

Abstract

U radu se obradjuje specijalna klasa konacnih diskretnih modela pod nazivom ‘celijski automati’ (eng. cellular automata) koji se baziraju na cisto lokalnoj interakciji ali proizvode uzorke koji se mogu promatrati na globalnijoj skali, te vecem nivou apstrakcije. Upravo ova cisto lokalna interakcija omogucava da se pomocu navedenih entiteta omoguci modeliranje sirokog spektra realnih pojava s obzirom da veliki broj stvari koje se pokusavaju modelirati spada upravo u ovu kategoriju.

Kroz rad se navodi nekoliko klasa, primjera i primjena celijskih automata, te se uz pomoc grafickog simulatora pokusavaju prikazati najbitniji koncepti potrebni za shvatanje nacina funkcionisanja te potencijalne primjene ovih modela. Rad sluzi kao uvod u siroku tematiku i primjene ove vrste modela.

Takodjer, ponudjen je i formalni matematicki tretman iz raznih aspekta koji ukljucuju teoriju haosa i teoriju igara, te aspekt na koji je stavljen najveći fokus – teorija kompjutacije s obzirom da se celijski automati, tacnije odredjene instance istih, mogu koristiti kao univerzalna Turingova masina sto otvara siroke primjene ove vrste sistema.

Okvir rada

// pregled poglavlja i strukture etc.

Pregled oznaka i nacini enumeracije

Slike su numerisane redom i enumeracije slike nema posebno znacenje.

Definicije su numerisane sa x.y.z gdje x.y predstavlja podpoglavlje u kojem se definicija nalazi, dok z predstavlja redni broj definicije u tom podpoglavlju. Numeracija podpoglavlja ide do dva nivoa dubine.

Lista figura sa opisima

1. Uvod

U uvodnom dijelu pokusacemo razmotriti pozadinska razmatranja koja vode do izucavanja klase konacnih diskretnih modela nazvanih celijski automati.

Prvo cemo dati pregled kroz specifican primjer da se uoce neke od osnovnih karakteristika ovih sistema.

Nakon cemo pregledati historijski koja matematicka pitanja su navela da se razmotre neke specijalne klase, te ce se nako toga obratiti paznja na unificarnje specijalni klasa, te posebno historijski pregled rada Stephena Wolframa za kojeg se moze reci da je dao jedna od najvećih doprinosa samom polju.

Bice dat i pregled potencijalnih primjena te motivacija za izucavanje oblasti celijskih automata, te se ovaj dio moze smatrati neformalnim uvodom u cjelokupnu oblast koji moze dati osnovnu ideju bilo kome ko zeli da se zainteresuje u oblast.

1.1 Osnovni pregled

Pocnimo prvo od pokusaja shvatanja kakve vrste modela predstavljaju celijski automati. Zato cemo prvo krenuti od konkretnog specificnog primjera kroz koji je moguće shvatiti osnovne odlike celijskih automata na koje cemo se kasnije nadograditi kako budemo gradili formalnu apstrakciju ovog konkretnog primjera.

Najjednostavniji takav je primjer je beskonacna dvodimenzionalna ravna ploca prekrivena kvadratima koje cemo nazvati *celije*. Kvadrati se medjusobno dodiruju stranicama, te tako svaki kvadrat ima punu vezu preko stranice sa tacno cetiri susjedne celije. Za svaku celiju kazemo da moze biti u dva stanja - on i off. Kako cemo za prikaz automata pretežno koristiti graficke interpretacije, ova dva stanja mozemo “zakodirati” bojom same celije, pa cemo tako uspostaviti konvenciju da crna prestavlja on, dok bijela prestavlja off stanje. Sva trenutna razmatranja bice formalno definisana kasnije u radu te ovaj dio razmatranja služi samo za intuitivni prikaz osnovnih ideja iza ovakvih vrsta modela.

// slika 1 – celija sa susjedstvom (4x4 grid)

Na slici 1 imamo prikazanu dosad opisanu celiju sa svojim susjednim celijama, te svaka od njih ima svoje definisano stanje. Celije su oznacene brojevima 1-16 radi njihovog referiranja unutar teksta, te ovi brojevi ne predstavljaju dio modela.

// slika 2 – 2D grid sa pocetnom konfiguracijom

Na slici 2 prikazan je skupe celija koje zajedno cine 2d *grid*. Primijetimo da je nemoguće simulirati beskonacni grid konacnim kompjutacionim metodama, pa se u praksi gotovo uvijek ogranicavamo na konacne dimenzije. Ovdje nastaje problem sta raditi sa rubnim

celijama, te će ta tematika biti detaljnije kasnije obradjena.

Ovakav raspored nazvamo *konfiguracija*. U konfiguraciji, svaka ćelija ima svoje početno stanje, pa je tako za svaku ćeliju definisano da li je ona inicijalno on ili off – crna ili bijela. Stanje je na početku izabrano proizvoljno i može se mijenjati u zavisnosti od potreba, te će različita početna stanja dati nekada i drastično različita ponašanja. Na slici 2 prikazan je primjer jednog takvog početnog stanja. Skup ovako organizovanih ćelija sa svojim početnim stanjima u konačnom gridu nazivamo *inicijalna konfiguracija grida*.

Naravno, dosadasnja definicija ćelijski automata ne bi imala nikakvog smisla, s obzirom da imamo samo početnu konfiguraciju i grid. Međutim, ono što čini ćelijske automate pravim modelima koji se mogu koristiti u razne svrhe je takozvana *evolucija ćelijskih automata*. Nakon što se uspostavi inicijalna konfiguracija grida, ovaj sistem može da se stavi u evoluciju. To znači da će svaka od ćelija da mijenja svoje stanje prema nekim pravilima, te će cijelokupan sistem da se mijenja prema tim pravilima u diskretnim vremenskim intervalima.

// slika 3 – tipovi susjedstva (moore, von Neumann, custom)

Sljedeće stanje svake individualne ćelije zavisi od njenog trenutnog stanja, kao i okolnih ćelija, te se prema ovim parametrima i formiraju pravila. Susjedne koje okružuju datu ćeliju a uzimaju se u obzir prilikom rancunanja sljedećeg stanja kolektivno se nazivaju *susjedstvo* (eng. *neighborhood*). Evidentno je da izbor susjedstva nije jedinstven. Na slici 3 prikazano je nekoliko načina izbora susjedstva. Najpoznatija dva ovakva tipa su Moore-ovo i von Neumann-ovo susjedstvo prikazano na slici, ali nije isključeno i kreiranje proizvoljnog susjedstva.

// slika 4 – ruleset picker za jednostavno 4-von neumannovo susjedstvo

Nakon što se izabere koje ćelije učestvuju u formiranju susjedstva, formira se i skup pravila koji govori o tome kako ćelija evoluirala na osnovu svog stanja i stanja svojih susjeda. Na slici 4 je u gornjem dijelu naveden i način specifikacije pravila za von Neumannovo-ovo susjedstvo gdje se za svaku pojedinačnu kombinaciju susjedstva na osnovu trenutnog stanja ćelija bira iduće stanje iste. Tako sa slike možemo uočiti npr. ukoliko je ćelija okružena sa tri bijele ćelije iznad, lijevo i desno, te crnom ćelijom ispod, ukoliko je trenutno stanje off prelazi u on, dok ukoliko je trenutno stanje on prelazi u off.

// slika 5 - tri koraka game of life-a

Kako su stanja binarna, ukoliko n predstavlja broj susjeda koji formiraju susjedstvo, tada je moguće 2^{n+1} mogućih pravila evolucije. Tako za jednostavno Moore-ovo susjedstvo od 8 susjednih ćelija postoji $2^{8+1} = 2^9 = 1.34 \times 10^{154}$ mogućih pravila evolucije. Ovo opažanje će biti kasnije detaljnije objašnjeno.

Na slici 5 data su 3 koraka evolucije sistema sa jednim od specifičnih pravila, te je ovo dobro poznata konfiguracija nazvana Game of Life.

Vidimo da opisana cisto lokalna interakcija moze da kreira poprilično kompleksne oblike, te ce se dalje ispitati kolika je ta kompleksnost i na koji nacin se mogu koristiti ova zapazanja za neke generalnije kompjutacije.

// slike 6 i 7 – selekcija 1d pravila – kompleksne strukture [3]

Na slikama 6 i 7 prikazana je kompleksnost koja moze da proizidje iz jednostavnih pravila prezentovanih iznad.

// TODO: dodati iz uvod iz [3] radi upoznavanja zanimljivih osobina

1.2 Historija

Da bismo dobili opstu sliku oblasti, te razloga nastanka i razvoja iste, potrebno je istraziti kako je doslo do pitanja i problema koja su dovela do toga da se naucna javnost zainteresira.

Kao i sve ostale oblasti nauke i matematike, tako su i celijski automati nastali pokusajem odgovaranja na specifican skup pitanja koji se kasnije prosirio u cijelu oblast nakon sto se uvidjela njihova moguca generalnost i opstija primjena.

Pioniri ove konkretne oblasti bili 1940-ih su istrazivaci u Los Alamos Nacionalnoj Laboratoriji *Stanislaw Ulam* i *John von Neumann*. Zanimljiva je cinjenica da su se bavili ovom oblasti sa strane u vidu hobija. Oni su se, posebno von Neumann, inicijalno zanimali pokusajem da naprave sistem u kojem bi bilo moguće da entitet unutar samog sistema vrsi *samoreplikaciju*. Ovo bi znacilo da odredjeni objekat pravi identicnu kopiju samoga sebe, te bi ta kopija takodjer pravila svoju kopiju ad infimum.

Prvobitno je von Neumann zelio da kreira robota koji bi vrsio samoreplikaciju (sto danas nazivamo kinematskim – realnim modelom samoreplikacije), medjutim nakon teskoca pri nalazenju ogromnog broja dijelova koje bi robot morao da ima na raspolaganju da bi izvrsio ovaj zadatak, te nakon prijedloga Ulama, odlucio je da iskoristi apstraktni diskretni model za ovaj zadatak. Model koji je koristio moze se smatrati prvim primjerom koristenja celijskih automata.

Ovo predstavlja pocetak oblasti diskretnih sistema celijskih automata. Rezultat je bio ono sto danas nazivamo von Neumann-ov univerzalni konstruktor. On se sastoji od celijskih automata koji imaju 29 stanja i potrebno je ~200.000 celija da bi se izgradio univerzalni konstruktor. Von Neumann je dao okvirni dizajn i dokaz postojanja, ali nikada nije implementirao ovaj sistem. [1] Tek 1990-ih godina je grupa predvodjena italijanskim naucnikom *Pasaventom* uspjela da napravi pravu implementaciju ovoga sistema, iako je ideja konceptualno zaceta gotovo 50 godina prije prve implementacije.

Nesto kasnije, 1950-ih, ista dvojica naucnika iz Los Alamos laboratorija iskoristili su celijske automate u prvom pokusaju modeliranja realnosti koristeći iste. Kreirali su model

koji *predviđa kretanje fluida* na način da smatraju fluid sastavljenim od diskretnih jedinica – ćelijskih automata, čije kretanje zavisi od susjednih jedinica. Na ovaj način moguće je aproksimirati kretanje cjelokupnog fluida modeliranjem samo lokalne interakcije susjednih čestica. Ovim modelom pokazano je da ćelijski automati imaju i širu primjenu van čisto teoretskih razmatranja za koja su ranije korišteni, te ovo predstavlja svojevrsni početak generalne primjene ove vrste modela u nauci.

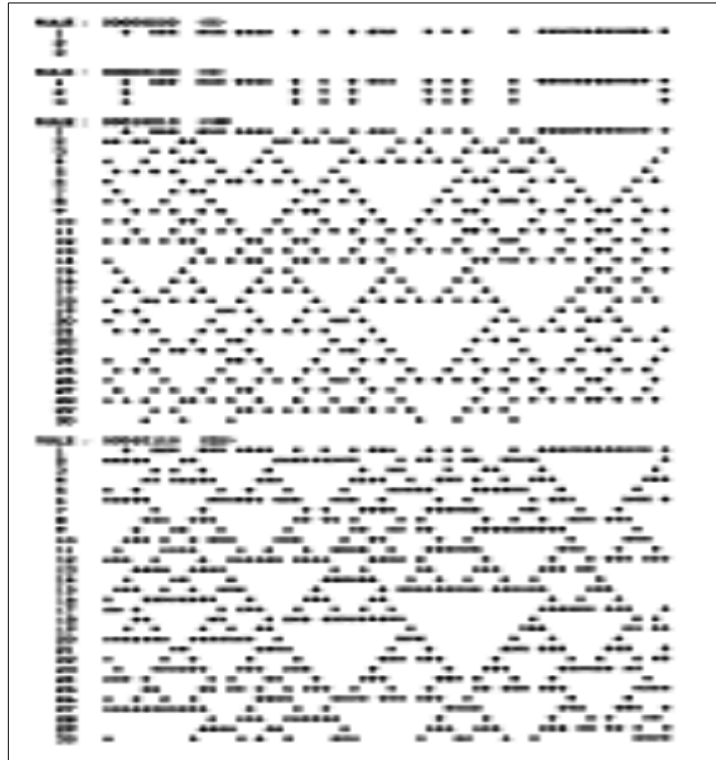
Trebalo bi napomenuti i da je neke od ranih istraživanja u ovom polju vrsio i pionir u oblasti vjestačkog života, norvesko-italijanski naučnik *Nils Aall Baricelli* koji je jedan od prvih prepoznao potencijalnu univerzalnost ćelijskih automata kao modela koji mogu predstavljati realne pojave.

Jos neku od ranih primjena ćelijski automati nasli su u modeliranju *propagacije talasa u medijima*. Rani ovakav model ćelijskih automata konstruisan je 1940-ih. Međutim kako je taj model koristio kontinualnu funkciju kao signal, ne može se smatrati diskretnim modelom ćelijskih automata, tako da su prvi pravi ovakav model korišteni u svrhe modeliranja impulsa kardio sistema u ćovjećijem tijelu konstruisali *J. M. Greenberg* i *S. P. Hastings* 1978-e. Ovaj model je i dalje često korišten i referenciran u istraživačkim radovima.

Prvih dvadeset godina od pojavljivanja modela ćelijskih automata, gotovo niko nije izvršavao rigorozno naučno i matematičko ispitivanje osobina ovih sistema. Jedan od pionirskih radova u polju bio je rad američkog matematičara *Gustava A. Hedlunda*, koji je kroz matematičku oblast dinamika simbola (koju je sam i osnovao, eng. *symbolic dynamics*) posmatrao ćelijske automate kao mijenjajuće nizove simbola uz određena pravila prelaza. Ovime je došao do nekih od najkorisnijih rezultata u ovom polju. Njegov rad iz 1969-e zajedno sa Curtis-Hedlund-Lyndon teoremom za koju snosi djelomične zasluge, koja klasificira globalni prostor pravila automata, i dalje predstavlja jednu od osnova za bilo koga ko planira da se upusti u ozbiljnije ispitivanje ove vrste sistema.

Pravu popularizaciju oblast je dozijvela pametno konstruisanim primjerom od strane britanskog matematičara i fizičara *John Conway*-a 1970-e godine. On je svoj specijčan model baziran na dvodimenzionalnim ćelijskim automatima prikladno nazvao *Igra života* (eng. *Game of Life*). Ovaj model je standardni dvodimenzionalni sistem ćelijskih automata sa dva stanja, međutim uprkos jednostavnim pravilima, nakon što se sistem pusti u rad, počinju da se pojavljuju visoko kompleksne strukture koje ispoljavaju osobine koje bi mogli pripisati i nekim živim bićima, kreću se, jedu jedni druge, razmnozavaju se i slično – zbog ovoga je model i dobio svoje ime. Upravo zbog ovih osobina gdje se visoka kompleksnost javlja iz poprilično jednostavnih pravila, Igra a samim tim i oblast ćelijskih automata dozijvela je veliku popularizaciju, te većina ljudi za ćelijske automate sazna upravo iz ovog primjera. Iako se ovaj model smatrao pretežno dijelom rekreativne matematike te sredstvom popularizacije ideje ćelijskih automata za širu javnost, nešto kasnije je Berlekamp u saradnji sa još nekoliko matematičara dokazao univerzalnost Igre života, tj. da sistem može ekvivalentno da se koristi u svrhu univerzalne kompjutacije kao i bilo koji drugi računar prema Turingovoj tezi.

Mozemo napomenuti da je i njemacki racunarski pionir *Konrad Zuse* 1969-e u svojoj knjizi *Racunajuci svemir* raspravljao neke od sirih filozofskih implikacija sistema celijskih automata. On je naveo da je moguće da cijelokupan univerzum zapravo jedan veliki celijski automat koji se sinhrono update-a u vremenskim koracima, te je ova ideja otvorila prostor za potpuno novu oblast nazvanu digitalna fizika.



Slika 8. Preuzeto iz [2]

Najdetaljnije i rigoroznije ispitivanje osobina celijskih automata izvorsio je u svojim radovima tokom dvadeset godina istrazivanja britanski matematicar i fizicar *Stephen Wolfram*, koji se smatra jednom od najbitnijih figura za ovu oblast. Poceo je sa svojim istrazivanjima 1981-e u pokusajima da razmotri kako se kompleksni uzorci u prirodi formiraju naizgled narusavajuci drugi zakon termodinamike. Tada je izvrsavao simulacije na ranim racunarima, te nekon sto je u simulacije unio odredjene klase celijskih automata, bio je zapanjen kolika kompleksnost proizilazi iz jednostavnih pravila koje je postavio(slika 8). Ovo naizgled kontraintuitivno ponasanje koje ga je zapanjilo navelo ga je da u iducim decenijama prebaci svoju sferu rada sa fizike na matematiku i kompjutersku nauku. U seriji od preko dvadeset radova Wolfram je izvrsio klasifikaciju i opisao osobine pretežno jednodimenzionalnih celijskih automata, te predlozio mnoge njihove primjene kao alternativu trenutno koristenim modelima poput parcijalnih diferencijalnih jednacina. Takodjer je doprinio dokazivanju univerzalnosti jednog od pravila jednodimenzionalnih celijskih automata. Svoje pronalaskе, stavove i historiju istrazivanja kompajlirao je 2002. u knjizi *Nova vrsta nauke* (eng. *A New Kind of Science*),

gdje se zalaze za celijske automate kao buducnost modeliranja prirodnih pojava te bilo kakve vrste apstraktnih sistema [2]. Dotice se i filozofskih implikacija celijskih automata. Wofram i dalje nastavlja da popularizira ovu temu kroz serije govora, te je poznat kao i kreator *Wolfram Alpha* i *Mathematica* softverskih sistema.

1.3 Motivacija

Nauka stoljecima pokusava da koristeći klasicne metode matematike u priminjenim disciplinama poput fizike objasni i razjasni svijet oko nas, kao i da iz datih pocetnih uslova nekog sistema da predvidjanja za buducnost istog, tj. da predvidi ponasanje bez potrebe da se sam sistem pusti u izvorsavanje ili simulaciju. Medjutim, i nakon toliko vremena izucavanja, i dalje postoje neka fundamentalna pitanja koja su ostala neodgovorena i cini se kao da ih je nauka zaobilazila te odgovarala samo na pitanja koja su se uklapala u stereotipni nacin dotadasnjeg razmisljanja da se stvari modeliraju kontinualnim alatima parcijalnih diferencijalnih jednacina. Treba uzeti u razmatranje mogucnost da mozda postoji fundamentalno ogranicenje ovakvog pristupa te da treba razmotriti neke nove metode koje bi mozda dale bolje rezultate [2]. Wolfram, fizicar po struci i jedna od ikonskih figura polja celijskih automata, u svojim radovima i knjigama daje primjere novih sistema i njihovih primjena na mjestima gdje tradicionalni naucni pristup ne uspijeva dati zadovoljavajuci odgovor (pogledati npr. [4] i [2]).

Ukoliko osmotrimo prirodne konstrukte oko nas, mozemo primijetiti da postoji nekoliko osnovnih karakteristika koje mozemo uociti. Za pocetak, koncept lokalne interakcije siroko je rasprostranjen s obzirom da u toku razmatranja necega uzimamo u obzir uticaje samo onih elemenata koji su vremenski i prostorno dovoljno blizu da bi mogli proizvesti znacajne efekte na ishod ponasanja. Naravno da je moguće da neki dalji objekat ima uticaj. Medjutim, kako vecina posmatranih pojava zadovoljava svojstvo linearnosti gdje mala promjena u pocetnim uslovima izaziva i malu promjenu u rezultatima (za razliku od kaoticnih sistema koji se rjedje srecu), to je posmatranje svakodnevnih sistema kroz pretežno lokalnu interakciju poprilično opravdano. Na primjer, prilikom razmatranja hemijske reakcije na molekularnom nivou, nije potrebno uzimati u obzir Jupiterovu gravitacionu silu, iako je istina da ona vrsi uticaj na sistem, ma koliko on malen bio (iz ovih razloga potpuno izoliran sistem je cisto teoretski konstrukt).

Nadalje, uprkos pretežno cisto lokalnoj interakciji, sistemi ispoljavaju poprilično kompleksna svojstva kao cjeline. Tako da iz cisto lokalnih svojstava nastaju globalna svojstva u kojima ucestvuju svi dijelovi. Ovo mozemo najbolje shvatiti kroz primjer ljudskog ponasanja kao vrste, gdje ljudi vrse razmjenu informacija i interakciju jedni medju drugima lokalno, medjutim i ljudsko drustvo u cjelini mozemo okarakterizirati nekim globalnim osobinama viseg nivoa. Ili npr. sistemi idealizovanih gasova sastavljeni od cestica koje vrse medjusobno mali prostorni uticaj, ali alatima statisticke mehanike i sl. mozemo okarakterizirati osobine cijelog volumena gasa [5].

Enormna kompleksnost nedvojbeno je dio sistema koji se uzimaju u naucna razmatranja i vecine naseg okruzenja. Ovo ide toliko daleko da su neke kompleksnosti i dalje ostale

nerazjasnjene i uz korištenje najmodernijih tehnika i metoda naucnih disciplina. Svakodnevno se susrecu kompleksne strukture koje je danas tesko precizno modelirati, npr. propagacija toplote, modeli kretanja fluida ili pak nesto sto se na prvi pogled cini kao jednostavna stvar poput formiranje oblika snjeznih pahulja ili oblici formirani na morskim školjkama [2]. Jedan od osnovnih ciljeva naucnih istrazivanja upravo "pripitomljavanje" ovih kompleksnosti i pokusaja objasnjenja istih kroz skup jednostavnijih principa. Postavlja se pitanje da li kompleksnost obavezno zahtijeva i kompleksnost na nizim nivoima i strukturama sistema, ili je nekako moguće, iako kontraintuitivno, da poprilično jednostavna pravila mogu da proizvedu ogromnu složenost koja se može uociti.

Postavlja se pitanje kako spojiti ove naizgled kontradiktorne osobine koje uvidjamo. Kako to da s jednog strana spektra postoji cisto lokalna interakcija, ali nakon sto pogledamo sistem sa viseg nivoa apstrakcije uvidjamo da se i sam sistem ponasa kao cjelina koja vrsi nesto globalniju interakciju sa sistemima istog nivoa? Takodjer, kako je moguće da ogromnu kompleksnost koju posmatramo pokusavamo svesti na jednostavna pravila koja ce je opisati i iz kojih ova kompleksnost može da se izrodi?

Necemo ici toliko daleko da kazemo, kao sto Wolfram tvrdi u svojoj knjizi, da celijski automati mogu da posluze da modeliraju cijeli univerzum [2], medjutim ocigledno je da celijski automati i njihovo razumijevanje može dovesti bar na pravi put razjasnjenja nekih od navedenih pitanja. Celijski automati po svojoj definiciji se baziraju na cisto lokalnoj interakciji, medjutim kada se pusti nekoliko koraka simulacije određenih pravila cak i jednodimenzionalnih instanci, uocavaju se globalne osobine. Tako da bi iz tog razumijevanja mogli doci blize razumijevanju nastajanja globalnog ponasanja iz lokalno ogranicenog. S druge strane, pravila celijskih automata poprilično su jednostavna, pa su cak i anticki narodi mogli doci do istih [2]. Medjutim iako su pravila jednostavna, nivo kompleksnosti koji se javlja uopste nije niskog nivoa. Tako da i s te strane, celijski automati i njihovo razumijevanje može da nas pribilizi odgovoru poveznice izmedju nastanka kompleksnosti iz malobrojnog skupa jednostavnih pravila.

1.4 Primjeri

Da bi se formirala kompletnija opsta slika celijskih automata kao modela kompleksnosti, prikladno bi bilo dati nekoliko primjera kompleksnosti i vrsta struktura generisanih celijskim automatima. Ovo može dati jasniju sliku i nekome ko nije toliko zainteresovan u mateimaticke detalje samog modela na nacin da može prikazati raznovrstnost struktura koje celijski automati mogu proizvesti.

Rule 30 ~ sea shells

rule 110

game of life + game of life life forms

snowflake formation

2. Formalni matematički tretman

Nakon neformalnog uvoda, potrebno je i rigorozno matematički definisati šta se misli kada se koristi pojam ćelijskih automata. Zbog raznovrstnosti modela koji se mogu smatrati ćelijskim automatima, u literaturi ne postoji opsteprihvacena generalna definicija, već autori strogo matematički definišu samo konkretan model kojim se bave, npr. jednodimenzionalni binarni ćelijski automati u slučaju Stephena Wolframa (vidjeti npr. [2], [3], [4]).

Na početku biće data generalna definicija koja pokušava što opstije da pokrije sve diskretne modele koji se mogu smatrati ćelijskim automatima, mada su moguće neke iznimke zbog širine diskretnih modela i mogućnosti proširenja osnovnog modela po raznim parametrima.

Nakon toga ce se redom proci kroz specificne instance poput jednodimenzionalnih i dvodimenzionalnih binarnih automata koje su najvise izucavane tokom godina. Obratice se paznja i na njihovu formalnu definiciju, tretman i matematicke osobine.

Takodjer, definicije i teoreme – osobine automata bice popracene primjerima i grafickim simulacijama iz implementacijskog dijela rada koji ce sam po sebi biti kasnije pokriven.

2.0 Preliminarne matematicke definicije

U ovoj sekciji potrebno je definisati nekoliko preliminarnih matematickih pojmova koji ce kasnije biti iskoristeni u izucavanju osobina celijskih automata. Citatelj se trenutno ne treba zamarati ovim dijelom, vec se po potrebi vratiti na njega ukoliko bude koristen neki od pojmova ovdje definisan.

U informacionoj teoriji, potrebno je na neki nacin kvantificirati kolicinu informacija u nekom sistemu, te za to koristimo koncept entropije.

Definicija 2.0.1 Za sistem sa n mogucih dogadjaja sa vjerovatnocama desavanja i -tog dogadjaja p_i , entropija se definise kao $= - \sum_i^n p_i \log(\frac{1}{p_i})$.

Entropija ce nam biti korisna u razmatranjima celijskih automata i njihovih statistickih osobina sa aspekta teorija informacija i kodiranja.

Kako su celijski automati zapravo specijalna klasa automata iz teorije izracunljivosti, to je veoma korisno gledati njihove osobine i kroz ovu naucnu oblast, pa ce nadalje biti navedene neke od osnovnih definicija potrebne za daljnja razmatranja.

Definicija 2.0.2 Turingova masina definise se kao uredjena 7-orka $M = (Q, \Gamma, b, \Sigma, \delta, q_0, F)$, gdje su oznake redom:

- Q je skup stanja
- Γ je skup simbola
- b predstavlja specijalni 'blank' simbol
- $\Sigma \subseteq \Gamma \setminus \{b\}$ skup ulaznih simbola
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ funkcija prelaza stanja
- q_0 pocetno stanje
- F skup prihvacenih stana

Turingova masina predstavlja nacin formalizacije svih funkcija koje nazivamo izracunljivim. Intuitivno, izracunljiva je funkcija koju covjek moze da obavi sa papirom i

olovkom bez pazenja na vremensko prostorna ogranicenja. Ovo je koristan koncept pri formalnim razmatranjima izracunljivosti i sposobnostima izracunavanja nekog sistema.

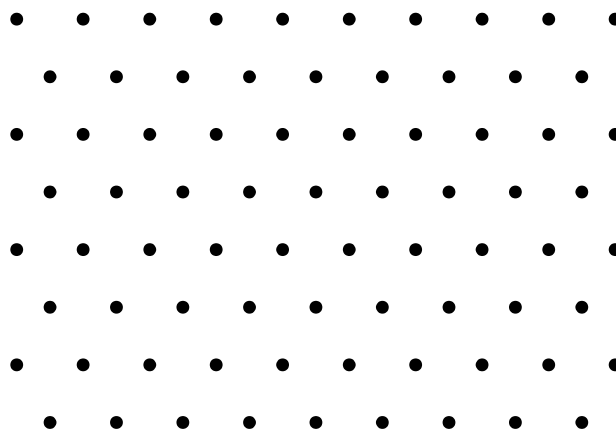
2.1 Generalna definicija celijskih automata

Pokusajmo prvo kroz nekoliko zapazanja iz konkretnih primjera uociti koje su to generalne osobine i koncepti koje bi definicija morala da pokrije, a nakon toga pokusacemo to da uklopimo u neki matematicki model.

Prvo idu neke preliminarne matematicke definicije kroz primjere koje su neophodne za razumijevanje krajnje definicije celijskih automata u generalnom slucaju. Pokusacemo kroz te definicije postepno proci kroz osobine apstraktnih pojmova za koje vezemo instance celijskih automata.

Sva desavanja modela moraju da se odvijaju na nekom apstraktnom prostoru, koji treba da zadovoljava odredjena svojstva. Dakle prvi koncept koji definicija treba da pokrije je prostor odvijanja modela. Prostor odvijanja modela celijskih automata je u pravilu diskretan, sto mozemo da uocimo iz nekoliko dosad razmotrenih primjera s obzirom da se uvijek radilo u pravougaonoj matrici kvadratnih mjesta ili jednodimenzionalnom ekvivalentu istih. Takodjer prostor mora da bude definisan na takav nacin da se na njemu mogu definisati i ostali bitni koncepti poput susjedstva.

Prvo onda definisimo prostor. Matematicki objekat koji moze da uhvati sve navede osobine je latica ili resetka (eng. lattice). Intuitivno resetka je skup tacaka sa ravnomjernim i homogenim rasporedom kao sto je prikazano npr. na slici 9. Naravno tu je i dosad poznati primjer na slici 1 u prethodnom poglavlju.



Slika 9.

Da bismo formalno predstavili sta je to ravnomjerno rasporedjenje, resetku mozemo

definisati kao skup tacaka kod kojih je udaljenost izmedju bilo koje dvije od njih cjelobrojna linearna kombinacija unaprijed odredenih vektora u prostoru gdje su tacke definisane (pretežno se radi sa skupom/prostorom \mathbb{R}^n). Kljucan detalj je cjelobrojnost jer time zadržavamo osobinu diskretnosti samog skupa, sto je kljucno za celijske automate kao diskretne modele kompjutacije.

Definicija 2.1.1 Latica ili resetka se definise kao $\Lambda := \left\{ \sum_i a_i v_i : a_i \in \mathbb{Z} \right\}$

gdje je $v_i \in \{v_1, v_2, v_3, \dots\}$ baza vektorskog prostora V nad skupom skupa gdje je resetka definisana.

Ovo samo znaci da izaberemo fiksni skup vektora udaljenosti od tacke i na tacku nadodjemo cjelobrojne umnoske vektora da bismo tako održali regularnost koja je zahtijevana za definiciju celijskih automata.

Celijski automati su kako prostorno tako i vremenski diskretni sistemi. U primjeru jednodimenzionalnih automata, vrijeme je bilo samo prirodan broj koji je govorio o kojoj se iteraciji automata radi, pa tako moramo moći definisati i diskretan koncept vremena u kojem automati evoluiraju kao dodatnu dimenziju pored prostorne koju smo već obradili.

Upravo iz diskretnosti dimenzije vremena za sisteme celijskih automata, javlja se i najjednostavniji način njegove definicije. Dovoljno je samo da se vrijeme može staviti u jedan-na-jedan (bijektivno) mapiranje sa skupom prirodnih brojeva, što se i uklapa u traženu diskretnost.

Definicija 2.1.2 Neka je dat skup T takav da neka postoji funkcija $b : T \rightarrow \mathbb{N}$ koja je bijektivna. Tada skup T nazivamo diskretnim vremenskim skupom.

Da bi se izbjegle zabune, zbog postojanja bijektivnosti moguće je u daljnjim razmatranjima umjesto apstraktnog skupa T koristiti skup prirodnih brojeva.

Nakon što smo definisali prostorne i vremenske dimenzije u kojima će celijski automati biti smješteni, još jedan osnovni koncept je stanje svakog automata koji se nalazi u prostor-vremenu.

Za jednodimenzionalni i dvodimenzionalni slučaj imali smo samo dva moguća stanja – 1 ili 0, crno ili bijelo u grafickoj reprezentaciji. Na osnovu ovoga, svaka ćelija mora biti u jednom od konačnog broja stanja, što je i jedino ograničenje na ovaj skup, da mu je kardinalnost konačna, tj. odgovara nekom prirodnom broju. Kasnije ćemo vidjeti da su moguće i neke generalizacije samog skupa stanja gdje npr. imamo celijske automate čija stanja odgovaraju realnim brojevima u intervalu $[0, 1]$ što očigledno nije konačan skup stanja.

Definicija 2.1.3 Skup Q nazivamo skupom stanja ukoliko $|Q| = n$, gdje $n \in \mathbb{N}$.

Potrebno je još razmotriti šta bi bilo povoljno da se koristi u definiciji susjedstva te

prelaza (evolucije) svake celije unutar automata. Ovi koncepti nista prostorne i vremenske transformacije konkretne celije. Npr. susjedstvo je samo mapiranje/dodjeljivanje nekoliko celija jednoj konkretnoj celiji u resetki. Evolucija je samo dodjeljivanje iduceg stanja celiji na osnovu prije dodijeljenog susjedstva i prijasnjeg stanja. Tako da matematicke funkcije ili mapiranja savrseno odgovaraju opisu objekta koji bi mogao da se koristi, sto ce i biti slucaj.

Sada je potrebno dati definiciju celijskih automata cija se evolucija odvija na diskretnom prostoru i u diskretnom vremenu sa konacnim brojem stanja uz postojanje susjedstva svake celije i tranzicijskog pravila za istu. Vidimo da smo pokrili sve koncepte koje smo ranije vidjeli u specifcnim instancama, tako da je moguće dati generalnu definiciju.

Definicija 2.1.4 Celijski automat mozemo definisati kao uredjenu sestorku $C := (\Lambda, Q, T, \sigma, \eta, \phi)$, pri cemu je:

- Λ resetka nad nekim skupom
- Q skup stanja
- T diskretni vremenski skup
- $\sigma : \Lambda \times T \rightarrow Q$
- $\eta : \Lambda \rightarrow \Lambda^c$
- $\phi : Q^{c+1} \rightarrow Q$ ($c \in \mathbb{N} \wedge c \geq 1$)

Funkcije σ , η i ϕ nazivaju se funkcije konfiguracije, susjedstva i prelaza re-spektivno. Prirodni broj c se naziva velicina ili kardinalnost susjedstva. Funkcija σ je rekurzivno definisana pomocu funkcije η kao $\sigma(\lambda, n) = \phi(\sigma(\eta(\lambda), n-1), \sigma(\lambda, n-1))$ gdje $\lambda \in \Lambda, n \in T$, ali mozemo uz prije navedenu diskretnost skupa T smatrati da $n \in \mathbb{N}$.

Potrebno je ukratko pojasniti detaljniju intuiciju iza funkcija σ, η, ϕ u Definiciji 3.4.

Funkcija σ predstavlja funkciju trenutne konfiguracije, sto znaci da pridruzuje stanje iz skupa stanja svakoj celiji u datoj vremenskoj instanci.

Funkcija η predstavlja funkciju susjedstva, te ona jednostavno pridruzuje svakoj celiji onoliko celija koliko je definisano velicinom susjedstva.

Na kraju, funkcija ϕ predstavlja funkciju prelaza celija celijskog automata, te ona definise u koje sljedece stanje prelazi celija celijskog automata uzimajuci u obzir stanja njenog susjedstva, te njeno trenutno stanje. Definisana je rekurzivno, jer svaka iteracija zavisi samo od prethodne, a veoma je moguće da je pojedina ili cak vecinu pravila prelaza nemoguće predstaviti elementarnim funkcijama na koje smo navikli. Ovo se vec dalo

zaključiti iz kompleksnosti uzoraka koje ova vrsta sistema generise.

Može se primijetiti da ćelijski automati posjeduju nekoliko osobina koje su zadovoljene za svaku instancu istih. To su homogenost, paralelizam i lokanost. Ovo je s obzirom da sve ćelije evoluiraju paralelno po istim pravilima koja su određena samo lokalnim susjedstvom i stanjem ostalih ćelija u njemu [1].

Ovim je završeno razmatranje generalne definicije ćelijskih automata, te će nadalje biti izučavane najinteresantnije i najistraženije instance istih koje ćemo pojedinačno pokušati uklopiti u ovaj model, te detaljno razmotriti specifične matematičke osobine svake od tih instanci.

2.2 Binarni jednodimenzionalni (elementarni) ćelijski automati

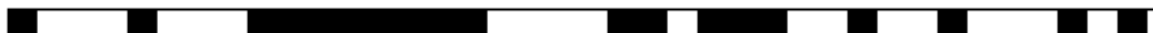
Najosnovniji tip sistema ćelijskih automata predstavljaju binarni jednodimenzionalni ćelijski automati. Ovo znači da svaka individualna ćelija posjeduje dva moguća stanja – binarni, te da svaka ćelija ima tačno dva susjeda, te da su ćelije međusobno poredane u jednodimenzionalnoj resetki – jednodimenzionalni.

Međutim, uprkos njihovoj jednostavnosti, upravo ova vrsta predstavlja najizučavaniji tip ćelijskih automata. Ovo možda možemo pripisati činjenici da se cjelokupna nauka vezana za ove vrste sistema i bazira na sto jednostavnijim gradivnim elementima koji proizvode kompleksne strukture, pa su tako najjednostavniji od njih vjerovatno i najzanimljiviji za izučiti. Ovo nije slučaj za većinu drugih oblasti sa drugim skroz drugim pogledom na kompleksnost.

Drugi naziv koji se koristi u literaturi za ovu vrstu ćelijskih automata je elementarni ćelijski automati, tako da će ova dva termina u daljem tekstu biti korištena naizmjenično.

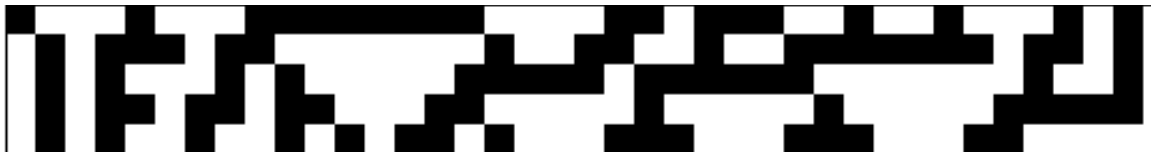
2.2.1 Definicija i način enumeracije pravila

Pokusajmo sada dati definiciju i prikazati kakvu vrstu sistema predstavljaju binarni jednodimenzionalni ćelijski automati. Najjednostavnije je shvatiti ove sisteme kao niz ćelija poredanih jedna do druge u liniji – zbog čega se i zovu jednodimenzionalni, prilikom čega svaka ćelija može biti u jednom od dva stanja – on ili off, što grafički predstavljamo crno i bijelom bojom respektivno. Generalno, grafički prikaz je često korišten alat u izučavanju ćelijskih automata jer može da da neke indikacije o osobinama izučavanih sistema.



Slika 10. jednodimenzionalni (generisano softverom)

Na slici 10 dat je prikaz jednog ovakvog niza celija u liniji gdje svaka celija ima slucajno dodijeljenu vrijednost. Ovakav skup celija sa odgovarajucim stanjima naziva se konfiguracija. Svaka celija evoluira, tj. mijenja svoje stanje u iducem koraku prema određenom pravilu koje zavisi od stanja te konkretne celije, kao i stanja nekih okolnih celija. Izmjena stanja svih celija vrsi se paralelno, tako da cjelokupan niz – sistem evoluira paralelno. Kasnije cemo vidjeti generalan nacin klasifikacije ovih pravila evolucije i njihovog definisanja. Upravo je i ovaj paralelizam cesto koristen argument u zagovaranju prakticnih primjena celijskih automata u sistemima paralelnih procesiranja (pogledati npr. [5]).



Slika 11. 5 koraka evolucije (pravilo 30) (generisano softverom)

Na slici 11 nadalje prikazo je 5 koraka evolucije sistema prema unaprije izabranom pravilu, gdje svaki red predstavlja sljedecu konfiguraciju sistema. Prvi red nazivamo pocetna konfiguracija, te svaki sljedeci red nastaje opisanim postupkom paralelne evolucije celija zasebno. Evolucija se moze nastaviti u nedogled, dok je ovdje postupak izvršen tek pet puta. Primijecujemo da je evolucija sistema izvršena u diskretnim vremenskim koracima, sto je jos jedna od navedenih kljucnih osobina celijskih automata.

Sada je potrebno doci do formalne definicije jednodimenzionalnih binarnih celijskih automata koja se javlja kao posebna instanca prije obradjene generalne definicije. Takodjer, potrebno je i naci nacin da se struktuirano navedu pravila izbora nacina evolucije ovih sistema.

Definicija 2.2.1 Jednodimenzionalni binarni celijski automat mozemo definisati kao uredjenu trojku (C, σ, ϕ) , gdje C predstavlja skup celija koje su prebrojive, tj. moguće je izvršiti enumeraciju istih. Radi jednostavnosti, moguće je umjesto C koristiti skup \mathbb{Z} . Funkcija σ predstavlja mapiranje $\sigma : \mathbb{N} \times \mathbb{Z} \rightarrow \{0, 1\}$ i naziva se pravilo evolucije. Funkcija ϕ predstavlja mapiranje $\phi : \{0, 1\}^n \rightarrow \{0, 1\}$, gdje se n naziva velicina susjedstva. Mapiranje σ definisano je rekursivno kao: $\sigma(t+1, i) = \phi(\sigma(S_{j \in J(i)} \sigma(t, j)))$, gdje $J(i)$ predstavlja susjedstvo konkretne celije izabrano prema nekom pravilu, te $\phi(0, i) \in \{0, 1\}$ nazivamo pocetnom konfiguracijom sistema.

Ovu potpuno formalnu definiciju intuitivno mozemo shvatiti na nacin da se za svaku posebnu celiju, koja je oznacena cijelim brojem koji je jedinstveno identifikuje s obzirom na raspored celija, prema određenom pravilu prvo bira susjedstvo. Susjedstvo predstavlja skup celija od cijih stanja zavisi i iduce stanje izabrane celije. Susjedstvo se bira na identican nacin za svaku celiju. Nakon toga koristeci pravilo ϕ na osnovu stanja celije i stanja njenih susjeda dodjeljujemo joj stanje u iducoj diskretnoj vremenskoj instanci.

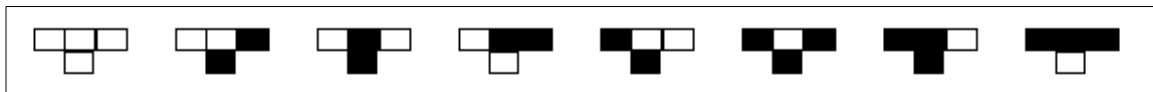
Stanja celija kako je vec navedeno pripadaju skupu od dva moguca stanje, $\{0, 1\}$.

Sada je potrebno razmotriti na koji nacin birati susjedstva, te na osnovu toga definisati moguca pravila koja mogu biti primjenjena na evoluciju sistema.

U pravilu, moguće je koristiti bilo koje pravilo za izbor susjedstva. Međutim, u praksi izučavanja pokazalo se da je najjednostavnije pravilo koje uzima samo simetrično susjedstvo od $2n$ najbližih celija sasvim dovoljno da se pokazuju i najkompleksnije osobine.

Prodiskutujemo sada broj mogućih pravila za svako izabrano susjedstvo. Cjelokupno susjedstvo će imati $2n + 1$ celiju uključujući i celiju u razmatranju. Za svaku od kombinacija stanja, tačnije 2^{2n+1} mogućih s obzirom na 2 moguća stanja, potrebno je definisati sljedeće stanje u koje se prelazi ukoliko se nađje na tu situaciju. Svako od idućih stanja također pripada skupu $\{0, 1\}$ tako da je ukupan broj pravila $2^{2^{2n+1}}$ za ovako izabrano susjedstvo.

Ispostavilo se da također i ovdje najjednostavnija pravila daju najzanimljivije rezultate, što smo vidjeli u već dosta primjera u ovoj oblasti, te za najjednostavnije susjedstvo sa $n = 1$ daje poprilično zanimljivu kompleksnost i moguća razmatranja. Najveći broj radova iz oblasti također je pisan upravo za ovaj slučaj (pogledati npr. većinu radova Stephena Wolframa uključujući [2], [3], [4] i mnoge druge).



Slika 12. rule picker (iz simulatora)

Sada, prema gore dobijenoj formuli imamo da je ukupan broj mogućih pravila $2^{2^3} = 2^8 = 256$. Ovo može grafički biti prikazano kao što je dato na slici 12 gdje imamo sljedeće stanje za svaku kombinaciju susjedstva.

Primijetimo i da proširenjem susjedstva na $n = 2$ broj pravila raste na $2^{2^5} = 2^{32} = 4294967296$, tako da nesto kompleksniji sistemi imaju eksponencijalno veći broj pravila. I ovo je također jedan od razloga izučavanja najjednostavnijeg slučaja zbog mogućnosti pregleda svih pravila, što bi za slučaj $n = 2$ bilo teže.

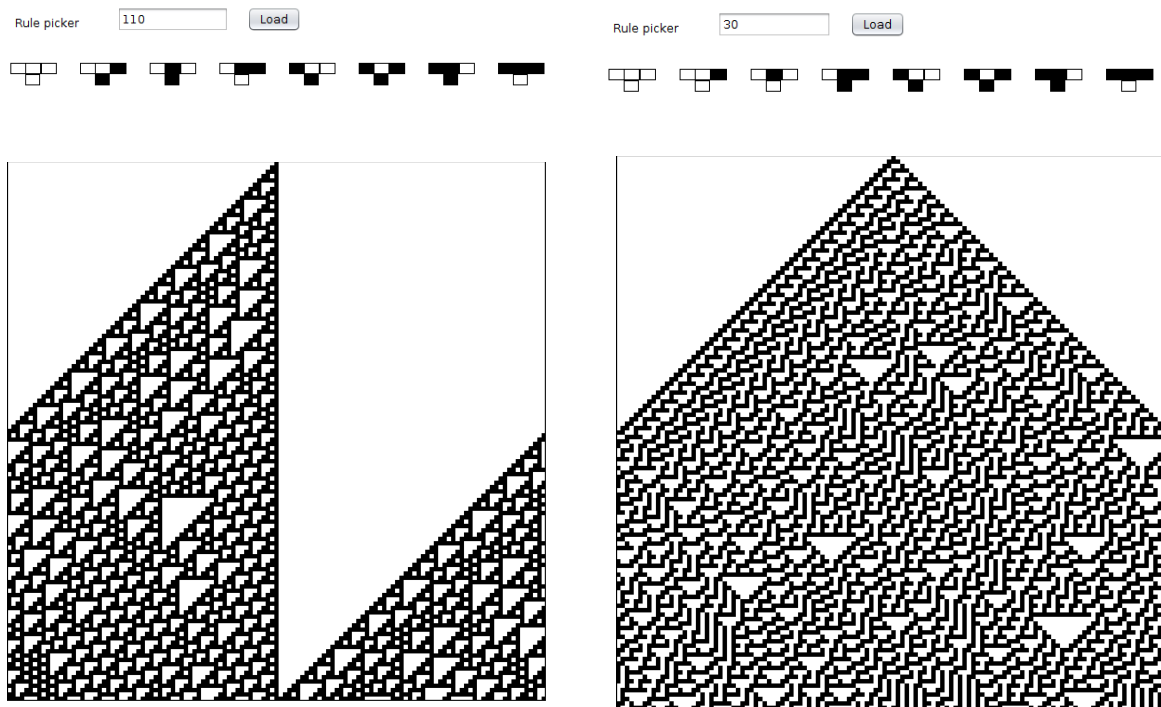
U daljem tekstu sva razmatranja odnose se pretežno na slučaj susjedstva $n = 1$ binarnih jednodimenzionalnih celijskih automata.

Nacin na koji definisemo koje pravilo se koristi pri evoluciji ovih sistema, a predložio ga je Stephen Wolfram na početku izučavanja ove oblasti, temelji se na enumeraciji pravila na osnovu bita u koje prelaze leksikografski poredane kombinacije susjedstva. To znači da su susjedstva poredana kao na slici 12, dok redoslijed on/off stanja u koji ona prelaze predstavlja binarno zakodiran broj pravila. Tako na primjer pravilo na slici 12 je predstavljeno brojem $01101110_2 = 110_{10}$, gdje se nule i jedinice redom uzimaju kao off

ili on stanja respektivno.

Ostaje jos i problem rubnih celija. Iako je u definiciji celijskog automata potencijalno beskonacna resetka, u realnim primjenama koristi se konacan broj celija u automatu, pa za jednodimenzionalni binarni slucaj koji ispitujemo postoje celije koje se nalaze na krajevima niza i koje nemaju lijeve i desne susjede. Tako da je potrebno na neki nacin definisati susjedstvo i za njih.

Za ovaj problem koristi se nekoliko mogucih rjesenja u zavisnosti od primjene i svrhe u koju se koristi celijski automat. Moguce je zamisliti da se kraj i pocetak niza spajaju u torusnu topologiju, tako da je niz neprekidan i prva i posljednja celija su susjedne. Ovo je najcesce koristen model u teoretskim izucavanjima. Takodjer moguće su i alternative, kao npr. imati fiksne vrijednosti rubnih celija sto je korisno u odredjenim simulacijama protoka toplote i slicnih fizikalnih pojava. Pokazuje se da izbor ovih rubnih uslova nema veci efekat na kvantitativnu i statisticku analizu (pogledati [6]).

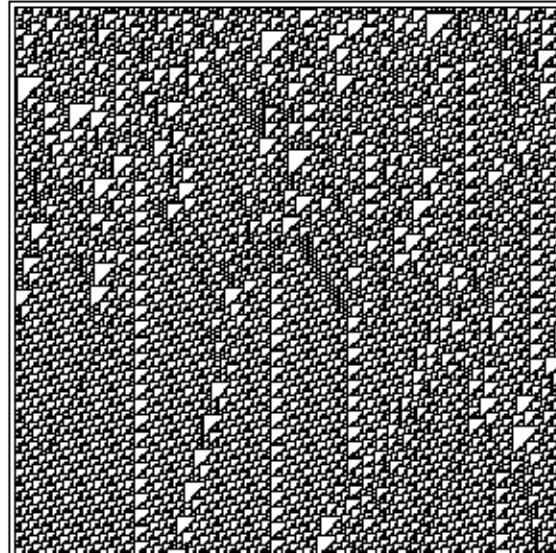
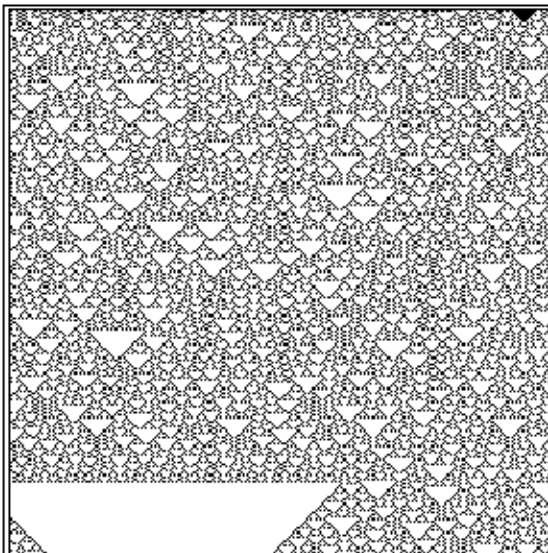
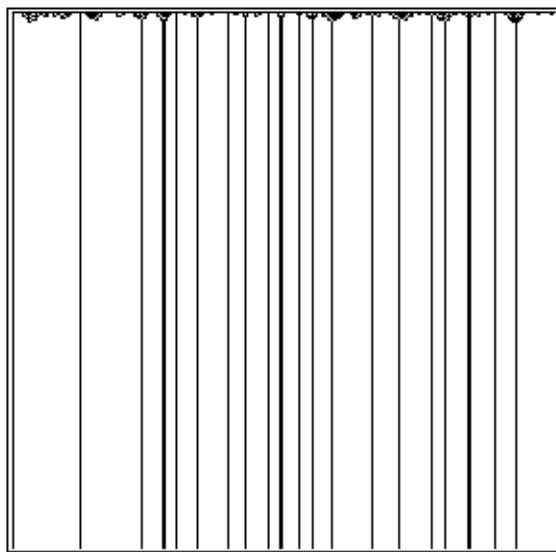
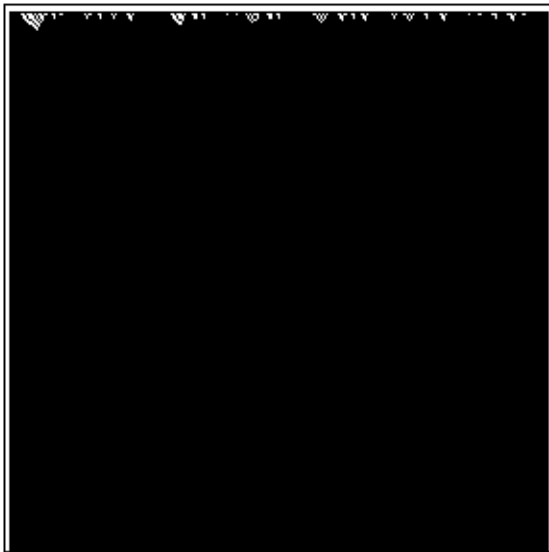


Slika 13. pravila 110 i 30 u evoluciji (simulator)

Na slici 13 prikazana je graficka evolucija od nekoliko stotina koraka pravila 110 i 30 sa torusnom topologijom niza respektivno da bismo poceli uocavati kompleksnost koja proizilazi iz nekih od najjednostavniji sistema izracunavanja koji mogu da se smisle. Kasnije cemo vidjeti da je moguće i formalnim putem pokazati da su ovi sistemi poprilično mocu sa strane mogucnosti izracunavanja generalnih funkcija – tj. simulacije ili ekvivalentnosti sa univerzalnom Turingovom masinom.

2.2.2 Pocetna statisticka zapazanja i klasifikacija ponasanja pravila celijskih automata

Elementarni celijski automati predstavljaju poprilično jednostavne sisteme sa ne tako mnogobrojnim skupom pravila za evoluciju istih. Prije smo vidjeli da taj broj iznosi svega 256 sto omogucava cak i brute force pretrazivanje skupa pravila i ispitivanje osobina svakog pravila ponaosob. Cak i neka od prvih izucavanja bazirala su se upravo na ovom principu (pogledati [2]). Upravo ova jednostavnost koja ne umanjuje kompleksnost struktura koje se formiraju omogucava detaljno teoretsko izucavanje elementarnih celijskih automata za razliku od nekih drugih sistema sa podjednako kompleksnim formacijama.

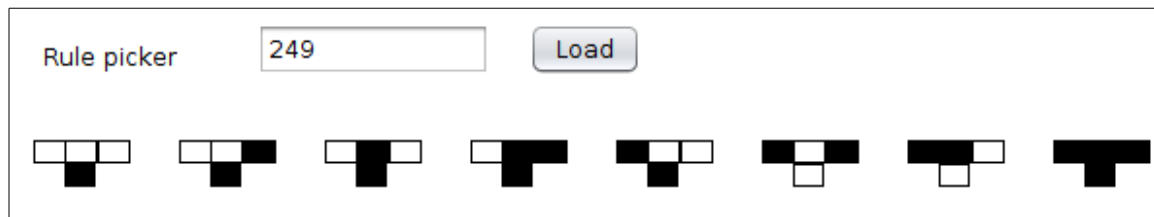


slika 14. evolucija pravila 249, 164, 146, 110 u klasama I, II, III i IV (simulator)

Ukoliko se izvrše simulacije sa slučajnim početnim uslovima svih mogućih pravila elementarnih ćelijskih automata (pogledati [2] za listu grafičkih navedenih grafičkih simulacija), moguće je uočiti određene uzorke u ponasanju istih. Tako naizgled čisto empirijskim ispitivanjem moguće je uočiti grube osobine nekih od pravila, što se kasnije može pokušati pretociti u formalnu klasifikaciju.

Klasifikacija sama po sebi je bitna jer predstavlja prvi korak u izučavanju neke vrste sistema. Ukoliko sve instance sistema posmatramo kao odvojene bez nekog načina na koji bismo svrstali svaku od njih u određenu klasu ponasanja, zapravo se radi o ad-hoc izučavanjima specijalnih slučajeva. Sama klasifikacija predstavlja viši nivo za posmatranje sistema koji se nalaze u njoj, te kroz klasifikaciju možemo da uočimo i neke ključne osobine višeg reda, što nas pomjera od početne tačke gdje smo samo definisali sistem i način njegovog funkcionisanja/evolucije.

Ukoliko pogledamo sliku 14, prikazana je vremenska evolucija četiri karakteristična pravila koja ujedno predstavljaju i osnovne uočljive klase ponasanja elementarnih automata koje ćemo sada navesti i nešto kasnije i obraditi [1]. Formalna obrada tematike ćelijskih automata predstavlja nešto teži zadatak s obzirom da se radi o nelinearnim sistemima za koje se većinom ne može naći eksplicitan matematički model predviđanja budućih stanja u zavisnosti od sadašnjeg. S obzirom na ovo u dosta slučajeva moguće je izvršiti jedino statistička i slična globalna ispitivanja bez dolaska na konkretan model predviđanja (pogledati npr. [6], [9] kao i većinu radova iz oblasti ćelijskih automata i dinamike kompleksnih nelinearnih sistema).



Slika 15. grafički prikaz prelaza pravila 249 (simulator)

Prva (empirijski!) uočljiva klasa – Klasa I automata predstavljena je pravilima koja konvergiraju u homogene strukture bez obzira na početno stanje sistema. Također, sve informacije kao i slučajnost (eng. randomness) u početnim uslovima gube se u narednim koracima. Tako da ova klasa automata ne predstavlja nikakvu korist sa strane kompjutacije s obzirom da bez obzira na ulaz koji bismo zakodirali kao početno stanje automata, uvijek ćemo dobiti isti izlaz koji ne sadrži nikakve korisne informacije. Primjer ovakvog pravila je pravilo 249 koje bez obzira na strukturu početnih uslova konvergira u niz ćelija u off stanju (crna boja na grafičkom prikazu). Ovo možemo pripisati činjenici da većina kombinacija susjedstva ovog pravila vrši prelaz u off stanje, što je evidentno sa slike 15.

Iduću klasu – Klasu II predstavljaju pravila koja ulaze u cikluse ponavljajućih struktura.

Na slici 14 prikazano je medju ostalima i pravilo sa enumeracijom 164 koje konvergira u ponavljajuće strukture bez obzira na pocetno stanje. Razlika ove klase pravila sa Klasom I je to sto strukture nisu uvijek iste i nisu obavezno homogene, medjutim postoje ciklusi ponavljanja istih. Treba primijetiti da bilo koja konacna konfiguracija automata takodjer mora da ispolji ciklicno ponasanje s obzirom na ukupan broj mogucih stanja 2^N ukoliko N predstavlja broj celija u pocetnoj konfiguraciji. Iako ce se ciklus u najgorem mogucem slucaju javljati nakon upravo 2^N ponavljanja nakon sto su prodjena sva moguca stanja, sto bi znacilo da i za poprilično male konfiguracije imamo ogromne cikluse, empirijski rezultati pokazuju da je taj broj manji. Pretežno je ogranicen je sa $2^{N/2}$ dok je u dosta slucajeva cak potrebno ne vise od N iteracija da bi se dovršio ciklus [6].

Unutar Klase III nalazimo pravila poput 146 i 30 koja ispoljavaju naizgled nepredvidivo slucajno ponasanje sa ponavljajucim uzorcima pretežno ispoljenim u vidu trougaonih struktura. Ova klasa je poprilično osjetljiva na pocetne uslove i predstavlja dobar alat za izucavanje slucajnosti (eng. Randomness) [1]. Kasnije cemo vidjeti nesto rigoroznije razmatranje kao i implikacije za prakticnu primjenu ovog fenomena (pogledati npr. [7]).

Klasa IV predstavlja najkompleksniju klasu ponasanja celijskih automata, ali isto tako i najslabije definisanu. Neformalno, u ovoj klasi automata gotovo svi pocetni uslovi proizvode kompleksne strukture koje vrše kompleksnu medjusobnu interakcije. Postulirano je da ova klasa omogucava skladištenje i prenos informacija, sto su osnove za univerzalnu kompjutaciju. Pravilo 110 iz ove klase cak je i pokazano kao univerzalno kompjutaciono, tj. Turing ekvivalentno (za vise informacija i dokaz pogledati [8]). I ovo ce biti kasnije nesto detaljnije razmotreno.

Sva gore zapazanja su empirijska i neformalna, ali lahko uocljiva smo pogledom na rezultate simulacije. Bilo je nekoliko pokusaja formalne klasifikacije elementarnih automata (i ostalih), od kojih nijedan nije bio potpuno uspjesan u smislu da daje predvidjanja ponasanja pravila u zavisnosti od neke njegove osobine, medjutim svi oni su dali neke rigorozne rezultate koji mogu biti od koristi. Takodjer kroz proces pokusaja pronalaska formalnog okvira za klasifikaciju doslo se i do mnogo zakljucaka o ovoj vrsti sistema koji prije nisu bili poznati.

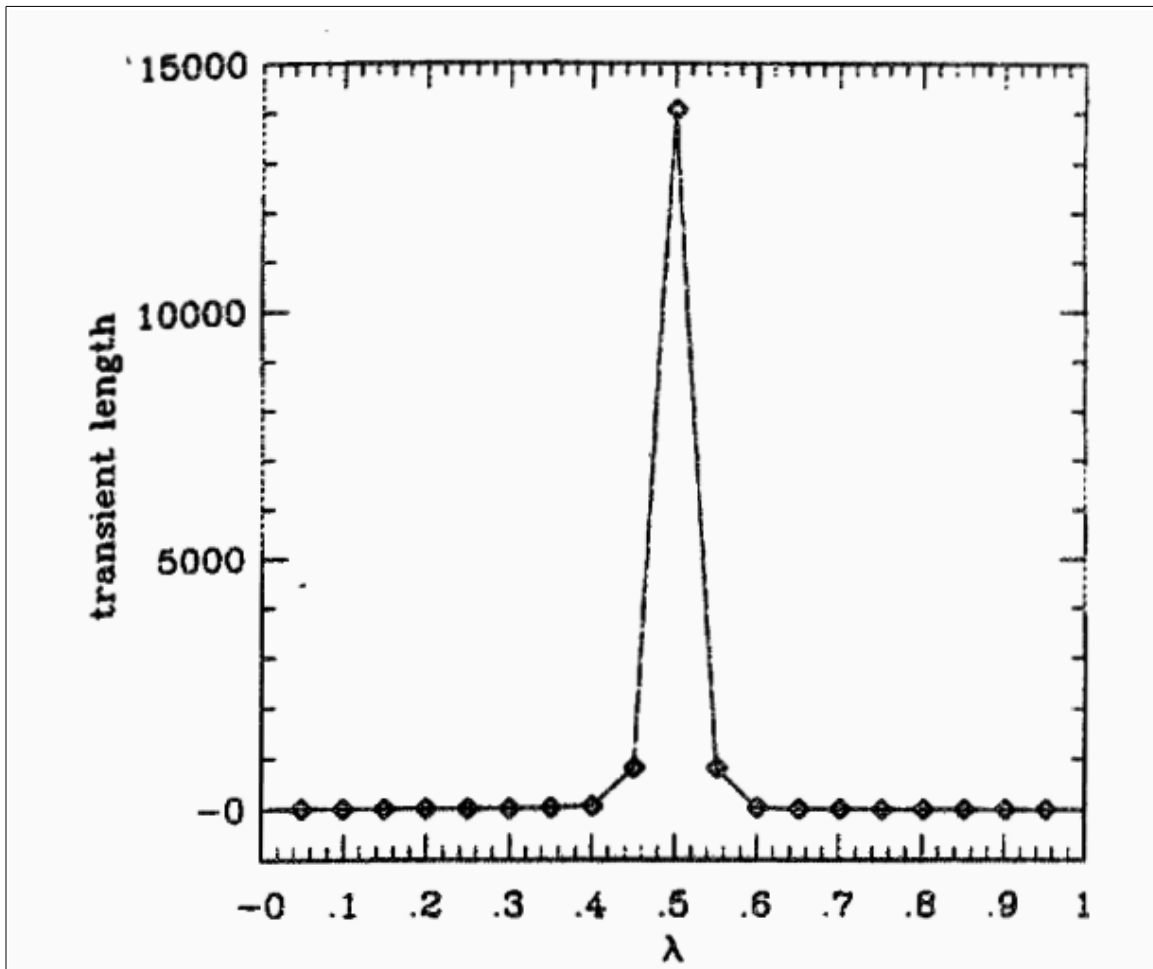
Obradicemo jedan takav pokusaj s obzirom da je dao poprilično dobre rezultate, te otvorio neka nova pitanja i hipoteze ne samo u oblasti izucavanja celijskih automata, vec dinamike kompleksnih sistema generalno. Tako cemo i ovdje navesti neke od tih rezultata kroz pokusaj formalne klasifikacije. Klasifikator koji navodimo naziva se Langtonov parametar [9]. Prema rijecima Langtona koji je i kreator ovog nacina klasifikacije, *“ovaj parametar je agregatna statistika koje je povezana sa, ali ne i sasvim pouzdana u predvidjanju kompleksnosti ponasanja”* [1].

Definicija 2.2.2 Za jednodimenzionalni celijski automat sa k stanja i velicinom susjedstva n , Langtonov parametar λ definise se kao $\lambda = \frac{k^n - \#_q}{k^n}$, gdje je $\#_q$ broj celija u funkciji prelaza koje su u unaprijed izabranom specijalnom (eng. quiescent) stanju.

Intuitivno, definicija 3.6 samo definise Langtonov parametar kao udio stanja koja nisu off (koje je izabrano kao specijalno za nas slucaj elementarnih automata) u funkciji prelaza za elementarne automate. Tako na primjer za pravilo 110 cija je funkcija prelaza data na slici 12 Langtonov parametar iznosi $\lambda = \frac{2^3-3}{2^3} = \frac{8-3}{8} = \frac{5}{8} = 0.625$ s obzirom da su 3 stanja off u prelazu.

Zapazanja koja cemo navesti Langton u svom radu struktuirano obradjuje za automate sa $n = 4$ i $k = 5$ [9] koji predstavljaju znatno kompleksnije sisteme u odnosu na obradjivane elementarne celijske automate, medjutim dobijeni rezultati govore o generalnim osobinama jednostavnih sistema koji proizvode kompleksne uzorke, te su itekako primjenljivi na obradjivanu tematiku.

Ukoliko eksprimentalno diskretno u razmacima od po 0.01 variramo parametar λ te za svaku tu vrijednost konstruisemo niz slucajnih pravila koja zadovoljavaju tu vrijednost parametra, te pustimo simulaciju za iste nad slucajno generisanim pocetnim uslovima, dobijaju se poprilično zanimljivi rezultati koji govore da postoji odredjena statisticka struktura u ponasanju ovih sistema u zavisnosti od parametra.

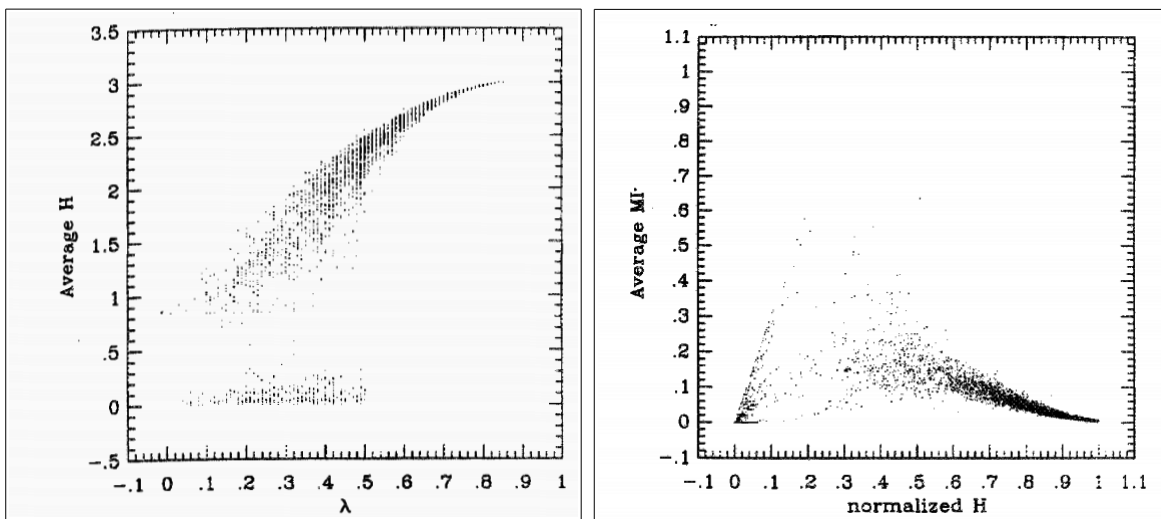


Slika 16. prosjecno vrijeme stabilizacije (preuzeto iz [9])

Langton postulira da unutar celijskih automata postoje klase ponasanja analogne agregatnim stanjima fizicke materije, gdje postoje i prelazne tacke agregatnih stanja koje su od posebne vaznosti. Ovo zapazjenje bazira na cinjenici da je prelaz stanja u dinamicnim sistemima (pa tako i agregatnim stanjima materije) ima direktno korelaciju sa vremenom stabilizacije sistema, sto predstavlja vrijeme koje je potrebno sistemu da udje u svoj "prirodni rezim rada". Naime, sto je sistem blizi prelazu stanja, to je i duze vrijeme njegove stabilizacije.

Ako izuzmemo statistiku iz prethodno opisanog eksperimenta, te ucrtamo na grafik prosjecno vrijeme stabilizacije, dobijamo grafik kao na slici 16. Vidimo da oko kriticne vrijednosti $\lambda = 0.5$ koju cemo nazvati λ_c vrijeme stabilizacije naglo raste, sto bi mogao biti indikator postojanja stanja/rezima rada celijskih automata, kao i postojanja prelaza stanja za kriticnu vrijednost $\lambda = \lambda_c$.

Pozeljno bi bilo sada ispitati nivo kompleksnosti sistema u zavisnosti od λ te vidjeti da li postoji potencijalna poveznica izmedju rezima rada automata, te kolicine informacija koju sistem prenosi, s obzirom da smo vidjeli da za odredjene sisteme sve pocetne informacije bivaju izgubljene, dok za druge izgledaju kao potpuno nasumicno generisane. Distribucija nivoa prenosa informacija u odnosu na parametar mogla bi dati indikacije i o zavisnosti ove osobine od stanja u kojem se sistem nalazi ukoliko prihvatimo postojanje stanja i njihovih prelaza. Nesto kasnije ovo bi se moglo iskoristiti i za izucavanje kompjucionog potencijala sistema, jer znamo da sistem ukoliko zelimo da ga iskoristimo za univerzalni tip izracunavanja mora da ima mogucnost skladistenja i propagacije informacija. Bitno je napomenuti da su ovo sve potencijalne korelacije, ali da ne moraju imati uzrocno-posljedicnu vezu. Neki su cak i opovrgnuli ova zapazanja kao netacna u kasnijim radovima, vidjeti npr [11].

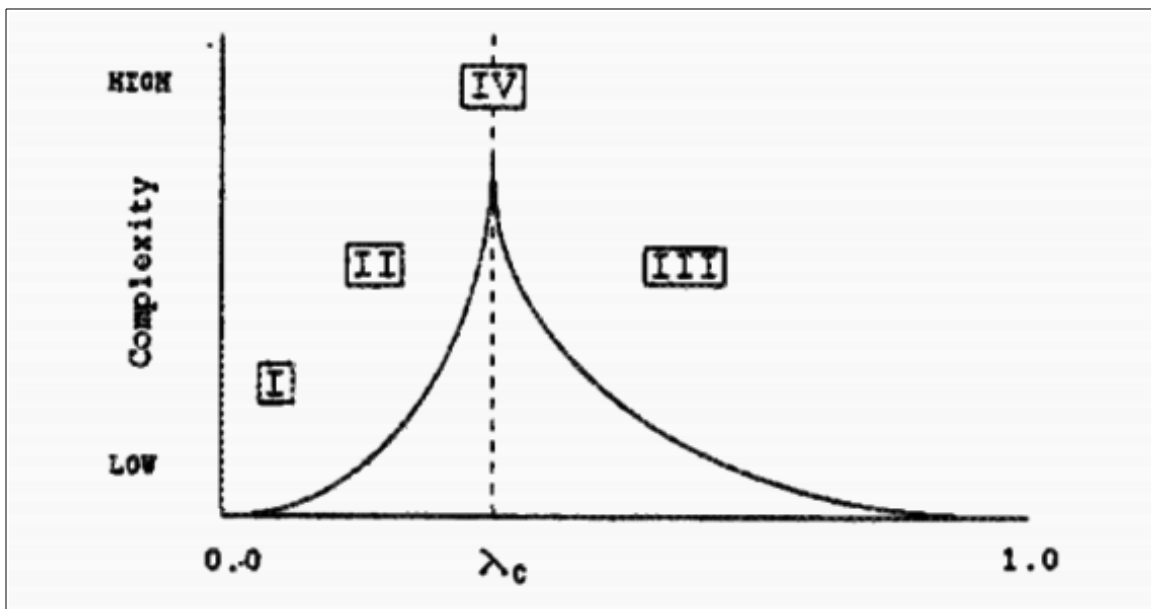


Slika 17. prosjecna entropija svake celije u zavisnosti od parametra (lijevo) prosjecna kolicina medjusobnih informacija celija za odredjenu entropiju (desno) (preuzeto iz [9])

Na slici 17 prikazana je na lijevoj strani prosjecna entropija (za definiciju pogledati preliminarne definicije u poglavlju 2.0 ili nesto strucniju literaturu, npr. [10]) svake celije u zavisnosti od parametra. Svaka tacka na grafu predstavlja odredjeno pravilo koje zadovoljava parametar. Mozemo uociti da sa porastom parametra sve se vise smanjuje razlika izmedju minimalne i maksimalne entropije za datu vrijednost parametra. Kljucno je i da je za vrijednost parametra 0.5 za koju je postulirano da predstavlja prelaz stanja, entropija varira u pojasu od priblizno 2.4 - 2.6.

Na desnoj strani na slici 17 nadalje je prikazan graf koji povezuje prosjecnu zajednicku dijeljenu kolicinu informacija (eng. mutual information) za odredjenu entropiju. Moze se uociti da postoji maksimalna vrijednost zajednicke kolicine povezanih informacija koje nose celije, te se ta vrijednost upravo poklapa sa vrijednoscu entropije u pojasu od priblizno 2.4 - 2.6, sto odgovara upravo postuliranoj vrijednosti parametra $\lambda = 0.5 = \lambda_c$.

Iz ovih vrijednosti moguće je zaključiti (ne sa potpunom sigurnoscu) da u kritičnoj vrijednosti parametra gdje se desava prelaz stanja ujedno se dostize i maksimum zajednickih informacija celija, sto predstavlja osnovu za univerzalnu kompjutaciju. Langton ovo naziva “kompjutacija na rubu haosa”, te predstavlja koncept da postoji tanka linija izmedju neodredjenosti, uredjenosti i haosa na kojoj postoje plodni uslovi za univerzalnu kompjutaciju. Ovo bi imalo i velike filozofske implikacije na cijele oblasti kompjuterske nauke i dinamike kompleksnih sistema s obzirom da bi kompjutacija kakvu znamo predstavljala samo specijalan slucaj znatno sireg koncepta [9].



Slika 18. Wolframove intuitivne klase ponasanja celijskih automata u zavisnosti od Langtonovog parametra (preuzeto iz [9])

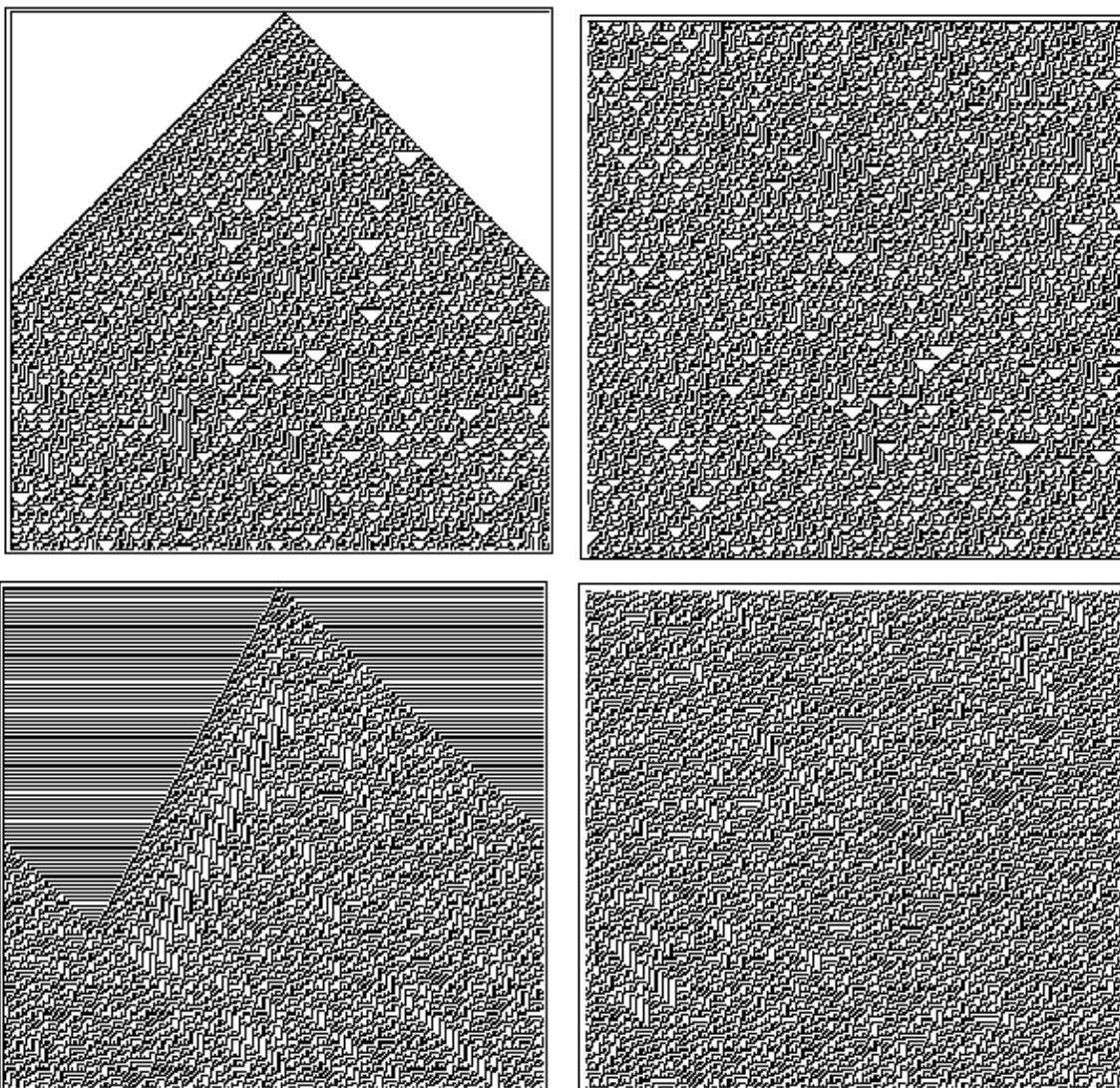
Ovo ima implikacije i na klasifikaciju ponasanja automata do koje smo zeljeli doci, jer bi prema tome automati Wolframove klase IV spadali upravo na rub prelaza, tj. u okolini kritične vrijednosti $\lambda = \lambda_c$, dok bi klase I i II bile u vrijednostima manjim od kritične s obzirom na nisku kolicinu informacije koje prenose, a klasa III bi bila locirana na

vrijednostima većima od kritične s obzirom na kaotično ponašanje iste. Ovaj raspored klasa prikazan je na slici 18.

Kako smo i rekli klasifikacija u većini slučajeva može biti prvi korak u daljnjem izučavanju nekog sistema, pa tako i ova navedena klasifikacija, bilo da se radi o intuitivnoj ili pokušajima formalne, daje naznake o zanimljivim osobinama elementarnih ćelijskih automata koje bi trebalo detaljnije ispitati. Tako klasifikacija predviđa automate koji imaju potpuno nepredvidivo ponašanje (eng. random), kao i automate “na rubu haosa” koji imaju moc univerzalne kompjutacije. Upravo ove klase koje odgovaraju Wolframovim klasama III i IV kao najzanimljivije i klase sa najvećim potencijalnim primjenama biće izučene u sljedećim razmatranjima.

2.2.3. Slučajnost (eng. randomness) elementarnih ćelijskih automata

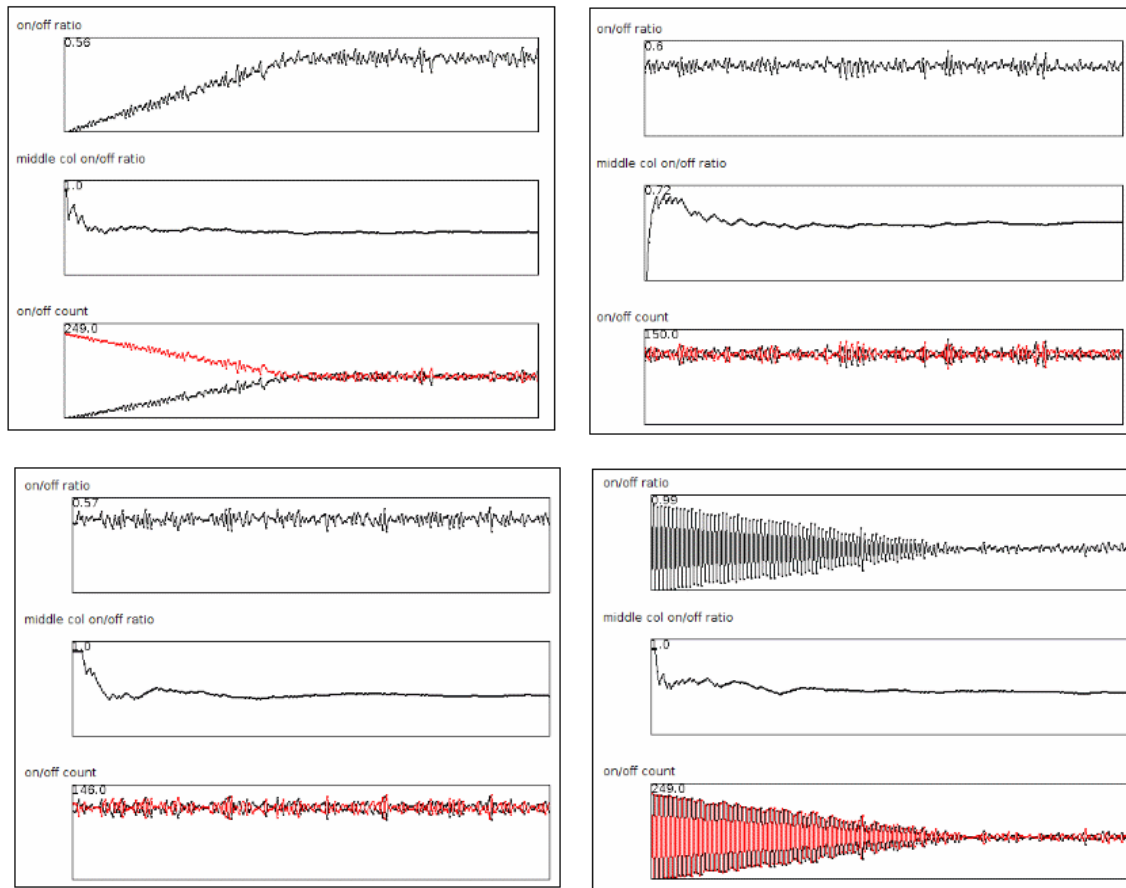
Izučavanje slučajnosti i slučajnih pojava ključno je u raznim dijelovima kompjuterske i ostale nauke s obzirom da neke od ključnih primjena danasnjice poput Monte-Carlo algoritamskih metoda i kriptografije su upravo bazirane na ovom konceptu. Odmah se postavlja pitanje da li bi se uočena slučajnost u klasi klasi III elementarnih ćelijskih automata mogla iskoristiti u ove svrhe, s obzirom da bi jednostavnost sistema koji generisu ovu slučajnost mogla biti presudna u izboru istih nad drugim metodama. Iz ovog razloga, korisno je izučiti nešto detaljnije i formalnije količinu slučajnosti koju mogu da proizvedu navedeni sistemi.



Slika 19. Strukture generisane pravilima 30 (prvi red) i 45 (drugi red) (generisano softwareom)

Ukoliko osmotrimo strukture generisane elementarnim pravilima 30 i 45 (i njihovim simetričnim ekvivalentima) kao što je prikazano na slici 19, može se uočiti da nema nekog predvidivog uzorka u ovim strukturama što bi mogao biti dobar indikator njihove potencijalne slučajne prirode [12].

Mnogi su testovi razvijeni da bi se testiralo koliko je zapravo neki sistem statistički slučajan. Treba napomenuti da se koncept slučajnosti razlikuje od statističke slučajnosti s obzirom da se statistička slučajnost javlja u sistemima koji su u svojoj prirodi deterministički, kao što je primjer i sa trenutno izučavanim elementarnim automatima, međutim ne postoji globalni uzorak ponašanja koji oni zadovoljavaju.



Slika 20. Statistika za simulacije na slici 19. (generisano softwareom)

Najprimitivnije formalno razmatranje koje bi moglo dati informacije o kolicini slucajnosti sistema je statistika odnosa on i off stanja celija u redu, s obzirom da ukoliko je sistem zaista slucajan taj bi odnos trebao da bude blizu $1/2 = 0.5$ s obzirom da sva stanja moraju biti podjednako moguca u potpuno slucajnom sistemu.

Razmotrimo sliku 20 koja daje po tri statistike za evoluciju pravila 30 i 45 sa uredjenim i slucajnim pocetnim uslovima (pogledati sliku 19) respektivno. Prva statistika za svaku konfiguraciju predstavlja odnos on celija u sistemu sa ukupnim brojem celija (nazvano on/off ratio) koji predstavlja procenat ili udio tih celija u svakom od redova. Apscisa predstavlja protok vremena automata, dok je na ordinati unijet ovaj odnos. Druga statistika predstavlja isti udio ali primjenjen ne na pojedinačne redove, već na srednju kolonu jer je predloženo da se upravo ovo koristi kao pseudoslučajni generator u kriptografskim primjenama [12][13]. Treća statistika je poprilično redundantna ako nas zanima samo odnos, te predstavlja stvarni broj on i off celija u svakom redu u svakom vremenskom koraku.

Vidimo da se za izvršeni broj koraka u svakom od četiri slučaja simulacije posmatrani odnos kako u svakom redu pojedinačno, tako i u specijalno posmatranoj srednjoj koloni blizu predviđenoj vrijednosti 0.5 koja bi bila postignuta za pseudoslučajni generator, tako da ovo može dati jake indikacije o slučajnosti navedenih pravila.

Wolfram je u [12] detaljnije i jaci matematičkim aparatom – informacionom teorijom i na prednom statistikom a ne samo indikacijama iz simulacija formalno obradio slučajnost ovih sistema. Dosao je do zaključaka da je pravilo 30 dovoljno slučajno za većinu primjena uz dovoljno veliku početnu konfiguraciju, dok je pravilo 45 znatno manje statistički slučajno, ali opet nepredvidivo u velikoj mjeri. Nesto kasnije u [14] pokazano je da za određene vrijednosti velicine inicijalne konfiguracije kriptanaliza može da obrne proces pseudoslučajne generacije.

Zaključujemo da je jedna od osobina pojedinih klasa i pravila elementarnih ćelijska automata tako strukturirana da ispoljava statističku slučajnost, što će kasnije biti izučeno s aspekta primjena ove vrste sistema.

2.2.4 Univerzalna izračunljivost elementarnih ćelijskih automata

Nakon predstavljanja bilo kojeg sistema koji ispoljava bar neki nivo kompleksnog ponašanja, uvijek je zanimljivo upitati se kolika je zapravo ta kompleksnost te je pokušati na neki način kvantificirati. Ovo znači ispitati da li je sistem sposoban simulirati bilo koji drugi sistem koji se smatra izračunljivim u matematičkom smislu.

Unutar teorije izračunljivosti, kao osnovni model izračunljivog sistema uzima se bilo koji algoritam ili procedura za koji se može konstruisati Turingova mašina koja ga simulira. Ovo direktno slijedi iz rezultata poznatog kao Church-Turingova teza koji govori da je pitanje izračunljivosti ekvivalentno pitanju da li je moguće to ponašanje simulirati na nekoj konstruisanoj Turingovoj mašini. Ovo razmatranje direktno uvodi Turingove mašine kao osnovni model izračunljivih sistema sa kojim se ostali sistemi trebaju porediti ukoliko se želi pokazati njihova računarska moc. Moguće je koristiti i neke druge modele za izučavanje ovog polja poput Alan Churchovog λ – *kalkulusa*.

Pitanje univerzalnosti unutar ovih okvira se svodi na mogućnost simulacije Turingovih mašina, tj. sistem se naziva univerzalno kompjutacion ili Turing ekvivalentan ukoliko je u mogućnosti simulirati svaku proizvoljnu Turingovu mašinu, što ima smisla ukoliko se razmotri zašto se Turingova mašina smatra za osnovni model preko kojeg se definiše izračunljivost.

Moguće je sada postaviti ovo pitanje i za elementarne ćelijske automate. Očigledno je i trivijalno pokazati da neka pravila nisu Turing ekvivalentna. Razmotrimo na primjer pravila iz Wolframovih klasa I i II. Ona gotovo pri svakoj inicijalnoj konfiguraciji dovode do homogenih ili ponavljajućih krajnjih stanja sistema, te kao takvi evidentno nisu u mogućnosti mapirati proizvoljan ulaz na proizvoljan izlaz. Pravila iz klase III kako je već pokazano pokazuju poprilično slučajno ponašanje, te kao takva također nisu dobar kandidat za Turing ekvivalentne sisteme s obzirom da nam za izračunljivost predvidivost igra ključnu ulogu. Ostaje nam klasa IV za koju je i postulirano da predstavlja klasu u kojoj se nalaze pravila dovoljno kompleksna s jedne strane, ali dovoljno strukturirana s

druge da bi se mogla iskoristiti u svrhu univerzalne izracunljivosti. Ova pravila takodjer spadaju u tanku liniju koju Langton [9] naziva “rub haosa” na kojoj bi mogle da se desavaju pojave sposobne za univerzalnu izracunljivost o kojoj smo raspravljali pri problemu klasifikacije elementarnih celijskih automata.

Matthew Cook je 2004. godine dao dokaz o univerzalnosti jednog od pravila elementarnih celijskih automata, i to pravila 110 za koje je Stephen Wolfram 1985. i postulirao da predstavlja Turing ekvivalentno, tj. univerzalno kompjutaciono pravilo [15].

Da bismo razumjeli okvirno u cemu lezi kljuc dokazivanja Turingove kompletnosti sistema elementarnog celijskog automata sa tranzicionima pravilom 110, potrebno je prvo navesti neke uvodne elemente.

Turingova kompletnost ima tzv. osobinu tranzitivnosti, tj. ukoliko je neki sistem A takav da je Turing kompletan, a neki drugi sistem B moze da ga simulira, tada je i sistem B Turing kompletan. Ovo moze biti korisno ukoliko je tesko dokazati direktnu mogucnost simulacije Turingove masine, ali je jednostavnije dokazati mogucnost simulacije nekog drugog sistema za koji je poznat da je Turing kompletan. Ovo je metod koji i Cook koristi u [15], te je izabran ciklicni tag sistem za koji je poznato da je Turing kompletan.

Tag sistem generalno predstavlja sistem izracunavanja koji se bazira na iteriranoj modifikaciji pocetnog stringa. Najbolje je prvo dati primjer i kroz njega pokusati shvatiti koncept, nakon cega ce biti data i formalna definicija.

Tag sistem za svaki simbol u alfabetu specifcnog sistema (alfabeti se naravno mogu razlikovati) daje string kojem se taj simbol pridružuje . U svakoj iteraciji, iz string se uklanja simbol sa pocetka stringa i na osnovu pravila i odgovarajuceg stringa za koje je simbol vezan, originalni string se nadopunjava.

Neka je dat pocetni string baa unutar alfabeta $\{a, b, c, H\}$, gdje H predstavlja terminalni simbol na koji ukoliko se naidje obustavlja daljnje izvršenje. Skup pravila je definisan kao $a \rightarrow ccbaH, b \rightarrow cca, c \rightarrow cc$. Tada 6 iteracija ovog sistema izgleda:

Iteracije:

- 1 baa
- 2 $acca$
- 3 $cacbaH$
- 4 $ccbaHcc$
- 5 $baHcccc$
- 6 $Hcccccca$ (halt).

Sistem staje u sestom koraku jer nailazi na halt simbol koji mu to govori.

Nesto modificirana verzija tag sistema koja je originalno koristena u [15] je ciklicni tag sistem, koji je u sustini ekvivalentan sa tag sistemom te postoji poprilično jednostavna pretvorba iz jednog u drugi, s razlikom da ciklicni tag sistem ne mora da provjerava

kojem simbolu odgovara koji string, već ima skup pravila koja ciklicno vrti – zato se i naziva ciklicni. Naime, za ciklicni tag sistem ne veže se pravilo za svaki simbol, već su pravila u fiksnom skupu i vrte se ukrug, te se početni znak uvijek mijenja trenutnim pravilom ukoliko znak nije takav da nalaze nemijenjanje, što je objašnjeno u sljedećem primjeru.

Neka je na primjer dat ciklicni tag sistem sa skupom pravila (produkcija) (010, 000, 1111). Alfabet ciklicnog tag sistema sastoji se od simbola $\{0, 1\}$, gdje se trenutno aktualno pravilo iz skupa produkcija primjenjuje na string ukoliko je skroz lijevi simbol 1, doke se u suprotnom simbol 0 samo uklanja sa početka stringa. Ukoliko je početni string bio 11001, tada niz iteracija na osnovu ciklicno primjenjenih produkcija izgleda:

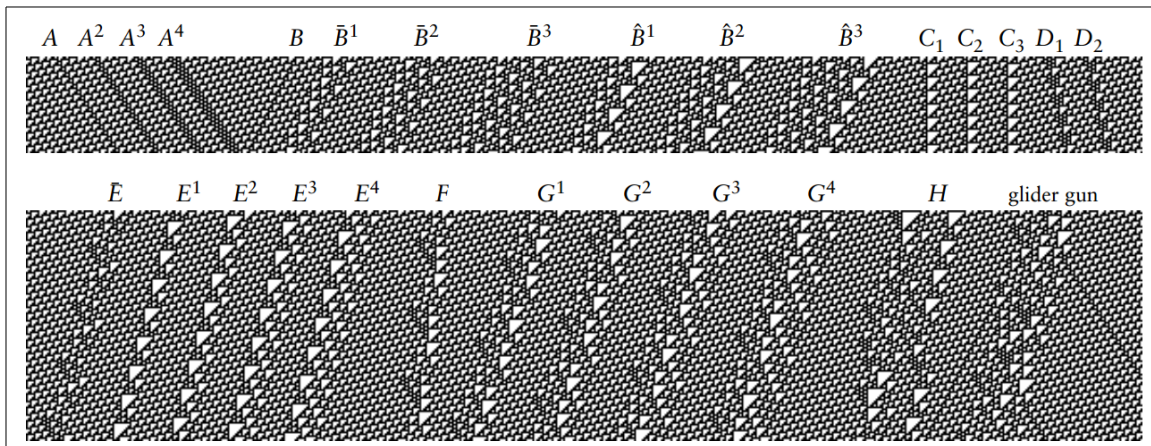
Iteracije:

trenutno pravilo: 010 trenutni string: 11001
 trenutno pravilo: 000 trenutni string: 1001010
 trenutno pravilo: 1111 trenutni string: 001010000
 trenutno pravilo: 010 trenutni string: 01010000
 trenutno pravilo: 000 trenutni string: 1010000
 trenutno pravilo: 1111 trenutni string: 010000000
 trenutno pravilo: 010 trenutni string: 10000000

Vidimo da se pravila primjenjuju redom u cikličnom redoslijedu, dok se string mijenja samo ukoliko je početni simbol 0.

Sada ako znamo za osobinu tranzitivnosti Turingove kompletnosti, te činjenicu da je ciklicni tag sistem Turing kompletan, tada ukoliko bismo pokazali da je pravilo 110 u mogućnosti na neki način emulirati ciklicni tag sistem, tada bi to bilo dovoljno da pokazuje da je i samo pravilo 110 univerzalno kompjutaciono.

Na ovom principu je i Mathew Cook dokazao univerzalnost, a ovdje će samo biti izložene osnovne ideje koje se kriju iza dokaza, jer je sam dokaz preobiman te se čitaoc za više detalja može referirati na [15].



Slika 20. Glideri pravila 110 (preuzeto iz [15] bez modifikacija)

Osnovno zapazanje je da pravilo 110, kao i neka druga pravila za koja se postulira univerzalnost, predstavljaju posebnu vrstu sistema u kojima se javljaju “cestice”, tacnije podskupovi konfiguracije koje propagiraju vremenom i prostorom u određenim periodima koji se nazivaju glideri. Da bi se bolje shvatilo na sta se misli mozemo pogledati sliku 20 na kojoj su dati osnovni tipovi glidera u pravilu 110.

Pogledajmo na primjer takozvani glider A (Cook je u dokazu dao nomenklaturu gliderima koje koristi sto se takodjer moze vidjeti na slici 20) koji zapravo predstavlja niz konfiguracija (111, 110, 100) koji se javlja periodicno, ali sa razlicitim pocetnim polozajem u svakom novom periodu, tako da mozemo reci da se glider krece. Upravo zavisnost vremena i novog pocetnog polozaja definise brzinu glidera. Glideri sa nenultom brzinom nazivaju se dinamicni. Postoje i stacionarni glideri koji imaju brzinu $v_g = 0$. takodjer mogu da vrse medjusobnu interakciju, takozvane kolizije, pri cemu se cesto desava da kolizija dva glidera proizvodi novi glider sa novim osobinama.

Upravo se na ovom skupu cinjenica temelji dokaz o univerzalnost. Tacnije, stacionarni glideri se koriste za cuvanje informacija u sistemu, dok se dinamicki glideri koriste da bi kolizijama mogli da modifikuju stacionarne glidera u kojima su informacije sacuvane. Glideri se biraju tako da modifikacije nastale kolizijama odgovaraju produkcijama ciklicnog tag sistema, sto dokazuje univerzalnost pravila 110.

Prakticna primjena i kodiranje realnim problema za izvorsavanje pravilom 110 moze se naci u [16], gdje se kombinuje pretvaranje Turingove masine u ciklicni tag sistem, nakon cega se on kodira za izvorsenje celijskim automatom. Iznenadjujuci je rezultat da ovo racunanje i pretvorba zahtijeva asimptotski polinomsko vrijeme, iako se smatralo da je izvorsenje barem reda eksponencijalne funkcije. Nije jos dovoljno istrazeno koliki je potencijal izracunavanja na ovaj nacin s obzirom na mogucnost fizicke implementacije celijskog automata za visoko paralelno izracunavanje, s obzirom da je jedna od osobina celijskih automata sinhroni (paralelni) prelaz stanja.

3. Prakticne primjene

U dosadasnjem dijelu rada vidjeli smo intuitivno i formalno prikazane osnovne osobine celijskih automata generalno, kao i pojedinih specifichnih klasa koje su zanimljive za razmatranje. Sada cemo pokusati pogledati na koji bi se nacin mogle iskoristiti te osobine u neke prakticne primjene, sto bi izvelo oblast iz cisto teoretske kompjuterske nauke u realnu inzenjersku aplikaciju.

Bice prvo razmotrene generalne oblasti primjene koje bi bile podrzane od strane osobina celijskih automata, te ce se nakon toga navesti i implementacije ili primjeri primjena nekih klasa specifichno.

3.1 Generalne oblasti primjene celijskih automata

Pogledajmo prvo neke generalne osobine celijskih automata koje bi mogle biti iskoristene za primjenu u određenim oblastima.

Pregledajmo prvo ključne osobine koje predstavljaju potencijalno plodno tlo za realne primjene celijskih automata. Za početak, celijski automati, bar ono što se klasično smatra njima, baziraju se na paralelnosti i homogenosti, to jest sinhronoj promjeni po identičnim pravilima stanja svih komponenti vodjeno globalnim diskretnim satom. Nesto slično vidimo i kod modernih računara gdje se elektronska logika ponasa na sličan način. Nadalje, celijski automati bazirani su na principu lokalnosti, što znači da pojedina ćelija ne zna ništa o cjelokupnom stanju sistema, već samo stanja svojih najbližih susjeda, što omogućava skladištenje jako male količine informacije i jednostavne komponente. Kao što je vidjeno, celijski automati iako jednostavni predstavljaju univerzalan izračunljiv sistem, tako da jednostavnost ne znači nužno i beskorisnost. Također, kako spadaju u klasu nelinearnih dinamičkih sistema, pojedine instance imaju haotično i random ponašanje, osobina koja također može biti od koristi u pojedinim oblastima.

Iz ovih osobina i njihovih kombinacija možemo zaključiti oblasti u kojima su te osobine korisne, te tu pokušati primijeniti celijske automate.

Iz osobina paralelnosti i univerzalnosti izračunavanja, može se pretpostaviti da bi bilo moguće iskoristiti celijske automate u svrhu paralelnog računanja koje sve više postaje paradigma na koju se pokušava prebaciti moderno računarstvo. I dan danas vidimo multicore procesore koji se baziraju na ovom principu, dok bi celijski automati ovaj koncept odveli u novi ekstrem gdje bi svaka ćelija predstavljala zaseban jednostavni procesor u visokoparalelnom računanju. Također, moguće je i implementirati celijskim automatima neke visoko paralelizabilne zadatke poput obrade digitalnih slika gdje bi se za svaki pixel pojedinačno uspostavljao celijski automat koji bi ga mijenjao u zavisnosti od susjednih pixela. Istraživanja u ovoj oblasti već su dala neke rezultate.

Ako pogledamo realne pojave iz raznih oblasti nauke poput sociologije ili fizike niskog nivoa, vidimo da je jedan od osnovnih koncepta koji se pojavljuju lokalnost i lokalna interakcija koja daje globalno prepoznatljivo ponašanje. Upravo ovo čini celijske automate pogodnim za razne vrste simulacija, od kojih će neke biti obradbe i u specifičnim implementacijama primjena.

Ako dodamo mogućnost izmjene, heterogenosti i napredovanja pravila prirodnom selekcijom, tada bi celijski automati mogli biti iskoristeni za primjene na polju vještačke inteligencije. Na samom početku oblasti sam Wolfram je izučavao potencijal celijskih automata u primjenama simulacija neuronskih mreža koje su veoma koristan i napredan koncept vještačke inteligencije danas široko korišten u raznim primjenama.

Također nepredvidivost nekih određenih instanci celijskih automata čini ih korisnim u primjenama gdje je potrebno da je teško moguće pogoditi početno stanje nekog sistema

na osnovu neke od njegovih kasnijih iteracija. Ocita oblast primjene ovog koncepta je kriptografija, gdje je potrebno naci nacin da se iz pocetne konfiguracije (takozvani seed) generise niz naizgled nasumicnih vrijednosti iz cije se statistike ne moze predvidjeti sljedeci clan tog niza. Ovo je korisno za primjene sifriranja informacija pri njihovom slanju kroz nesiguran komunikacioni kanal.

3.2 Primjene elementarnih celijskih automata

3.2.1. Primjena elementarnog pravila 30 u kriptografiji

3.3 Pregled primjena ostalih tipova

A. Pregled koristenog software-a

U ovom poglavlju bice ukratko prezentiran sadrzaj, arhitektura i prezentacija nacina koristena software-a napisanog kao praktican dio ovog rada. Bice pregledani ciljevi software-a te razlog za pisanje, te neke specificnosti poput arhitekture i nekih izbora koji su pravljani pri specifinoj implementaciji logike sistema. Bice prikazan i nacin koristenja sistema i najbitniji feature-i kroz graficki interfejs.

A.1 Namjena i razlog za sistem

Pri izucavanju oblasti celijskih automata, a posebno elementarnih koji su najvećim dijelom obradivana instanca ovih sistema u literaturi i radovima, poseban je naglasak na jednostavnost sistema koji proizvodi kompleksne krajnje rezultate, kako je navedeno u nekoliko navrata u ovom radu. Tako da prilikom implementacije software-a za ovu oblast, nije naglasak na visoko optimizovanom izracunavanju, vec je vecina potreba javlja se prilikom simuliranja sistema iz datog pocetnog stanja prema odredjenim pravilima evolucije, jer su sistemi u razmatranju vec sami po sebi poprilično jednostavni i samim time optimizovani jer se baziraju na jednostavnom lookup-u predefinisanih pravila. Iz ovog razloga pisani software vecinom predstavlja graficke simulacije sistema elementarnih celijskih automata.

Generalna metodologija izucavanja bilo koje vrste sistema nalaze da se prvo izuce specificne instance da bi se dobila neka osnovna ideja o generalnim osobinama koje se prostiru kroz sve njih, te da se kasnije postulati dobijeni ovim postupkom pokusaju dokazati. Ovo smo mogli vidjeti na primjerima klasifikacije automata, ili uocavanju kompleksnosti glidera pravila 110 pri dokazivanju njegove univerzalnosti. Da bi se omogucilo izucavanje specificnih instanci prije zakljucka o generalizaciji, pisani software trebao bi da omoguci izvrsavanje simulacija sa raznim parametrima da bi se mogle uociti pravilnosti u sistemu kroz te specificne instance. Zato bi trebao da korisniku omoguci veliku slobodu izbora, sto bi za slucaj simulacije elementarnih celijskih automata znacilo

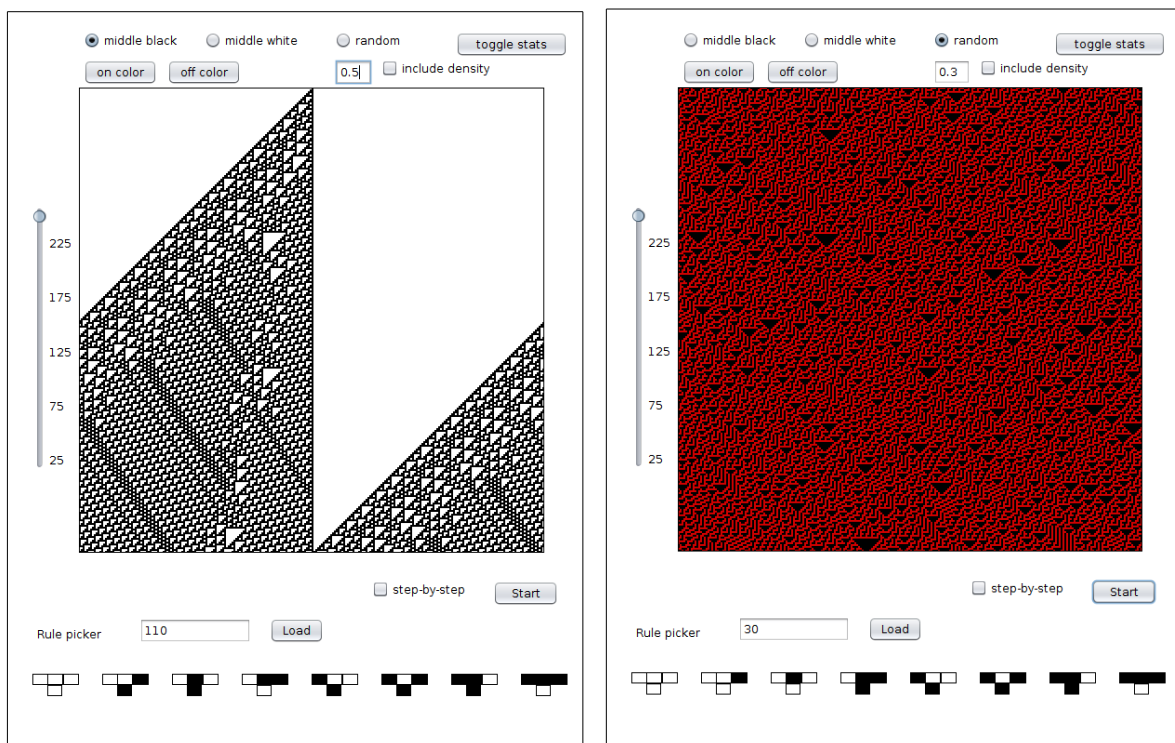
mogućnost izbora početnog stanja, pravila evolucije grafički i numerički, te broj koraka simulacije koje je potrebno izvršiti.

Nadalje, kako smo vidjeli ćelijski automati tesko se obrađuju konkretnim formulama i modelima s obzirom na njihovu nepredvidivu prirodu koja proizilazi iz njihove nelinearnosti, pa je tako statistika jedan od najmoćnijih alata koji može da pomogne. Tako da bi software koji bi vrsio simulaciju trebao da sadrži i neki vid statistike o izvršenoj simulaciji da bi se mogli donijeti neki zaključci.

Takodjer, ovako opisan software imao bi i edukacionu vrijednost gdje bi se mogle na praktičnim primjerima kroz grafičko okruženje pokazati osobine ćelijskih automata koje bi možda zaintrigirale studente za daljnja izučavanja ovakvih sistema.

A.2 Prikaz software-a i ključni featuri

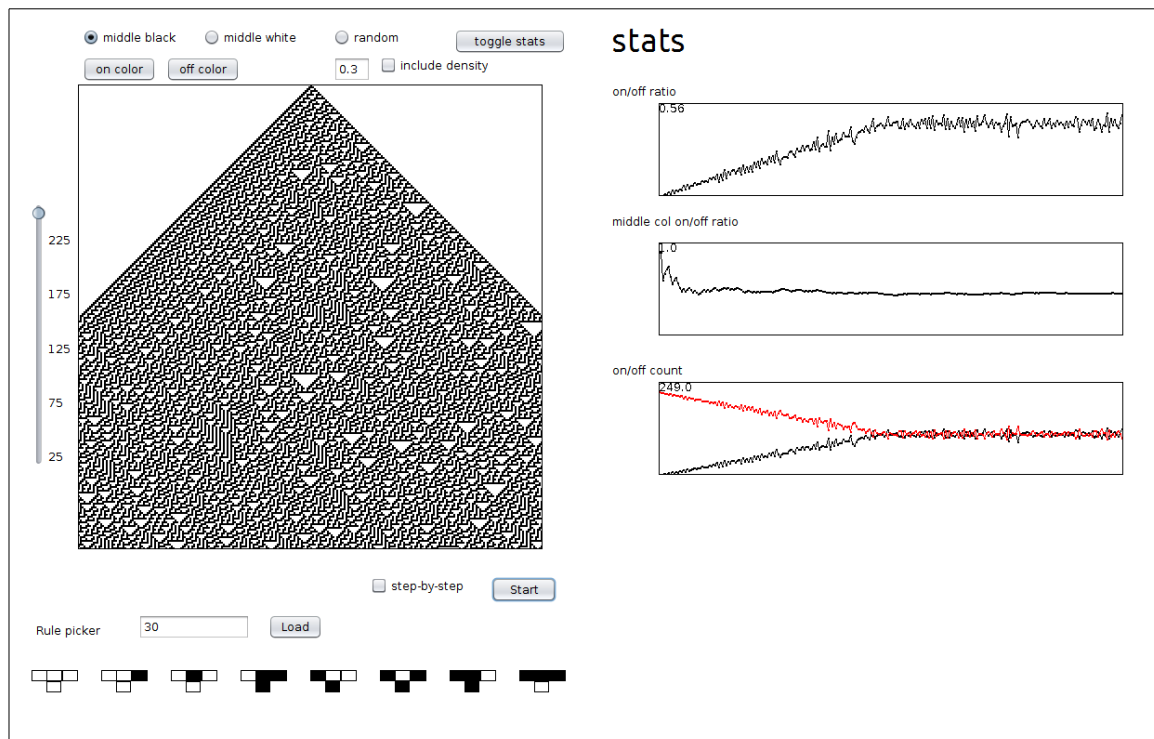
Pokušaćemo kroz grafički interface software-a proći kroz njegove osnovne elemente i slučajeve upotrebe.



Slika 21. Osnovni prozor GUI software-a

Ako pogledamo sliku 21, vidimo osnovni prozor grafičkog interface-a software-a, te je moguće uočiti elemente prozora za prikaz simulacije, prostora za izbor pravila na numerički način čistim ukucavanjem njegovog Wolfram koda, ili izborom prelaza stanja za svaku od mogućih kombinacija susjedstva grafički. Takodjer u gornjem dijelu prozora

moгуće je izabrati način formiranja početne konfiguracije gdje je moguće izabrati srednju on ili off ćeliju, te nasumičnu početnu konfiguraciju koja zadovoljava određenu distribuciju. Također postoje i kontrole koje omogućavaju mijenjanje boja on i off stanja radi boljeg grafičkog prikaza u nekim slučajevima. Vidimo da korisnik ima veliku slobodu izbora parametara simulacije. Na lijevoj strani slike 21 prikazan je software sa defaultnim početnim parametrima, dok na desnoj strani možemo vidjeti kako izgleda simulacija sa izmijenjenim parametrima boje pravila i početne konfiguracije.

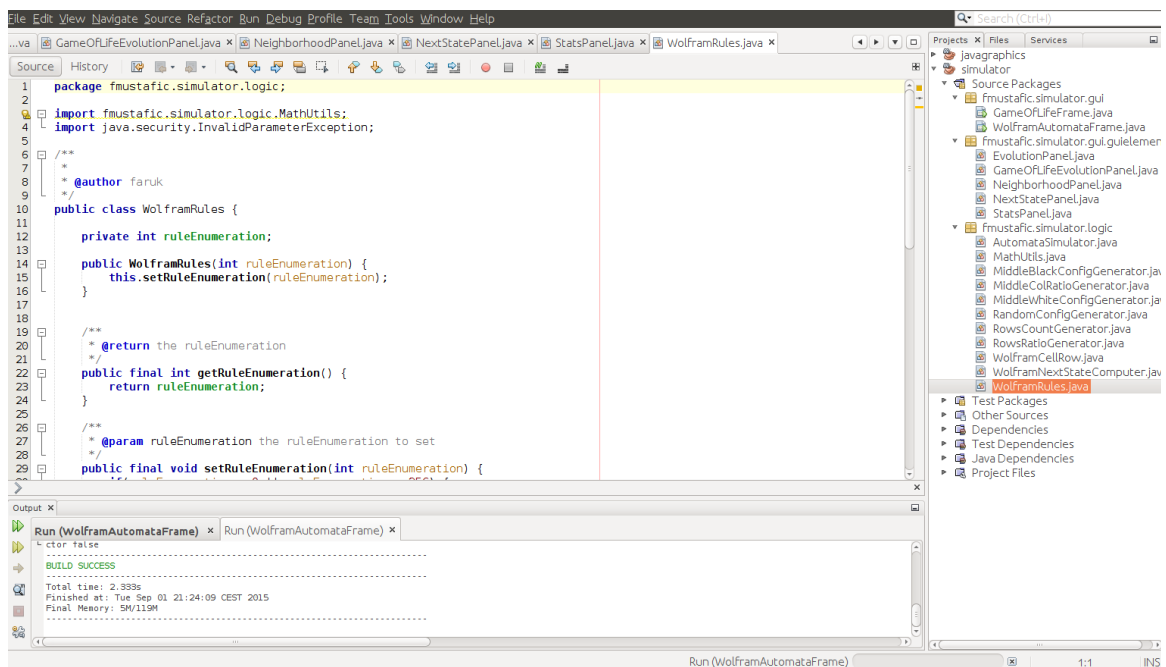


Slika 22. Otvorena statistika

Na slici 22 prikazan je grafički interface software-a sa proširenim prozorom za statistiku koji se otvara/zatvara na odgovarajuće toggle dugme. Na istoj slici također možemo vidjeti i primjer praktičnog korištenja software-a, gdje statistika za Wolframovo pravilo 30 daje naznake da je srednja kolona nasumična s obzirom da broj on i off ćelija u njoj konvergira u odnos 0.5 kako sistem dalje evoluira, što je raspravljeno detaljnije u poglavlju 2 gdje se formalno tretira nasumičnost elementarnih ćelijskih automata.

A.3 Tehnološke implementacione odluke

Sam software pisan je u Java programskom jeziku, Netbeans IDE okruženju, te koristeci standardne swing GUI elemente koji su dio programskog jezika.



Slika 23. Prikaz razvojnog okruzenja

Izbor ove kombinacije tehnologija opravdava se sa nekoliko faktora. Programski jezik Java izabran je iz nekoliko razloga. Kao prvo, smatrano je da open-source kao koncept omogućava ubrzan razvoj software-a, te tako Java kao programski jezik koji je poznat kao jedan od ključnih u ovoj oblasti omogućava dijeljenje koda sa ostatkom open-source zajednice u svrhu njegovog razvijanja. Nadalje, Java kao jezik i tehnologija uz Java virtualnu masinu omogućava cross-platform development, sto je takodjer bio jedan od ciljeva, tako da software moze da se pokrene na mnoštvu operativnih sistema koji podržavaju Java platformu. Java takodjer kao jezik visokog nivoa ima jednu od najrazvijenijih grafičkih biblioteka, sto je omogućilo olaksan razvoj grafičkog interfejsa programa. Takodjer, s obzirom na osobine paralelnosti celijskih automata kao sistema, uzeta je u obzir i Javina podrška paralelizaciji zadataka koja bi mogla biti korisna u ovoj oblasti za ubrzanje i optimizaciju izracunavanja. Nedostatak Java je nemogućnost eksplicitne manipulacije memorijom, medjutim u slucaju ovog software-a to je vise pozitivna nego negativna osobina s obzirom da programer svakako nema potrebe za eksplicitnom alokacijom, dok je garbage collection feature koji takodjer olaksava i ubrjava proces razvoja s obzirom da se ne mora paziti na detalje niskog nivoa.

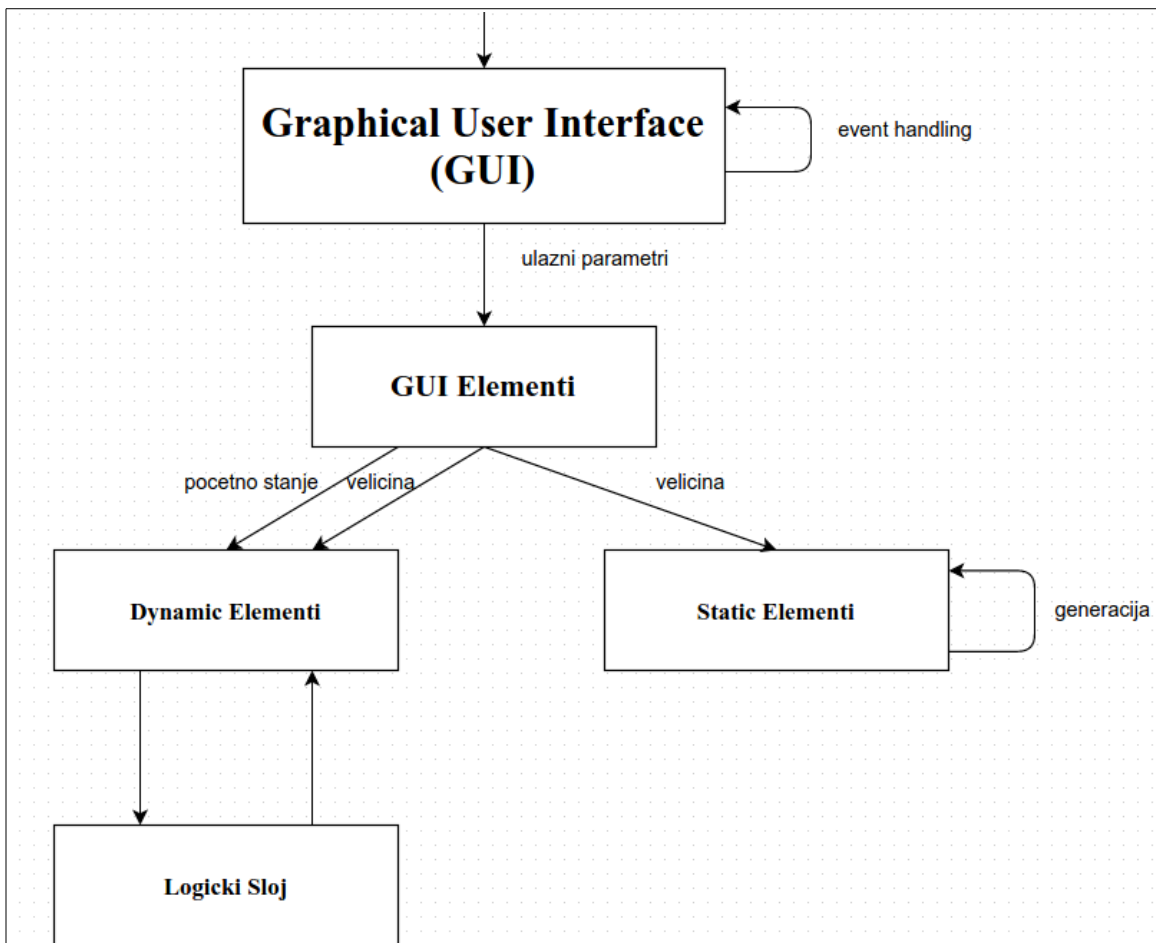
Netbeans razvojno okruzenje prikazano na slici 23 izabrano je iz razloga jednostavnosti integrisanog GUI dizajnera, te vece stabilnosti na platformi Linux od najveceg konkurenta Eclipse IDE-a.

Mozemo napomenuti da je razvoj vrsen na Linux operativnom sistemu, ali to je sustinski nebitno s obzirom na portabilnost i cross-platform osobine Java programskog jezika.

A.4 Arhitektura software-a

A.4.1 Pregled generalne arhitekture

Potrebno je izvršiti i pregled arhitekture software-a da bi se detaljnije shvatile specifičnosti istog, s obzirom da je uvijek dobro imati neku vrstu pregleda visokog nivoa. Kroz ovaj pregled, u ovom slučaju generalna arhitektura sistema, daje mogućnost stvaranja slike sistema na visem nivou, gdje se pojedini dijelovi posebno obrađuju i detaljnije popunjavaju implementacionim specifičnostima.



Slika 24. Pregled arhitekture sistema najviseg nivoa

Na slici 24 prikazana je pomenuta arhitektura. Vidimo da na najvisem nivou imamo graficki korisnicki interfejs (eng. GUI – Graphical User Interface), koji je jedini sloj koji direktno dolazi u kontakt s korisnikom. Tacnije, kako je vidjeno u pregledu feature software-a, korisnik daje ulazne parametre programu kroz upravo ovaj interfejs. Pozitivna osobina grafickih interfejsa je to sto se lako moze kontrolisati korisnicki ulaz kroz graficke elemente, pa je tako na primjer velicina automata koji se simulira kontrolisana sliderom koji ima definisane minimume i maksimume, sto uklanja potrebu za

eksplicitnom validacijom ulaza.

Na ovom sloju grafickog interfejsa, sva logika se svodi na procesiranje takozvanih eventa – događaja na grafickom interfejsu, te poziva odgovarajucih nizih slojeva u zavisnosti od akcije zahtijevane događajem. Ovo je prikazano na dijagramu sa desne strane, gdje cemo uspostaviti konvenciju da strelice oznacavaju prenos informacija, dok strelice koje se vraćaju u objekat definisu vrstu logickog procesiranja samog sloja/elementa. Događaji su generisani od strane korisnika njegovom interakcijom sa grafickim elementima.

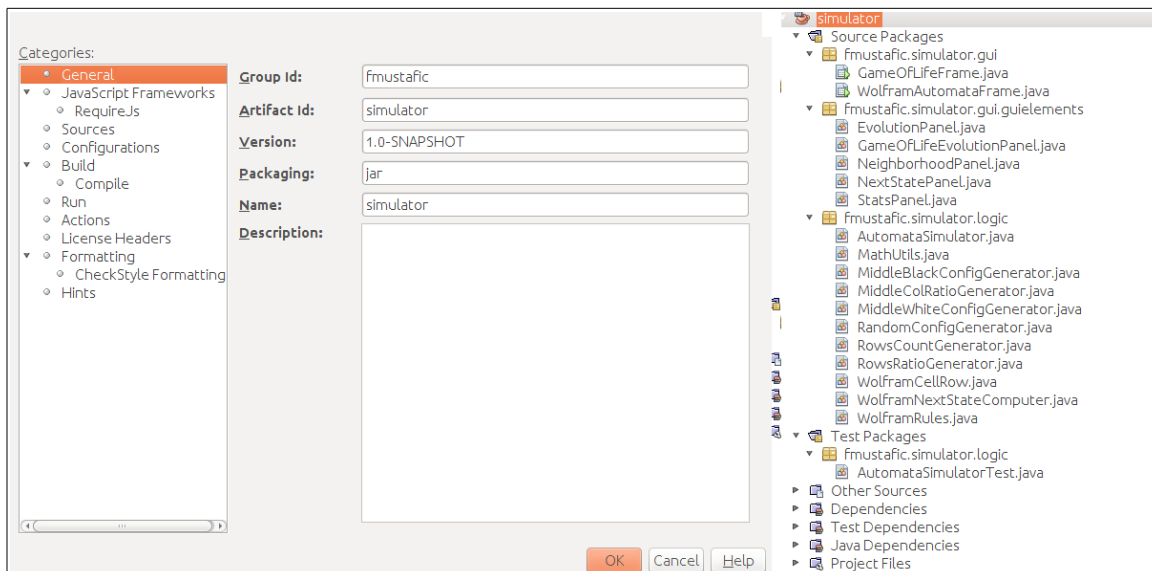
Graficki interfejs sastoji se od standardnih grafickih elemenata poput dugmadi i slidera, medjutim javlja se potreba i za specifinim custom elementima kao sto je u nasem slucaju panel za prikaz evolucije celijskih automata. Upravo to je iduci sloj arhitekture na slici prikazan kao GUI Elementi blok. Na ovom sloju postoje specifine implementacije grafickih elemenata koji nisu dio standardnog toolkita nekog jezika, te se potreba za njima javlja pretežno iz domene problema koji se razmatra. Na ovom sloju mozemo uociti potrebu za dvije vrste elemenata – staticni i dinamicni (takodjer prikazano na slici 24).

Staticki elementi su takvi da sam njihov izgled i raspored elemenata unutar njih ne zavisi od ulaznih parametara, te je jedina varijabla koja je bitna za njih njihova velicina. Logika unutar ovih elemenata svodi se na jednostavno procesiranje velicine i zavisnosti pozicije dijelova elementa u zavisnosti od velicine.

Dinamicni elementi su nesto kompleksniji, te logika za njihovo generisanje nije toliko jednostavna i zavisi pretežno od domena problema koji se razmatra – celijski automati u nasem slucaju. Tako da je kao prvo potrebno dati odgovarajuce informacije o pocetnim uslovima od kojih zavisi izgled ovih elemenata, te je potrebno omoguciti jos neki sloj koji bi sluzio za procesiranje i generisanje cjelokupnog izgleda, s obzirom da bi bila losa praksa “natrpati” svo ovo procesiranje u graficke elemente. Iz ovog razloga za dinamicne elemente imamo jos dodatni sloj arhitekture, koji predstavlja logiku generisanja strukture istih, te odvaja graficki od logickog dijela aplikacije sto omogucava razna implementaciona razmatranja za logicki sloj specifino.

A.4.2 Specifna implementacija arhitekture

Obradjena arhitektura implementirana je specifino prema implementacionim odlukama u proslom poglavlju, te ce ovdje biti dat kratak pregled ove specifine implementacije za slucaj simulacije elementarnih celijskih automata sa raznim parametrima.



Slika 25. Specifična implementacija arhitekture

Na slici 25 prikazana je ova specifična implementacija. Na lijevoj strani prikazane su osnovne informacije o projektu unutar NetBeans okruznja, dok su sa desne strane dati source-code file-ovi koji predstavljaju implementaciju.

Java paketi predstavljaju način grupisanja logički povezanih file-ova, te ako pogledamo desnu stranu slike 25, vidimo da postoje tri paketa od kojih svaki ugrubo odgovara sloju pomenute arhitekture.

U GUI sloju koji odgovara `fmustafic.simulator.gui` paketu u projektu nalaze se Frame klase koje predstavljaju ugrađene Java containere za prikaz ostalih elemenata. Također na njima se nalaze osnovni GUI elementi poput buttona, te logika u ovim file-ovima predstavlja većinski event handling.

U sloju grafičkih elemenata, potrebno je bilo custom definisati nekoliko istih. Vidimo da paket `fmustafic.simulator.gui.elements` sadrži elemente potrebne za prikaz evolucije automata, grafički izbor pravila evolucije te prikaz statistike koji se nalaze u source code fileovima `EvolutionPanel`, `NeighborhoodPanel` i `StatsPanel` respektivno. Svaka od ovih klasa predstavlja Panel tip koji također predstavlja vid containerskog elementa u Javi.

Na kraju paket `fmustafic.simulator.logic` koji odgovara najnižem sloju arhitekture sadrži klase odgovorne za logičko procesiranje evolucije ćelijskih automata, te generisanje slučajnih početnih uslova za elementarni ćelijski automat.

A.4.3. Implementacione odluke po slojevima

Razmotrimo još i specifične odluke u konkretnoj implementaciji koje su se morale donijeti po slojevima arhitekture.

Na najvisem sloju arhitekture, odabrana je javax.swing biblioteka za konstrukciju i implementaciju grafickih elemenata.

Dalje, na sloju ispod za generisanje specijalnih grafickih elemenata iz problem domena, koristen je Java Graphics objekat koji omogucava pixelwise manipulaciju elementa, te postavljanje svakog pixela elementa na zeljeno stanje pojedinačno. Na prva dva sloja nije bilo previse razmatranja s obzirom da postoji poprilično ustaljena procedura za kreiranje grafickog interfejsa.

Na sloju logike, s obzirom na visok nivo paralelizma u razmatranim sistemima, bilo je razmotrena i opcija implementacije preko Java podrške paralelizmu, te ispod mozemo vidjet snippet koda koji prikazuje slucaj upotrebe iste.

```
...
WolframNextStateComputer right =
    new WolframNextStateComputer(
        this.currentState,
        this.from + (this.to - this.from) / 2,
        this.to,
        this.rule
    );

left.fork();
Boolean[] rightResult = right.compute();
Boolean[] leftResult = left.join();

Boolean[] nextState = new Boolean[leftResult.length + rightResult.length];

System.arraycopy(leftResult, 0, nextState, 0, leftResult.length);

for(int i = leftResult.length; i < nextState.length; i++) {
    nextState[i] = rightResult[i - leftResult.length];
}

return nextState;
...
```


Reference

- [1] Schiff, Intro CA
- [2] Wolfram, A New Kind of Science
- [3] Cellular Automata »Wolfram, S. Los Alamos Science 9 (1983): 2–21.
- [4] Universality and Complexity in Cellular Automata »Wolfram, S. Physica D: Nonlinear Phenomena 10, no. 1–2 (1984): 1–35.
- [5] Cellular Automaton Supercomputing »Wolfram, S. In High-Speed Computing: Scientific Applications and Algorithm Design [Wilhelmson, R. B. (Ed.)]. University of Illinois Press, 40–48, 1988.
- [6] Statistical Mechanics of Cellular Automata »Wolfram, S. Reviews of Modern Physics 55, no. 3 (1983): 601–644
- [7] Cryptography with Cellular Automata »Wolfram, S. In "Advances in Cryptology: CRYPTO '85 Proceedings" [Williams, H. C. (Ed.)]. Lecture Notes in Computer Science 218. Springer-Verlag, 429–432, 1986.
- [8] Cook, M. "Universality in Elementary Cellular Automata." Complex Systems 15, 1–40, 2004.
- [9] Christopher G. Langton (1990). "Computation at the edge of chaos"
- [10] Entropy and Information Theory First Edition, Corrected Robert M. Gray
- [11] Dynamics, Computation, and the "Edge of Chaos": A Re-Examination Melanie Mitchell¹, James P. Crutchfield², and Peter T. Hraber¹, 1994
- [12] Random Sequence Generation by Cellular Automata »Wolfram, S. Advances in Applied Mathematics 7, no. 2 (1986): 123–169.
- [13] Cryptography with Cellular Automata »Wolfram, S. In "Advances in Cryptology: CRYPTO '85 Proceedings" [Williams, H. C. (Ed.)]. Lecture Notes in Computer Science 218. Springer-Verlag, 429–432, 1986.

- [14] Advances in Cryptology — EUROCRYPT '91 Volume 547 of the series Lecture Notes in Computer Science pp 186-199 Analysis of Pseudo Random Sequences Generated by Cellular Automata Willi Meier Othmar Staffelbach
- [15] Universality in Elementary Cellular Automata Matthew Cook Department of Computation and Neural Systems, Caltech, Mail Stop 136-93, Pasadena, California 91125, USA
- [16] A Concrete View of Rule 110 Computation, Matthew Cook, arxiv.org, 2009