratno. Zatim su Culik[3] i Fris 1977. godine, oslanjajući se na Nielsen-ove rezultate, dokazali da su ovi problemi odlučivi. Za detalje pogledati [17] i [18].

Definicija 2.30 *$\mathcal{T}0\mathcal{L}$ sistem je uređena trojka*

$$\widetilde{G} = (W, \widetilde{P}, w_0),$$

gde je $\widetilde{P}$ konačan skup konačnih substitucija, takav da je za svako $\sigma \in \widetilde{P}$, trojka (W, σ, w_0) $0\mathcal{L}$ sistem. Jezik $\mathcal{T}0\mathcal{L}$ sistema, $L(G)$, sastoji se iz reči w_0 i svih reči u svim jezicima $\sigma_1 \ldots \sigma_k(w_0)$, gde je $k \geq 1$ i svako σ_i pripada skupu $\widetilde{P}$ – neki σ_i mogu biti isti. Ako su sve konačne substitucije u $\widetilde{P}$ morfizmi onda je G deterministički, odnosno $\mathcal{DT}0\mathcal{L}$.

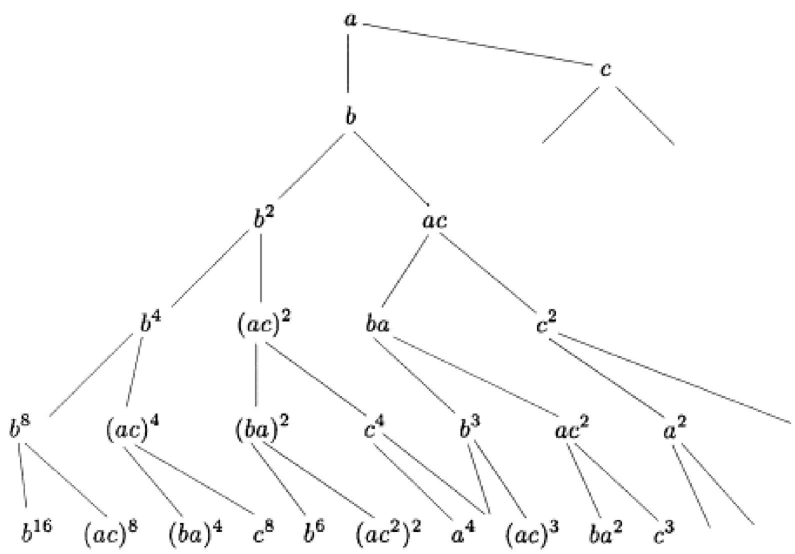
[2]Mogens Nielsen, danski informatičar rođen 1949. godine.
[3]Karel Culik II, profesor emeritus Univerziteta u Južnoj Karolini, rođen 1934. godine.

Prema gornjoj definiciji, nema ograničenja u primeni tablica – tablice se mogu koristiti u proizvoljnom redosledu i proizvoljno mnogo puta, pa se jezik $\mathcal{DT}0\mathcal{L}$ sistema neće generisati nizovno. $\mathcal{T}0\mathcal{L}$ sistem se zadaje navođenjem početne reči i svih njegovih tablica. Produkcije koje pripadaju istoj tablici navode se unutar zagrada [,]. Jedna ista produkcija se može pojavljivati u više tablica. Za razliku od problema ekvivalencije $\mathcal{D}0\mathcal{L}$ sistema, **problem ekvivalencije $\mathcal{DT}0\mathcal{L}$ sistema** je neodlučiv. Ovo je dokazao Rozenberg[4] 1972. godine ([19]).

Primer 2.31 *Razmatramo $\mathcal{DT}0\mathcal{L}$ sistem PAL čija je početna reč $w_0 = a$ i tablice produkcija*

$$T_1 = [a \to b,\ b \to b^2,\ c \to a] \text{ i } T_2 = [a \to c,\ b \to ac,\ c \to c].$$

Umesto linearnim nizom, izvođenja se mogu predstaviti drvetom kao na sledećoj slici:



Slika 2.3: Drvo izvođenja za $\mathcal{T}0\mathcal{L}$ sistem PAL.

Ovde grana ukazuje na to koja od tablica je korišćena: levi potomak reči je rezultat primene tablice T_1, a desni tablice T_2. Ako potomak reči nije označen (kao što nisu označeni potomci a i c reči c, na trećem nivou) to znači da se on već pojavio u prethodnim nivoima izvođenja. Nastavak se u tom slučaju može izostaviti ako nekoga interesuje samo određivanje jezika. Sve reči $a^i, b^i, c^i,\ i \geq 1$ se pojavljuju negde u drvetu, što zaključujemo na sledeći način. Uočimo prvo da za sve $i \geq 0$ reč ba^{i+1} dobijamo tako što na reč ba^i primenimo prvo T_2, pa T_1. Sada iz reči ba^i dobijamo reč b^{i+2} primenom T_1, reč c^{i+2} primenom T_2 dva puta, i reč a^{i+2} iz c^{i+2} primenom T_1. Dalje, krajnji levi ekstremitet drveta sadrži sve reči $b^{i^2},\ i \geq 0$ itd.

[4]Grzegorz Rozenberg, poljski informatičar, rođen 1942. godine.

2.5 Mehanizmi filtriranja

Jedna od bitnih odlika $\mathcal{T}0\mathcal{L}$ i $0\mathcal{L}$ sistema jeste da ne možemo izostaviti nijednu reč dobijenu u procesu izvođenja. To znači da pri modeliranju biljaka i pojava L-sistemima iz ovih klasa u slučaju pojave strukture koja se ne uklapa u model ne bismo imali način da taj deo uklonimo.

U teoriji L-sistema, kao i u teoriji formalnih jezika, uobičajena je upotreba ***mehanizama filtriranja***. Oni omogućavaju da se iz skupa svih reči dobijenih u procesu izvođenja izdvoje samo one reči koje nam odgovaraju. U ovom odeljku ćemo se upoznati sa ukupno pet mehanizama filtriranja i njihovim osnovnim svojstvima.

Najpoznatiji od svih mehanizama filtriranja je mehanizam proširivanja. Nosi oznaku $\mathcal{E}$ (engl. „extended") i omogućava upotrebu neterminalnih simbola. Stoga, $\mathcal{E}0\mathcal{L}$ sistem je $0\mathcal{L}$ sistem čija se azbuka sastoji iz skupa terminalnih i skupa neterminalnih simbola koji su disjunktni. $\mathcal{E}0\mathcal{L}$ i $0\mathcal{L}$ sistemi rade na isti način, ali u jezik $\mathcal{E}0\mathcal{L}$ sistema ulaze samo reči nad skupom terminalnih simbola. Neterminalne simbole označavamo isto kao i kod formalnih gramatika.

Definicija 2.32 *$\mathcal{E}0\mathcal{L}$ sistem je uređena četvorka $G = (W, \sigma, w_0, A)$ gde su W i A azbuke takve da $A \subseteq W$, σ je konačna substitucija nad W, $w_0 \in W^*$ početna reč i $G' = (W, \sigma, w_0)$ $0\mathcal{L}$ sistem. Njegov jezik je*

$$L(G) = \{w \in A^* \mid w = \sigma^i(w_0) \text{ za neko } i \geq 0\}.$$

Ako je σ morfizam uređena četvorka G je $\mathcal{ED}0\mathcal{L}$ sistem.

Primer 2.33 *$\mathcal{E}0\mathcal{L}$ sistem SYNCHRO zadat je aksiomom ABC i produkcijama:*

$$\begin{array}{l}
A \to AA',\ A \to a,\ A' \to A',\ A' \to a,\ a \to F,\\
B \to BB',\ B \to b,\ B \to B',\ B \to b,\ b \to F,\\
C \to CC',\ C \to c,\ C \to C',\ C \to c,\ c \to F,\\
F \to F.
\end{array}$$

*Ovaj sistem je **sinhronizovan** u smislu da svi terminalni simboli moraju biti dostignuti istovremeno. Inače, simbol F dospeva u reč i ne može se eliminisati, pa reč ne ulazi u jezik. Sada je lako žaključiti da je*

$$L(SYNCHRO) = \{a^n b^n c^n \mid n \geq 1\}.$$

Dobijeni jezik je klasičan primer kontekstno osetljivog jezika koji nije kontekstno slobodan u teoriji jezika.

Mehanizmi filtriranja daju klase L-jezika koje su zatvorene za brojne operacije. Čiste klase L-jezika, kao što su $\mathcal{D}0\mathcal{L}$ i $0\mathcal{L}$ jezici imaju veoma slabo svojstvo zatvorenosti za operacije. Većina jezičko-teorijskih operacija može proizvesti reči koje su izvan klase jezika na koju se primenjuje operacija.

Definicija 2.34 *Klasa L-jezika se naziva potpuno AFL (engl. „**AFL- abstract family of languages**") akko je L zatvorena za svaku od sledećih operacija: sabiranje, konkatenacija, iteracija, morfizam, inverzni morfizam, presek sa regularnim jezicima. Klasa jezika L se naziva anti-AFL akko nije zatvorena ni za jednu od navedenih operacija.*

Teorema 2.35 *Klasa 0$\mathcal{L}$ jezika je anti-AFL, kao i klasa $\mathcal{D}$0$\mathcal{L}$ jezika. Klasa $\mathcal{E}$0$\mathcal{L}$ je zatvorena za sve AFL operacije, osim za inverzni morfizam.*

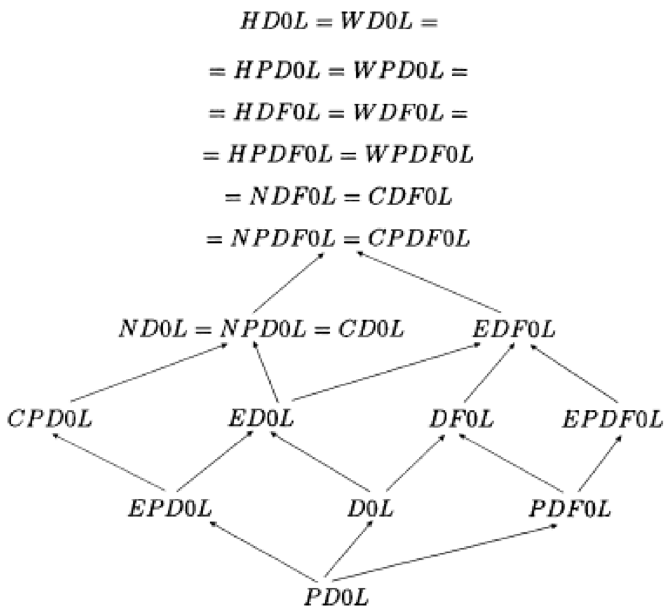
Ova teorema jasno pokazuje moć $\mathcal{E}$-mehanizma da klasu slabe strukture (u smislu slabih svojstava zatvorenosti) transformiše u jaku strukturu. Moć varira među klasama L-sistema. Razlika između $\mathcal{D}0\mathcal{L}$ i $\mathcal{ED}0\mathcal{L}$ jezika nije toliko velika. Ako je niz $S(G')$ $\mathcal{D}0\mathcal{L}$ sistema G' krajnje periodičan, takav će biti i niz $S(G)$ $\mathcal{ED}0\mathcal{L}$ sistema G koji ga filtrira. Takođe, rezultati u vezi periodičnosti iz teoreme 2.26 o azbukama vazi i za $\mathcal{ED}0\mathcal{L}$ nizove.

Sada ćemo definisati i preostala četiri mehanizma filtriranja koji nose oznake redom $\mathcal{H}$ (engl. ,,***homomorphism***"), $\mathcal{N}$(engl. ,,***nonerasing***"), $\mathcal{C}$(engl. ,,***coding***"), $\mathcal{W}$(engl. ,,***weak coding***"). Primena svakog od njih na jezik nekog L-sistema, daje morfičnu sliku tog jezika.

Definicija 2.36 *Uređena četvorka $G = (W, \sigma, w_0, g)$ takva da je $G' = (W, \sigma, w_0)$ 0$\mathcal{L}$ sistem je:*
(a) $\mathcal{H}0\mathcal{L}$ sistem ako je g morfizam na W,
(b) $\mathcal{N}0\mathcal{L}$ sistem ako je g λ–slobodan morfizam na W,
(c) $\mathcal{C}0\mathcal{L}$ sistem ako je g kodiranje na W,
(d) $\mathcal{W}0\mathcal{L}$ sistem ako je g slabo kodiranje na W,
Analogno se definišu $\mathcal{HD}0\mathcal{L}$, $\mathcal{ND}0\mathcal{L}$, $\mathcal{CD}0\mathcal{L}$, $\mathcal{WD}0\mathcal{L}$ sistemi, kada je G' $\mathcal{D}0\mathcal{L}$ sistem. Jezik koji generiše svaki od ovih sistema je

$$L(G) = \{g(w) \mid w \in L(G')\}.$$

Svi navedeni mehanizmi filtriranja se analogno definišu za $\mathcal{T}0\mathcal{L}$, $\mathcal{F}0\mathcal{L}$ i $\mathcal{TF}0\mathcal{L}$ sisteme.



Slika 2.4: Odnos determinističkih klasa 0$\mathcal{L}$ jezika.

Do sada smo se upoznali sa 16 klasa L-sistema kod kojih je zamena slova kontekstno slobodna (ali ne moraju generisati kontekstno slobodan jezik). S' obzirom na to da je pravilo da se na svaki L-sistem može primeniti najviše jedan mehanizam filtriranja sada ćemo imati $16 \cdot 6 = 96$ klasa L-sistema sa kontekstno slobodnom zamenom (šesta opcija je da ne bude primenjen nijedan mehanizam filtriranja). Sledeća tvrđenja se odnose na međusobne odnose njima odgovarajućih klasa jezika. Ove klase poredimo u skladu sa λ-konvencijom: dva jezika se smatraju jednakima ako se međusobno razlikuju samo po praznoj reči. Inače bi se reprodukcijske klase (one koje imaju svojstvo $\mathcal{P}$) automatski razlikovale od nereprodukcijskih (bez svojstva $\mathcal{P}$).

Teorema 2.37 *Dijagram na slici 2.4 karakteriše međusobne odnose klasa determinističkih kontekstno slobodnih L-sistema. Strelice označavaju striktnu inkluziju. Familije koje nisu povezane stazom su međusobno neuporedive.*

Ono što je iznenađujuće kod nedeterminističkih klasa je da je $\mathcal{CP}0\mathcal{L}$ pravi podskup klase $\mathcal{EP}0\mathcal{L}$, iako je $\mathcal{E}0\mathcal{L} = \mathcal{C}0\mathcal{L}$, što je suprotno od onoga što bi neko mogao da očekuje znajući deterministički slučaj. Ključni rezultati za nedeterminističke klase $0\mathcal{L}$ jezika su dati u sledećoj teoremi.

Teorema 2.38 *Sledeće klase su jednake sa klasom $\mathcal{E}0\mathcal{L}$:*

$$\begin{gathered}\mathcal{E}0\mathcal{L} = \mathcal{C}0\mathcal{L} = \mathcal{N}0\mathcal{L} = \mathcal{W}0\mathcal{L} = \mathcal{H}0\mathcal{L} = \mathcal{NP}0\mathcal{L} = \mathcal{EP}0\mathcal{L} = \\ \mathcal{WP}0\mathcal{L} = \mathcal{HP}0\mathcal{L} = \mathcal{EF}0\mathcal{L} = \mathcal{CF}0\mathcal{L} = \mathcal{NF}0\mathcal{L} = \mathcal{WF}0\mathcal{L} = \\ \mathcal{HF}0\mathcal{L} = \mathcal{EPF}0\mathcal{L} = \mathcal{NPF}0\mathcal{L} = \mathcal{WPF}0\mathcal{L} = \mathcal{HPF}0\mathcal{L}\end{gathered}$$

Klasa $\mathcal{E}0\mathcal{L}$ leži striktno između kontekstno slobodnih (CF) i kontekstno osetljivih (CS) jezika i sadrži međusobno neuporedive klase $\mathcal{CP}0\mathcal{L}$ i $\mathcal{F}0\mathcal{L}$.

Dakle, klasa $\mathcal{E}0\mathcal{L}$ jezika sadrži klasu kontekstno slobodnih jezika. Ovo važi jer se kontekstno slobodne gramatike hijerarhije Chomsky mogu transformisati u $\mathcal{E}0\mathcal{L}$ sisteme dodavanjem produkcija $a \to a$ za svako slovo $a \in W$. Da je ova inkluzija prava sledi iz primera 2.33 gde je dat $\mathcal{E}0\mathcal{L}$ sistem koji generiše kontekstno osetljiv jezik (videti i primer 2.12). Ova činjenica bi trebala biti u suprotnosti sa činjenicom da je veliki deo klase konačnih jezika van klase 0L jezika.

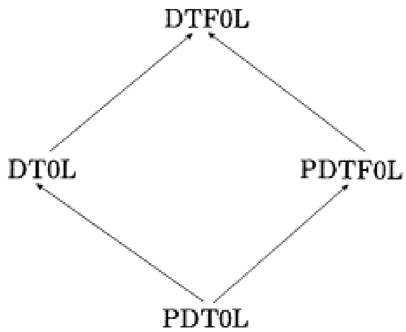
U teoriji formalnih jezika je uobičajeno da se gramatike redukuju u normalnu formu, tj. da se pokaže da se svaka gramatika može zameniti ekvivalentnom gramatikom koja ima neka poželjna svojstva. Sledeća teorema je ilustracija takve redukcije kod L-sistema. Za detalje o normalnim formama formalnih gramatika videti [13].

Teorema 2.39 *Za svaki $\mathcal{E}0\mathcal{L}$ jezik postoji $\mathcal{E}0\mathcal{L}$ sistem koji ga generiše, takav da je:*
(a) jedina produkcija za svako terminalno slovo a je $a \to F$, gde je F neterminalni simbol za koji je $F \to F$ jedina produkcija,
(b) početna reč je neterminalni simbol koji se ne pojavljuje ni u jednoj produkciji sa desne strane,
(c) desna strana svake produkcije je ili terminalna ili neterminalna reč,
(d) terminalna reč je dostižna iz svakog neterminala osim iz F (i početne reči ako je jezik prazan).

Uočimo da je posebno svojstvo $\mathcal{E}0\mathcal{L}$ sistema *SYNCHRO* o terminalima zapravo generalno svojstvo $\mathcal{E}0\mathcal{L}$ jezika (ovo se odnosi na to da sva terminalna slova moraju biti dostignuta istovremeno-svojstvo sinhronizacije).

Sledeće dve teoreme sumiraju rezultate o međusobnim odnosima klasa tabelarnih L-sistema. Prva teorema se bavi determinističkim tabelarnim klasama, a druga nedeterminističkim.

Teorema 2.40 *Važe sledeće inkluzije i jednakosti:*
$\mathcal{DT}0\mathcal{L} \subset \mathcal{CDT}0\mathcal{L} = \mathcal{NDT}0\mathcal{L} = \mathcal{EDT}0\mathcal{L} = \mathcal{WDT}0\mathcal{L} = \mathcal{HDT}0\mathcal{L}$,
$\mathcal{PDT}0\mathcal{L} \subset \mathcal{CPDT}0\mathcal{L} \subseteq \mathcal{NPDT}0\mathcal{L} \subseteq \mathcal{EPDT}0\mathcal{L} = \mathcal{WPDT}0\mathcal{L} = \mathcal{HPDT}0\mathcal{L}$,
$\mathcal{DTF}0\mathcal{L} \subset \mathcal{CDTF}0\mathcal{L} = \mathcal{NDTF}0\mathcal{L} = \mathcal{EDTF}0\mathcal{L} = \mathcal{WDTF}0\mathcal{L} = \mathcal{HDTF}0\mathcal{L}$,
$\mathcal{PDTF}0\mathcal{L} \subset \mathcal{CPDTF}0\mathcal{L} \subseteq \mathcal{NPDTF}0\mathcal{L} \subseteq \mathcal{EPDTF}0\mathcal{L} = \mathcal{WPDTF}0\mathcal{L} =$
$= \mathcal{HPDTF}0\mathcal{L}$.
Čiste klase (bez filtera) zadovoljavaju sledeći dijagram inkluzije:



Teorema 2.41 *Svaka od sledećija klasa jednaka je klasi* $\mathcal{ET}0\mathcal{L}$*:*
$\mathcal{ET}0\mathcal{L} = \mathcal{CT}0\mathcal{L} = \mathcal{NT}0\mathcal{L} = \mathcal{WT}0\mathcal{L} = \mathcal{HT}0\mathcal{L} = \mathcal{NPT}0\mathcal{L} = \mathcal{EPT}0\mathcal{L} =$
$= \mathcal{WPT}0\mathcal{L} = \mathcal{HPT}0\mathcal{L} = \mathcal{CTF}0\mathcal{L} = \mathcal{NTF}0\mathcal{L} = \mathcal{ETF}0\mathcal{L} = \mathcal{WTF}0\mathcal{L} =$
$= \mathcal{HTF}0\mathcal{L} = \mathcal{NPTF}0\mathcal{L} = \mathcal{EPTF}0\mathcal{L} = \mathcal{WPTF}0\mathcal{L} = \mathcal{HPTF}0\mathcal{L}$.
Klase $\mathcal{E}0\mathcal{L}$, $\mathcal{TF}0\mathcal{L}$ *i* $\mathcal{CPT}0\mathcal{L} = \mathcal{CPTF}0\mathcal{L}$ *su pravi podskupovi klase* $\mathcal{ET}0\mathcal{L}$.

Klasa $\mathcal{ET}0\mathcal{L}$ je jedna od najšire proučavanih L-klasa u kojima je zamena slova kontekstno slobodna. Ova klasa ima jaka svojstva zatvaranja.

Teorema 2.42 *Klasa* $\mathcal{ET}0\mathcal{L}$ *je potpuno AFL, dok je klasa* $\mathcal{T}0\mathcal{L}$ *anti-AFL. Za svaki* $\mathcal{ET}0\mathcal{L}$ *jezik postoji* $\mathcal{ET}0\mathcal{L}$ *sistem sa dve tabele koji ga generiše. Takođe, za svaki* $\mathcal{ET}0\mathcal{L}$ *jezik postoji* $\mathcal{ET}0\mathcal{L}$ *sistem takav da je* $a \to F$ *jedina produkcija u svakoj tabeli za svako terminalno slovo* a *i* $F \to F$ *je jedino pravilo u svakoj tabeli za neterminal* F.

Dva rezultata iz prethodne teoreme vezana za normalne forme, sinhronizacija i uslov o dve tabele, ne mogu uvek biti ostvarena istovremeno.

2.6 Beskonačne reči

Neka je W azbuka. Sa W^ω označavamo skup svih ***beskonačnih reči*** nad W. Za nepraznu reč u sa u^ω označena je beskonačna reč $uuu\ldots$. Takva reč se naziva

čisto periodična. Beskonačna reč oblika uv^ω, gde su u i v reči i $v \neq \lambda$, naziva se ***na kraju periodična***. Za reč $u \in W^*$ kažemo da je ***smrtnik*** morfizma h ako postoji $m \geq 0$ takvo da je $h^m(u) = \lambda$. Morfizam h ***proširuje*** (engl. h is prolongable on a) slovo a ako je $h(a) = au$ za neko $u \in W^*$ koje nije smrtnik morfizma h. U ovom slučaju $h^\omega(a)$ označava beskonačnu reč $auh(u)h^2(u)\ldots$. Beskonačna reč α je ***čisto morfična*** ako postoje morfizam h i slovo a takvi da h proširuje a i da je $\alpha = h^\omega(a)$. Beskonačna reč α je ***morfična*** ako postoje morfizam h, kodiranje e i slovo a takvi da h proširuje a i da je $\alpha = e(h^\omega(a))$. Svaka čisto periodična reč je i čisto morfična, i svaka na kraju periodična reč je morfična.

Definicija 2.43 *Jezik L takav da je za sve $u, v \in L$, u prefiks od v ili v prefiks od u, naziva se* ***prefiksni jezik****. Jezik L određuje beskonačnu reč α akko je L beskonačan i svako $u \in L$ prefiks od α.*

Na primer, beskonačan prefiksni jezik $\{\lambda, ab, abab, ababab\ldots\}$ određuje beskonačnu reč $(ab)^\omega$. Sledeća tvrđenja su osnovne posledice gornje definicije.

Teorema 2.44 *Jezik L određuje najvše jednu beskonačnu reč.*

Teorema 2.45 *Jezik L određuje beskonačnu reč akko je L beskonačan prefiksni jezik.*

Napominjemo da, dok jezik L određuje najviše jednu beskonačnu reč, beskonačnu reč može oređivati više od jednog jezika.

Teorema 2.46 *Ako jezik L određuje beskonačnu reč α i L' je beskonačan podskup od L, onda i L' određuje α.*

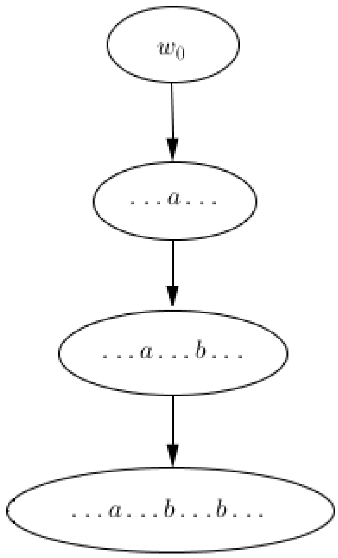
Za klasu jezika C, neka je

$$\omega(C) = \{\alpha | \alpha \text{ je beskonačna reč i neko } L \in C \text{ određuje } \alpha\}.$$

Za L-sistem G kažemo da je ***beskonačan*** ako je njegov jezik $L(G)$ beskonačan.

Skup svojstava S $0\mathcal{L}$ sistema je podskup od $\{D, P, F, T\} \cup \{E, C, N, W, H\}$ koji sadrži najviše jedan element drugog skupa, tj. najviše jedan mehanizam filtriranja. Neka je $L(S)$ klasa jezika $0\mathcal{L}$ sistema sa skupom svojstava S. Na primer, $L(\{C, D, T\}) = \mathcal{CDT}0\mathcal{L}$. Iz definicija svojstava dobijamo sledeće inkluzije.

Teorema 2.47 ***(Strukturne inkluzije)*** *Neka je S skup svojstava L-sistema. Tada:*
(a) $L(S \cup \{D\}) \subseteq L(S)$,
(b) $L(S \cup \{P\}) \subseteq (S)$,
(c) $L(S) \subseteq L(S \cup \{F\})$,
(d) $L(S) \subseteq L(S \cup \{T\})$.
Neka je S skup svojstava koji ne sadrži nijedan element iz skupa $\{E, C, N, W, H\}$. *Tada:*
(a) $L(S) \subseteq L(S \cup \{E\})$,
(b) $L(S) \subseteq L(S \cup \{C\})$,
(c) $L(S \cup \{C\}) \subseteq L(S \cup \{N\}) \subseteq L(S \cup \{H\})$,
(d) $L(S \cup \{C\}) \subseteq L(S \cup \{W\}) \subseteq L(S \cup \{H\})$.



Definicija 2.48 *$\mathcal{T}0\mathcal{L}$ sistem $G = (W, \widetilde{P}, w_0)$ se može ,,**napumpati**" ako postoje $a, b \in W$ takvi da:*
(a) neka reč $w \in L(G)$ sadrži a,
(b) za neku kompoziciju t, tabela iz $\widetilde{P}$, u $t(a)$ postoji reč w_1 u kojoj imamo različita pojavljivanja slova a i b, a u $t(b)$ reč w_2 u kojoj se pojavljuje slovo b.

Na gornjoj slici je prikazan ,,proces napumpavanja".

Teorema 2.49 *Pretpostavimo da se $\mathcal{T}0\mathcal{L}$ sistem $G = (W, \widetilde{P}, w_0)$ može napumpati. Tada je njegov jezik $L(G)$ beskonačan.*

Dokaz. Za reč $w \in L(G)$ i kompoziciju tabela t iz prethodne definicije važi $t^i(w) \subseteq L(G)$ za sve $i \geq 0$. Indukcijom se može jednostavno pokazati da za sve $i \geq 0$, $t^i(w)$ sadrži reč u kojoj se pojavljuje slovo a i bar i kopija slova b. Stoga je $L(G)$ beskonačan. $\square$

Sledeću teoremu navodimo bez dokaza (dokaz se može naći u [6]).

Teorema 2.50 *Pretpostavimo da je $\mathcal{T}0\mathcal{L}$ sistem $G = (W, \widetilde{P}, w_0)$ beskonačan. Tada se G može napumpati.*

Posledica 2.51 *$\mathcal{T}0\mathcal{L}$ sistem $G = (W, \widetilde{P}, w_0)$ je beskonačan akko se može napumpati.*

Dokaz. Direktno iz prethodne dve teoreme. $\square$

Posledica 2.52 *$0\mathcal{L}$ sistem $G = (W, \sigma, w_0)$ je beskonačan akko postoje $a, b \in W$ takvi da:*
(a) neka reč $w \in L(G)$ sadrži a,
(b) za neko $i \geq 0$ u $\sigma^i(a)$ postoji reč w_1 u kojoj imamo različita pojavljivanja slova a i b, a u $\sigma^i(b)$ reč w_2 u kojoj se pojavljuje slovo b.

Posledica 2.53 *$\mathcal{DT}0\mathcal{L}$ sistem $G = (W, \widetilde{P}, w_0)$ je beskonačan akko postoje $a, b \in W$ takvi da:*
(a) neka reč $w \in L(G)$ sadrži a,
(b) za neku kompoziciju $\tilde{h}$, morfizama iz $\widetilde{P}$, u $\tilde{h}(a)$ postoji reč w_1 u kojoj imamo različita pojavljivanja slova a i b, a u $\tilde{h}(b)$ reč w_2 u kojoj se pojavljuje slovo b.

Prethodne dve posledice slede direktno iz posledice 2.51 i činjenice da su svi $0\mathcal{L}$ i $\mathcal{DT}0\mathcal{L}$ sistemi ujedno i $\mathcal{T}0\mathcal{L}$ sistemi.

Dalje ćemo pokazati da svi beskonačni jezici iz pojedinih klasa L-jezika imaju beskonačne podskupove koji pripadaju nekim manjim klasama L-jezika.

Teorema 2.54 *Svaki beskonačan $\mathcal{T}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup.*

Dokaz. Neka je $G = (W, \widetilde{P}, w_0)$ proizvoljan $\mathcal{T}0\mathcal{L}$ sistem koji generiše beskonačan jezik L(G). Iz teoreme 2.50 sledi da se ovaj sistem može napumpati za neko $a, b \in W$, $w, w_1, w_2 \in W^*$ i kompoziciju tabela t. Neka je h morfizam na W takav da je $h(a) = w_1$, $h(b) = w_2$ osim ako je $a = b$, za sva ostala slova $c \in W$ $h(c) = \tilde{w}$ za neko $\tilde{w} \in t(c)$. Razmatramo jezik L' $\mathcal{D}0\mathcal{L}$ sistema $G' = (W, h, w)$. Za sve $c \in W$, $h(c)$ je u $t(c)$, pa pošto je w u L, imamo $L' \subset L$. Indukcijom se jednostavno pokazuje da za sve $i \geq 0$, $h^i(w)$ sadrži slovo a i bar i kopija slova b. Stoga je L' beskonačan $\mathcal{D}0\mathcal{L}$ podskup od L. $\Box$

Teorema 2.55 *Svaki beskonačan $\mathcal{PT}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup.*

Dokaz. Neka je $G = (W, \widetilde{P}, w_0)$ $\mathcal{PT}0\mathcal{L}$ sistem koji generiše beskonačan jezik. Dokaz se izvodi slično kao u prethodnoj teoremi, pri čemu treba imati na umu da su sve tabele iz $\widetilde{P}$ λ-slobodne, pa će i kompozicija t biti takva, a morfizam h će biti λ-slobodan. $\Box$

Teorema 2.56 *Neka je $G = (W, h, w_0, A)$ beskonačan $\mathcal{ED}0\mathcal{L}$ sistem. Tada postoje $k \geq 0$ i $p \geq 1$ takvi da je jezik $\mathcal{D}0\mathcal{L}$ sistema $(W, h^p, h^k(w_0))$ beskonačan podskup od $L(G)$.*

Dokaz. Pošto je $L(G)$ beskonačan postoji $m \geq 0$ takvo da niz $w_0, h(w_0), h^2(w_0), \ldots, h^m(w_0)$ sadrži više od $2^{|A|}$ reči jezika $L(G)$. Za svako $w \in L(G)$ važi $alph(w) \subseteq A$. Stoga postoje $B \subseteq A$ i i, j takvi da $0 \leq i < j \leq m$ i $alph(h^i(w_0)) = alph(h^j(w_0)) = B$. Tada za svaku reč w za koju je $alph(w) = B$ važi $alph(h^{j-i}(w)) = B$. Uzmimo $k = i$ i $p = j - i$. Tada je za sve $n \geq 0$, $alph(h^{k+np}(w_0)) = B$. Stoga je za svako $n \geq 0$, $h^{k+pn}(w_0)$ u $L(G)$, pa imamo da za $\mathcal{D}0\mathcal{L}$ sistem $G' = (W, h^p, h^k(w_0))$ važi $L(G') \subseteq L(G)$. Pretpostavimo da $L(G')$ nije beskonačan odnosno da se neka reč w pojavljuje dva puta u nizu $S(G')$. Tada se ta reč pojavljuje dva puta i u nizu $S(G)$ odakle sledi da je $L(G)$ konačan, kontradikcija. Dakle, $L(G')$ je beskonačan podskup od $L(G)$. $\Box$

Posledica 2.57 *Svaki beskonačan $\mathcal{ED}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup.*

Posledica 2.58 *Svaki beskonačan $\mathcal{EPD}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup.*

Dokaz. Neka je $G = (W, h, w_0, A)$ beskonačan $\mathcal{EPD}0\mathcal{L}$ sistem. Prema prethodnoj teoremi postoje $k \geq 0$ i $p \geq 1$ takvi da je jezik $\mathcal{D}0\mathcal{L}$ sistema $G' =$

$(W, h^p, h^k(w_0))$ beskonačan podskup od $L(G)$. Pošto je h λ-slobodan morfizam i h^p će biti λ-slobodan morfizam. Stoga je $L(G')$ beskonačan $\mathcal{PD}0\mathcal{L}$ podskup od $L(G)$. $\square$

Teorema 2.59 *Svaki beskonačan $\mathcal{ET}0\mathcal{L}$ jezik ima beskonačan $\mathcal{CD}0\mathcal{L}$ podskup.*

Dokaz. Neka je L proizvoljan beskonačan $\mathcal{ET}0\mathcal{L}$ jezik. Iz teoreme 2.41 imamo da važi $\mathcal{ET}0\mathcal{L} = \mathcal{CT}0\mathcal{L}$, pa postoje kodiranje e i $\mathcal{T}0\mathcal{L}$ jezik L' takvi da je $L = e(L')$. Pošto je L beskonačan, mora biti beskonačan i L'. Tada prema teoremi 2.54 L' ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup L''. S'obzirom na to da je L'' beskonačan, a e kodiranje, $e(L'')$ je takođe beskonačan. Dalje, iz $L'' \subseteq L'$ sledi $e(L'') \subseteq e(L')$, pa konačno imamo da je $e(L'')$ beskonačan $\mathcal{CD}0\mathcal{L}$ podskup od L. $\square$

Teorema 2.60 *Neka je S skup svojstava $0\mathcal{L}$ sistema koji ne sadrži F. Tada svaki beskonačan jezik $L(S \cup \{F\})$ ima beskonačan podskup $L(S)$.*

Dokaz. Neka je G proizvoljan $0\mathcal{L}$ sistem sa skupom svojstava $S \cup \{F\}$. Pošto G ima konačan skup aksioma, $L(G)$ će biti konačna unija $L(S)$ jezika. Tada, pošto je $L(G)$ beskonačan, bar jedan od ovih $L(S)$ jezika mora biti beskonačan. Samim tim $L(G)$ ima beskonačan $L(S)$ podskup. $\square$

Teorema 2.61 *Neka su C i D klase jezika takve da svaki beskonačan jezik u C ima beskonačan podskup koji je u D. Tada je $\omega(C) \subseteq \omega(D)$.*

Dokaz. Neka je $\alpha \in \omega(C)$ uzeto proizvoljno. Tada u C postoji jezik L koji određuje reč α. Prema teoremi 2.45 L je beskonačan jezik i on prema pretpostavci ima beskonačan podskup L' u klasi D. Sada na osnovu teoreme 2.46 imamo da L' određuje α, odakle sledi da je $\alpha \in \omega(D)$. Konačno, $\omega(C) \subseteq \omega(D)$. $\square$

Dalje ćemo izvršiti kategorizaciju beskonačnih reči određenih svakom od klasa $0\mathcal{L}$ sistema, pomoću gornjih rezultata o beskonačnim podskupovima. 96 klasa $0\mathcal{L}$ sistema podelićemo u tri skupa, nazvana $Set_1,\ Set_2,\ Set_3$ i pokazati da za svako $C_1 \in Set_1,\ C_2 \in Set_2,\ C_3 \in Set_3$ važi

$$\omega(C_1) = \omega(\mathcal{PD}0\mathcal{L}),\ \omega(C_2) = \omega(\mathcal{D}0\mathcal{L}),\ \omega(C_3) = \omega(\mathcal{CD}0\mathcal{L}).$$

Uzmimo $Set_1 = \{\mathcal{PD}0\mathcal{L}, \mathcal{PDF}0\mathcal{L}, \mathcal{PF}0\mathcal{L}, \mathcal{PDT}0\mathcal{L}, \mathcal{PDTF}0\mathcal{L}, \mathcal{PT}0\mathcal{L}, \mathcal{PTF}0\mathcal{L}, \mathcal{P}0\mathcal{L}, \mathcal{EPD}0\mathcal{L}, \mathcal{EPDF}0\mathcal{L}\}$.

Teorema 2.62 *Za svako $C \in Set_1$, svaki beskonačan C jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup.*

Dokaz. Uzmimo proizvoljno $C \in Set_1$. Prema teoremi 2.47 o strukturnim inkluzijama imamo da je $C \subseteq \mathcal{PTF}0\mathcal{L}$ ili $C \subseteq \mathcal{EPDF}0\mathcal{L}$. Prema teoremi 2.60 svaki beskonačan $\mathcal{PTF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PT}0\mathcal{L}$ podskup. Prema teoremi 2.55 svaki beskonačan $\mathcal{PT}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. Stoga svaki beskonačan $\mathcal{PTF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. Prema teoremi 2.60 svaki beskonačan $\mathcal{EPDF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{EPD}0\mathcal{L}$ podskup. Prema posledici 2.58 svaki beskonačan $\mathcal{EPD}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. Stoga svaki $\mathcal{EPDF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. Sada je jasno da svaki beskonačan C jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. $\square$

Teorema 2.63 *Za svako $C \in Set_1$ važi $\omega(C) = \omega(\mathcal{PD}0\mathcal{L})$.*

Dokaz. Uzmimo $C \in Set_1$ proizvoljno. Prema teoremi o strukturnim inkluzijama imamo $\mathcal{PD}0\mathcal{L} \subseteq C$. Stoga $\omega(\mathcal{PD}0\mathcal{L}) \subseteq \omega(C)$. Prema prethodnoj teoremi svaki beskonačan C jezik ima beskonačan $\mathcal{PD}0\mathcal{L}$ podskup. Dalje, prema teoremi 2.61 imamo $\omega(C) \subseteq \omega(\mathcal{PD}0\mathcal{L})$. Sada je jasno da važi $\omega(C) = \omega(\mathcal{PD}0\mathcal{L})$. □

Uzmimo

$$Set_2 = \{\mathcal{D}0\mathcal{L}, \mathcal{DF}0\mathcal{L}, 0\mathcal{L}, \mathcal{F}0\mathcal{L}, \mathcal{DT}0\mathcal{L}, \mathcal{DTF}0\mathcal{L}, \mathcal{T}0\mathcal{L}, \mathcal{TF}0\mathcal{L}, \mathcal{ED}0\mathcal{L}, \mathcal{EDF}0\mathcal{L}\}.$$

Teorema 2.64 *Za svako $C \in Set_2$, svaki beskonačan C jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup.*

Dokaz. Uzmimo proizvoljno $C \in Set_2$. Prema teoremi 2.47 o strukturnim inkluzijama imamo da je $C \subseteq \mathcal{TF}0\mathcal{L}$ ili $C \subseteq \mathcal{EDF}0\mathcal{L}$. Prema teoremi 2.60 svaki beskonačan $\mathcal{TF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{T}0\mathcal{L}$ podskup. Prema teoremi 2.54 svaki beskonačan $\mathcal{T}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. Stoga svaki beskonačan $\mathcal{TF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. Prema teoremi 2.60 svaki beskonačan $\mathcal{EDF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{ED}0\mathcal{L}$ podskup. Prema posledici 2.57 svaki beskonačan $\mathcal{ED}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. Stoga svaki $\mathcal{EDF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. Konačno, svaki beskonačan C jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. □

Teorema 2.65 *Za svako $C \in Set_2$ važi $\omega(C) = \omega(\mathcal{D}0\mathcal{L})$.*

Dokaz. Neka je $C \in Set_2$ uzeto proizvoljno. Prema teoremi o strukturnim inkluzijama $\mathcal{D}0\mathcal{L} \subseteq C$. Stoga $\omega(\mathcal{D}0\mathcal{L}) \subseteq \omega(C)$. Prema prethodnoj teoremi svaki beskonačan C jezik ima beskonačan $\mathcal{D}0\mathcal{L}$ podskup. Onda prema teoremi 2.61 imamo $\omega(C) \subseteq \omega(\mathcal{D}0\mathcal{L})$. Konačno, imamo da važi $\omega(C) = \omega(\mathcal{PD}0\mathcal{L})$. □

Uzmimo
$Set_3 = \{\mathcal{CD}0\mathcal{L}, \mathcal{ND}0\mathcal{L}, \mathcal{WD}0\mathcal{L}, \mathcal{HD}0\mathcal{L}, \mathcal{CPD}0\mathcal{L}, \mathcal{NPD}0\mathcal{L}, \mathcal{WPD}0\mathcal{L}, \mathcal{HPD}0\mathcal{L}, \mathcal{EDT}0\mathcal{L}, \mathcal{CDT}0\mathcal{L}, \mathcal{NDT}0\mathcal{L}, \mathcal{WDT}0\mathcal{L}, \mathcal{HDT}0\mathcal{L}, \mathcal{EPDT}0\mathcal{L}, \mathcal{CPDT}0\mathcal{L}, \mathcal{NPDT}0\mathcal{L}, \mathcal{WPDT}0\mathcal{L}, \mathcal{HPDT}0\mathcal{L}, \mathcal{EDTF}0\mathcal{L}, \mathcal{CDTF}0\mathcal{L}, \mathcal{NDTF}0\mathcal{L}, \mathcal{WDTF}0\mathcal{L}, \mathcal{HDTF}0\mathcal{L}, \mathcal{EPDTF}0\mathcal{L}, \mathcal{CPDTF}0\mathcal{L}, \mathcal{NPDTF}0\mathcal{L}, \mathcal{WPDTF}0\mathcal{L}, \mathcal{HPDTF}0\mathcal{L}, \mathcal{ET}0\mathcal{L}, \mathcal{CT}0\mathcal{L}, \mathcal{NT}0\mathcal{L}, \mathcal{WT}0\mathcal{L}, \mathcal{HT}0\mathcal{L}, \mathcal{EPT}0\mathcal{L}, \mathcal{CPT}0\mathcal{L}, \mathcal{NPT}0\mathcal{L}, \mathcal{WPT}0\mathcal{L}, \mathcal{HPT}0\mathcal{L}, \mathcal{ETF}0\mathcal{L}, \mathcal{EPTF}0\mathcal{L}, \mathcal{CPTF}0\mathcal{L}, \mathcal{NPTF}0\mathcal{L}, \mathcal{WPTF}0\mathcal{L}, \mathcal{HPTF}0\mathcal{L}, \mathcal{C}0\mathcal{L}, \mathcal{N}0\mathcal{L}, \mathcal{W}0\mathcal{L}, \mathcal{H}0\mathcal{L}, \mathcal{E}0\mathcal{L}, \mathcal{CTF}0\mathcal{L}, \mathcal{NTF}0\mathcal{L}, \mathcal{WTF}0\mathcal{L}, \mathcal{HTF}0\mathcal{L}, \mathcal{EP}0\mathcal{L}, \mathcal{CP}0\mathcal{L}, \mathcal{NP}0\mathcal{L}, \mathcal{WP}0\mathcal{L}, \mathcal{HP}0\mathcal{L}, \mathcal{EF}0\mathcal{L}, \mathcal{CF}0\mathcal{L}, \mathcal{NF}0\mathcal{L}, \mathcal{WF}0\mathcal{L}, \mathcal{HF}0\mathcal{L}, \mathcal{CDF}0\mathcal{L}, \mathcal{NDF}0\mathcal{L}, \mathcal{WDF}0\mathcal{L}, \mathcal{HDF}0\mathcal{L}, \mathcal{EPF}0\mathcal{L}, \mathcal{CPF}0\mathcal{L}, \mathcal{NPF}0\mathcal{L}, \mathcal{WPF}0\mathcal{L}, \mathcal{HPF}0\mathcal{L}, \mathcal{CPDF}0\mathcal{L}, \mathcal{NPDF}0\mathcal{L}, \mathcal{HPDF}0\mathcal{L}, \mathcal{WPDF}0\mathcal{L}\}$.

Teorema 2.66 *Za svako $C \in Set_3$, svaki beskonačan C jezik ima beskonačan $\mathcal{CD}0\mathcal{L}$ podskup.*

Dokaz. Uzmimo proizvoljno $C \in Set_3$. Na osnovu teoreme o strukturnim inkluzijama imamo $C \subseteq \mathcal{ETF}0\mathcal{L}$ ili $C \subseteq \mathcal{HTF}0\mathcal{L}$. Prema teoremi 2.41 imamo $\mathcal{ETF}0\mathcal{L} = \mathcal{HTF}0\mathcal{L} = \mathcal{ET}0\mathcal{L}$. Stoga je $C \subseteq \mathcal{ET}0\mathcal{L}$. Dalje, prema teoremi 2.59 svaki beskonačan $\mathcal{ET}0\mathcal{L}$ jezik ima beskonačan $\mathcal{CD}0\mathcal{L}$ podskup. konačno, svaki beskonačan C jezik ima beskonačan $\mathcal{CD}0\mathcal{L}$ podskup. □

Teorema 2.67 *Za svako $C \in Set_3$ važi $\omega(C) = \omega(\mathcal{CD}0\mathcal{L})$.*

Dokaz. Neka je $C \in Set_3$ uzeto proizvoljno. Prema prethodnoj teoremi svaki beskonačan C jezik ima beskonačan $\mathcal{CD}0\mathcal{L}$ podskup. Onda prema teoremi

2.61 imamo $\omega(C) \subseteq \omega(\mathcal{CD}0\mathcal{L})$. Prema teoremi o strukturnim inkluzijama $\mathcal{CPD}0\mathcal{L} \subseteq C$ ili $\mathcal{EP}0\mathcal{L} \subseteq C$ ili $\mathcal{EPDT}0\mathcal{L} \subseteq C$. Prema teoremi 2.38 važi $\mathcal{EP}0\mathcal{L} = \mathcal{C}0\mathcal{L}$, pa važi $\mathcal{CPD}0\mathcal{L} \subseteq \mathcal{EP}0\mathcal{L}$. Prema teoremi 2.40 važi $\mathcal{CPDT}0\mathcal{L} \subseteq \mathcal{EPDT}0\mathcal{L}$, pa imamo $\mathcal{CPD}0\mathcal{L} \subseteq \mathcal{EPDT}0\mathcal{L}$. Stoga $\mathcal{CPD}0\mathcal{L} \subseteq C$. Dalje, prema toremi 2.37 imamo $\mathcal{CPDF}0\mathcal{L} = \mathcal{CDF}0\mathcal{L}$, pa važi $\mathcal{CD}0\mathcal{L} \subseteq \mathcal{CPDF}0\mathcal{L}$. Stoga $\omega(\mathcal{CD}0\mathcal{L}) \subseteq \omega(\mathcal{CPDF}0\mathcal{L})$. Prema teoremi 2.60 imamo da svaki beskonačan $\mathcal{CPDF}0\mathcal{L}$ jezik ima beskonačan $\mathcal{CPD}0\mathcal{L}$ podskup. Onda prema teoremi 2.61 imamo $\omega(\mathcal{CPDF}0\mathcal{L}) \subseteq \omega(\mathcal{CPD}0\mathcal{L})$. Stoga $\omega(\mathcal{CD}0\mathcal{L}) \subseteq \omega(\mathcal{CPD}0\mathcal{L}) \subseteq \omega(C)$. Konačno, $\omega(\mathcal{CD}0\mathcal{L}) = \omega(C)$. □

Iz gornjih teorema imamo da za proizvoljne $C_1, C_2 \in Set_i, i = 1,2,3$ važi $\omega(C_1) = \omega(C_2)$, ali treba imati na umu da ovo ne implicira da je klasa C_1 prefiksnih jezika jednaka klasi C_2 prefiksnih jezika.

Dalje ćemo pokazati da važi $\omega(\mathcal{PD}0\mathcal{L}) \subset \omega(\mathcal{D}0\mathcal{L}) \subset \omega(\mathcal{CD}0\mathcal{L})$.

Teorema 2.68 $\omega(\mathcal{PD}0\mathcal{L}) \subset \omega(\mathcal{D}0\mathcal{L})$.

Dokaz. Prema teoremi o strukturnim inkluzijama, $\omega(\mathcal{PD}0\mathcal{L}) \subseteq \omega(\mathcal{D}0\mathcal{L})$. Da bi smo pokazali da ovde imamo striktnu inkluziju koristimo beskonačnu reč iz [20]. Neka je $A = \{0,1,2\}$ i f morfizam na A zadat sa

$$f(0) = 01222, f(1) = 10222, f(2) = \lambda.$$

Neka je $\alpha = f^\omega(0) = 01222102221022201222\ldots$. Tada je α čisto morfična reč, stoga je α u $\omega(\mathcal{D}0\mathcal{L})$. U [20] je pokazano da ne postoji $\lambda-$slobodan morfizam g nad A takav da je $g^\omega(0) = \alpha$. Ovaj rezultat se generalizuje da bi se pokazalo da α nije u $\omega(\mathcal{PD}0\mathcal{L})$. Prvo se pokaže da je morfizam g identičko preslikavanje ako je $\lambda-$slobodan morfizam nad A i važi $g(\alpha) = \alpha$, kao što sledi.

Neka je τ Thue-Morse-ova reč $\tau = 01101001\ldots = u^\omega(0)$, gde je u morfizam nad $\{0,1\}$ takav da je $u(0) = 01$ i $u(1) = 10$. Neka je d morfizam nad A takav da je $d(0), d(1) = 1$ i $d(2) = \lambda$. Kao što je zapaženo u [20] $d(\alpha) = \tau$. Primetimo da ni 2222 ni 212 nisu podreč od α.

Pretpostavimo da je g $\lambda-$slobodan morfizam i da važi $g(\alpha) = \alpha$. Pošto α počinje sa 0 imamo $g(0) = 0x$ za neko $x \in A^*$. Ako je $x \neq \lambda$, onda $g^\omega(0) = \alpha$ što nas dovodi u kontradikciju sa rezultatom iz [20] da ne postoji takav $\lambda-$slobodan morfizam g. Stoga je $g(0) = 0$.

Pretpostavimo da $g(2)$ nije u 2^*. Neka je $s = d(g(2))$. Tada s nije prazan. Pošto je 222 podreč od α i $g(\alpha) = \alpha$, $g(222)$ je podreč od α. Dalje, pošto je $d(\alpha) = \tau$, τ sadrži $d(g(222)) = sss$, kontradikcija jer je poznato da τ ne sadrži kubove. Stoga je $g(2)$ u 2^*. Sada, pošto znamo da α sadrži $g(222)$, kao i da 2222 nije podreč od α i da g je $\lambda-$slobodan morfizam, zaključujemo da je $g(2) = 2$.

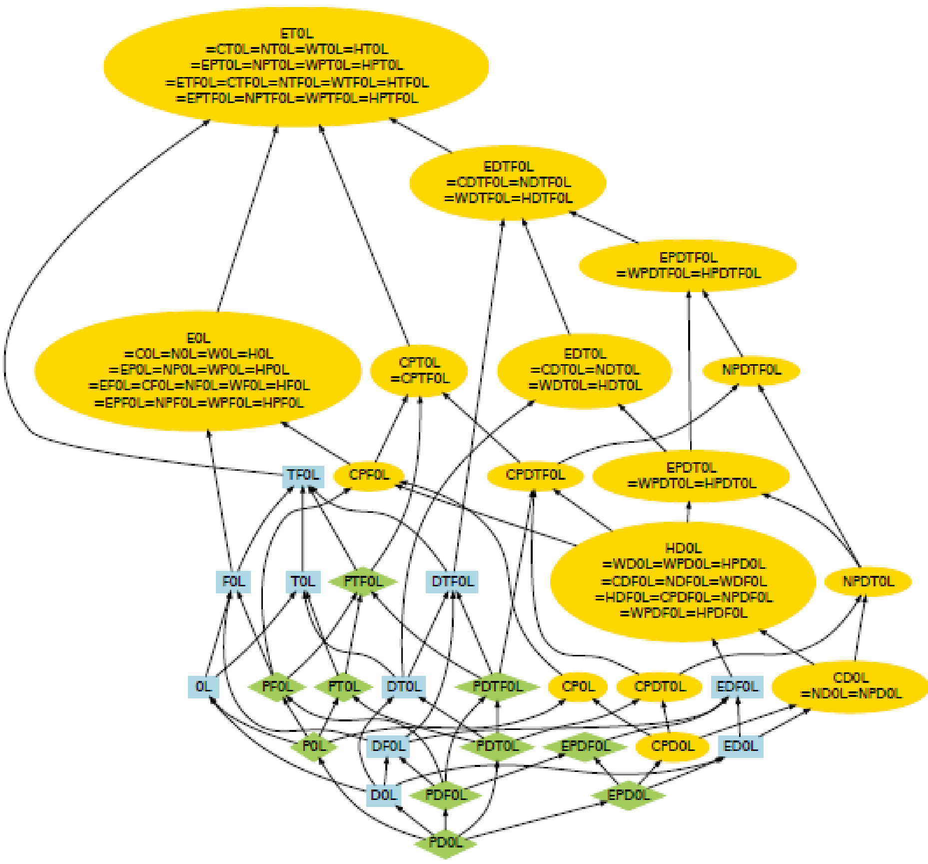
Pretpostavimo $g(1) \neq 1$. Tada, pošto α počinje sa 012, $g(1) = 12z$ za neko $z \in A^*$. Pošto je 2221 podreč od α, $g(2221) = 22212z$ je podreč od α, kontradikcija, jer α ne sadrži 212 kao podreč. Dakle, $g(1) = 1$. Konačno, imamo $g(0) = 0, g(1) = 1, g(2) = 2$, odnosno g je identičko preslikavanje.

Pretpostavimo da je α u $\omega(\mathcal{PD}0\mathcal{L})$. Tada postoji $\mathcal{PD}0\mathcal{L}$ sistem $G = (A, h, w_0)$ takav da $L(G)$ određuje α. Tada je h $\lambda-$slobodan morfizam nad A i $h(\alpha) = \alpha$, pa prema pokazanom, zaključujemo da je h identičko preslikavanje. Ali tada je $h(w_0) = w_0$, pa je $L(G)$ konačan, kontradikcija. Dakle, α nije u $\omega(\mathcal{PD}0\mathcal{L})$ i važi $\omega(\mathcal{PD}0\mathcal{L}) \subset \omega(\mathcal{D}0\mathcal{L})$. □

Teorema 2.69 $\omega(\mathcal{D}0\mathcal{L}) \subset \omega(\mathcal{CD}0\mathcal{L})$.

Dokaz. Prema teoremi o strukturnim inkluzijama $\omega(\mathcal{D}0\mathcal{L}) \subseteq \omega(\mathcal{CD}0\mathcal{L})$. Neka je $\alpha = abba^\omega$. Pošto je α krajnje periodična, α je morfična, stoga je α u $\omega(\mathcal{CD}0\mathcal{L})$. Pretpostavimo da je α u $\omega(\mathcal{D}0\mathcal{L})$. Tada postoji $\mathcal{D}0\mathcal{L}$ sistem $G = (A, h, w_0)$ čiji jezik $L(G)$ određuje α. Jasno, $h(a)$ ne uključuje b, a ako bi bilo $h(a) = \lambda$ imali bi da je $L(G)$ konačan, kontradikcija. Pa prema tome, pošto $h(a)$ mora biti prefiks od α, imamo $h(a) = a$. Tada je $h(b)$ prefiks od α, stoga $h(b) = \lambda$ ili $h(b) = b$. Ali tada je $L(G)$ konačan, kontradikcija. Dakle, α nije u $\omega(\mathcal{D}0\mathcal{L})$. Konačno, $\omega(\mathcal{D}0\mathcal{L}) \subset \omega(\mathcal{CD}0\mathcal{L})$. □

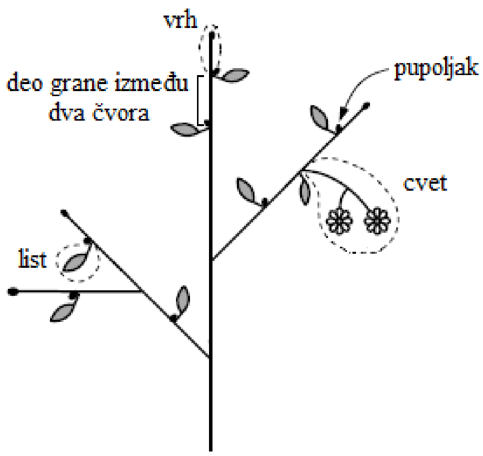
Kao posledicu teoreme 2.60 imamo da dodavanje svojstva F skupu svojstava neke klase ne utiče na klasu beskonačnih reči određenih tom klasom jezika. Za sva ostala svojstva $(\mathcal{D}, \mathcal{P}, \mathcal{T}, \mathcal{E}, \mathcal{C}, \mathcal{N}, \mathcal{W}, \mathcal{H})$, prema prethodnim teoremama, postoji bar jedan slučaj gde dodavanje nekog od njih skupu svojstava klase utiče na klasu beskonačnih reči određenu tom klasom jezika. Na primer, za $\mathcal{D}$ imamo $\omega(\mathcal{ED}0\mathcal{L}) \subset \omega(\mathcal{E}0\mathcal{L})$, za svojstvo $\mathcal{P}$ imamo $\omega(\mathcal{PD}0\mathcal{L}) \subset \omega(\mathcal{D}0\mathcal{L})$, za $\mathcal{E}$ imamo $\omega(0\mathcal{L}) \subset \omega(\mathcal{E}0\mathcal{L})$ itd. Na sledećoj slici dat je dijagram odnosa svih 96 klasa $0\mathcal{L}$ jezika sa kojima smo se do sada susreli. Strelice označavaju inkluziju. Zelene klase (romboidi) određuju tačno $\omega(\mathcal{PD}0\mathcal{L})$, plave klase (pravougaonici) određuju tačno $\omega(\mathcal{D}0\mathcal{L})$, a žute klase (elipse) određuju $\omega(\mathcal{CD}0\mathcal{L})$ klasu beskonačnih reči.



Slika 2.5: Dijagram odnosa klasa $0\mathcal{L}$ jezika.

2.7 L-sistemi i modeliranje razvoja biljaka

Kao što smo na početku ovog poglavlja rekli, L-sistemi su uvedeni radi modeliranja rasta i razvoja biljaka. Jedan od osnovnih postulata teorije L-sistema jeste da biljka ima ***modularnu strukturu***, odnosno da se može smatrati konačnim skupom diskretnih komponenti-***modula***, koje se pojavljuju iznova i iznova tokom razvoja biljke, kao što su npr. vrhovi, listovi, pupoljci, cvetovi, neki delovi grana itd. Smatra se da je broj vrsta modula kod svake biljke konačan, bez obzira na njenu veličinu. Kod jednostavnih višećelijskih organizama, npr. algi, vrste modula su sve vrste ćelija koje taj organizam čine.



Slika 2.6: Primer modularne strukture biljke.

Svaka vrsta modula označena je nekim slovom, pri čemu su različite vrste označene različitim slovima. Kako je broj vrsta konačan, skup W ovih slova (azbuka) će takođe biti konačan. ***Cilj modeliranja biljaka na modularnom nivou jeste da se opiše razvoj cele biljke, kao integracija razvoja pojedinačnih modula***. Na slici 2.6 dat je primer modularne strukture biljke, gde su vrste modula: deo grane između dva čvora, vrh, list, pupoljak i cvet, pri čemu uzimamo u obzir da je u biologiji čvor deo stabljike (grane) iz koga raste nova grana ili list. Jasno, pridužena azbuka W će imati pet slova.

Sa stanovišta teorije grafova modularna struktura biljke može se opisati pomoću ***osnog stabla***. Moduli su predstavljeni ivicama ovog stabla, pri čemu je ivica označene slovom iz azbuke W koje se koristi za označavanje vrste modula koju ta ivica predstavlja.

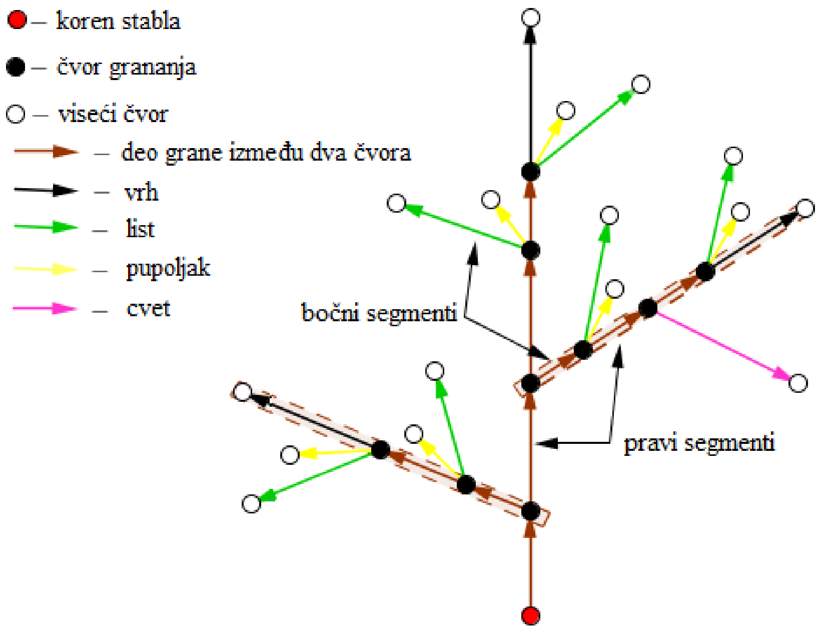
Formalno, osno stablo je poseban tip ***korenskog stabla***. Korensko stablo je stablo čije su ivice orijentisane i koje ima jedan posebno označen čvor koji se naziva ***koren*** (za detalje videti [14] i str. 31. u [3]). Među ivicama koje izlaze iz čvora osnog stabla najviše jedna ivica će biti označena kao ***prav segment***, sve ostale će biti ***bočni segmenti***. Put u osnom stablu naziva se ***osa*** ako:

(a) prva ivica puta izlazi iz korena stabla ili je bočni segment nekog čvora,

(b) svaka sledeća ivica puta je prav segment,

(c) iz poslednjeg čvora puta ne izlazi prav segment.

Prvi čvor ose se naziva ***osnova***, a poslednji ***vrh***.

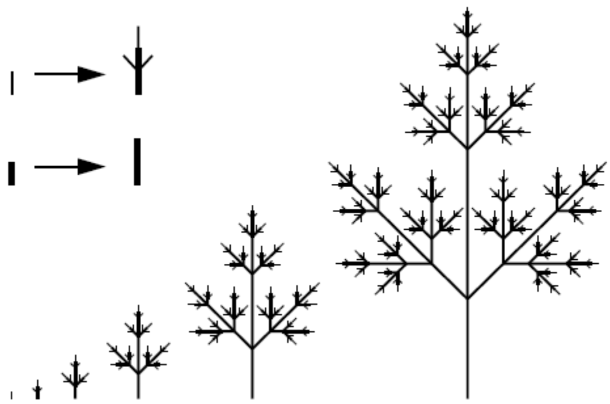


Slika 2.7: Osno drvo za biljku sa slike 2.6.

Osa sa svim svojim potomcima (ivice do kojih se može stići iz čvorova na osi) se naziva ***grana***. Grana je podstablo osnog stabla. Za osu kažemo da je nultog reda ako izlazi iz korena osnog drveta. Osa koja izlazi iz čvora ose n-tog reda je reda $n+1$. Osa grane čiji je red najmanji naziva se ***glavna*** osa grane. Red grane odgovara redu njene glavne ose. Na slici 2.7 dato je osno drvo za biljku sa slike 2.6, gde smo, radi jednostavnosti slike, umesto slovima vrste modula označili bojama. Takođe, naznačeni su pojedini pravi i bočni segmenti, a ose uokvirene isprekidanom linijom su ose prvog reda.

Još jedna pretpostavka teorije L-sistema jeste da se svi delovi biljke–moduli razvijaju istovremeno i da razvoj počinje od neke početne strukture kao što je npr. seme ili klica. Imajuči ovo na umu, postaje jasno da se L-sistemi iz definicije 2.14 mogu iskoristiti za ostvarenje cilja modeliranja biljaka na modularnom nivou. Kako je azbuka W, vrsta modula konačna, i imamo početnu strukturu (početnu reč), razvoj biljke kroz vreme može se simulirati pomoću konačnog skupa produkcija P, koje opisuju razvoj vrsta modula u diskretnom vremenskom intervalu. U najjednostavnijem slučaju, kada razvoj pojedinačnih vrsta modula ne zavisi od ostalih vrsta modula, produkcije se sastoje iz jednog modula, koji nazivamo ***prethodnik*** sa jedne strane i konfiguracije od nula, jednog ili više modula, koju nazivamo ***sledbenik***, sa druge strane (kontekstno slobodan L-sistem). U jednom koraku izvođenja modul prethodnik biva zamenjen modulom sledbenikom. Produkcije se primenjuju istovremeno na sve module u svakom koraku izvođenja. Korak izvođenja odgovara nekom vremenskom intervalu (intervalu koji je recimo u stvarnosti potreban da se taj razvoj odigra).

Niz struktura dobijenih u uzastopnim koracima izvođenja iz neke početne strukture naziva se ***razvojni niz***. U terminima teorije grafova, produkcija menja ivicu osnog drveta, koju nazivamo ***prethodnik***, sa osnim poddrvetom koje nazivamo ***sledbenik***. Sledbenik će biti ,,ugrađen" u strukturu na mesto prethodnika tako što će čvor koji je bio početni (krajnji) čvor prethodnika sada biti osnova (vrh) glavne ose sledbenika. Produkcije će istovremeno zameniti sve ivice osnog drveta. Sledeća slika ilustruje razvoj lista koji ima dva tipa modula, vrhove (predstavljeni tankom linijom) i delove između dva čvora (predstavljeni debelom linijom). Vidimo kako se pomoću jednostavnih produkcija počev od vrha dolazi do složene granajuće strukture.



Slika 2.8: Primer razvojnog niza lista.

U sklopu teorije L-sistema, osno drvo se opisuje kao niz simbola nad azbukom $W \cup \{[,],+,-\}$ gde su [,], + i – pomoćni simboli koji se koriste pri opisivanju granajućih struktura. Niz oznaka uzastopnih pravih segmenata predstavlja osu. Niz oznaka unutar zagrada $[\dots]$ predstavlja granu. Znaci + i – služe za označavanje smera grananja, na levu i na desnu stranu ose, respektivno. Na primer, neka su a i b slova azbuke W i w reč nad W. Zapis $\dots a[+w]b \dots$ znači da je b prav segment koji se nadovezuje na a u osi $\dots ab \dots$ i da je w bočna grana koja izlazi sa leve strane ose iz krajnjeg čvora ivice a. Na primer, L-sistem sa slike 2.8 može se zapisati kao:

$$w_0 = a,$$

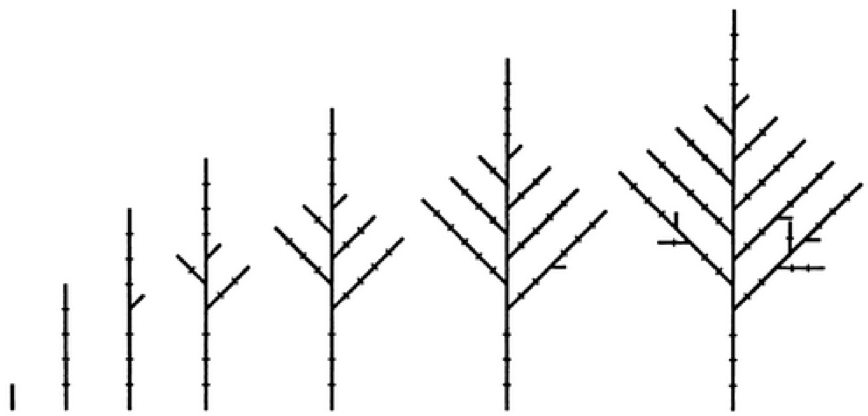
$$a \to b[+a][-a]ba,$$

$$b \to bb,$$

gde a označava vrh, a b deo grane između dva čvora. Način crtanja grana se može dati i opisno, kao što je slučaj u sledećem primeru, bez upotrebe simbola + i $-$.

$\mathcal{D}0\mathcal{L}$ sistemi su najjednostavniji među L-sistemima i daju jasan uvid u njihove osnovne ideje i tehnike. Prvi L-sistemi upotrebljeni za modeliranje razvoja biljaka bili su upravo iz klase $\mathcal{D}0\mathcal{L}$ i jedan od njih je dat u sledećem primeru. Upitanju je L-sistem za modeliranje razvoja crvene alge ***Callithamnion roseum*** iz prvih radova Aristida Lindenmayera o L-sistemima.

Primer 2.70 *Razmatramo $\mathcal{D}0\mathcal{L}$ sistem $G = (W, P, w_0)$ za modeliranje razvoja crvene alge Callithamnion roseum gde je:*
$W = V \cup \{[,\ ]\}$, *pri čemu je* $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$,
$P = \{1 \to 23, 2 \to 2, 3 \to 24, 4 \to 25, 5 \to 65, 6 \to 7, 7 \to 8, 8 \to 9[3], 9 \to 9\}$,
$w_0 = 1$.
Zagrade [,] su pomoćni simboli koji služe za označavanje grana kao što je maločas objašnjeno. Ovde se grane crtaju naizmenično sa obe strane ose, pri čemu se prva grana crta sa desne strane. Počev od $w_0 = 1$ dobijamo razvojni niz:
$w_0 = 1$,
$w_1 = 23$,
$w_2 = 224$,
$w_3 = 2225$,
$w_4 = 22265$,
$w_5 = 222765$,
$w_6 = 2228765$,
$w_7 = 2229[3]8765$,
$w_8 = 2229[24]9[3]8765$,
$w_9 = 2229[225]9[24]9[3]8765$,
$w_{10} = 2229[2265]9[225]9[24]9[3]8765$,
$w_{11} = 2229[22765]9[2265]9[225]9[24]9[3]8765$,
$w_{12} = 2229[228765]9[22765]9[2265]9[225]9[24]9[3]8765$,
$w_{13} = 2229[229[3]8765]9[228765]9[22765]9[2265]9[225]9[24]9[3]8765$,
$\ldots$
Na slici 2.9 prikzane su redom sledeće faze razvoja ove alge: $w_0, w_4, w_7, \ldots$ i w_{15} (koraci u izvođenju).

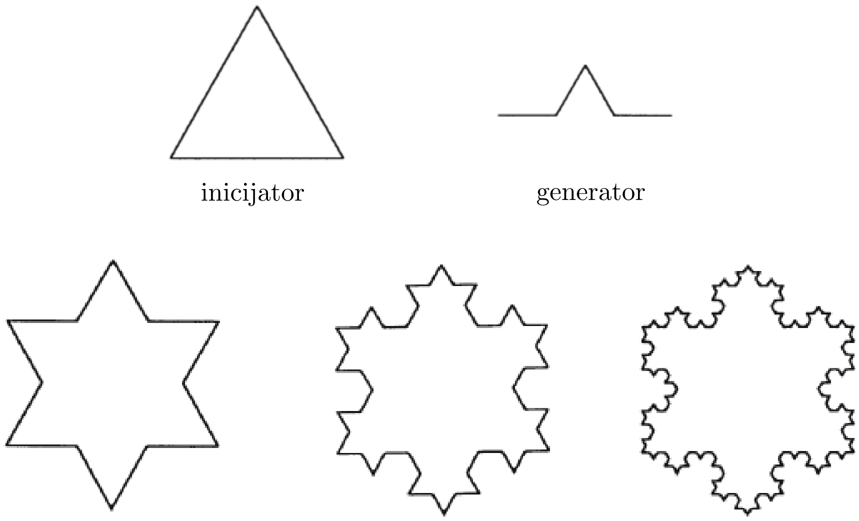


Slika 2.9: Neke od faza razvoja alge Callithamnion Roseum.

Ovaj sistem pripada i klasi $\mathcal{PD}0\mathcal{L}$ jer nema produkcija oblika $a \to \lambda$.

2.8 Fraktali

Pored primene u modeliranju rasta i razvoja biljaka L-sistemi imaju široku primenu u generisanju fraktala. Među prve opisane fraktalne krive spadaju krive dobijene prvobitnim ***Koch*[5]*-ovim konstrukcijama***. Koch-ove konstrukcije takođe spadaju, kao i L-sistemi i formalne gramatike, u sisteme prerade, s'tim što za razliku od njih ne prerađuju reči već mnogouglove. Prvobitno se ovaj sistem sastojao od pravilnog mnogougla – ***inicijatora*** i ***generatora***. U svakom koraku konstruisanja sve stranice mnogougla dobijenog u prethodnom koraku zamenjujemo sa generatorom, pri čemu generator može biti transliran, rotiran, skaliran da bi se uklopio na originalnu poziciju segmenta koji zamenjuje i konstruisanje počinje od inicijatora. Na slici 2.10 dati su inicijator i generator prve krive konstruisane na ovakav način, tzv. ***Koch-ove pahuljice***, kao i konstrukcije dobijene u prva tri koraka konstruisanja. Danas postoje brojne modifikovane Koch-ove konstrukcije. Sve krive konstruisane Koch-ovim konstrukcijama čine klasu Koch-ovih krivih.



Slika 2.10: Konstrukcija Koch-ove pahuljice.

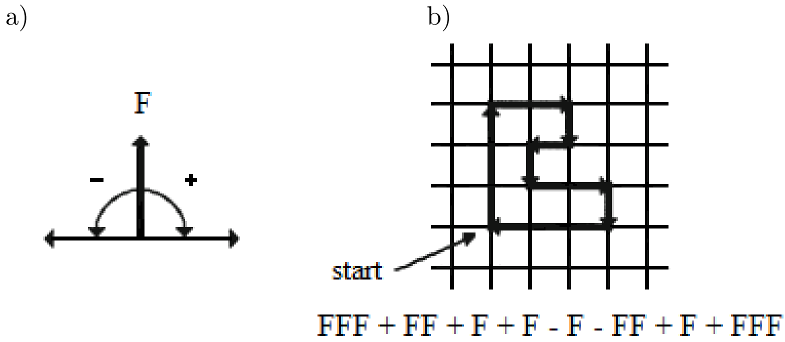
Mnogi fraktali (ili barem njihove konačne aproksimacije) mogu se posmatrati kao konačni nizovi primitivnih elemenata – duži (linijskih segmenata). Pri tome dužine duži i uglovi među njima igraju suštinsku ulogu. Stoga, reči generisane L-sistemom moraju sadržati neophodne informacije o geometriji fraktala, da bi mogle biti primenjene u njihovom generisanju. Za grafičku interpretaciju reči generisanih L-sistemom Szilard i Quinton su predložili upotrebu programskog jezika ***LOGO-style kornjača***, koji funkcioniše na sledeći način.

Stanje kornjače je definisano uređenom trojkom $(x,\ y,\ \alpha)$, gde Dekartove koordinate $(x,\ y)$ predstavljaju ***poziciju*** kornjače, a ugao α ***pravac*** kretanja kornjače. Ako se zada ***dužina koraka*** d i ***priraštaj ugla*** δ kornjača će odreagovati na komande predstavljene sledećim simbolima:

[5] Niels Fabian Helge von Koch (1870-1924), švedski matematičar.

F	Pomeri se napred za korak dužine d. Stanje kornjače se menja u $(x',\ y',\ \alpha)$, gde je $x' = x + d\cos\alpha$ i $y' = y + d\sin\alpha$. Nacrtaj duž koja povezuje tačke $(x,\ y)$ i $(x',\ y')$.
f	Pomeri se napred za korak dužine d bez crtanja duži koja povezuje tačke $(x,\ y)$ i $(x',\ y')$.
$+$	Skreni desno pod uglom δ. Sledeće stanje kornjače je $(x,\ y,\ \alpha+\delta)$. Ovde podrazumevamo da je pozitivna orjentacija uglova u smeru kazaljke na satu.
$-$	Skreni levo pod uglom δ. Novo stanje kornjače je $(x,\ y,\ \alpha-\delta)$.

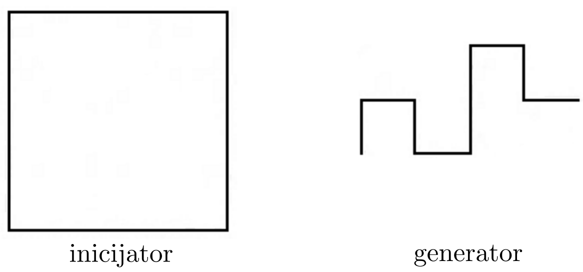
Sve ostale simbole kornjača ignoriše, odnosno kornjača ne menja svoje stanje kada naiđe na nepoznat simbol.

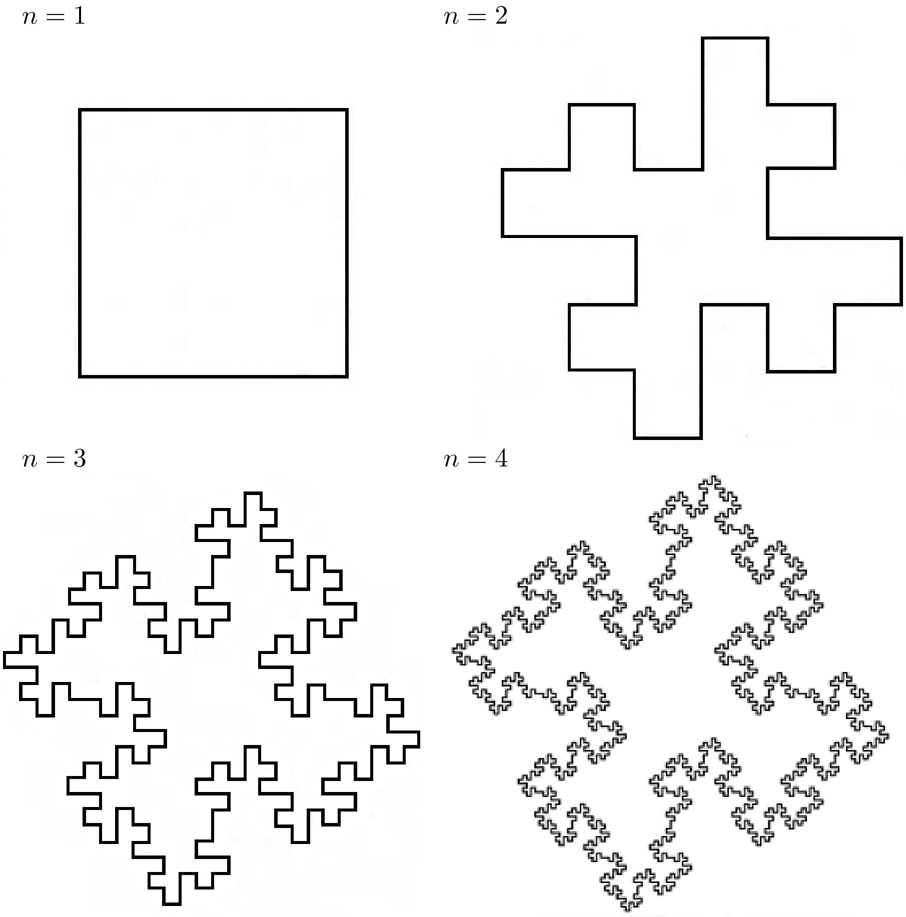


Slika 2.11: (a) Kornjačina interpretacija simbola F, $+$, $-$. (b) Primer interpretacije reči kada je priraštaj ugla $\delta = 90°$. Kornjača je na početku okrenuta na gore.

Neka je w reč nad azbukom $W = \{F,\ f,\ +,\ -\}$, $(x_0,\ y_0,\ \alpha_0)$ početno stanje kornjače, i d i δ fiksni parametri. Slika (skup duži) koju nacrta kornjača interpretirajući reč w naziva se ***kornjačina interpretacija*** reči w.

Ovako uvedene rigorozne metode za interpretaciju reči kao slika, možemo primeniti u interpetaciji reči generisanih L-sistemima nad azbukom W. Na slici 2.8 su predstavljene prve četiri aproksimacije ***kvadranog ostrva Kocha*** koje je prvobitno dobijeno pomoću Koch-ove konstrukcije čiji su inicijator i generator dati na sledećoj slici.



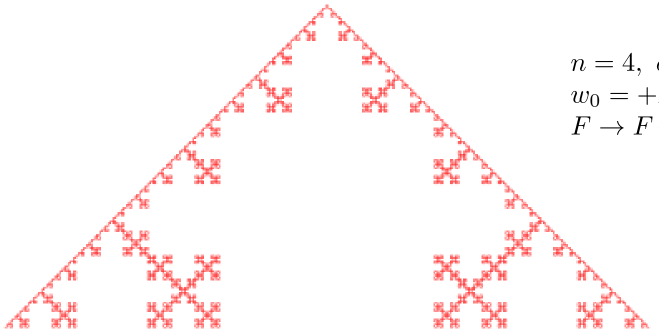


Slika 2.12: generisanje kvadratnog ostrva Koch-a.

Ove figure se pored Koch-ove konstrukcije mogu dobiti i kornjačinom interpretacijom reči generisanih L-sistemom nad azbukom $W = \{F,\ f,\ +,\ -\}$, koji je zadat početnom rečju $w_0 = F + F + F + F$ i produkcijom:

$$F \to F + F - F - FF + F + F - F.$$

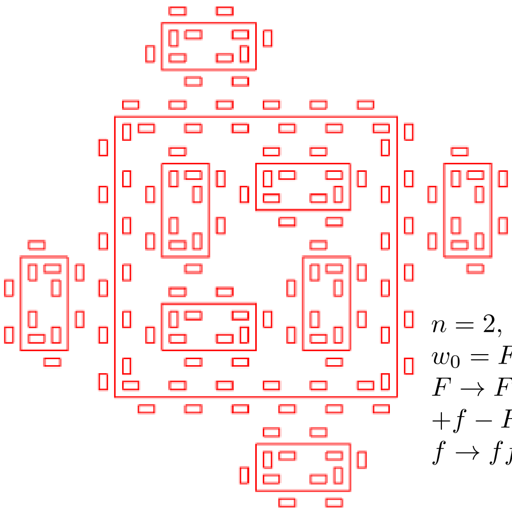
Slike odgovaraju rečima dobijenim u prva četiri koraka izvođenja redom. Priraštaj ugla je $\delta = 90°$. Dužina koraka d je u svakoj sledećoj aproksimaciji četiri puta manja nego u prethodnoj. Zbog toga je rastojanje između krajnjih tačaka dela mnogougla predstavljenog sledbenikom produkcije jednako dužini duži predstavljene prethodnikom produkcije. Ovaj primer ukazuje na usku povezanost između Koch-ove konstrukcije (sa jednim generatorom) i L-sistema. Nije teško uočiti da inicijator odgovara početnoj reči L-sistema, dok je generator predstavljen sledbenikom jedine produkcije (desna strana produkcije); leva strana je jednaka F. Stoga se L-sistemi mogu se posmatrati kao kodiranja Koch-ovih konstrukcija. Na sledećoj slici je dat još jedan primer Koch-ove krive i L-sistema koji je generiše.



$n = 4,\ \delta = 90°$
$w_0 = +F$
$F \to F - F + F + F - F$

Slika 2.13: Kvadratna modifikacija ***Koch-ove pahuljice***.

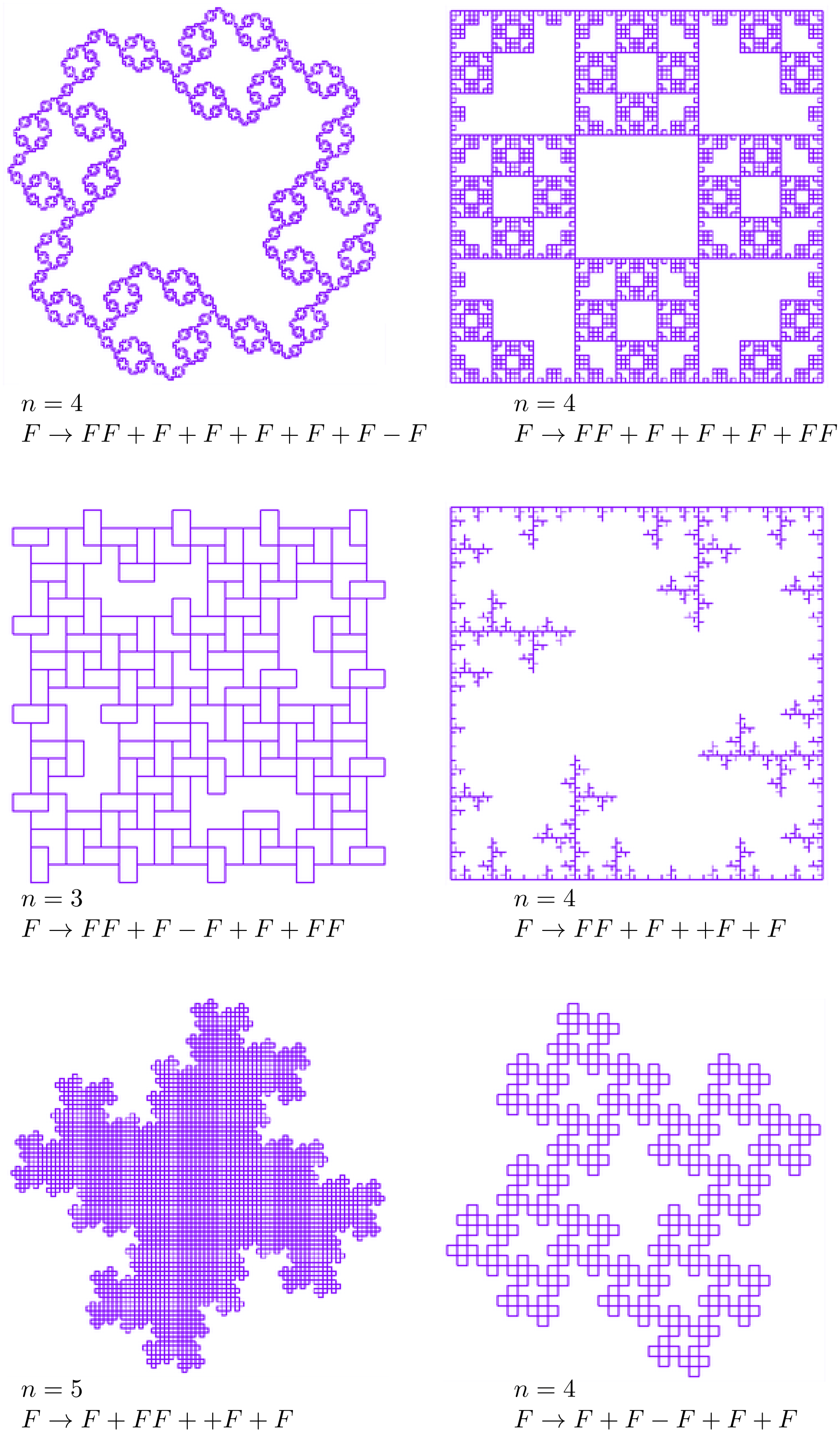
Mala komplikacija nastaje ako fraktal nije povezan. Takav fraktal i L-sistem koji ga generiše dati su na slici 2.14. Druga produkcija ovog L-sistema (sa prethodnikom f) u ovom slučaju služi da drži komponente fraktala na odgovarajućem odstojanju.



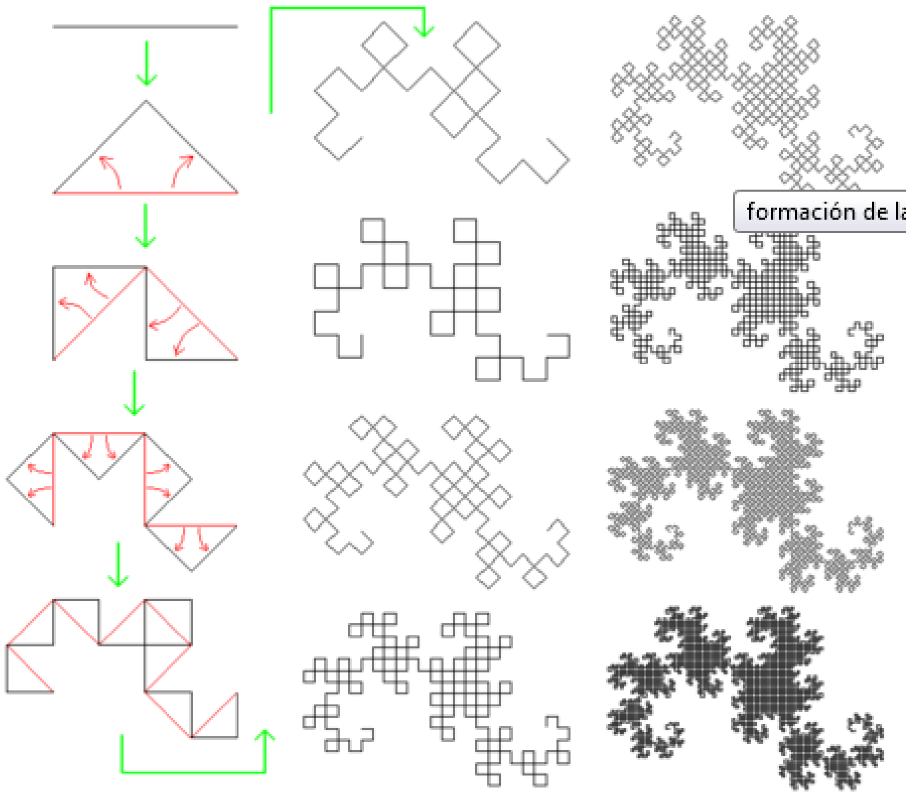
$n = 2,\ \delta = 90°$
$w_0 = F - F - F - F$
$F \to F - f + FF - F - FF - Ff - FF +$
$+ f - FF + F + FF + Ff + FFF$
$f \to ffffff$

Slika 2.14: ***Kombinacija ostrva i jezera***.

L-sistemima se mogu generisati mnoge Koch-ve krive, jer se oni lako modifikuju. Na primer, možemo krenuti od jednog L-sistema i ispitivati šta se dešava kada ubacimo, izbacimo, zamenimo neke simbole u sledbeniku produkcije. Neki fraktali dobijeni na ovaj način i L-sistemi koji ih generišu dati su na slici 2.15. Ovde svi L-sistemi imaju istu početnu reč $w_0 = F + F + F + F$ i isti priraštaj ugla $\delta = 90°$. Već smo rekli, u opisu Koch-ove konstrukcije, da generator može biti transliran, rotiran, skaliran da bi se uklopio na originalnu poziciju segmenta koji zamenjuje. Ako bi se omogućila i refleksija, još neke krive bi mogle biti konstruisane. To je slučaj, recimo kod ***zmajeve krive***, čijih je prvih trinaest koraka Koch-ove konstrukcije dato na slici 2.16.



Slika 2.15: Niz Koch-ovih krivih dobijenih sukcesivnom modifikacijom sledbenika u produkciji.



Slika 2.16: ***Zmajeva kriva***. Crvenom linijom su u svakom koraku označeni zamenjeni (izbrisani) segmenti iz prethodnog koraka.

Ovih trinaest figura dobili bi smo kornjačinom interpretacijom L-sistema čija je početna reč $w_0 = F_l$ i produkcije

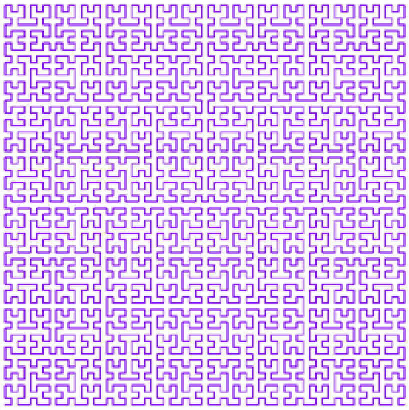
$$F_l \to F_l + F_r + \quad i \quad F_r \to -F_l - F_r,$$

ako bi kornjača dva različita simbola, F_l *i* F_r, interpretirala kao komandu ,,vrati se nazad za korak dužine d". Ipak, ovo nije u skladu sa originalnom definicijom kornjačine interpretacije. Srećom upotreba mnogih simbola sa istom interpretacijom može se izbeći transformacijom L-sistema. Pretpostavićemo, privremeno, da prethodnik produkcije može sadržati vise od jednog slova (da podreč može biti zamenjena primenom samo jedne produkcije). L-sistem koji generiše zmajevu krivu može se tada zapisati i ovako

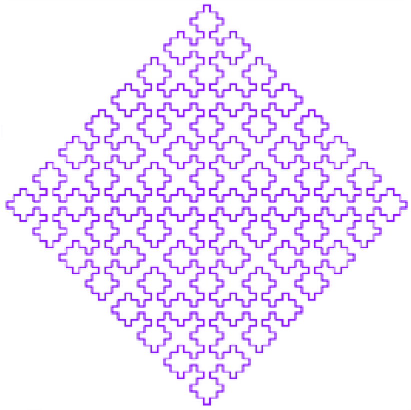
$$w_0 = Fl, \; Fl \to Fl + rF+, \; rF \to -Fl - rF,$$

gde kornjača ne interpretira simbole r (engl. ,,right") i l (engl. ,,left"). Produkcija $Fl \to Fl + rF+$ zamenjuje slovo l sa $l + rF+$ dok F ostaje F. Slično produkcija $rF \to -Fl - rF$ slovo F ostavlja nepromenjenim, a umesto slova r ubacuje $-Fl-r$. Sada obe produkcije imaju prethodnike koji se sastoje iz jednog slova i nepotrebna upotreba više simbola sa istom grafičkom interpretacijom je izbegnuta.

Na slici su dati primeri još nekih fraktalnih krivih konstruisani od „levih" i „desnih" elemenata. Njihovi L-sistemi imaju sličnu strukturu kao ovaj kod zmajeve krive.

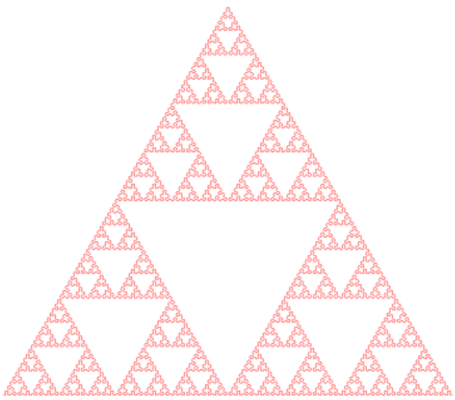


$n = 6,\ \delta = 90°$
$w_0 = X$
$X \to -YF + XFX + FY-$
$Y \to +XF - YFY - FX+$

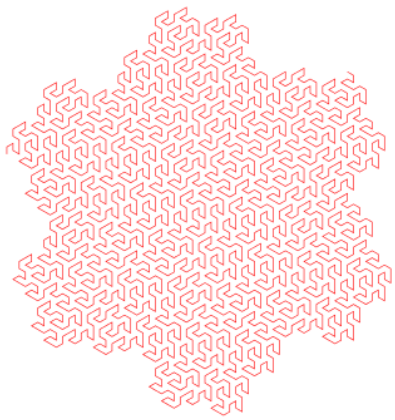


$n = 4,\ \delta = 90°$
$w_0 = F + XF + F + XF$
$X \to XF - F + F - XF + F +$
$+XF - F + F - X$

Slika 2.17: „Klasične" krive koje popunjavaju prostor i odgovarajući L- sistemi: ***Hilbertova kriva*** (levo), ***kvadratna kriva Sierpinskog*** (desno).



$n = 7,\ \delta = 60°$
$w_0 = YF$
$X \to YF + XF + Y$
$Y \to XF - YF - X$



$n = 4,\ \delta = 60°$
$w_0 = XF$
$X \to X + YF + +YF - FX -$
$-FXFX - YF+$
$Y \to -FX + YFYF + +YF +$
$FX - -FX - Y$

Slika 2.18: ***Kriva Sierpinskog*** (levo) i ***šestougaona kriva Gospera*** (desno).

Priraštaj ugla δ može uzeti i druge vrednosti pored $90°$. U tim slučajevima se takođe mogu dobiti interesantne slike. Na prethodnoj slici su date dve fraktalne krive dobijene za $\delta = 60°$ i odgovarajući L-sistemi.

Osnovna kornjačina interpretacija reči sa kojom smo se u ovom odeljku upoznali može se proširiti u mnogim pravcima. Ubacivanje dodatnih simbola u azbuku $W = \{F, f, +, -\}$ omogućava primenu u modeliranju biljka, trodimenzionalnu interpretaciju reči itd. Za više detalja o tome videti [3].

Glava 3

Odnos ćelijskih automata i L-sistema

Nakon što smo se, u prethodna dva poglavlja, upoznali sa osnovnim idejama teorije ćelijskih automata i teorije L-sistema lako uočavamo sledeće ***sličnosti između ova dva sistema***:

- Oba imaju početnu informaciju koja se može smatrati njihovim početnim stanjem:
  - kod L-sistema početna reč,
  - kod CA početna konfiguracija.
- Oba imaju komponente koje određuju kako će se sistem menjati:
  - kod L-sistema skup produkcija,
  - kod CA lokalno pravilo ažuriranja.
- Oba generišu svoje sledeće stanje primenom transformacije na sve komponente istovremeno:
  - L-sistem istovremeno menja sva slova u reči u skladu sa produkcijama,
  - CA istovremeno menja stanja svih svojih ćelija u skladu sa lokalnim pravilom ažuriranja.

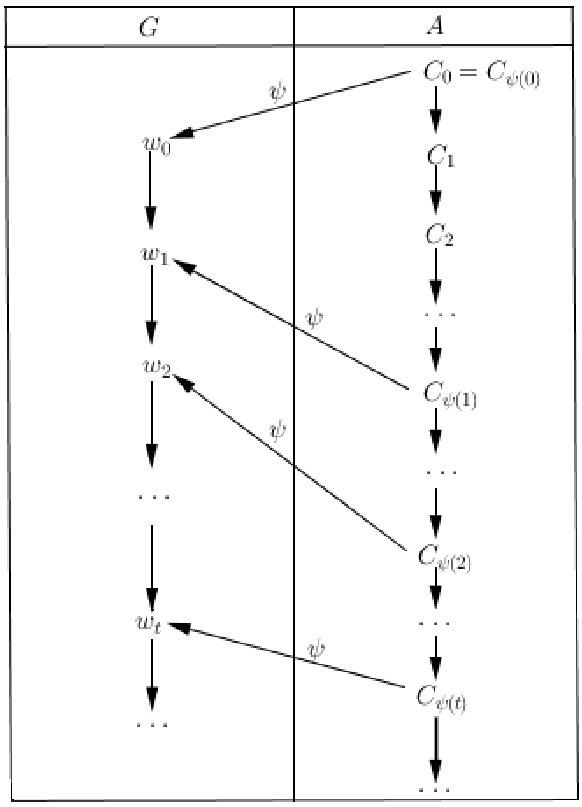
Ove sličnosti navode na sledeće pitanje:
Da li je moguće za dati CA konstruisati njemu ekvivalentan L-sistem i obratno, ako smatramo da su CA i L-sistem ekvivalentni ako generišu iste reči u istom redosledu?
Uprkos brojnim pokušajima, nauka još uvek nije dala odgovor na ovo pitanje. Dokazano je samo za neke specijalne slučajeve da je odgovor potvrdan i da postoji strukturna veza između L-sistema i CA. U ovom poglavlju ćemo dati uvid u neke od najkonkretnijih rezultata vezanih za ovo pitanje, koje naučnici smatraju dobrom osnovom za dalja istraživanja. Upitanju su pre svega rezultati M. Alfonseca, A. L. A. Dalhouma i A. Ortega iz perioda 2000-2004. godine o konstruisanju n-ekvivalentnog CA datom L-sistemu.

3.1 Jednodimenzionalan CA n-ekvivalentan $\mathcal{PD}0\mathcal{L}$ sistemu

3.1.1 Neformalan opis

U ovom delu ćemo se baviti kreiranjem CA ekvivalentnog datom $\mathcal{PD}0\mathcal{L}$ sistemu. Kažemo da je ***CA n-ekvivalentan L-sistemu*** ako je moguće naći prvih n reči jezika generisanog tim L-sistemom u konačnom broju uzastopnih konfiguracija ćelijskog automata, tako da redosled generisanja reči bude očuvan. Odnosno, mora postojati način da se uspostavi veza između

- Početne reči L-sistema i početne konfiguracije CA.
- Svake sledeće reči u jeziku L-sistema i njoj odgovarajuće konfiguracije CA, pod uslovom da se redosled generisanja reči očuva, tj. da se konfiguracija koja odgovara prvoj generisanoj reči mora dobiti pre konfiguracije koja odgovara drugoj generisanoj reči i tako dalje.



Slika 3.1: ***Mehanizam povezivanja ψ.***

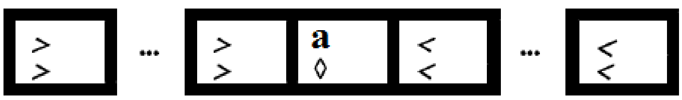
Na slici 3.1 prikazan je način povezivanja reči generisanih L-sistemom sa konfiguracijama generisanim jednodimenzionalnim CA pomoću mehanizma ψ. Radi jednostavnosti, način na koji CA simulira $\mathcal{PD}0\mathcal{L}$ sistem objašnjen je na sledećem primeru.

Dat je $\mathcal{PD}0\mathcal{L}$ sistem G čija je početna reč $w_0 = a$ i produkcije:

$$a \to bb,\ b \to ab.$$

Radi konstrukcije n-ekvivalentnog jednodimenzionalnog CA donete su sledeće odluke:

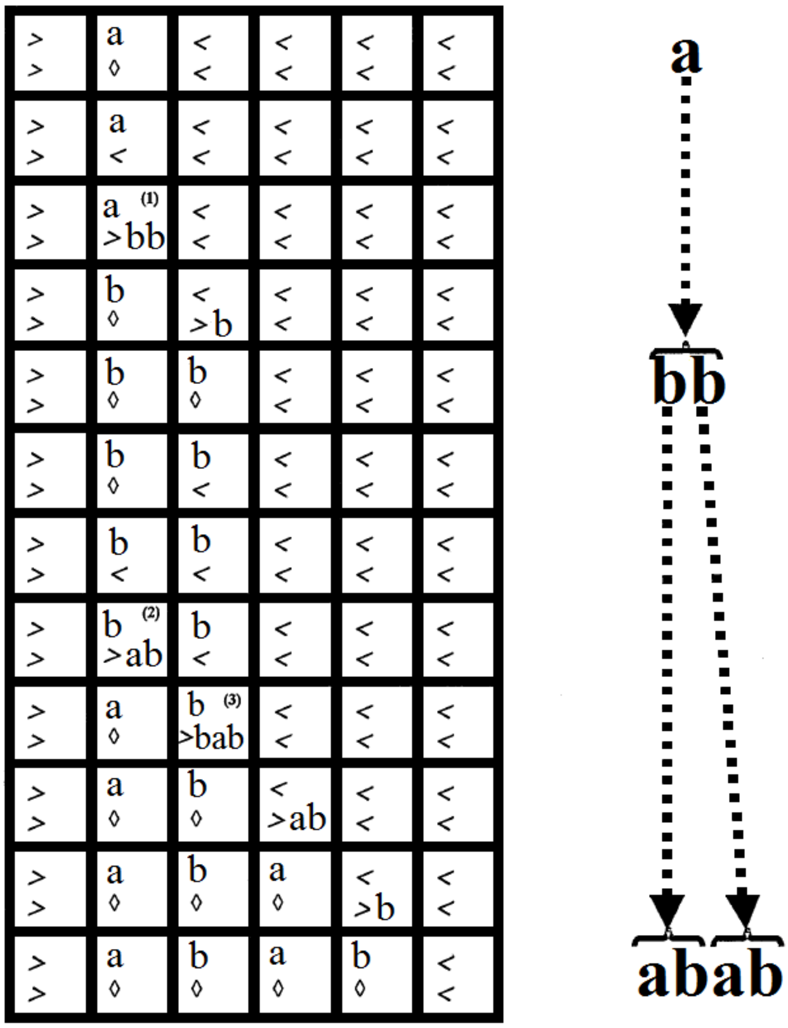
- U ćelijskom prostoru $\mathbb{Z}$ CA će postojati ćelije koje sadrže slova reči generisanih L-sistemom.
- Stanje svake ćelije CA će sadržati dve informacije, prva je ***tačno jedan simbol*** koji ćelija u datoj konfiguraciji sadrži, a druga je ***reč premeštanja*** koja omogućava ubacivanje novih slova na datu poziciju i pomeranje starih u desno na njihovu krajnu poziciju onda kada je to potrebno, jer je, kao što smo ranije već rekli, niz dužina reči svakog $\mathcal{PD}0\mathcal{L}$ jezika neopadajući. Prva informacija (simbol) se uvek nalazi iznad druge, tj. iznad reči premeštanja u grafičkom prikazu ćelija (prostorno-vremenski dijagram).
- Reči premeštanja zapravo obezbeđuju sprovođenje mehanizma za izvođenje reči jezika datog $\mathcal{PD}0\mathcal{L}$ sistema, odnosno omogućavaju da CA simulira proces izvođenja reči koji se kod L-sistema ostvaruje pomoću produkcija. Reči premeštanja su u ovom primeru reči nad azbukom $\{a, b, <, >\}$. Simboli $<$ i $>$ su respektivno ***levi*** i ***desni marker***.
- Početak i kraj reči koja se izvodi moraju biti markirani. Stanja svih ćelija sa leve strane ćelije koja sadrži prvo slovo reči su ista, i kao prvu i kao drugu informaciju imaju samo levi marker. Analogno, stanja svih ćelija sa desne strane ćelije koja sadrži poslednje slovo reči su ista, i kao prvu i kao drugu informaciju imaju samo desni marker.
- Ćelije koje ne sadrže ni levi ni desni marker sadrže simbol $\diamondsuit$. Ovaj simbol se naziva ***prazna reč premeštanja***. Ako konfiguracija sadrži končan skup susednih ćelija koje kao prvu informaciju sadrže slovo, a kao drugu praznu reč premeštanja onda ona sadrži reč izvedenu L-sistemom.



Slika 3.2: Početna konfiguracija CA.

Na slici 3.4 je data početna konfiguracija ovog CA. Ova konfiguracija predstavlja početnu reč datog $\mathcal{PD}0\mathcal{L}$ sistema.

Ponašanje CA sumirano je prostorno-vremenskim dijagramom na slici 3.3, koji prikazuje prvih dvanaest konfiguracija orbite početne konfiguracije sa prethodne slike koji simuliraju prva dva koraka izvođenja $\mathcal{PD}0\mathcal{L}$ sistema.



Slika 3.3: Ćelijski automat – $\mathcal{PD}0\mathcal{L}$ sistem poređenje: Prvih dvanaest konfiguracija orbite početne konfiguracije CA (levo) simulira prva dva koraka izvođenja $\mathcal{PD}0\mathcal{L}$ sistema (desno).

Ponašanje ovog** CA **može se sumirati i na sledeći način:

- Prva faza, desni marker se „širi" na levo dok ne dostigne levi marker. Kada je ovaj uslov ispunjen, automat je spreman za proces izvođenja sledeće reči jezika $\mathcal{PD}0\mathcal{L}$ sistema, odnosno za primenu produkcija.
- Druga faza, produkcije se primenjuju u suprotnom smeru, s leva na desno. Istovremeno, slova reči jezika $\mathcal{PD}0\mathcal{L}$ sistema koju CA izvodi biće van svog mesta, dok ih CA svojim mehanizmom u nekoliko narednih konfiguracija ne smesti na njihovu krajnju poziciju.
- Treća faza, CA je svako slovo smestio na krajnju poziciju, tj. naredna reč jezika $\mathcal{PD}0\mathcal{L}$ sistema je izvedena i automat je spreman za izvođenje

sledeće reči.

Sledeće stanje svake od ćelija zavisi samo od njenog stanja i stanja njenog prvog levog i prvog desnog suseda. Pretpostavljamo da vreme teče ,,na dole".
Sledeće situacije opisuju različite faze automata:

- širenje desnog markera je predstavljeno u svakoj ćeliji koja kao prvu informaciju ima slovo, a druga informacija, odnosno reč premeštanja je desni marker $<$.

- Produkcije su primenjene na ćelije čija je prva informacija slovo, a reč premeštanja počinje levim markerom $>$. Sa (1) i (2) su označene ćelije u kojima su, respektivno, primenjene produkcije $a \to bb$ i $b \to ab$.

  Nakon primene produkcije, nova slova moraju dostići svoju konačnu poziciju. Ćelije čija je prva informacija desni marker $<$, a reč premeštanja počinje sa levim markerom $>$ prikazuje ovu situaciju, kao i ćelija označena sa (3) u kojoj imamo istovremenu primenu produkcije $b \to ab$ i pomeranje nekih simbola ka njihovoj konačnoj poziciji. Nije teško primetiti da imamo ,,širenje" levog markera $>$ kada su simboli van svoje pozicije. Konačna pozicija prvog slova koje se u reči premeštanja nalazi nakon desnog markera biće upravo ćelija koja tu reč sadrži. Stoga će sledeće stanje te ćelije predstavljati slovo koje je našlo svoju konačnu poziciju, odnosno stanje će kao prvu informaciju imati upravo to slovo, a kao drugu simbol $\diamondsuit$.

- Kada smesti sva slova na svoju poziciju reč je izvedena i automat je spreman za izvođenje sledeće reči, odnosno za ponavljanje čitavog ovog procesa ispočetka.

Ovaj metod je generalizovan i formalizovan u sledećem odeljku.

3.1.2 Formalan opis

Teorema 3.1 *(Alfonseca*[1]*, Dalhoum*[2] *& Ortega*[3]*) Za svaki $\mathcal{PD}0\mathcal{L}$ sistem $G = (W, P, w_0)$ i svako $n \in \mathbb{N}$ postoji jednodimenzionalan CA koji je n-ekvivalentan sa G.*

Dokaz. Dokaz je konstruktivne prirode. Neka je G prizvoljan $\mathcal{PD}0\mathcal{L}$ sistem i n proizvoljan prirodan broj. U sledećim paragrafima biće konstruisan jednodimenzionalni CA koji je n-ekvivalentan sa G.

Da bi dokaz bio jasniji, koristimo sledeću notaciju:

- $W_{m_l} = \{>\}$ – azbuka levog markera.

- $W_{m_d} = \{<\}$ – azbuka desnog markera.

- $W_{e_m} = W \cup W_{m_l} \cup W_{m_d}$ – azbuka W sistema G proširena markerima.

- $W_e = W \cup W_{m_l} \cup W_{m_d} \cup \{\diamondsuit\}$ – azbuka W_{e_m} proširena simbolom prazne reči premeštanja $\diamondsuit$.

[1]Manuel Alfonseca, španski informatičar i pisac, rođen 1946. godine.
[2]Abdel Latif Abu Dalhoum, Jordanski informatičar rođen 1971. godine.
[3]Alfonso Ortega De La Puente, španski informatičar.

- $W_{e,k} = \bigcup_{i=1}^{k} W_e^i$ – skup svih reči dužine ne veće od k nad azbukom W_e.
- $W_k = \bigcup_{i=1}^{k} W^i$ – skup svih reči dužine ne veće od k nad azbukom W.

Ćelijski prostor automata koji konstruišemo mora biti $\mathbb{Z}$, a za vektor susedstva uzimamo $N = (-1, 0, 1)$. Kao što je sugerisano u neformalnom opisu, mora se omogućiti da svaka ćelija sadrži sledeće informacije:

- Jedan simbol iz azbuke W_{e_m}.
- Jednu reč nad azbukom W_e koju treba izmestiti u procesu izvođenja(reč premeštanja).

Prema tome, svaka moguća stanja ćelija CA biće uređeni parovi definisani na sledeći način:

$$S \subseteq W_{e_m} \times W_{e,k+1},$$

gde

- je $k \leq \max_{a \in W}\{|\rho(a)|\} \cdot |h^n(w_0)|$, pri čemu je za sve $a \in W$ $\rho(a)$ sledbenik produkcije čiji je prethodnik a.
- Uređeni parovi $(>, >)$ i $(<, <)$ se koriste za popunjavanje dela ćelijskog prostora $\mathbb{Z}$ koji ostaje neiskorišcen u procesu izvođenja reči jezika $\mathcal{PD}0\mathcal{L}$ sistema pomoću CA.
- Uređeni parovi $(a, <)$, $a \in W$ se koriste za „širenje” desnog markera.
- Uređeni parovi $(a, \diamondsuit)$, $a \in W$ se koriste kada slovo a dostigne konačnu (krajnju) poziciju.
- Uređeni parovi $(b, > \alpha)$, $b \in W \cup \{<\} \wedge \alpha \in W_k$ se koriste za primenu produkcija i izmeštanje podreči, odnosno za dovođenje slova na njihovu konačnu poziciju nakon primene produkcija.

Početna konfiguracija c_0 ovog CA mora predstavljati $w_0 = a_1 a_2 \ldots a_n$, $a_i \in W$, tj. početnu reč sistema G, stoga nju definišemo na sledeći način:

$$c_0(i) = \begin{cases} (a_j, \diamondsuit) & \text{ako } i = j, \\ (>, >) & \text{ako } i < 0, \\ (<, <) & \text{ako } i \geq |w_0|, \end{cases} \tag{3.1}$$

pri čemu uzimamo da je $w_0 = a_0 a_1 \ldots a_{|w_0|-1}$ i $a_i \in W$, za sve $1 \leq i \leq |w_0| - 1$. Lokalno pravilo ažuriranja f definisano je tabelom 3.1, po slučajevima.

Ponašanje CA može biti podeljeno u sledeće faze:

1. Širenje desnog markera

- Ćelije u početnoj konfiguraciji uzimaju samo tri vrste stanja:

  $(>, >)$, $(<, <)$ i stanja oblika $(a, \diamondsuit)$, gde je $a \in W$.

- Slučajevi 1–4 iz tablice funkcije f održavaju stanje $(>, >)$, dok slučajevi 5 i 6 održavaju stanje $(<, <)$. Slučajevi 7 i 8 održavaju stanja oblika $(a, \diamondsuit)$. Primena slučaja 9 i 10 funkcije f na stanja oblika $(a, \diamondsuit)$ inicira „širenje” desnog markera. Sledeće korake u njegovom širenju obavljaju slučajevi 11 i 12. Slučajevi 13–16 beleže da je desni marker već prošao ovu poziciju.

1.	$f((>,>),(>,>),(>,>)) = (>,>)$	
2.	$f((>,>),(>,>),(a,\diamondsuit)) = (>,>)$	$\forall a \in W$
3.	$f((>,>),(>,>),(a,<)) = (>,>)$	$\forall a \in W$
4.	$f((>,>),(>,>),(a,>\alpha)) = (>,>)$	$\forall a \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
5.	$f(<,<),(<,<),(<,< = (<,<)$	
6.	$f((a,\diamondsuit),(<,<),(<,<)) = (<,<)$	$\forall a \in W$
7.	$f((>,>),(a,\diamondsuit),(b,\diamondsuit)) = (a,\diamondsuit)$	$\forall a,b \in W$
8.	$f((b,\diamondsuit),(a,\diamondsuit),(c,\diamondsuit)) = (a,\diamondsuit)$	$\forall a,b,c \in W$
9.	$f((>,>),(a,\diamondsuit),(<,<)) = (a,<)$	$\forall a \in W$
10.	$f((b,\diamondsuit),(a,\diamondsuit),(<,<)) = (a,<)$	$\forall a,b \in W$
11.	$f((>,>),(a,\diamondsuit),(b,<)) = (a,<)$	$\forall a,b \in W$
12.	$f((b,\diamondsuit),(a,\diamondsuit),(c,<)) = (a,<)$	$\forall a,b,c \in W$
13.	$f((b,\diamondsuit),(a,<),(<,<)) = (a,<)$	$\forall a,b \in W$
14.	$f((b,\diamondsuit),(a,<),(c,<)) = (a,<)$	$\forall a,b,c \in W$
15.	$f((b,<),(a,<),(c,<)) = (a,<)$	$\forall a,b,c \in W$
16.	$f((b,<),(a,<),(<,<)) = (a,<)$	$\forall a,b \in W$
17.	$f((>,>),(a,<),(b,<)) = (a,>\rho(a))$	$\forall a,b \in W$
18.	$f((>,>),(a,<),(<,<)) = (a,>\rho(a))$	$\forall a \in W$
19.	$f((>,>),(a,>b\alpha),(c,<)) = (b,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
20.	$f((>,>),(a,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
21.	$f((d,\diamondsuit),(a,>b\alpha),(c,<)) = (b,\diamondsuit)$	$\forall a,b,c,d \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
22.	$f((c,>),(a,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
23.	$f((a,\diamondsuit),(a,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
24.	$f((b,>c\alpha),(a,<),(c,<)) = (a,>\alpha\rho(a))$	$\forall a,b,c \in W, \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
25.	$f((a,>b\alpha),(<,<),(<,<)) = (<,>\alpha)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
26.	$f((<,>b\alpha),(<,<),(<,<)) = (<,>\alpha)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
27.	$f((b,\diamondsuit),(a,\diamondsuit),(<,>\alpha)) = (a,\diamondsuit)$	$\forall a,b \in W, \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
28.	$f((>,>),(a,\diamondsuit),(<,>\alpha)) = (a,\diamondsuit)$	$\forall a \in W, \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
29.	$f((>,>),(a,\diamondsuit),(b,>\alpha)) = (a,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$
30.	$f((b,\diamondsuit),(a,\diamondsuit),(c,>\alpha)) = (a,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ takve da $\|\alpha\| \leq k$

Tabela 3.1: Lokalno pravilo ažuriranja f CA n-ekvivalentnog datom $\mathcal{PD}0\mathcal{L}$ sistemu.

2. Produkcije su primenjene i ubačene podreči traže svoju krajnju poziciju

- Kada se oba markera sretnu na početku reči, širenje desnog markera je završeno. Kada se ovaj uslov ispuni, postoji produkcija koja se može primeniti. Primenu produkcija započinju slučajevi 17 i 18 pravila ažuriranja f. Produkcija se primenjuje na slovo koje predstavlja prvu informaciju koju ta ćelija sadrži. Dakle primena produkcija počinje tamo gde se širenje desnog markera završava, i nastavlja duž reči sa leva na desno.

- Kad god je produkcija primenjena, ceo njen sledbenik se smešta u drugu informaciju ćelije (slučajevi 17–18) koja će u sledećoj konfiguraciji orbite kao prvu informaciju sadržati prvo slovo sledbenika (ovo omogućavaju slučajevi 19–23). Preostala podreč se mora pomerati na desno dok svako slovo ne dostigne svoju konačnu poziciju (poziciju koju ima u reči koju izvodimo). To nam omogućava primena slučajeva 24–26. Ponekad, ove podreči mogu biti pomerane u isto vreme kad se produkcija primenjuje na sledeću ćeliju. U toj situaciji se koristi 24. slučaj pravila ažuriranja f. Slučajevi 25 i 26 se koriste kada je samo pomeranje moguće. Kad god slovo dostigne svoju konačnu poziciju ono biva sačuvano na toj poziciji pomoću slučajeva 27–30 kao i slučajeva 7 i 8 do sledećeg koraka izvođenja.

3. Konfiguracija pridružena sledećoj reči je generisana

- Nakon nekoliko koraka, CA generiše konfiguraciju pridruženu sledećoj reči izvedenoj u $\mathcal{PD}0\mathcal{L}$ sistemu. Kao što je ranije objašnjeno ova konfiguracija sadrži končan skup susednih ćelija koje kao prvu informaciju imaju tačno jedno slovo, druga informacija im je prazna reč premeštanja, a sve ćelije sa leve (desne) strane ovog skupa sadrže samo leve (desne) markere.

$\square$

Primer 3.2 *Metod opisan u prethodnom dokazu primenićemo na $\mathcal{PD}0\mathcal{L}$ sistem G dat u neformalnom opisu. Njegova početna reč je $w_0 = a$ i produkcije:*

$$a \to bb,\ b \to ab.$$

CA n-ekvivalentan $\mathcal{PD}0\mathcal{L}$ sistemu definisan je sa:

$$A = (\mathbb{Z}, S, N, f),$$

gde je

$$S = \{(>,>),(<,<),(a,\diamondsuit),(b,\diamondsuit),(a,<),(b,<),(<,>\alpha),(a,>\alpha),(b,>\alpha)\},$$

$\alpha \in W_k, k \leq 2|h^n(a)|$, $N = (-1,0,1)$ *a pravilo ažuriranja dato tabelom 3.1, ako za početnu konfiguraciju uzmemo*

$$c_0(i) = \begin{cases} (a,\diamondsuit) & ako\ i = 0, \\ (>,>) & ako\ i < 0, \\ (<,<) & ako\ i \geq 1. \end{cases}$$

Ovaj CA će simulirati prvih n koraka izvođenja polaznog $\mathcal{PD}0\mathcal{L}$ sistema G za svako $n \in \mathbb{N}$.

3.2 Jednodimenzionalan CA n-ekvivalentan $\mathcal{D}0\mathcal{L}$ sistemu

3.2.1 Neformalan opis

S obzirom na to da je svaki $\mathcal{PD}0\mathcal{L}$ sistem istovremeno i $\mathcal{D}0\mathcal{L}$ sistem, CA n-ekvivalentan datom $\mathcal{D}0\mathcal{L}$ sistemu posedovaće sve osobine koje poseduje CA opisan u prethodnom odeljku, ali pored njih mora imati i sledeću osobinu:

- Mora posedovati mehanizam koji će omogućiti brisanje slova na koja se primenjuje produkcija oblika $a \to \lambda$ (ako takva postoje) i pomeranje u levo za jedno mesto svih slova u reči koja se nalaze sa desne strane tog slova (onda kada je to potrebno), jer za razliku od $\mathcal{PD}0\mathcal{L}$ sistema skup produkcija $\mathcal{D}0\mathcal{L}$ sistema može sadržati i produkcije oblika $a \to \lambda$.

Način na koji CA simulira $\mathcal{D}0\mathcal{L}$ sistem objasnićemo pomoću primera. Dat je $\mathcal{D}0\mathcal{L}$ sistem čija je početna reč $w_0 = acccb$ i produkcije

$$a \to bc, b \to ac, c \to \lambda.$$

Radi konstrukcije n-ekvivalentnog jednodimenzionalnog CA donete su sledeće odluke:

- U ćelijskom prostoru $\mathbb{Z}$ CA će postojati ćelije koje sadrže slova reči generisanih L-sistemom.

- Stanje svake ćelije CA će sadržati dve informacije, prva je ***tačno jedan simbol*** koji ćelija u datoj konfiguraciji sadrži, a druga je ***reč premeštanja*** koja omogućava:

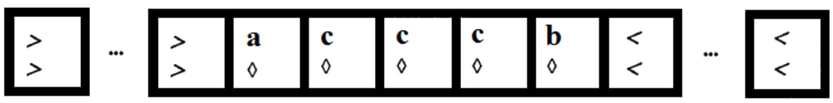
  - ubacivanje novih slova na datu poziciju i pomeranje starih u desno na njihovu krajnju poziciju onda kada je to potrebno, jer se dužina reči može povećati kod $\mathcal{D}0\mathcal{L}$ sistema.

  - brisanje slova na koje se primenjuje produkcija oblika $a \to \lambda$ (u ovom primeru je to slovo c) i pomeranje simbola koji se nalaze sa desne strane tog slova u levo ako je to potrebno, jer će se dužina reči smanjiti ako je broj obrisanih simbola veći od broja ubačenih simbola usled primene produkcija čiji sledbenik nije λ. Prva informacija (simbol) se uvek nalazi iznad druge (reč premeštanja) u grafičkom prikazu ćelija.

- Reči premeštanja zapravo obezbeđuju sprovođenje mehanizma za izvođenje prvih n reči jezika datog $\mathcal{D}0\mathcal{L}$ sistema, odnosno omogućavaju da CA simulira proces izvođenja reči koji se kod L-sistema ostvaruje pomoću produkcija. Reči premeštanja su reči nad azbukom $\{a, b, <, >, \leftarrow, <_{\leftarrow}, <_{\rightarrow}\}$. Simboli $<$ i $>$ su respektivno ***levi*** i ***desni marker*** kao kod $\mathcal{PD}0\mathcal{L}$ sistema, dok se simboli $\{\leftarrow, <_{\leftarrow}, <_{\rightarrow}\}$ koriste pri brisanju slova usled primene produkcije čiji je sledbenik λ i pomeranje svih simbola sa njegove desne strane u levo onda kada je to potrebno.

- Početak i kraj reči koja se izvodi moraju biti markirani. Stanja svih ćelija sa leve strane ćelije koja sadrži prvo slovo reči su ista, i kao prvu i kao drugu informaciju imaju samo levi marker $(>, >)$. Analogno, stanja svih ćelija sa desne strane ćelije koja sadrži poslednje slovo reči su ista, i kao prvu i kao drugu informaciju imaju samo desni marker $(<, <)$.

- Ćelije koje ne sadrže ni levi ni desni marker, kao ni simbole $\{\leftarrow, <_{\leftarrow}, <_{\rightarrow}\}$, sadrže kao drugu informaciju simbol $\Diamond$. Ovaj simbol se naziva ***prazna reč premeštanja***. Ako konfiguracija sadrži končan skup susednih ćelija koje kao prvu informaciju imaju tačno jedno slovo, kao drugu praznu reč premeštanja $\Diamond$, a sve ćelije sa leve (desne) strane ovog skupa sadrže samo $(>, >)$ $((<, <))$ onda ona sadrži reč izvedenu L-sistemom.

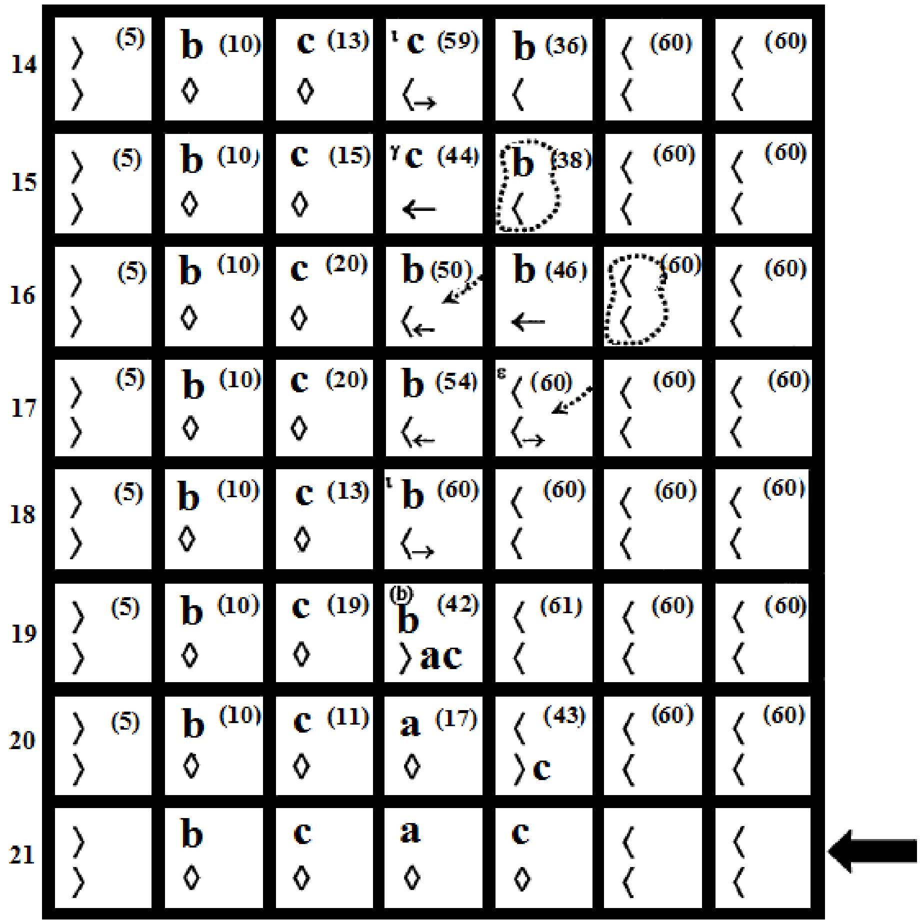


Slika 3.4: Početna konfiguracija c_0 CA.

Na slici 3.4 je data početna konfiguracija CA n-ekvivalentnog datom $\mathcal{D}0\mathcal{L}$ sistemu. Ova konfiguracija predstavlja početnu reč datog $\mathcal{D}0\mathcal{L}$ sistema.

Ponašanje CA sumirano je prostorno-vremenskim dijagramom na slici 3.5, koji prikazuje prvih 21 konfiguracija orbite početne konfiguracije sa prethodne slike koji simuliraju prvi korak izvođenja $\mathcal{D}0\mathcal{L}$ sistema.

	1	2	3	4	5	6	7
1	〉 (5) 〉	a (10) ◊	c (11) ◊	c (11) ◊	c (11) ◊	b (22) ◊	〈 (60) 〈
2	〉 (5) 〉	a (10) ◊	c (11) ◊	c (11) ◊	c (24) ◊	b (26) 〈	〈 (60) 〈
3	〉 (5) 〉	a (10) ◊	c (11) ◊	c (24) ◊	c (27) 〈	b (29) 〈	〈 (60) 〈
4	〉 (5) 〉	a (10) ◊	c (24) ◊	c (29) 〈	c (28) 〈	b (29) 〈	〈 (60) 〈
5	〉 (5) 〉	a (23) ◊	c (27) 〈	c (28) 〈	c (28) 〈	b (29) 〈	〈 (60) 〈
6	〉 (6) 〉	a (30) 〈	c (28) 〈	c (28) 〈	c (28) 〈	b (29) 〈	〈 (60) 〈
7	〉 (7) 〉	[(a)] a (39) 〉bc	c (34) 〈	c (28) 〈	c (28) 〈	b (29) 〈	〈 (60) 〈
8	〉 (5) 〉	b (12) ◊	$^{\alpha}$c (41) 〉c	$^{\beta}$c (35) 〈	c (28) 〈	b (29) 〈	〈 (60) 〈
9	〉 (5) 〉	b (10) ◊	c (15) ◊	$^{\gamma}$c (44) ←	c (37) 〈	b (29) 〈	〈 (60) 〈
10	〉 (5) 〉	b (10) ◊	c (20) ◊	$^{\delta}$c (50) 〈←	c (45) ←	b (38) 〈	〈 (60) 〈
11	〉 (5) 〉	b (10) ◊	c (20) ◊	c (52) 〈←	b (49) 〈←	b (46) ←	〈 (60) 〈
12	〉 (5) 〉	b (10) ◊	c (20) ◊	c (52) 〈←	b (56) 〈←	$^{\varepsilon}$〈 (60) 〈→	〈 (60) 〈
13	〉 (5) 〉	b (10) ◊	c (20) ◊	c (54) 〈←	b (64) 〈→	〈 (60) 〈	〈 (60) 〈



Slika 3.5: Prvih 21 konfiguracija orbite početne konfiguracije c_0 CA simulira prvi korak izvođenja $\mathcal{D}0\mathcal{L}$ sistema.

Ponašanje ovog CA ***može se sumirati i na sledeći način***:

- Prva faza, desni marker se ,,širi" na levo dok ne dostigne levi marker. Kada je ovaj uslov ispunjen, automat je spreman za proces izvođenja sledeće reči jezika $\mathcal{D}0\mathcal{L}$ sistema, odnosno za primenu produkcija (na slici 3.5 konfiguracije $1-6$).

- Druga faza, produkcije se primenjuju u suprotnom smeru, s leva na desno. Istovremeno, slova reči jezika $\mathcal{D}0\mathcal{L}$ sistema koju CA izvodi biće van svog mesta, dok ih CA svojim mehanizmom u nekoliko narednih konfiguracija ne smesti na njihovu krajnju poziciju (na slici 3.5 konfiguracije $7-20$).

- Treća faza, CA je svojim mehanizmom smestio sva slova na krajnju poziciju, tj. naredna reč jezika $\mathcal{D}0\mathcal{L}$ sistema je izvedena i automat je spreman za izvođenje sledeće reči (na slici 3.5 konfiguracija 21).

Sledeće stanje svake od ćelija zavisi samo od njenog stanja i stanja njenog prvog levog i prvog desnog suseda. Pretpostavljamo da vreme teče ,,na dole".

Sledeće situacije opisuju prethodno navedene faze automata:

- U prvoj fazi, desni marker koji je druga informacija ćelije se širi na levo kada levi sused te ćelije kao prvu informaciju ima slovo, a kao drugu praznu reč premeštanja.

- U drugoj fazi, produkcije se primenjuju jedna po jedna i primena svake produkcije je praćena smeštanjem slova na krajnju poziciju:

  – Nakon primene produkcija novi simboli moraju biti smešteni na njihovu krajnju poziciju. To je označeno sa $>$ na početku reči premeštanja.

  – Produkcije čiji je sledbenik neprazna reč su primenjene u ćelijama koje kao prvu informaciju imaju slovo, a reč premeštanja im počinje sa $>$. U ćeliji označenoj sa (a) primenjena je produkcija $a \to bc$, a u ćeliji označenoj sa (b) produkcija $b \to ac$.

  – Prvo slovo iza simbola $>$ u reči premeštanja postaje prva informacija te ćelije u narednoj konfiguraciji, jer je našlo svoju konačnu poziciju. Levi marker $>$ i preostali deo reči premeštanja su pomereni jedno mesto u desno (isto kao kod $\mathcal{PD}0\mathcal{L}$ sistema).

  – Ako je sledbenik primenjene produkcije λ, moguće je da podreč na koju još nisu primenjene produkcije (ćelije između trenutne pozicije i desnog markera) mora biti pomerena za jedno mesto u levo. U ovom slučaju ćelija na koju je primenjena ovakva produkcija šalje signal $\leftarrow$ na desno (ćelije označene se γ). Ovaj signal se „širi" na desno dok god ne dostigne ćeliju čije je stanje $(<,<)$. Ćelija šalje ovaj signal kada reč premeštanja njenog prvog levog suseda pripada skupu $\{> a, a \in W\}$.

  – Ćelije preko kojih je prešao signal $\leftarrow$ u narednoj konfiguraciji kao drugu informaciju imaju $<_{\leftarrow}$, a kao prvu informaciju imaju kopiju prve informacije prvog desnog suseda. Isprekidane strelice označavaju ovu situaciju.

  – Kada signal $\leftarrow$ dostigne $(<,<)$ pomera se na levo dok ne pronađe ćeliju koja kao drugu informaciju ima $\diamondsuit$. Tada se simbol $<_{\rightarrow}$ koristi za označavanje ćelija preko kojih je signal $\leftarrow$ prešao u povratku. Tek nakon što signal $\leftarrow$ dostigne praznu reč premeštanja $\diamondsuit$ automat primenjuje sledeću produkciju.

  – Ovaj postupak pomeranja u levo nije sproveden nakon primene produkcije $c \to \lambda$ u ćeliji označenoj sa α jer je ovde mesto izbrisanog slova c popunjeno poslednjim slovom sledbenika prethodno primenjene produkcije.

  – Za identifikovanje ćelije koja sadrži simbol koji je na svojoj konačnoj poziciji koristimo praznu reč premeštanja $\diamondsuit$.

- U trećoj fazi su sva slova dostigla svoju krajnju poziciju i automat je spreman za izvođenje sledeće reči. U ovoj konfiguraciji se pojavljuju samo stanja oblika $(<,<), (a,\diamondsuit)$ i $(>,>)$.

Ovaj metod je genralizovan i formalizovan u sledećem odeljku.

3.2.2 Formalan opis

Teorema 3.3 *(Alfonseca, Dalhoum & Ortega) Za svaki $\mathcal{D}0\mathcal{L}$ sistem $G = (W, P, w_0)$ i svako $n \in \mathbb{N}$ postoji jednodimenzionalan CA koji je n-ekvivalentan sa G.*

Dokaz. Neka je $n \in \mathbb{N}$ proizvoljan prirodan broj i G proizvoljan $\mathcal{D}0\mathcal{L}$ sistem. Slično kao i za $\mathcal{PD}0\mathcal{L}$ sistem konstruišemo jednodimenzionalni CA koji je n-ekvivalentan sa G. Azbuke levog i desnog markera, W_{m_l} i W_{m_d}, su iste kao dokazu prethodne teoreme, dok su ostale azbuke nešto drugacije jer uvodimo još i

- $W_{m_s} = \{<_\rightarrow, <_\leftarrow, \leftarrow\}$ – ***azbuku signala***, koja se koristi za skraćivanje reči usled primene produkcija obilika $a \rightarrow \lambda,\ a \in W$.
- $W_m = W \cup W_{m_l} \cup W_{m_d}$ – azbuka W sistema G proširena markerima.
- $W_{e_m} = W_m \cup W_{m_s}$ – azbuka W sistema G proširena markerima i signalima.
- $W_e = W_{e_m} \cup \{\diamondsuit\}$ – azbuka W_{e_m} proširena simbolom $\diamondsuit$ prazne reči premeštanja.
- $W_{e,k} = \bigcup_{i=1}^{k} W_e^i$ – skup svih reči dužine ne veće od k nad azbukom W_e.
- $W_k = \bigcup_{i=1}^{k} W^i$ – skup svih reči dužine ne veće od k nad azbukom W.

Ćelijski prostor CA koji konstruišemo je $\mathbb{Z}$, a vektor susedstva $N = (-1, 0, 1)$. Mora se omogućiti da svaka ćelija sadrži sledeće informacije:

- Jedan simbol iz azbuke W_m.
- Jednu reč nad azbukom W_e koju treba izmestiti u procesu izvođenja (reč premeštanja).

Stoga, svaka moguća stanja ćelija CA biće uređeni parovi definisani na sledeći način:

$$S \subseteq W_m \times (W_{m_l} \cdot (W_k \cup \{\lambda\}) \cup W_{m_s} \cup W_m \cup \diamondsuit),$$

gde

- je $k \leq \max_{a \in W}\{|\rho(a)|\} \cdot |h^n(w_0)|$, pri čemu je za sve $a \in W$ $\rho(a)$ sledbenik produkcije čiji je prethodnik a.
- Uređeni parovi $(>, >)$ i $(<, <)$ se koriste, respektivno, kao levi i desni markeri koji popunjavaju deo ćelijskog prostora $\mathbb{Z}$ koji ostaje neiskorišćen u procesu izvođenja reči jezika $\mathcal{D}0\mathcal{L}$ sistema pomoću CA.
- Uređeni parovi $(a, <)$, $a \in W$ se koriste za „širenje" desnog markera.
- Uređeni parovi $(a, \diamondsuit)$, $a \in W$ se koriste kada slovo a dostigne krajnju poziciju.
- Uređeni parovi $(b, > \alpha)$, $b \in W \cup \{<\} \wedge \alpha \in W_k$ se koriste za primenu produkcija čiji je sledbenik neprazan i za dovođenje slova sledbenika produkcije na njihovu konačnu poziciju nakon primene produkcije.
- Uređeni parovi (c, d), $c \in W \cup \{>\} \wedge d \in W_s$ se koriste za skraćivanje izvedenih reči nakon primene produkcije oblika $a \rightarrow \lambda$.

Početna konfiguracija ovog CA mora predstavljati početnu reč $w_0 = a_1 a_2 \ldots a_n$, $a_i \in W$ sistema G i nju definišemo isto kao kod $\mathcal{PD}0\mathcal{L}$ sistema (vidi (3.1)).

1.	$f((>,>),(>,>),(>,>)) = (>,>)$	
2.	$f((>,>),(>,>),(<,<)) = (>,>)$	
3.	$f((>,>),(>,>),(<,<_{\leftarrow})) = (>,>)$	
4.	$f((>,>),(>,>),(a,\leftarrow)) = (>,>)$	$\forall a \in W$
5.	$f((>,>),(>,>),(a,\diamondsuit)) = (>,>)$	$\forall a \in W$
6.	$f((>,>),(>,>),(a,<)) = (>,>)$	$\forall a \in W$
7.	$f((>,>),(>,>),(a,>\alpha)) = (>,>)$	$\forall a \in W$, $\forall \alpha \in W^*$ za koje je $\lvert\alpha\rvert \leq k$
8.	$f(<,<),(<,<),(<,<) = (<,<)$	
9.	$f(>,>),(<,<),(<,<) = (<,<)$	
10.	$f((>,>),(a,\diamondsuit),(b,\diamondsuit)) = (a,\diamondsuit)$	$\forall a,b \in W$
11.	$f((b,\diamondsuit),(a,\diamondsuit),(c,\diamondsuit)) = (a,\diamondsuit)$	$\forall a,b,c \in W$
12.	$f((>,>),(a,\diamondsuit),(b,>\alpha)) = (a,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
13.	$f((b,\diamondsuit),(a,\diamondsuit),(c,<_{\rightarrow})) = (a,\diamondsuit)$	$\forall a,b,c \in W$
14.	$f((>,>),(a,\diamondsuit),(c,<_{\rightarrow})) = (a,\diamondsuit)$	$\forall a,b \in W$
15.	$f((b,\diamondsuit),(a,\diamondsuit),(c,\leftarrow)) = (a,\diamondsuit)$	$\forall a,b,c \in W$
16.	$f((>,>),(a,\diamondsuit),(b,\leftarrow)) = (a,\diamondsuit)$	$\forall a,b \in W$
17.	$f((b,\diamondsuit),(a,\diamondsuit),(<,>\alpha)) = (a,\diamondsuit)$	$\forall a,b \in W$, $\forall \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
18.	$f((>,>),(a,\diamondsuit),(<,>\alpha)) = (a,\diamondsuit)$	$\forall a,b \in W\ \forall \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
19.	$f((b,\diamondsuit),(a,\diamondsuit),(c,>\alpha)) = (a,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
20.	$f((b,\diamondsuit),(a,\diamondsuit),(c,<_{\leftarrow})) = (a,\diamondsuit)$	$\forall a,b \in W$
21.	$f((>,>),(a,\diamondsuit),(b,<_{\leftarrow})) = (a,\diamondsuit)$	$\forall a,b \in W\ \forall \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
22.	$f((b,\diamondsuit),(a,\diamondsuit),(<,<)) = (a,<)$	$\forall a,b \in W$
23.	$f((>,>),(a,\diamondsuit),(b,<)) = (a,<)$	$\forall a,b \in W$
24.	$f((b,\diamondsuit),(a,\diamondsuit),(c,<)) = (a,<)$	$\forall a,b,c \in W$
25.	$f((>,>),(a,\diamondsuit),(<,<)) = (a,<)$	$\forall a \in W$
26.	$f((b,\diamondsuit),(a,<),(<,<)) = (a,<)$	$\forall a,b \in W$
27.	$f((b,\diamondsuit),(a,<),(c,<)) = (a,<)$	$\forall a,b,c \in W$
28.	$f((b,<),(a,<),(c,<)) = (a,<)$	$\forall a,b,c \in W$
29.	$f((b,<),(a,<),(<,<)) = (a,<)$	$\forall a,b \in W$
30.	$f((>,>),(a,<),(b,<)) = (a,>\rho(a))$	$\forall a,b \in W$ za koje je $\rho(a) \neq \lambda$
31.	$f((>,>),(a,<),(b,<)) = (a,\leftarrow)$	$\forall a,b \in W$, za koje je $\rho(a) = \lambda$
32.	$f((>,>),(a,<),(<,<)) = (a,>\rho(a))$	$\forall a \in W$ za koje je $\rho(a) \neq \lambda$
33.	$f((>,>),(a,<),(<,<)) = (a,\leftarrow)$	$\forall a \in W$ za koje je $\rho(a) = \lambda$
34.	$f((b,>c\alpha),(a,<),(c,<)) = (a,>\alpha\rho(a))$	$\forall a,b,c \in W$, $\forall \alpha \in W^+$ za koje je $1 \leq \lvert\alpha\rvert \leq k$
35.	$f((b,>c),(a,<),(d,<)) = (a,\leftarrow)$	$\forall a,b,c,d \in W$
36.	$f((b,<_{\rightarrow}),(a,<),(<,<)) = (a,<)$	$\forall a,b \in W$
37.	$f((b,\leftarrow),(a,<),(c,<)) = (a,\leftarrow)$	$\forall a,b,c \in W$
38.	$f((b,\leftarrow),(a,<),(<,<)) = (a,\leftarrow)$	$\forall a,b \in W$
39.	$f((>,>),(a,>b\alpha),(c,<)) = (b,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
40.	$f((>,>),(a,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
41.	$f((d,\diamondsuit),(a,>b\alpha),(c,<)) = (b,\diamondsuit)$	$\forall a,b,c,d \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
42.	$f((c,\diamondsuit),(a,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b,c \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
43.	$f((a,\diamondsuit),(<,>b\alpha),(<,<)) = (b,\diamondsuit)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
44.	$f((b,\diamondsuit),(a,\leftarrow),(c,<)) = (c,<_{\leftarrow})$	$\forall a,b,c \in W$
45.	$f((b,<_{\leftarrow}),(a,\leftarrow),(c,<)) = (c,<_{\leftarrow})$	$\forall a,b,c \in W$
46.	$f((b,<_{\leftarrow}),(a,\leftarrow),(<,<)) = (<,<_{\rightarrow})$	$\forall a,b \in W$
47.	$f((>,>),(a,\leftarrow),(<,<)) = (<,<_{\rightarrow})$	$\forall a \in W$
48.	$f((>,>),(a,\leftarrow),(b,<)) = (a,<_{\leftarrow})$	$\forall a,b \in W$
49.	$f((b,<_{\leftarrow}),(a,<_{\leftarrow}),(c,\leftarrow)) = (a,<_{\leftarrow})$	$\forall a,b,c \in W$
50.	$f((b,\diamondsuit),(a,<_{\leftarrow}),(c,\leftarrow)) = (a,<_{\leftarrow})$	$\forall a,b,c \in W$
51.	$f((b,<_{\leftarrow}),(a,<_{\leftarrow}),(c,<_{\rightarrow})) = (a,<_{\rightarrow})$	$\forall a,b,c \in W$
52.	$f((b,\diamondsuit),(a,<_{\leftarrow}),(c,<_{\leftarrow})) = (a,<_{\leftarrow})$	$\forall a,b,c \in W$
53.	$f((b,<_{\leftarrow}),(a,<_{\leftarrow}),(c,<_{\leftarrow})) = (a,<_{\leftarrow})$	$\forall a,b,c \in W$
54.	$f((b,\diamondsuit),(a,<_{\leftarrow}),(c,<_{\rightarrow})) = (a,<_{\rightarrow})$	$\forall a,b,c \in W$
55.	$f((b,\diamondsuit),(a,<_{\leftarrow}),(<,<_{\rightarrow})) = (a,<_{\rightarrow})$	$\forall a,b \in W$
56.	$f((b,<_{\leftarrow}),(a,<_{\leftarrow}),(<,<_{\rightarrow})) = (a,<_{\rightarrow})$	$\forall a,b \in W$
57.	$f((>,>),(a,<_{\leftarrow}),(b,<_{\leftarrow})) = (a,<_{\leftarrow})$	$\forall a,b \in W$
58.	$f((a,<_{\leftarrow}),(<,<_{\rightarrow}),(<,<)) = (<,<)$	$\forall a \in W$
59.	$f((b,\diamondsuit),(a,<_{\rightarrow}),(c,<)) = (a,\leftarrow)$	$\forall a,b \in W$
60.	$f((b,x),(<,<),(<,<)) = (<,<)$	$\forall a \in W,\ \forall x \neq> \alpha,\ \alpha \in W^+$
61.	$f((a,>b\alpha),(<,<),(<,<)) = (<,>\alpha)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
62.	$f((>,>),(<,<_{\rightarrow}),(<,<)) = (<,<)$	
63.	$f((b,\diamondsuit),(a,<_{\rightarrow}),(<,<)) = (a,>\rho(a))$	$\forall a \in W$
64.	$f((b,<_{\leftarrow}),(a,<_{\rightarrow}),(<,<)) = (a,<)$	$\forall a,b \in W$
65.	$f((<,>b\alpha),(<,<),(<,<)) = (<,>\alpha)$	$\forall a,b \in W,\ \forall \alpha \in W^+$ za koje je $\lvert\alpha\rvert \leq k$
66.	$f((>,>),(a,<_{\leftarrow}),(b,\leftarrow)) = (a,<_{\leftarrow})$	$\forall a,b \in W$

Tabela 3.2: Lokalno pravilo ažuriranja f CA n-ekvivalentnog datom $\mathcal{D}0\mathcal{L}$ sistemu.

Lokalno pravilo ažuriranja f definisano je pomoću gornje tabele.

- Slučajevi $1 - 7$ očuvavaju levi marker.
- Slučajevi 8, 9 i 60 očuvavaju desni marker.
- Slučajevi $10 - 21$ zadržavaju slova na njihovim krajnjim pozicijama.
- Slučajevi $22 - 25$ sprovode ,,širenje" levog markera.
- Slučajevi $26 - 29$ očuvavaju slova koja ,,čekaju" da se na njih primeni odgovarajuća produkcija.
- Slučajevi $30 - 34$ primenjuju produkcije.
- Slučajevi $35-38$, $44-59$ i $62-64$ i 66 omogućavaju da se nakon primene produkcije oblika $a \to \lambda$ na neko slovo, podreč koja se nalazi sa desne strane tog slova pomeri u levo za jedno mesto.
- Slučajevi $39 - 43$ smeštaju slova na njihovu krajnju poziciju.
- Slučajevi 61 i 65 pomeraju podreč sledbenika koja nije smeštena na svoju krajnju poziciju na desno.

Ova funkcija definiše mehanizam za brisanje simbola opisan u prethodnom odeljku. □

Primer 3.4 *Jednodimenzionalni CA n-ekvivalentan $\mathcal{D}0\mathcal{L}$ sistemu iz neformalnog opisa čija je početna reč $w_0 = acccb$ i produkcije:*

$$a \to bc,\ b \to ac,\ c \to \lambda$$

definisan je sa

$$A = (\mathbb{Z},\ S,\ N,\ f)$$

gde je skup stanja

$$S = \left\{ \begin{array}{l} (<,<),\ (>,>),\ (a,\diamondsuit),\ (b,\diamondsuit),\ (c,\diamondsuit),\ (a,<),\ (b,<), \\ (c,<),\ (a,>\alpha),\ (b,>\alpha),\ (c,>\alpha),\ (c,\leftarrow),\ (c,<_{\leftarrow}), \\ (b,<_{\leftarrow}),(a,<_{\leftarrow}),\ (<,<_{\rightarrow}),\ (c,<_{\rightarrow}),\ (b,<_{\rightarrow}),\ (a,<_{\rightarrow}) \end{array} \right\},$$

$\alpha \in W_k, k \leq 2|h^n(a)|$, $N = (-1, 0, 1)$, *ako je početna konfiguracija*

$$c_0(i) = \begin{cases} (>,>) & ako\ i < 0, \\ (a,\diamondsuit) & ako\ i = 0, \\ (c,\diamondsuit) & ako\ i = 1, \\ (c,\diamondsuit) & ako\ i = 2, \\ (c,\diamondsuit) & ako\ i = 3, \\ (b,\diamondsuit) & ako\ i = 4, \\ (<,<) & ako\ i \geq 5. \end{cases}$$

Funkciju prelaska definisana je pomoću prethodne tablice. Na slici 3.5 je dat prostorno-vremenski dijagram ovog CA gde je predstavljeno prvih 21 konfiguracija orbite početne konfiguracije c_0, koje simuliraju prvi korak izvođenja datog $\mathcal{D}0\mathcal{L}$ sistema. Broj (xy) u gornjem desnom uglu svake od ćelija na slici 3.5 predstavlja redni broj slučaja iz prethodne tablice koji je na tu ćeliju primenjen. Na *ćeliju označenu sa* (α) *primenjuje se produkcija* $c \to \lambda$. *Podreč koja se nalazi sa njene desne strane neće biti pomerena za jedno mesto u levo, jer će poslednje slovo prethodno primenjene produkcije popuniti mesto izbrisanog slova* c. *Na ćeliju označenu sa* (β) *primenjuje se produkcija* $c \to \lambda$. *Podreč koja se nalazi sa njene desne strane mora biti pomerena za jedno mesto u levo, jer nema slova iz prethodno primenjenih produkcija koje bi popunilo mesto izbrisanog slova* c. *Ćelije označene sa* (γ) *šalju signal* $\leftarrow$ *na desno dok on ne dostigne ćeliju čije je stanje* $(<,<)$. *Kada ovaj signal pređe preko neke ćelije njena prva informacija postaje prva informacija ćelije koja se nalazi ispred nje. Ovo pomeranje čitave podreči koja se nalazi sa desne strane ćelije označene sa* (β) *završava se kada se signal* $\leftarrow$ *vrati do ove ćelije nakon što se odbije od kraja reči markiranog sa* $(<,<)$. (δ) *Ćelije koje čekaju povratak signala* $\leftarrow$ *kao drugu informaciju imaju* $<_{\leftarrow}$. (ε) *Signal* $\leftarrow$ *se odbija od* $(<,<)$ *i ćelije preko kojih je prešao u povratku kao drugu informaciju imaju* $<_{\rightarrow}$. (ι) *Kada signal* $\leftarrow$ *u povratku pronađe ćeliju čija je druga informacija* $\diamondsuit$ *pomeranje u levo je završeno i nova produkcija može biti primenjena.*

Primetimo da nisu svi slučajevi iz prethodne tablice našli svoju primenu u ovih 21 konfiguracija. Njihovu primenu imamo u sledećim primerima.

Primer 3.5 *Razmatramo* $\mathcal{D}0\mathcal{L}$ *sistem čija je početna reč* $w_0 = acccccb$, *a produkcije*

$$a \to \lambda,\ b \to \lambda,\ c \to \lambda.$$

Jednodimenzionalan CA n-ekvivalnetan ovom $\mathcal{D}0\mathcal{L}$ *sistemu je*

$$A = (\mathbb{Z},\ S,\ N,\ f)$$

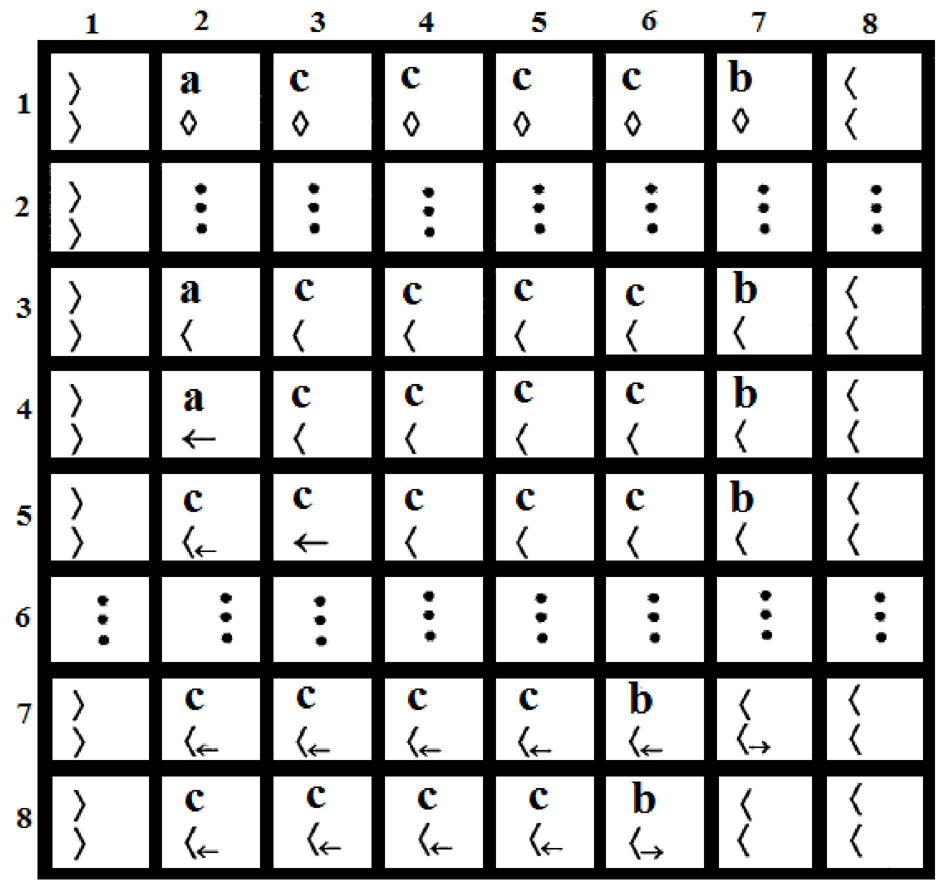
za početnu konfiguraciju

$$c_0(i) = \begin{cases} (>,>) & ako\ i < 0, \\ (a,\diamondsuit) & ako\ i = 0, \\ (c,\diamondsuit) & ako\ i = 1, \\ (c,\diamondsuit) & ako\ i = 2, \\ (c,\diamondsuit) & ako\ i = 3, \\ (c,\diamondsuit) & ako\ i = 4, \\ (b,\diamondsuit) & ako\ i = 5, \\ (<,<) & ako\ i \geq 6. \end{cases}$$

Skup stanja

$$S = \left\{ \begin{array}{l} (<,<),\ (>,>),\ (a,\diamondsuit),\ (b,\diamondsuit),\ (c,\diamondsuit),\ (a,<),\ (b,<),\ (c,<),\ (c,\leftarrow), \\ (c,<_{\leftarrow}),(b,<_{\leftarrow}),(a,<_{\leftarrow}),\ (<,<_{\rightarrow}),\ (c,<_{\rightarrow}),\ (b,<_{\rightarrow}),\ (a,<_{\rightarrow}) \end{array} \right\},$$

vektor susedstva $N = (-1,0,1)$ *i za pravilo ažuriranja* f *koristimo prethodnu tabelu. Na sledećoj slici je prikazano kako ovaj CA simulira dati* $\mathcal{D}0\mathcal{L}$ *sistem. Redovi 1-3 predstavljaju širenje levog markera, u 4-5 je signal* $\leftarrow$ *poslat na desno, a u 7-8 ovaj signal se vraća nakon što je dostigao* $(<,<)$. *Ovaj sistem briše sva slova početne reči i njegov jezik sadrži samo praznu reč* λ. *Primetimo da ovde imamo primenu slučajeva 31,48,57 i 66 iz tabele lokalnog pravila ažuriranja* f *koji nisu imali primenu u prethodnom primeru.*



Primer 3.6 *Neka je $\mathcal{D}0\mathcal{L}$ sistem zadat početnom rečju $w_0 = a$ i produkcijama*

$$a \to bca, \;\; b \to b, \;\; c \to b.$$

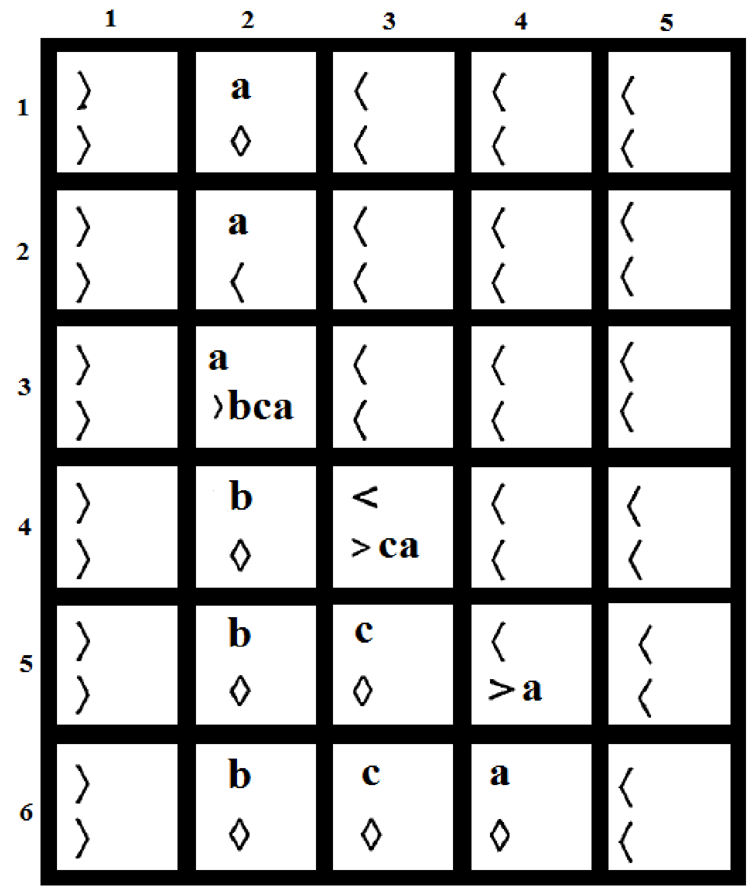
Njemu n-ekvivalentan jednodimenzionalni CA je $A = (\mathbb{Z},\ S,\ N,\ f)$ za početnu konfiguraciju

$$c_0(i) = \begin{cases} (>,>) & ako\ i < 0, \\ (a,\diamondsuit) & ako\ i = 0, \\ (<,<) & ako\ i > 0, \end{cases}$$

pri čemu su N i f isti kao i ranije, dok je skup stanja

$$S = \left\{ \begin{array}{l} (<,<),\ (>,>),\ (a,\diamondsuit),\ (b,\diamondsuit),\ (c,\diamondsuit),\ (a,<),\ (b,<), \\ (c,<),\ (a,>\alpha),\ (b,>\alpha),\ (c,>\alpha) \end{array} \right\},$$

$\alpha \in W_k, k \leq 2|h^n(a)|$, Na sledećoj slici je prikazano kako ovaj CA simulira dati $\mathcal{D}0\mathcal{L}$ sistem. Primetimo da je ovaj sistem istovremeno i $\mathcal{PD}0\mathcal{L}$ sistem i da se njemu n-ekvivalentan CA konstruiše kao sto je opisano u odeljku 3.1.2.



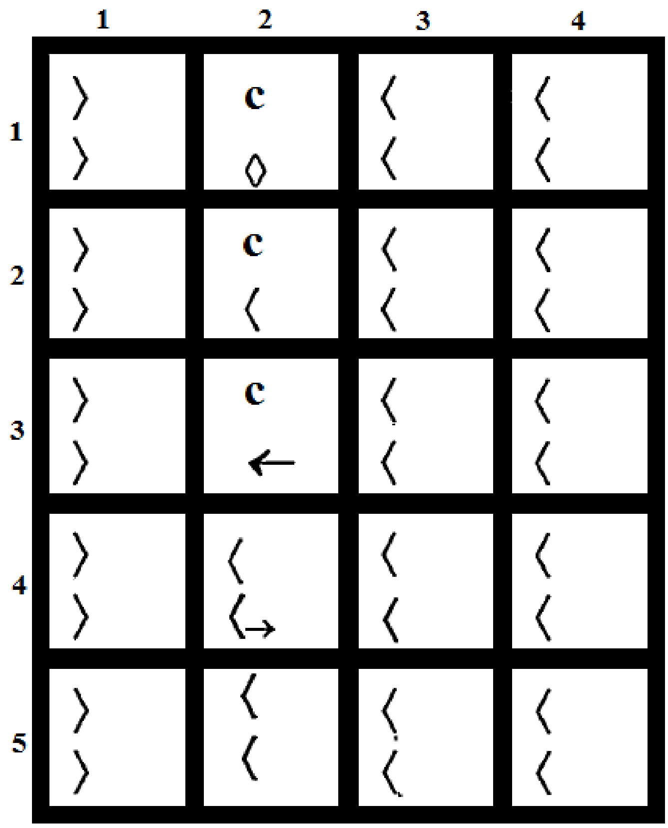
Primer 3.7 *Jednodimenzionalan CA n-ekvivalentan* $\mathcal{D}0\mathcal{L}$ *sistemu čija je početna reč* $w_0 = c$ *i jedina produkcija*

$$c \to \lambda$$

je $A = (\mathbb{Z},\ S,\ N,\ f)$ *za početnu konfiguraciju*

$$c_0(i) = \begin{cases} (>,>) & \textit{ako } i < 0, \\ (c, \Diamond) & \textit{ako } i = 0, \\ (<,<) & \textit{ako } i > 0. \end{cases}$$

Na sledećoj slici je prikazan način na koji ovaj CA simulira dati $\mathcal{D}0\mathcal{L}$ *sistem. Ovde su primenjeni slučajevi 25,33,47 i 62 iz tablice funkcije* f *na ćelije u drugoj vrsti redom.*



Manuel Alfonseca i Abdel Latif Abu Dalhoum su takođe pokazali da za proizvoljan determinstički kontekstno osetljiv L-sistem ($\mathcal{DIL}$ sistem) i proizvoljno $n \in \mathbb{N}$ postoji jednodimenzionalan CA koji je njemu n-ekvivalentan. Konstrukcija ovog CA je osetno drugačija od konstrukcija u slučaju $\mathcal{PD}0\mathcal{L}$ i $\mathcal{D}0\mathcal{L}$ sistema, jer je sistem kontekstno osetljiv. S' obzirom na to da smo u drugom poglavlju daleko više pažnje posvetili kontekstno slobodnim L-sistemima nego kontekstno osetljivim ovom n-ekvivalencijom se u ovom radu nećemo baviti. Prethodno pomenti autori imaju u planu da prošire ove rezultate na beskonačan broj koraka izvođenja i kompleksnije tipove L-sistema kao što su npr. tabelarni L-sistemi, a sve to sa ciljem da dobiju jedan opšti rezultat.

Rezultati Alfonseca-e, Ortega-e i Suarez-a vezani za obratan smer u kojem se za dati CA konstruiše njemu ekvivalentan L-sistem osetno prevazilaze granice rezultata prikazanih u ovom radu, pa se njima ovde nećemo baviti, ali ćemo dati jedan interesantan primer.

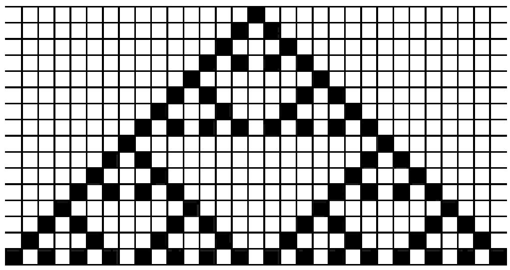
Primer 3.8 *Neka je dat elementarni CA čiji je Wolfram-ov broj 90. U primeru 1.16 smo videli da pomoću ovog CA za početnu konfiguraciju c_0 kod koje je $c_0(0) = 1$ i $c_0(i) = 0$ za sve $i \neq 0$ dobijamo trougao Sierpinski-og.*

Dalje razmatramo deterministički L-sistem čija je azbuka ista kao skup stanja datog CA odnosno $W = \{0, 1\}$, a skup produkcija P dobijen direktno iz lokalnog *pravila ažuriranja f datog CA (pravilo 90), odnosno*

$$P = \{111 \to 0, 110 \to 1, 101 \to 0, 100 \to 1, 011 \to 1, 010 \to 0, 001 \to 1, 000 \to 0\}.$$

Na simbole u reči sa čije se desne (leve) strane nalazi prazna reč λ primenjivaćemo odgovarajuću produkciju u čijem se prethodniku sa desne (leve) strane tog simbola nalazi 0. Ako za početnu reč uzmemo npr. $w_0 = 0^{15}10^{15}$ prvih 16 reči izvedenih L-sistemom $G_{90} = (W, P, w_0)$ će se ,,poklapati" sa prvih 16 konfiguracije orbite početne konfiguracije c_0 datog CA. Na sledećoj slici je predstavljeno redom prvih 16 reči izvedenih ovim L-sistemom, pri čemu najviši red predstavlja početnu reč, a svaki sledeći narednu reč dobijenu u procesu izvođenja. Crn kvadrat predstavlja simbol 1, a beo simbol 0.

Ovaj L-sistem je $(1,1)\mathcal{DIL}$, odnosno deterministički kontekstno osetljiv sistem u kojem zamena svakog slova u reči zavisi od slova koje se nalazi sa njegove leve i slova koje se nalazi sa njegove desne strane.



Primetimo da nas ovde ograničava činjenica da ćelijski prostor $\mathbb{Z}$ datog CA ima beskonačno mnogo ćelija, a da početna reč w_0 L-sistema mora biti konačna. Stoga i ovde možemo govoriti samo o nekom obliku n-ekvivalencije.

U osnovi ideje za konstrukciju L-sistema ekvivalentnog datom CA je upravo poistovećivanje skupa stanja sa azbukom, a pravila ažuriranja f sa skupom produkcija. Čitaoce koje detaljnije zanima ova tema upućujemo na [22] i [23]. Vezano temu ekvivalencije CA i L-sistema jako su poznati i rezultati A. Stauffer-a i M. Sipper-a iz 1997. godine o opisivanju komponenti ćelijskog prostora CA pomoću L-sistema ([9]), kao i rad J.R.Koza-e iz 1993. u kojem se primenom genetičkog programiranja i na L-sisteme i na CA pokazuje da mora postojati strukturna veza između njih ([21]).

Zaključak

Literatura

[1] J. Kari, *Cellular Automata*, University of Turku, spring 2016.

[2] P. Rendell, *Turing machine universality of the game of life*, PhD, University of the West of England, 2014.

[3] P. Prusinkiewicz, J. Hanan, *Lindenmayer systems, Fractals, and Plants*, Springer-Verlag, New York, 1989.

[4] P. Prusinkiewicz, J. Hanan, R. Mech *An L-system-based plant modeling language*, Lecture Notes in Computer Science 1779, Springer, Berlin, 2000, 395-410.

[5] P.M.B. Vitányi, *Lindenmayer systems: Structure, languages, growth functions*, Mathematisch centrum, Amsterdam, 1978.

[6] T. Smith, *On infinite words determined by L systems*, in: Theoretical computer science 595, 2015, 1-10.

[7] L. Kari, G. Rozenberg, A. Salomaa, *L systems*, in: Handbook of Formal Languages, Vol. 1, Springer-Verlag, New York, 1997, 253-328.

[8] A. Lindenmayer, *Mathematical Models for Cellular Interactions in Development, I and II*, in: J. Theor. Biol. 18. (1968), 280-315.

[9] A. Stauffer, M. Sipper, *On the relationship between cellular automata and L-systems: The self-replication case*, in: Physica D 116, 1998, 71-80.

[10] M. Alfonseca, A. Ortega, *Representation of some cellular automata by means of equivalent L Systems*, in: Comlexity international Vol. 7, 2000, 1-16.

[11] A.L.Abu Dalhoum, A. Ortega, M. Alfonseca, *Cellular Automata equivalent to D0L Systems*, in: WSEAS Transactions on Computers, Vol. 4:2, 2003, 1159-1167.

[12] Zu-Guo Yu, *Fuzzy L languages*, Xiangtan University, China, 1998.

[13] R.Sz. Madarász, S. Crvenković, *Uvod u teoriju automata i formalnih jezika*, Univerzitet u Novom Sadu, 1995.

[14] V. Petrović, *Teorija Grafova*, Univerzitet u Novom Sadu, 1998.

[15] K.J. Falconer, *The geometry of fractal sets*, Cambridge university press, 1985.

[16] M. Churchill, *Introduction to Fractal Geometry*, Keble Summer Essay, 2004.

[17] M. Nielsen, *On the decidability of Some Equivalence Problems for D0L-Systems*, Information and control 25, 1974, 166-193.

[18] K. Culik, *The decidability of the Equivalence Problem for D0L-Systems*, Information and control 35, 1977, 20-39.

[19] G. Rozenberg, *The equivalence problem for deterministic T0L-systems is undecidable*, Information processing letters, 1972, 201-204.

[20] J. Cassaigne, F. Nicolas, *Quelques proprietes des mots substitutifs*, Bulletin of the Belgian Mathematical Society 10, 2003, 661-677.

[21] J.R. Koza, *Discovery of Rewrite Rules in Lindenmayer Systems and State Transition Rules in Cellular Automata via Genetic Programming*, Symposium on Pattern Formation, 1993.

[22] M.Alfonseca, A. Ortega, A. Suarez, *Cellular automata and probabilistic L-systems: An Example in ecology*, Universidad autonoma de Madrid, 1999.

[23] A.L.A. Dalhoum, *Genetic Evolution and equivalence of some comlex systems: Fractals, Cellular Automata and Lindenmayer Systems*, Universidad Autonoma de Madrid, 2004.

Biografija

Marijana Čotrić rođena je 14.10.1991. godine u Loznici. Osnovnu školu "Kralj Aleksandar I Karađorđević" završila je 2006. godine, kao nosilac Vukove diplome i učenik generacije. Iste godine upisuje Tehničku školu u Loznici, smer elektrotehničar računara koju završava 2010. godine, takođe kao nosilac Vukove diplome i učenik generacije. Školske 2010/2011. upisuje osnovne studije na Prirodno-matematičkom fakultetu u Novom Sadu, smer Diplomirani profesor matematike. Iste završava 2014. godine i upisuje master studije na istom fakultetu, smer Master profesor matematike. Položila je sve ispite predviđene planom i programom sa prosečnom ocenom 9.4 i time stekla pravo na odbranu ovog master rada.

u Novom Sadu, oktobra 2016. *Marijana Čotrić*

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
DEPARTMAN ZA MATEMATIKU I INFORMATIKU
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:
RBR
Identifikacioni broj:
IBR
Tip dokumentacije: Monografska dokumentacija
TD
Tip zapisa: Tekstualni štampani materijal
TZ
Vrsta rada: Master rad
VR
Autor: Marijana Čotrić
AU
Mentor: dr Rozália Sz. Madarász:
MN
Naslov rada: Ćelijski automati i L-sistemi
MR
Jezik publikacije: Srpski (latinica)
JP
Jezik izvoda: srpski i engleski
JI
Zemlja publikovanja: Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2016.
GO
Izdavač: Autorski reprint
IZ
Mesto i adresa: Novi Sad, Departman za matematiku i informatiku, Prirodno-matematički fakultet u Novom Sadu, Trg Dositeja Obradovića 4
MA
Fizički opis rada: x poglavlja/ x strana / x literatura/ x slika
FO

Naučna oblast: Matematika
NO
Naučna disciplina: Primenjena algebra
ND
Ključne reči: Ćelijski automati, L-sistemi,

PO
UDK:
Čuva se: Biblioteka Departmana za matematiku i informatiku
ČU
Važna napomena:
VN
Izvod:
IZ

UNETI OPIS RADA

Datum prihvatanja teme od strane NN veća: 4.5.2016.
DP
Datum odbrane:
DO
Članovi komisije:
KO

Predsednik: dr Siniša Crvenković, redovni profesor Prirodno-matematičkog fakulteta u Novom Sadu
Član: dr Ivica Bošnjak, vanredni profesor Prirodno-matematičkog fakulteta u Novom Sadu
Član: dr Rozália Sz. Madarász, redovni profesor Prirodno-matematičkog fakulteta u Novom Sadu, mentor

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCE
DEPARTAMENT OF MATHEMATICS AND INFORMATICS
KEY WORDS DOCUMENTATION

Accession number:
ANO
Identification number:
INO
Document type: Monograph type
DT
Type of record: Printed text
TR
Contents Code: Master thesis
CC
Author: Marijana Čotrić
AU
Mentor: dr Rozália Sz. Madarász
MN
Title: Cellular automata and L-systems
XI
Language of text: Serbian (latin)
LT
Language of abstract: Serbian and English
LA
Country of publication: Serbia
CP
Locality of publication: Vojvodina
LP
Publication year: 2016.
PY
Publisher: Author's reprint
PU
Publ. place: Novi Sad, Department of mathematics and informatics, Faculty of Science, Trg Dositeja Obradovića 4
PP
Physical description: x chapters/ x pages / x references/ x photos
PD

Scientific field: Mathematics
SF
Scientific discipline: Applied algebra
SD
Key words: Cellular automata, L-systems,
SKW UC:
Holding data: Library of the Department of Mathematics and Informatics
HD
Note:
N
Abstract:
AB

UNETI OPIS RADA

Accepted by Scientific Board on: 4.5.2016.
ASB
Defended:
DE
Thesis defend board:
DB

President: dr Siniša Crvenković, Full professor, Faculty of Science, University of Novi Sad
Member: dr Ivica Bošnjak, Associate professor, Faculty of Science, University of Novi Sad
Member: dr Rozália Sz. Madarász, Full professor, Faculty of Science, University of Novi Sad, mentor