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 najveci 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. Numercija podpoglavlja ide do dva nivoa dubine.

Lista figura sa opisima

Slika 1. *Celija (plavo) sa susjdestvom od 8 celija (crvene celije)*

Slika 2. *Dvodimenzionalni grid "Igre zivota" sa pocetnom konfiguracijom [preuzeto sa http://dana.loria.fr/doc/game-of-life.html]. Uocljive su pravilne strukture u konfiguraciji.*

Slika 3. *Tipovi dvodimenzionalnih susjdestva. Moore-ovo (lijevo) i von Neumannovo susjedstvo (desno). Oba imaju primjene u zavisnosti od domena problema.*

Slika 4. *Graficki izbor pravila (prelaza stanja) u zavisnosti od stanja susjedstva jednodimenzionalnog automata sa dva moguca stanja – on/off (crno/bijelo na slici). [generisano sofverom pisanim u sklopu rada]*

Slika 5. *Deset koraka evolucije specificnih pocetnih uslova 10x10 grida "Igre zivota" [preuzeto sa https://datasciencelab.wordpress.com/2014/01/03/yet-another-game-of-life/]*

Slike 6. i 7. *Kompleksne strukture generisane jednostavnim pravilima [generisano sofverom pisanim u sklopu rada]*

Slika 8. *Neki od prvih simulacija sistema celijskih automata od strane Stephena Wolframa. [2]*

Slika P1. *Dva primjera kompleksnih struktura. Prirodno generisana oklopa struktura skoljke (lijevo) i elementarno pravilo 30 (desno). Uocljiva je slicnost izmedju ova dva naizgled nepovezana sistema.*

Slika P2. *Struktura generisana pravilom 110 (lijevo) [generisano softverom] i "Igra zivota" (desno) [preuzeto sa http://www.marekfiser.com/Projects/Conways-Game-of-Life-on-GPU-using-CUDA]*

Slika P3. *Generisanje kompleksne strukture snjeznih pahuljica pomocu celijskih automata [preuzeto sa http://radicalart.info/AlgorithmicArt/grid/cellular/2D/]*

Slika 9. *Primjer strukture resetke [preuzeto sa http://mathworld.wolfram.com/CircleLatticePoints.html]*

Slika 10. *Primjer proizvoljne pocetne konfiguracije jednodimenzionalnog celijskog automata sa dva stanja [generisano softverom]*

Slika 11. *Pet koraka evolucije pocetnih uslova jednodimenzionalnog celijskog automata sa dva stanja. Specificno pravilo evolucije primijenjeno je pravilo 30 [generisano softverom]*

Slika 12. *Graficki izbor pravila elementarnog celijskog automata. [generisano softverom]*

Slika 13. *Prikaz vise koraka evolucije pravila 110 i 30. Pravila su graficki prikazana na vrhu slike. Uocljive su kompleksne strukture. [generisano softverom]*

Slika 14. *Evolucija pravila 249, 164, 146, 110 respektivno koja se nalaze u postuliranim klasama I, II, III i IV. [generisano softverom]*

Slika 15. *Graficki prikaz prelaza stanja pravila 249. [generisano softverom]*

Slika 16. *Prosjecno vrijeme stabilizacije automata ponasanja u zavisnosti od Langtonovog parametra. [9]*

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

Slika 18. *Wolframove intuitivne klase ponasanja celijskih automata u zavisnosti od Langtonovog parametra. [9]*

Slika 19.a *Naizgled slucajne strukture generisane pravilima 30 (prvi red) i 45 (drugi red). [generisano softverom]*

Slika 20. *Neki od glideri pravila 110. [15]*

Slika 21. *Osnovni graficki interfejs softvera pisanog u sklopu rada.*

Slika 22. *Prosireni graficki interfejs softvera pisanog u sklopu rada sa prikazanim dijelom za generisanje statistike.*

Slika 23. *Prikaz razvojnog okruzenja u kojem je vrsen razvoj softvera.*

Slika 24. *Pregled arhitekture sistema najviseg nivoa.*

Slika 25. *Specificna implementacija arhitekture.*

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.

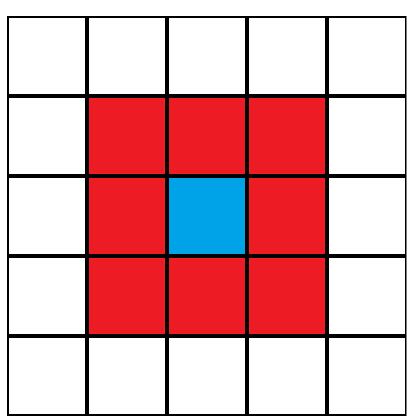
Nakon cemo pregledati histroijski koja matematicka pitanja su navela da se razmotre neke specijalne klase, te ce se nako toga obratiti paznja na unificarnje specijalni klasa, te posebno historijski pregled rada Stephena Wolframa za kojeg se moze reci da je dao jedna od najvecih doprinosa samom polju.

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

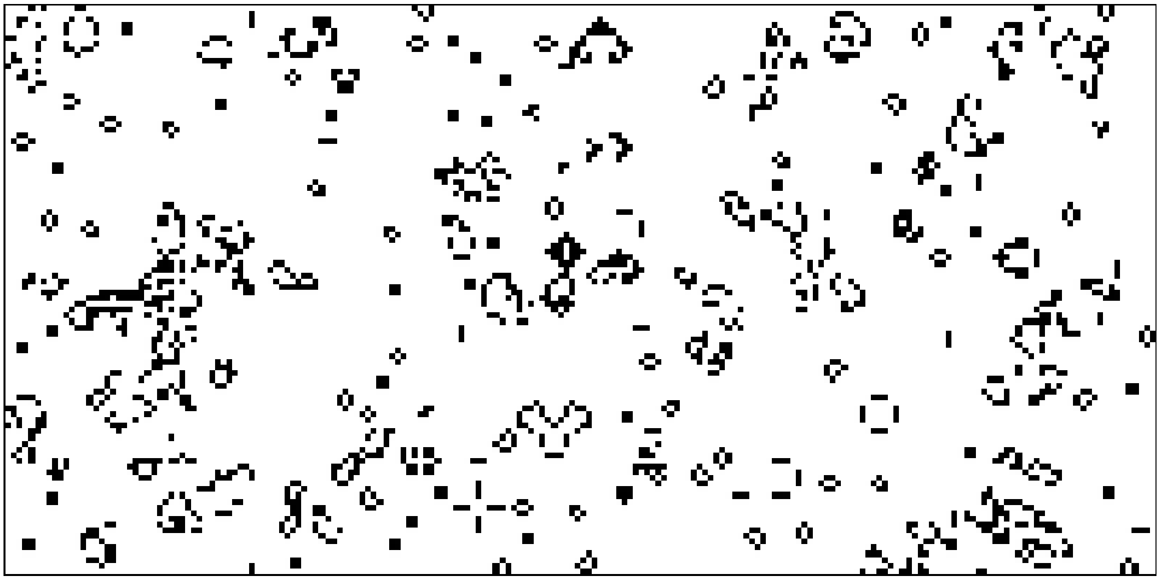
Pocnimo prvo od pokusaja shvatanja kakve vrste modela predstavljaju celijski automati. Zato cemo prvo krenuti od konkretnog specificnog primjera kroz koji je moguce shvatiti osnovne odlike celijskih automata na koje cemo se kasnije nadograditi kako budemo gradili formalnu apstrakciju ovog konkretnog 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 pretezno 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 sluzi samo za intuitivni prikaz osnovnih ideja iza ovakvih vrsta modela.



Slika 1. *Celija (plavo) sa susjdestvom od 8 celija (crvene celije)*

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.

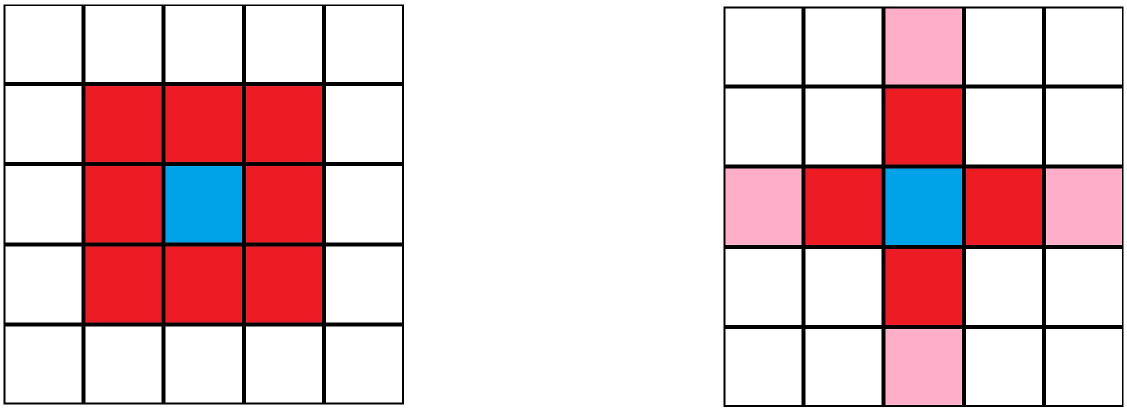


Slika 2. *Dvodimenzionalni grid "Igre zivota" sa pocetnom konfiguracijom [preuzeto sa http://dana.loria.fr/doc/game-of-life.html]. Uocljive su pravilne strukture u konfiguraciji.*

Na slici 2 prikazan je skupe celija koje zajedno cine 2d *grid*. Primijetimo da je nemoguce 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 ce ta tematika biti detaljnije kasnije obradjena.

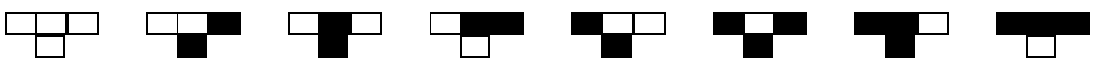
Ovakav raspored nazvacemo *konfiguracija*. U konfiguraciji, svaka celija ima svoje pocetno stanje, pa je tako za svaku celiju definisano da li je ona inicijalno on ili off – crna ili bijela. Stanje je na pocetku izabrano proizvoljno i moze se mijenjati u zavisnosti od potreba, te ce razlicita pocetna stanja dati nekada i drasticno razlicita ponasanja. Na slici 2 prikazan je primjer jednog takvog pocetnog stanja. Skup ovako organizovanih celija sa svojim pocetnim stanjima u konacnom gridu nazivamo *inicijalna konfiguracija grida*.

Naravno, dosadasnja definicija celijski automata ne bi imala nikakvog smisla, s obzirom da imamo samo pocetnu konfiguraciju i grid. Medjutim, ono sto cini celijske automate pravim modelima koji se mogu koristiti u razne svrhe je takozvana *evolucija celijskih automata*. Nakon sto se uspostavi inicijalna konfiguracija grida, ovaj sistem moze da se stavi u evoluciju. To znaci da ce svaka od celija da mijenja svoje stanje prema nekim pravilima, te ce cijelokupan sistem da se mijenja prema tim pravilima u diskretnim vremenskim intervalima.



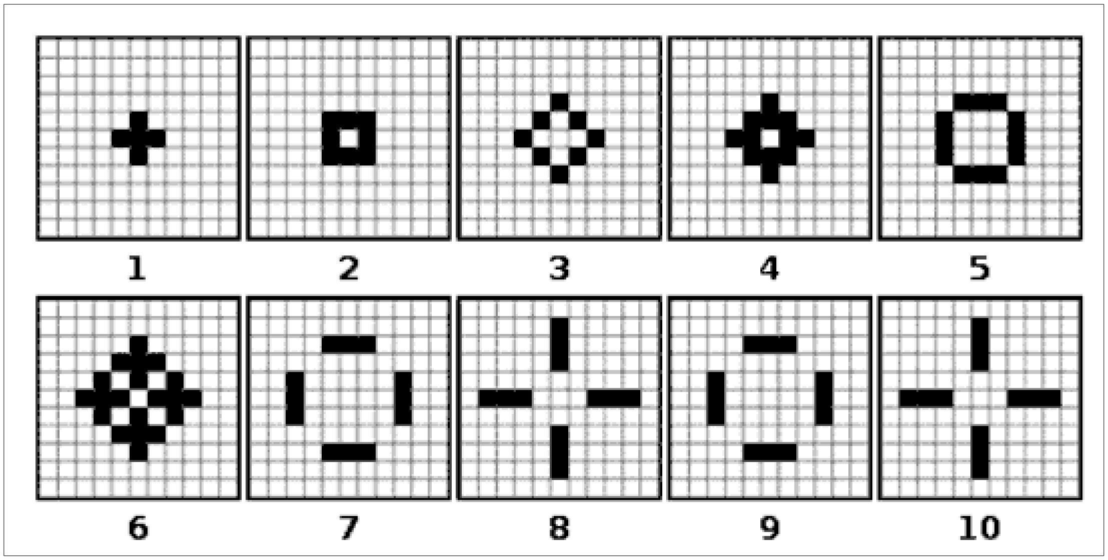
Slika 3. *Tipovi dvodimenzionalnih susjdestva. Moore-ovo (lijevo) i von Neumannovo susjedstvo (desno). Oba imaju primjene u zavisnosti od domena problema.*

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



Slika 4. *Graficki izbor pravila (prelaza stanja) u zavisnosti od stanja susjedstva jednodimenzionalnog automata sa dva moguca stanja – on/off (crno/bijelo na slici). [generisano sofverom pisanim u sklopu rada]*

Nakon sto se izabere koje celije ucestvuju u formiranju susjsedsva, formira se i skup pravila koji govori o tome kako celija evoluira na osnovu svog stanja i stanja svojih susjeda. Na slici 4 je u gornjem dijelu naveden i nacin specifikacije pravila za jednodimenzionalno sujedstvo (jer bi broj mogucnosti za dvodimenzionalno bio ogroman) gdje se za svaku pojedinacnu kombinaciju susjedstva na osnovu trenutnog stanja celija bira iduce stanje iste. Tako sa slike mozemo uociti npr. ukoliko je celija okruzena sa dvije crne celije, prelazi u off.

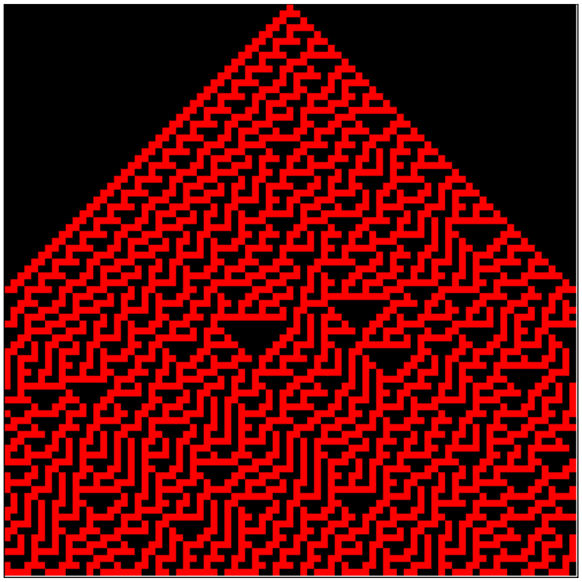


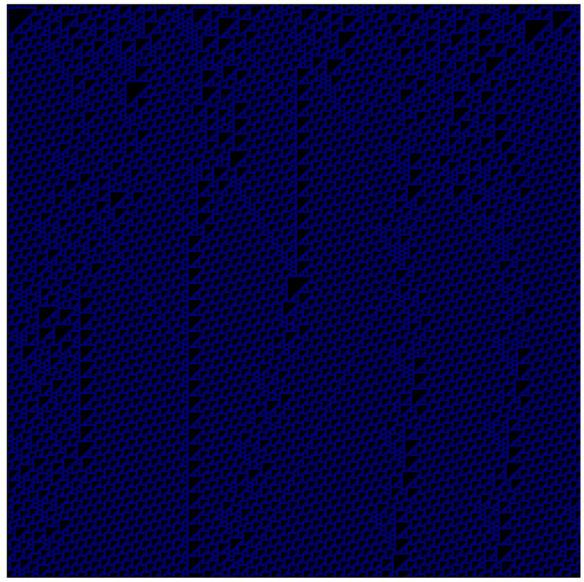
Slika 5. *Deset koraka evolucije specificnih pocetnih uslova 10x10 grida "Igre zivota" [preuzeto sa https://datasciencelab.wordpress.com/2014/01/03/yet-another-game-of-life/]*

Kako su stanja binarna, ukoliko n predstavlja broj susjeda koji formiraju susjedstvo, tada je moguce $2^{2^{n+1}}$ mogucih pravila evolucije. Tako za jednostavno Moore-ovo susjedstvood 8 susjednih celija postoji $2^{2^{8+1}} = 2^{2^9} = 1.34 \times 10^{154}$ mogucih pravila evolucije. Ovo opazanje ce biti kasnije detaljnije objasnjeno.

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

Vidimo da opisana cisto lokalna interakcija moze da kreira poprilicno 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. *Kompleksne strukture generisane jednostavnim pravilima [generisano sofverom pisanim u sklopu rada]*

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

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 Laboratorji *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 moguce 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 koristeci iste. Kreirali su model koji *predvidja kretanje fluida* na nacin da smatraju fluid sastavljenim od diskretnih jedinica – celijskih automata, cije kretanje zavisi od susjednih jedinica. Na ovaj nacin moguce je aproksimirati kretanje cjelokupnog fluida modeliranjem samo lokalne interakcije susjednih cestica. Ovim modelom pokazano je da celijski automati imaju i siru primjenu van cisto teoretskih razmatranja za koja su ranije koristeni, te ovo predstavlja svojevrsni pocetak generalne primjene ove vrste modela u nauci.

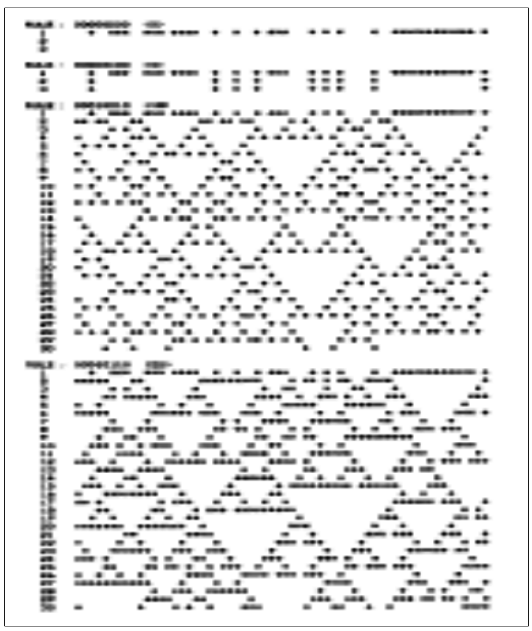
Trebalo bi napomenuti i da je neke od ranih istrazivanja u ovom polju vrsio i pionir u oblasti vjestackog zivota, norvesko-italijanski naucnik *Nils Aall Baricelli* koji je jedan od prvih prepoznao potencijalnu univerzalnost celijskih automata kao modela koji mogu predstavljati realne pojave.

Jos neku od ranih primjena celijski automati nasli su u modeliranju *propagacije talasa u medijima*. Rani ovakav model celijskih automata konstruisan je 1940-ih. Medjutim kako je taj model koristio kontinualnu funkciju kao signal, ne moze se smatrati diskretnim modelom celijskih automata, tako da su prvi pravi ovakav model koristen u svrhe modeliranja impulsa kardio sistema u covjecijem tijelu konstrukisali *J. M. Greenberg* i *S. P. Hastings* 1978-e. Ovaj model je i dalje cesto koristen i referenciran u istrazivackim radovima.

Prvih dvadeset godina od pojavljivanja modela celijskih automata, gotovo niko nije izvrsavao rigorozno naucno i matematicko ispitivanje osobina ovih sistema. Jedan od pionirskih radova u polju bio je rad americkog matematicara *Gustava A. Hedlunda*, koji je kroz matematicku oblastu dinamika simbola (koju je sam i osnovao, eng. *symbolic dynamics*) posmatrao celijske automate kao mijenjajuce nizove simbola uz odredjena pravila prelaza. Ovime je dosao do nekih od najkorisnih rezultata u ovom polju. Njegov rad iz 1969-e zajedno sa Curtis-Hedlund-Lyndon teoremom za koju snosi djelomicne 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 dozivjela pametno konstruisanim primjerom od strane britanskog matematicara i fizicara *John Conway*-a 1970-e godine. On je svoj specifican model baziran na dvodimenzionalnim celijskim automatima prikladno nazvao *Igra zivota* (eng. *Game of Life*). Ovaj model je standardni dvodimenzionalni sistem celijskih automata sa dva stanja, medjutim uprkos jednostavnim pravilima, nakon sto se sistem pusti u rad, pocinju da se pojavljuju visoko kompleksne strukture koje ispoljavaju osobine koje bi mogli pripisati i nekim zivim bicima, krecu se, jedu jedni druge, razmnozavaju se i slicno – zbog ovoga je model i dobio svoje ime. Upravo zbog ovih osobina gdje se visoka kompleksnost javlja iz poprilicno jednostavnih pravila, Igra a samim tim i oblast celijskih automata dozivjela je veliku popularizaciju, te vecina ljudi za celijske automate sazna upravo iz ovog primjera. Iako se ovaj model smatrao pretezno dijelom rekreativne matematike te sredstvom popularizacije ideje celijskih automata za siru javnost, nesto kasnije je Berlekamp u saradnji sa jos nekoliko matematicara dokazao univerzalnost Igre zivota, tj. da sistem moze ekvivalentno da se koristi u svrhu univerzalne kompjutacije kao i bilo koji drugi racunar 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 moguce 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. *Neki od prvih simulacija sistema celijskih automata od strane Stephena Wolframa. [2]*

Najdetalnjije i rigoroznije ispitivanje osobina celijskih automata izvrsio 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 pretezno 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 pronalaske, 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 koristeci 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 izvrsavanje 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 moguce 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 pretezno lokalnu interakciju poprilicno opravdano. Na primjer, prilikom razmtranja 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 pretezno cisto lokalnoj interakciji, sistemi ispoljavaju poprilicno 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].

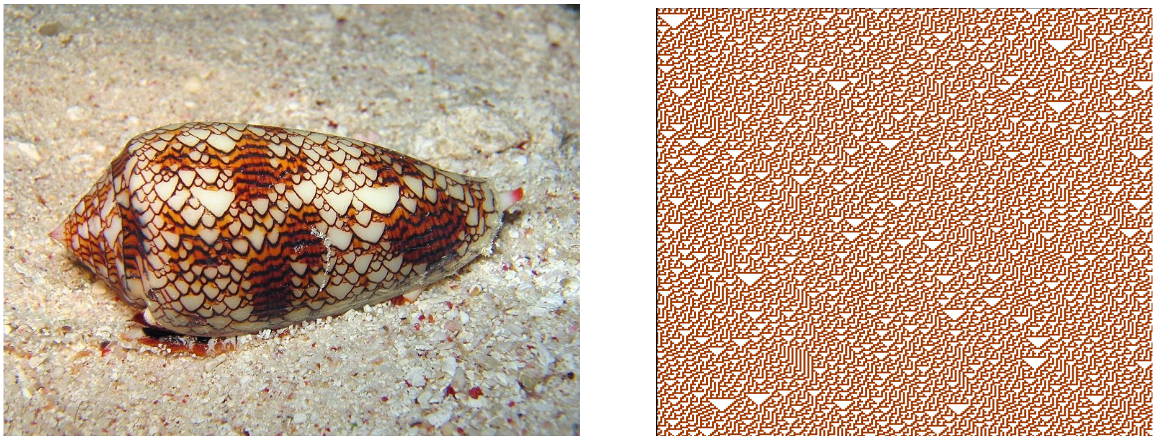
Enormna kompleksnost nedvojbeno je dio sistema koji se uzimaju u naucna razmatranja i vecine naseg okruzenja. Ovo ide toliko daleko da su neke kompleksnosti i dalje ostale nerazjasnjene i uz koristenje najmodernijih tehnika i metoda naucnih disciplina. Svakodnevno se susrecu kompleksne strukture koje 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 skoljkama [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 moguce, iako kontraituitivno, da poprilicno jednostavna pravila mogu da proizvedu ogromnu slozenost koja se moze 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 moguce da ogromnu kompleksnost koju posmatramo pokusavamo svesti na jednostavna pravila koja ce je opisati i iz kojih ova kompleksnost moze 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 moze 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 odredjenih 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 poprilicno 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 moze da nas pribilizi odgovoru poveznice izmedju nastanka kompleksnosti iz malobrojnog skupa jednostavnih pravila.

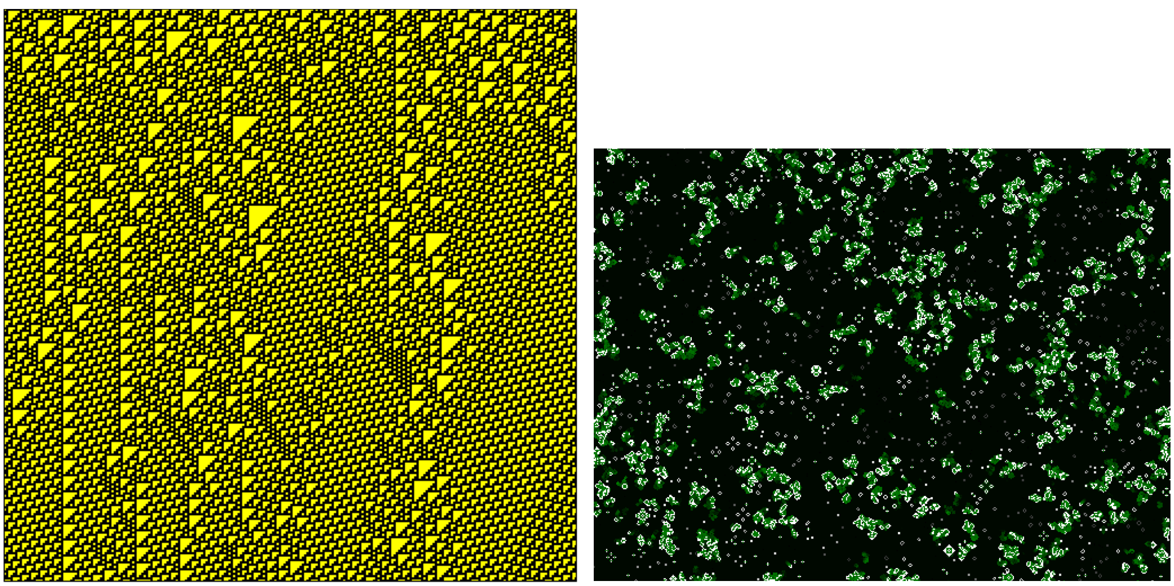
1.4 Primjeri

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 moze dati jasniju sliku i nekome ko nije toliko zainteresovan u matematicke detalje samog modela na nacin da moze prikazati raznovrstnost struktura koje celijski automati mogu proizvesti.



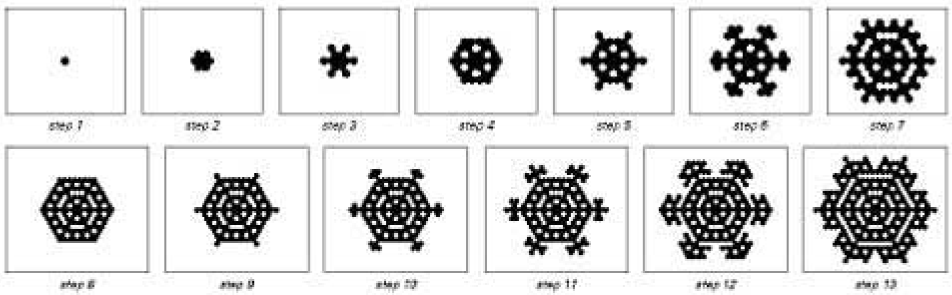
Slika P1. *Dva primjera kompleksnih struktura. Prirodno generisana oklopa struktura skoljke (lijevo) i elementarno pravilo 30 (desno). Uocljiva je slicnost izmedju ova dva naizgled nepovezana sistema.*

Na slici P1 mozemo uociti slicnost izmedju strukture morske skoljke generisane prirodnim putem i evolucijom elementarnog pravila 30. Smatralo se da kompleksne strukture uzoraka na zivotinjskim ljusturama moraju nastati kao rezultat nekog kompleksnog procesa, medjutim ako pogledamo strukture generisane jednostavnim pravilom elementarnih celijskih autoamta, mozemo vidjeti da i najjednsotavnija pravila mogu da proizvedu strukture velike kompleknosti. Zbog ovoga je oblast celijskih autoamta jedna od zanimljivijih s obzirom da povezuje naizgled nespojive cjeline lokalne jednostavnosti i globalne kompleksnosti. [2]



Slika P2. *Struktura generisana pravilom 110 (lijevo) [generisano softverom] i "Igra zivota" (desno) [preuzeto sa http://www.marekfiser.com/Projects/Conways-Game-of-Life-on-GPU-using-CUDA]*

Na slici P2 s lijeve strane mozemo vidjeti strukture generisane elementarnim pravilom 110 za koje je postulirano i kasnije dokazano da je kompleksno kao i bilo koji sistem univerzalnog izracunavanja. S desne strane mozemo vidjeti jednu konfiguraciju Conwayove Igre zivota simulirane na CUDA grafickoj kartici sa milionima individualnih celija koje izgledaju kao da formiraju clustere i nesto kao zivuce strukture.



Slika P3. *Generisanje kompleksne strukture snjeznih pahuljica pomocu celijskih automata [preuzeto sa http://radicalart.info/AlgorithmicArt/grid/cellular/2D/]*

Formiranje kristala snjeznih pahuljica nije jos dovoljno izucena oblast s obzirom da nemamo rjesenja pojedinih diferencijalnih jednacina propagacije toplote, medjutim pokazuje se da jednostavni dvodimenzionalni model celijskih autoamata moze da generise strukture koje poprilicno oslikavaju izglede pravih pahuljica bez potrebe da rijese te jednacine kao sto se moze vidjeti na slici P3.

2. Formalni matematicki tretman

Nakon neformalnog uvoda, potrebno je i rigorozno matematicki definisati sta se misli kada se koristi pojam celijskih automata. Zbog raznovrstnosti modela koji se mogu smatrati celijskim automatima, u literaturi ne postoji opsteprihvacena generalna definicija, vec autori strogo matematicki definisu samo konkretan model kojim se bave, npr. jednodimenzinalni binarni celijski automati u slucaju Stephena Wolframa (vidjeti npr. [2], [3], [4]).

Na pocetku bice data generalna definicija koja pokusava sto opstije da pokrije sve diskretne modele koji se mogu smatrati celijskim automatima, mada su moguce neke iznimke zbog sirine diskretnih modela i mogucnosti prosirenja 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 desavanaj 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
- $\delta : (Q \setminus F) \times \Gamma \to Q \times \Gamma \times \{L, R\}$ 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 sposobonostima izracunavanja nekog sistema.

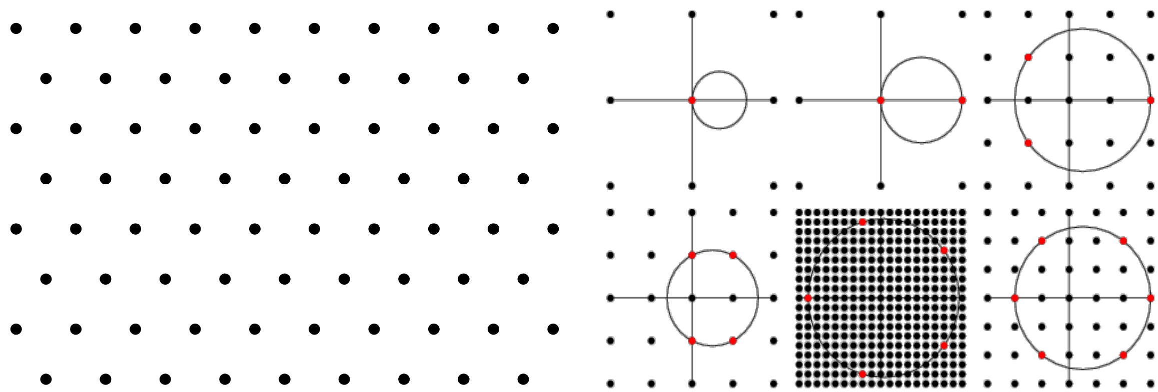
2.1 Generalna definicija celijskih automata

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. *Primjer strukture resetke [preuzeto sa http://mathworld.wolfram.com/CircleLatticePoints.html]*

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 odredjenih vektora u prostoru gdje su tacke definisane (pretezno se radi sa skupom/prostorom R^n). Kljucan detalj je cjelobrojnost jer time zadrzavamo osobinu diskretnosti samog skupa, sto je kljucno za celijske automate kao diskretne modele kompjutacije.

Definicija 2.1.1 Latica ili resetka se definise kao $\Lambda := \left\{ \sum_i \mathrm{a}_i \ v_i \ : \ a_i \in \mathbb{Z} \right\}$ gdje je $v_i \in \{v_1, v_2, v_3, ...\}$ baza vektorskog prostora V nad skupom skupa gdje je resetka definisana.

Ovo samo znaci da izaberemo fiksan skup vektora udaljenosti od tacke i na tacku nadodjemo cjelobrojne umnoske vektora da bismo tako odrzali 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 moci definisati i diskretan koncept vremena u kojem automati evoluiraju kao dodatnu dimenziju pored prostorne koju smo vec obradili.

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

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

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

Nakon sto smo definisali prostorne i vremenske dimenzije u kojima ce celijski automati biti smjesteni, jos jedan osnovni koncept je stanje svakog automata koji se nalazi u prostor-vremenu.

Za jednodimenzionalni i dvodimenzionalni slucaj imali smo samo dva moguca stanja – 1 ili 0, crno ili bijelo u grafickoj reprezentaciji. Na osnovu ovoga, svaka celija mora biti u jednom od konacnog broja stanja, sto je i jedino ogranicenje na ovaj skup, da mu je kardinalnost konacna, tj. odgovara nekom prirodnom broju. Kasnije cemo vidjeti da su moguce i neke generalizacije samog skupa stanja gdje npr. imamo celijske automate cija stanja odgovaraju realnim brojevima u intervalu $[0, 1]$ sto ocigledno nije konacan skup stanja.

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

Potrebno je jos razmotriti sta bi bilo povoljno da se koristi u definiciji susjedstva te prelaza (evolucije) svake celije unutar automata. Ovi koncepti nista prostorne i vremenske transformacije konretne celije. Npr. susedstvo je samo mapiranje/dodjeljivanje nekoliko celija jednoj konretnoj 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 specificnim instancama, tako da je moguce 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
- $\sigma : \Lambda \times T \to Q$
- $\eta : \Lambda \to \Lambda^c$
- $\phi : Q^{c+1} \to Q$ $(c \in \mathbb{N} \wedge c \geqslant 1)$

Funkcije σ, η i ϕ nazivaju se funkcije konfiguracije, susjedstva i prelaza respektivno. 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 susjdestva, te njeno trenutno stanje. Definisana je rekurzivno, jer svaka iteracija zavisi samo od prethodne, a veoma je moguce da je pojedina ili cak vecinu pravila prelaza nemoguce predstaviti elementarnim funkcijama na koje smo navilkli. Ovo se vec dalo zakljuciti iz kompleksnosti uzoraka koje ova vrsta sistema generise.

Moze se primijetiti da celijski automati posjeduju nekoliko osobina koje su zadovoljene za svaku instancu istih. To su homogenost, paralelizam i lokanost. Ovo je s obzirom da sve celije evoluiraju paralelno po istim proavilima koja su odredjena samo lokalnim susjedstvom i stanjem ostalih celija u njemu [1].

Ovim je zavrseno razmatranje generalne definicije celijskih automata, te ce nadalje biti izucavane najinteresantnije i najistrazenije instance istih koje cemo pojedinacno pokusati uklopiti u ovaj model, te detaljno razmotriti specificne matematicke osobine svake od tih instanci.

2.2 Binarni jednodimenzionalni (elementarni) celijski automati

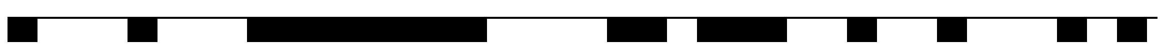
Najosnovniji tip sistema celijskih automata predstavljaju binarni jednodimenzionalni celijsi automati. Ovo znaci da svaka individualna celija posjeduje dva moguca stanja – binarni, te da svaka celija ima tacno dva susjeda, te da su celije medjusobno poredane u jednodimenzionalnoj resetki – jednodimenzionalni.

Medjutim, uprkos njihovoj jednostavnosti, upravo ova vrsta predstavlja najizucavaniji tip celijski automata. Ovo mozda mozemo pripisati cinjenici da se cjelokupna nauka vezana za ove vrste sistema i bazira na sto jednostavnijim gradivnim elementima koji produkuju kompleksne strukture, pa su tako najjednostavniji od njih vjerovatno i najzanimljivi za izuciti. Ovo nije slucaj za vecinu drugih oblasti sa drugim skroz drugim pogledom na kompleksnost.

Drugi naziv koji se koristi u literaturi za ovu vrstu celijskih automata je elementarni celijski automati, tako da ce ova dva termina u daljem tekstu biti koristena naizmjenicno.

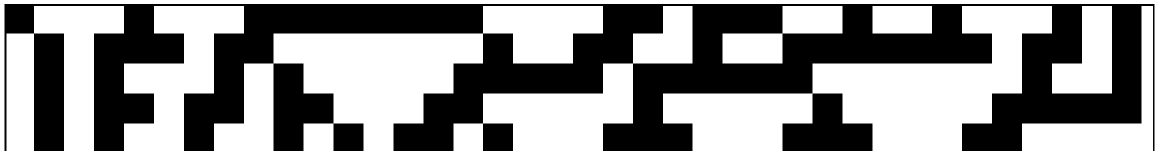
2.2.1 Definicija i nacin enumeracije pravila

Pokusajmo sada dati definiciju i prikazati kakvu vrstu sistema predstavljaju binarni jednodimenzionalni celijski automati. Najjednostavnije je shvatiti ove sisteme kao niz celija poredanih jedna do druge u liniji – zbog cega se i zovu jednodimenzionalni, prilikom cega svaka celija moze biti u jednom od dva stanja – on ili off, sto graficki predstvaljamo crno i bijelom bojom respektivno. Generalno, graficki prikaz je cesto koristen alat u izucavanju celijskih automata jer moze da da neke indikacije o osobinama izucavanih sistema.



Slika 10. *Primjer proizvoljne pocetne konfiguracije jednodimenzionalnog celijskog automata sa dva stanja [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 odredjenom 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. *Pet koraka evolucije pocetnih uslova jednodimenzionalnog celijskog automata sa dva stanja. Specificno pravilo evolucije primijenjeno je pravilo 30 [generisano softverom]*

Na slici 11 nadalje prikazano 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 izvrsen tek pet puta. Primijecujemo da je evolucija sistema izvrsena u diskretnim vremenskih 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. moguce je izvrsiti enumeraciju istih. Radi jednostavnosti, moguce je umjesto C koristiti skup $\mathbb{Z}$. Funkcija σ predstavlja mapiranje $\sigma : \mathbb{N} \times \mathbb{Z} \to \{0, 1\}$ i naziva se pravilo evolucije. Funkcija ϕ predstavlja mapiranje $\phi : \{0, 1\}^n \to \{0, 1\}$, gdje se n naziva velicina susjetstva. Mapiranje σ definisano je rekurzivno 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.

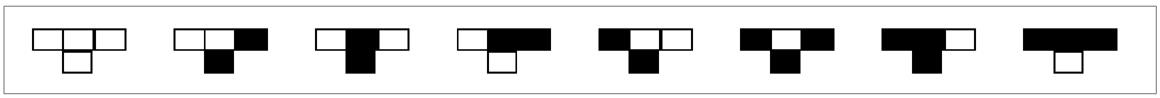
Ovu potpuno formalnu definiciju intuitivno mozemo shvatiti na nacin da se za svaku posebnu celiju, koja je oznacena cijelim brojem koji je jedinstveno identifikuje s obzirom na raspored celija, prema odredjenom pravilu prvo bira susjdestvo. Susjedstvo predstavlja skup celija od cijih stanja zavisi i 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, moguce je koristit bilo koje pravilo za izbor susjedstva. Medjutim, u praksi izucavanja pokazalo se da je najjednostavnije pravilo koje uzima samo simetricno susjedstvo od 2n najblizih celija sasvim dovoljno da se pokazu i najkompleksnije osobine.

Prodiskutujmo sada broj mogucih pravila za svako izabrano susjedstvo. Cjelokupno susjedstvo ce imati $2n + 1$ celiju ukljucujuci i celiju u razmatranju. Za svaku od kombinacija stanja, tacnije 2^{2n+1} mogucih s obzirom na 2 moguca stanja, potrebno je definisati sljedece stanje u koje se prelazi ukoliko se naidje na tu situaciju. Svako od iducih stanja takodjer pripada skupu $\{0, 1\}$ tako da je ukupan broj pravila $2^{2^{2n+1}}$ za ovako izabrano susjedstvo.

Ispostavilo se da takodjer i ovdje najjednostavnija pravila daju najzanimljivije rezultate, sto smo vidjeli u vec dosta primjera u ovoj oblasti, te za najjednostavnije susjedstvo sa $n = 1$ daje poprilicno zanimljivu kompleksnost i moguca razmatranja. Najveci broj radova iz oblasti takodjer je pisan upravo za ovaj slucaj (pogledati npr. vecinu radova Stephena Wolframa ukljucujuci [2], [3], [4] i mnoge druge).



Slika 12. *Graficki izbor pravila elementarnog celijskog automata. [generisano softverom]*

Sada, prema gore dobijenoj formuli imamo da je ukupan broj mogucih pravila $2^{2^3} = 2^8 = 256$. Ovo moze graficki biti prikazano kao sto je dato na slici 12 gdje imamo sljedece stanje za svaku kombinaciju susjedstva.

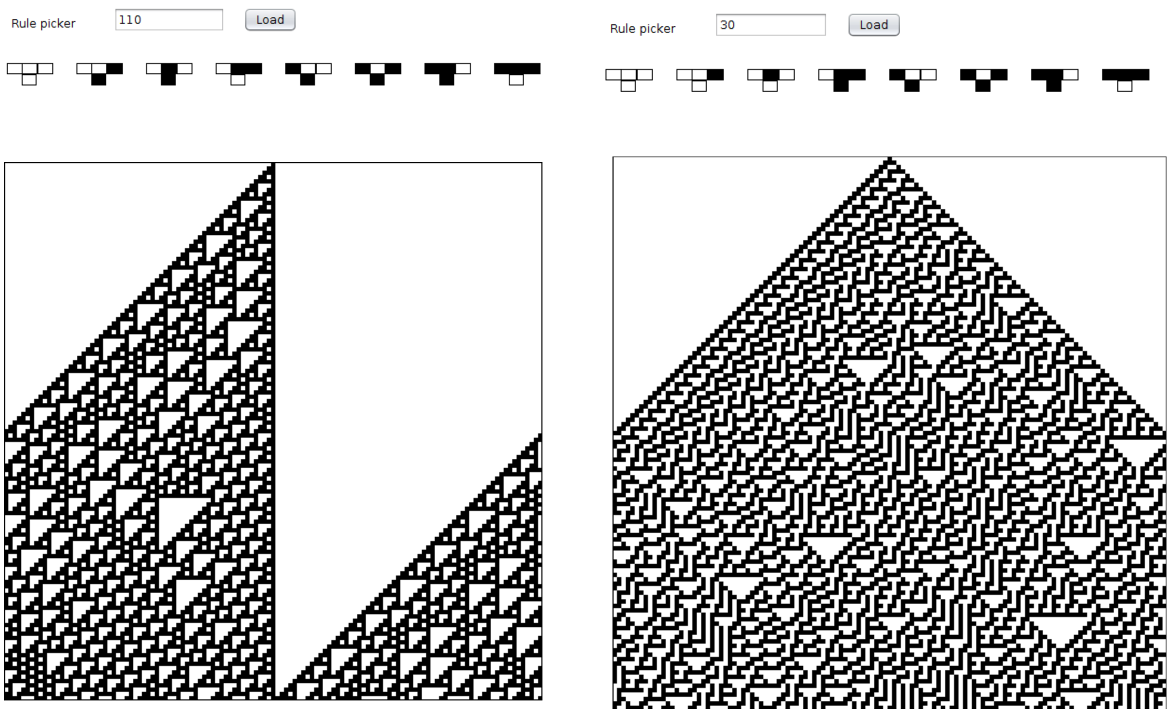
Primijetimo i da prosirenjem susjedstva na $n = 2$ broj pravila raste na $2^{2^5} = 2^{32} = 4294967296$, tako da nesto kompleksniji sistemi imaju eksponencijalno veci broj pravila. I ovo je takodjer jedan od razloga izucavanja najjednostavnijeg slucaja zbog mogucnosti pregleda svih pravila, sto bi za slucaj $n = 2$ bilo teze.

U daljem tekstu sva razmatranja odnosice se pretezno na slucaj susjedstva $n = 1$ binarnih jednodimenzionalnih celijskih automata.

Nacin na koji definisemo koje pravilo se koristi pri evoluciji ovih sistema, a predlozio ga je Stephen Wolfram na pocetku izucavanja ove oblasti, temelji se na enumeraciji pravila na osnovu bita u koje prelaze leksikografski poredane kombinacije susdjedstva. To znaci 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 moguce 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]).

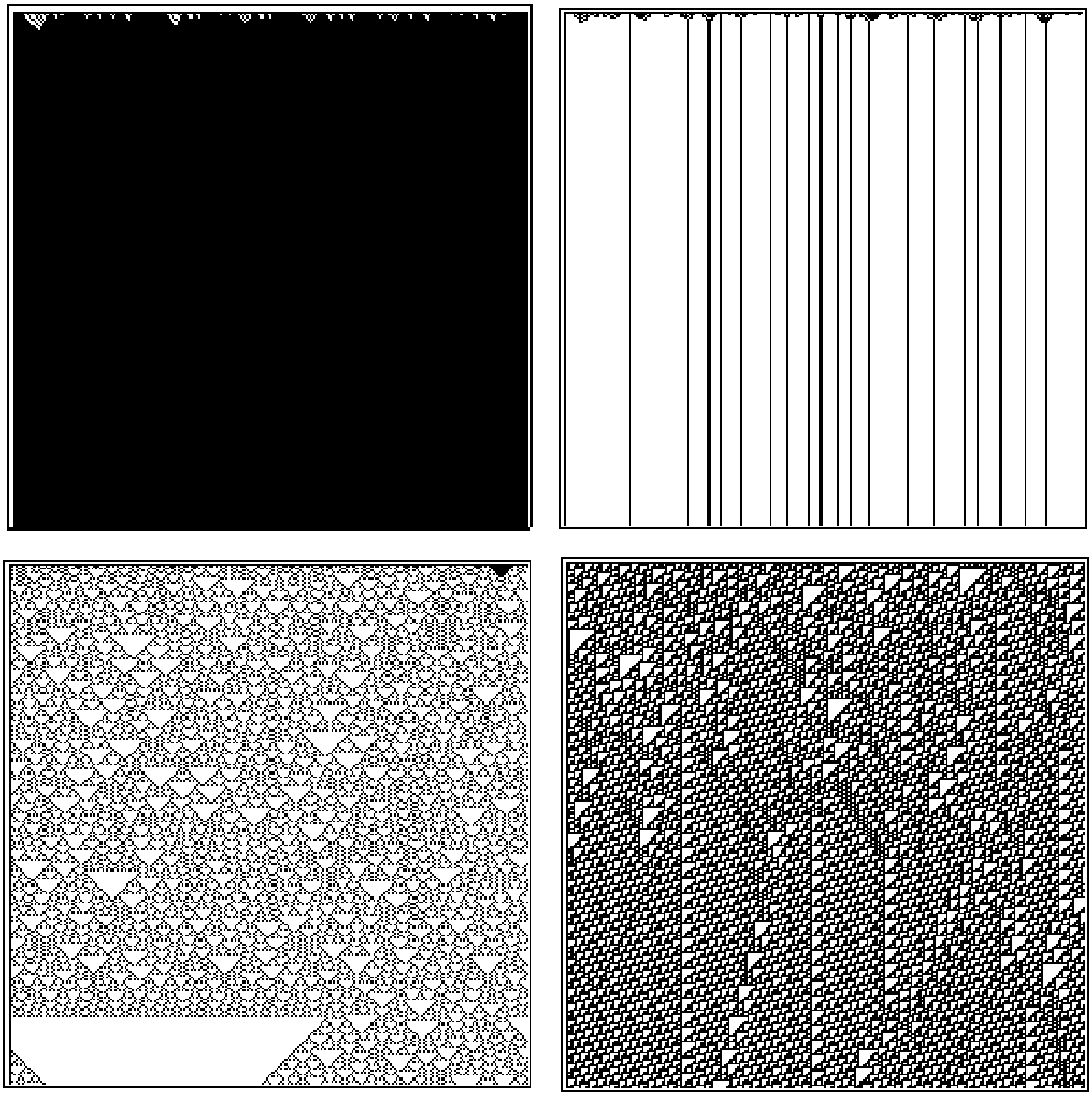


Slika 13. *Prikaz vise koraka evolucije pravila 110 i 30. Pravila su graficki prikazana na vrhu slike. Uocljive su kompleksne strukture. [generisano softverom]*

Na slici 13 prikazana je graficka evolucija od nekoliko stotina koraka pravila 110 i 30 sa torusnom topologijom niza respektivno da bismo poceli uocavati kompleknost koja proizilazi iz nekih od najjednostavniji sistema izracunavanja koji mogu da se smisle. Kasnije cemo vidjeti da je moguce i formalnim putem pokazati da su ovi sistemi poprilicno 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 poprilicno 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.

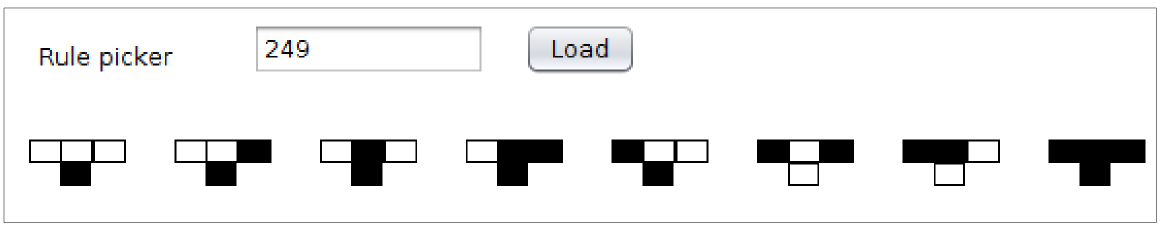


Slika 14. *Evolucija pravila 249, 164, 146, 110 respektivno koja se nalaze u postuliranim klasama I, II, III i IV. [generisano softverom]*

Ukoliko se izvrse simulacije sa slucajnim pocetnim uslovima svih mogucih pravila elementarnih celijskih automata (pogledati [2] za listu grafickih navedenih grafickih simulacija), moguce je uociti odredjene uzorke u ponasanju istih. Tako naizgled cisto empirijskim ispitivanjem moguce je uociti grube osobine nekih od pravila, sto se kasnije moze pokusati pretociti u formalnu klasifikaciju.

Klasifikacija sama po sebi je bitna jer predstavlja prvi korak u izucavanju neke vrste sistema. Ukoliko sve instance sistema posmatramo kao odvojene bez nekog nacina na koji bismo svrstali svaku od njih u odredjenu klasu ponasanja, zapravo se radi o ad-hoc izucavanjima specijalnih slucajeva. Sama klasifikacija predstavlja visi nivo za posmatranje sistema koji se nalaze u njoj, te kroz klasifikaciju mozemo da uocimo i neke kljucne osobine viseg reda, sto nas pomjera od pocetne tacke gdje smo samo definisali sistem i nacin njegovog funkcionisanja/evolucije.

Ukoliko pogledamo sliku 14, prikazana je vremenska evolucija cetiri karakteristicna pravila koja ujedno predstavljaju i osnovne uocljive klase ponasanja elementarnih automata koje cemo sada navesti i nesto kasnije i obraditi [1]. Formalna obrada tematike celijskih automata predstavlja nesto tezi zadatak s obzirom da se radi o nelinearnim sistemima za koje se vecinom ne moze naci eksplicitan matematicki model predvidjanja buducih stanja u zavisnosti od sadasnjeg. S obzirom na ovo u dosta slucajeva moguce je izvrsiti jedino statisticka i slicna globalna ispitivanja bez dolaska na konkretan model predvidjanja (pogledati npr. [6], [9] kao i vecinu radova iz oblasti celijskih automata i dinamike kompleksnih nelinearnih sistema).



Slika 15. *Graficki prikaz prelaza stanja pravila 249. [generisano softverom]*

Prva (empirijski!) uocljiva klasa – Klasa I automata predstavljena je pravilima koja konvergiraju u homogene strukture bez obzira na pocetno stanje sistema. Takodjer, sve informacije kao i slucajnost (eng. randomness) u pocetnim 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 pocetno stanje automata, uvijek cemo dobiti isti izlaz koji ne sadrzi nikakve korisne informacije. Primjer ovakvog pravila je pravilo 249 koje bez obzira na strukturu pocetnih uslova konvergira u niz celija u off stanju (crna boja na grafickom prikazu). Ovo mozemo pripisati cinjenici da vecina kombinacija susjedstva ovog pravila vrsi prelaz u off stanje, sto je evidentno sa slike 15.

Iducu klasu – Klasu II predstavljaju pravila koja ulaze u cikluse ponavljajucih struktura. Na slici 14 prikazano je medju ostalima i pravilo sa enumeracijom 164 koje konvergira u ponavljajuce 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 poprilicno male konfiguracije imamo ogromne cikluse, empirijski rezultati pokazuju da je taj broj manji. Pretezno je ogranicen je sa $2^{N/2}$ dok je u dosta slucajeva cak potrebno ne vise od N iteracija da bi se dovrsio ciklus [6].

Unutar Klase III nalazimo pravila poput 146 i 30 koja ispoljavaju naizgled nepredvidivo slucajno ponasanje sa ponavljajucim uzorcima pretezno ispoljenim u vidu trougaonih struktura. Ova klasa je poprilicno osjetljiva na pocetne uslove i predstvlja 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 predstvlja najkompleksniju klasu ponasanja celijskih autoamta, ali isto tako i najslabije definisanu. Neformalno, u ovoj klasi automata gotovo svi pocetni uslovi proizvode kompleksne strukture koje vrse kompleksnu medjusobnu interakcije. Postulirano je da ova klasa omogucava skladistenje 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.

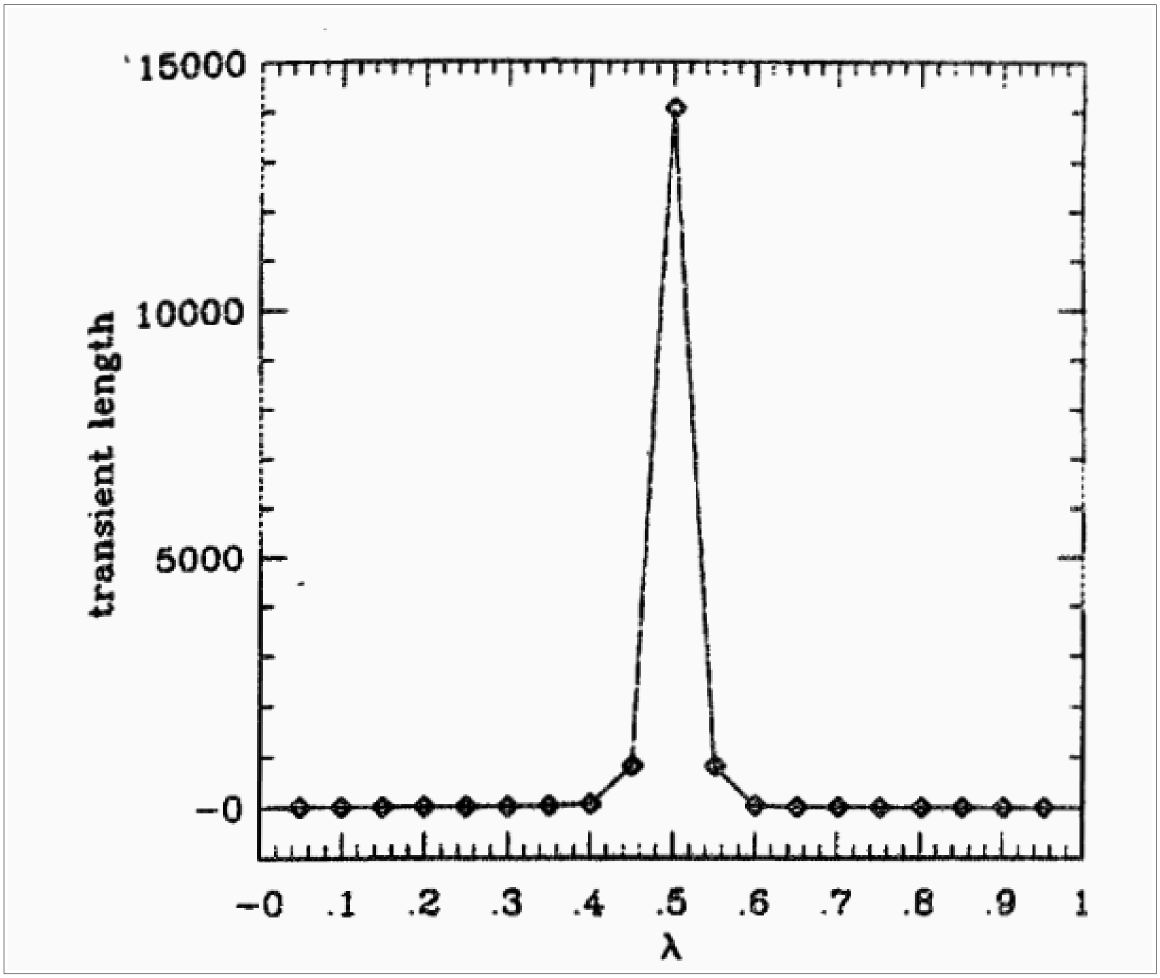
Obradicemo jedan takav pokusaj s obzirom da je dao poprilicno dobre rezultate, te otvorio neka nova pitanja i hipoteze ne samo u oblasti izucavanja celijskih automata, vec dinamike kompleksnih sistema generalno. Tako cemo i ovdje navesti neke od tih rezultata kroz pokusaj formalne klasifikacije. Klasifikator 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 poprilicno zanimljivi rezultati koji govore da postoji odredjena statisticka struktura u ponasanju ovih sistema u zavisnosti od parametra.

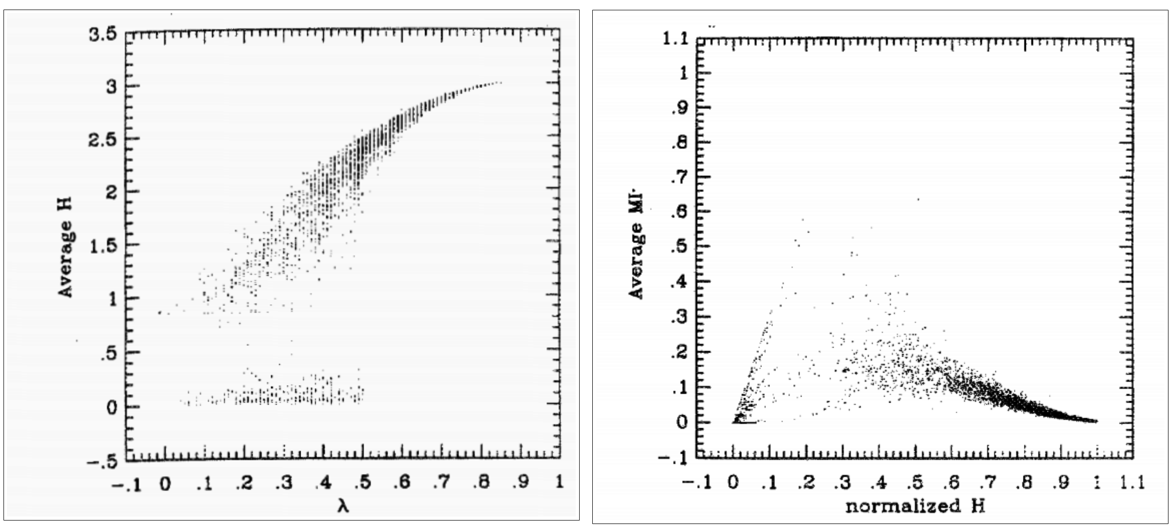


Slika 16. *Prosjecno vrijeme stabilizacije automata ponasanja u zavisnosti od Langtonovog parametra. [9]*

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 zapazenje bazira na cinjenici da je prelaz stanja u dinamickim 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 eksprimenta, 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 kompjutacionog 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].

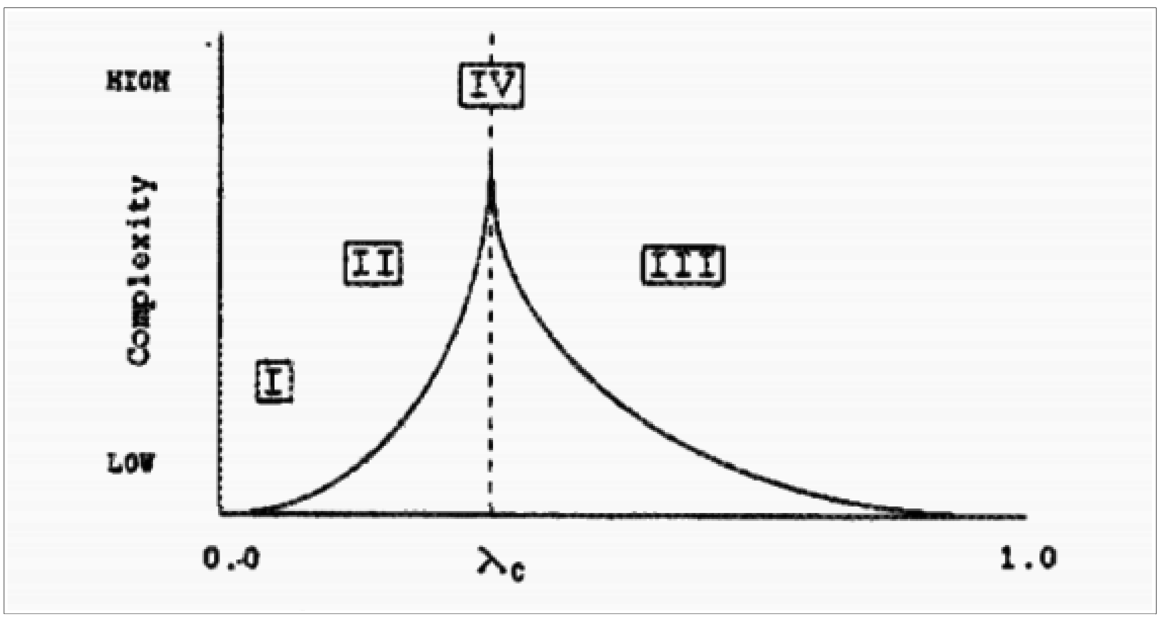


Slika 17. *Prosjecna entropija svake celije u zavisnosti od parametra (lijevo). Prosjecna kolicina medjusobnih informacija celija za odredjenu entropiju (desno). [9]*

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. mututal 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 moguce je zakljuciti (ne sa potpunom sigurnoscu) da u kriticnoj vrijednosti parametra gdje se desava prelaz stanja ujedno se dostize i maksimum zajednickih informacija celija, sto predstvalja 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].



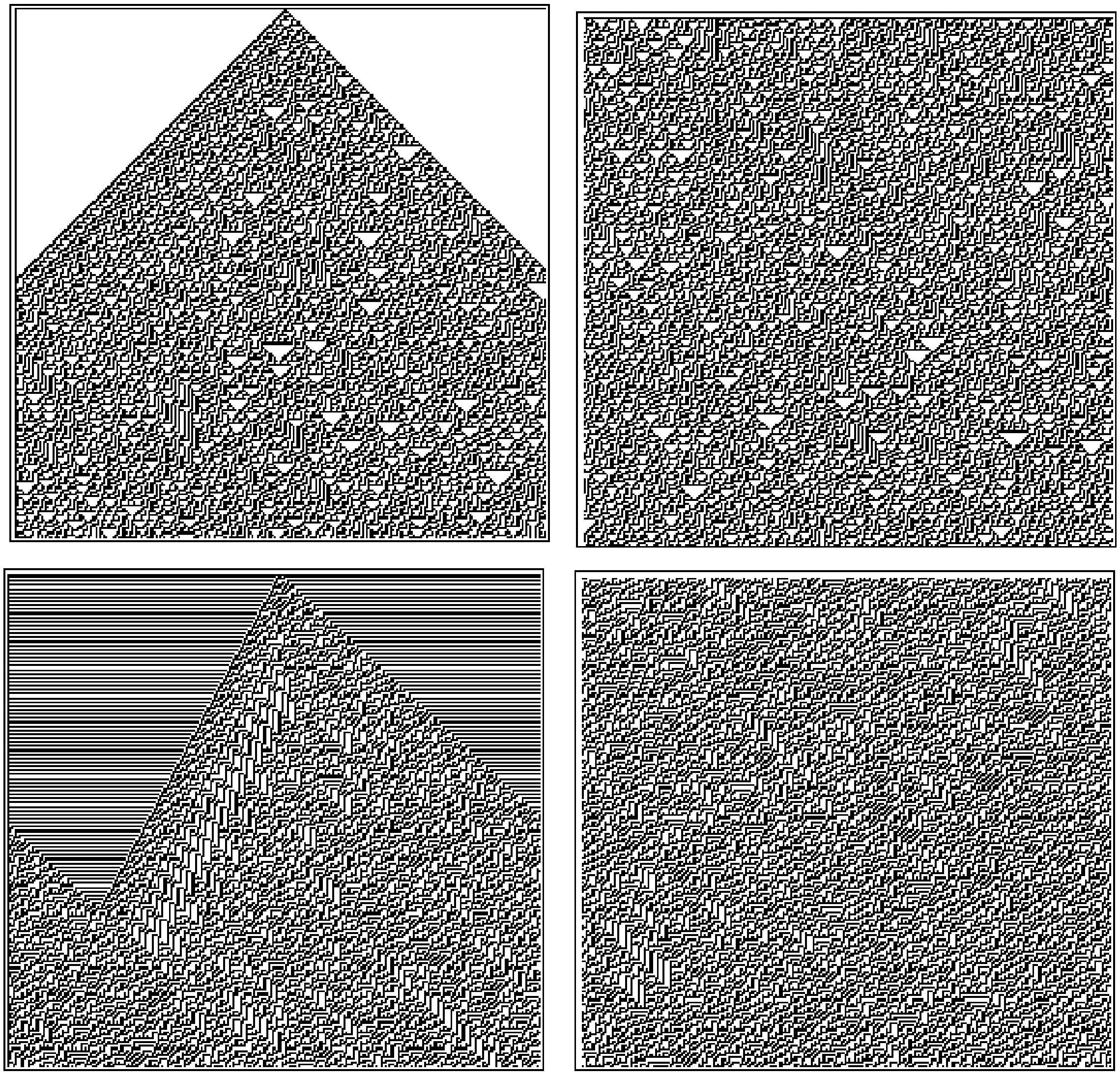
Slika 18. *Wolframove intuitivne klase ponasanja celijskih automata u zavisnosti od Langtonovog parametra. [9]*

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 kriticne vrijednosti $\lambda = \lambda_c$, dok bi klase I i II bile u vrijednostima manjim od kriticne s obzirom na nisku kolicinu informacije koje prenose, a klasa III bi bila locirana na vrijednostima vecima od kriticne s obzirom na kaoticno ponasanje iste. Ovaj raspored klasa prikazan je na slici 18.

Kako smo i rekli klasifikacija u vecini slucajeva moze biti prvi korak u daljnjem izucavanju nekog sistema, pa tako i ova navedena klasifikacija, bilo da se radi o intuitivnoj ili pokusajima formalne, daje naznake o zanimljivim osobinama elementarnih celijskih automata koje bi trebalo detaljnije ispitati. Tako klasifikacija predvidja automate koji imaju potpuno nepredvidivo ponasanje (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 najvecim potencijalnim primjenama bice izucene u sljedecim razmatranjima.

2.2.3. Slucajnost (eng. randomness) elementarnih celijskih automata

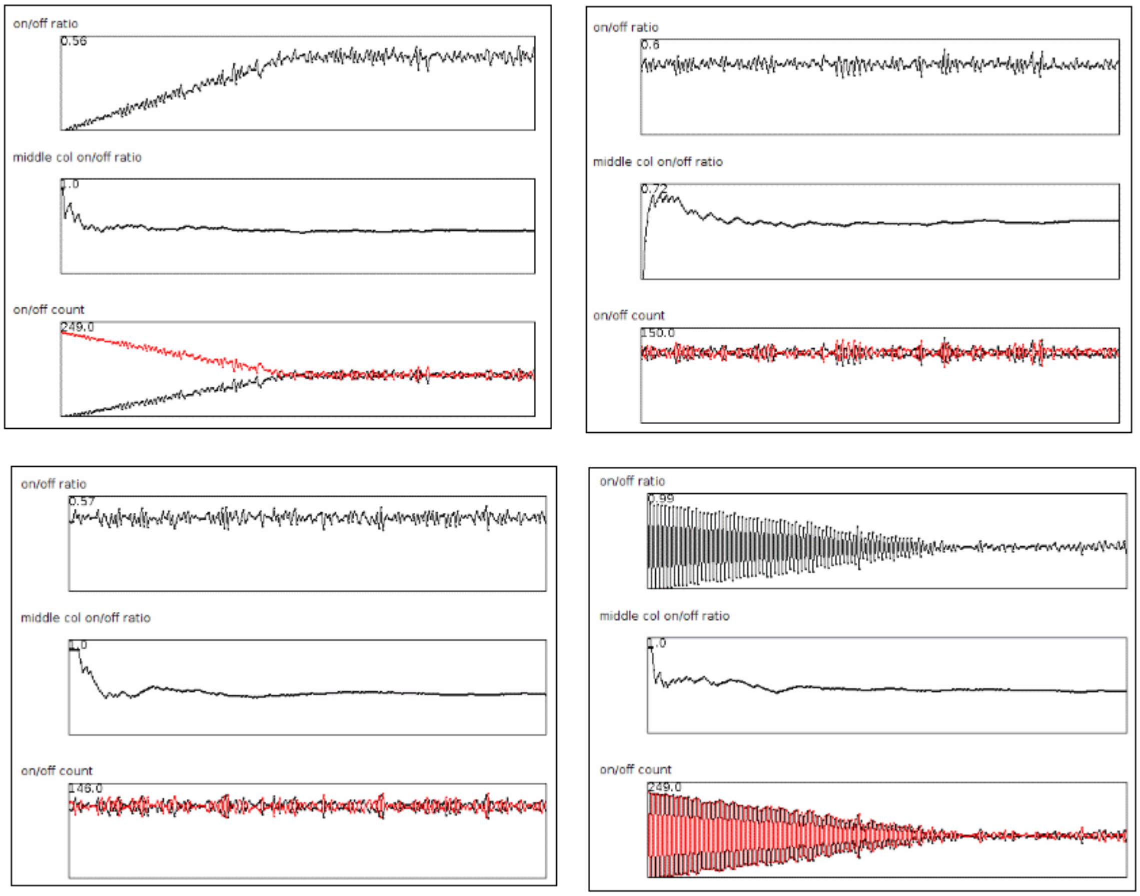
Izucavanje slucajnosti i slucajnih pojava kljucno je u raznim dijelovima kompjuterske i ostale nauke s obzirom da neke od kljucnih primjena danasnjice poput Monte-Carlo algoritamskih metoda i kriptografije su upravo bazirane na ovom konceptu. Odmah se postavlja pitanje da li bi se uocena slucajnost u klasi klasi III elementarnih celijskih automata mogla iskoristiti u ove svrhe, s obzirom da bi jednostavnost sistema koji generisu ovu slucajnost mogla biti presudna u izboru istih nad drugim metodama. Iz ovog razloga, korisno je izuciti nesto detaljnije i formalnije kolicinu slucajnosti koju mogu da proizvedu navedeni sistemi.



Slika 19.a *Naizgled slucajne strukture generisane pravilima 30 (prvi red) i 45 (drugi red). [generisano softverom]*

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

Mnogi su testovi razvijeni da bi se testiralo koliko je zapravo neki sistem statisticki slucajan. Treba napomenuti da se koncept slucajnosti razlikuje od statisticke slucajnosti s obzirom da se statisticka slucajnost javlja u sistemima koji su u svojoj prirodi deterministicki, kao sto je primjer i sa trenutno izucavanim elementarnim automatima, medjutim ne postoji globalni uzorak ponasanja koji oni zadovoljavaju.



Slika 19.b *Statistika generisana za simulacije prikazane na slici 19. [generisano softverom]*

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 pojedinacne redove, vec na srednju kolonu jer je predlozeno da se upravo ovo koristi kao pseudoslucajni generator u kriptografskim primjenama [12][13]. Treca statistika je poprilicno redundatna ako nas zanima samo odnos, te predstavlja stvarni broj on i off celija u svakom redu u svakom vremenskom koraku.

Vidimo da se za izvrseni broj koraka u svakom od cetiri slucaja simulacije posmatrani odnos kako u svakom redu pojedinacno, tako i u specijalno posmatranoj srednjoj koloni blizi predvidjenoj vrijednosti 0.5 koja bi bila postignuta za pseudoslucajni generator, tako da ovo moze dati jake indikacije o slucajnosti navedenih pravila.

Wolfram je u [12] detaljnije i jacim matematickim aparatom – informacionom teorijom i na prednom statistikom a ne samo indikacijama iz simulacija formalno obradio slucajnost ovih sistema. Dosao je do zakljucaka da je pravilo 30 dovoljno slucajno za vecinu primjena uz dovoljno veliku pocetnu konfiguraciju, dok je pravilo 45 znatno manje statisticki slucajno, ali opet nepredvidivo u velikoj mjeri. Nesto kasnije u [14] pokazano je da za odredjene vrijednosti velicine inicijalne konfiguracije kriptoanaliza moze da obrne proces pseudoslucajne generacije.

Zakljucujemo da je jedna od osobina pojedinih klasa i pravila elementarnih celijska automata tako struktuirana da ispoljava statisticku slucajnost, sto ce kasnije biti izuceno s aspekta primjena ove vrste sistema.

2.2.4 Univerzalna izracunljivost elementarnih celijskih automata

Nakon predstavljanja bilo kojeg sistema koji ispoljava bar neki nivo kompleksnog ponasanja, uvijek je zanimljivo upitati se kolika je zapravo ta kompleksnost te je pokusati na neki nacin kvantificirati. Ovo znaci ispitati da li je sistem sposoban simulirati bilo koji drugi sistem koji se smatra izracunljivim u matematickom smislu.

Unutar teorije izracunljivosti, kao osnovni model izracunljivog sistema uzima se bilo koji algoritam ili procedura za koji se moze konstruisati Turingova masina koja ga simulira. Ovo direktno slijedi iz rezultata poznatog kao Church-Turingova teza koji govori da je pitanje izracunljivsto ekvivalentno pitanju da li je moguce to ponasanje simulirati na nekoj konstruisanoj Turingovoj masini. Ovo razmatranje direktno uvodi Turingove masine kao osnovni model izracunljivih sistema sa kojim se ostali sistemi trebaju porediti ukoliko se zeli pokazati njihova racunarska moc. Moguce je koristiti i neke druge modele za izucavanje ovog polja poput Alan Churchovog $\lambda - kalkulusa.$

Pitanje univerzalnosti unutar ovih okvira se svodi na mogucnost simulacije Turingovih masina, tj. sistem se naziva univerzalno kompjutacion ili Turing ekvivalentan ukoliko je u mogucnosti simulirati svaku proizvoljnu Turingovu masinu, sto ima smisla ukoliko se razmotri zasto se Turingova masina smatra za osnovni model preko kojeg se definise izracunljivost.

Moguce je sada postaviti ovo pitanje i za elementarne celijske automate. Ocigledno 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 ponavljajucih krajnjih stanja sistema, te kao takvi evidentno nisu u mogucnosti mapirati proizvoljan ulaz na proizvoljan izlaz. Pravila iz klase III kako je vec pokazano pokazuju poprilicno slucajno ponasanje, te kao takva takodjer nisu dobar kandidat za Turing ekvivalentne sisteme s obzirom da nam za izracunljivost predvidivost igra kljucnu 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 specificnog sistema (alfabeti se naravno mogu razlikovati) daje string kojem se taj simbol pridruzuje . 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 izvrsenje. Skup pravila je definisan kao $a \to ccbaH, b \to cca, c \to cc$. Tada 6 iteracija ovog sistema izgleda:

Iteracije:
1 baa
2 acca
3 caccbaH
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 poprilicno jednostavna pretvorba iz jednog u drugi, s razlikom da ciklicni tag sistem ne mora da provjerava kojem simbolu odgovara koji string, vec ima skup pravila koja ciklicno vrti – zato se i naziva ciklicni. Naime, za ciklicni tag sistem ne veze se pravilo za svaki simbol, vec su pravila u fiksiranom skupu i vrte se ukrug, te se pocetni znak uvijek mijenja trenutnim pravilom ukoliko znak nije takav da nalaze nemijenjanje, sto je objasnjeno u sljedecm primjeru.

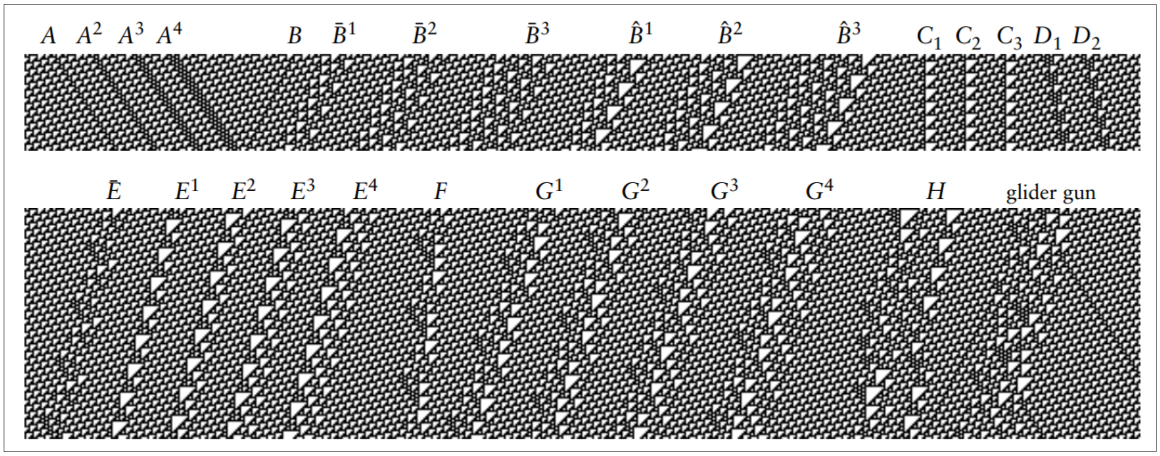
Neka je na primjer dat ciklicni tag sistem sa skupom pravila (produkcija) $(010, 000, 1111)$. Alfabet ciklicnog tag sistema sastoji se od simbola $\{0, 1\}$, gdje se trenutno aktualno pravilo iz skupa produkcija primjenjuje na string ukoliko je skroz lijevi simbol 1, doke se u suprotnom simbol 0 samo uklanja sa pocetka stringa. Ukoliko je pocetni 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 ciklocnom redoslijedu, dok se string mijenja samo ukoliko je pocetni simbol 0.

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

Na ovom principu je i Mathew Cook dokazao univerzalnost, a ovdje ce samo biti izlozene osnovne ideje koje se kriju iza dokaza, jer je sam dokaz preobiman te se citaoc za vise detalja moze referirati na [15].



Slika 20. *Neki od glideri pravila 110. [15]*

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 odredjenim 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 glier 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 izvrsavanje pravilom 110 moze se naci u [16], gdje se kombinuje pretvaranje Turingove masine u ciklicni tag sistem, nakon cega se on kodira za izvrsenje celijskim automatom. Iznenadjujuci je rezultat da ovo racunanje i pretvorba zahtijeva asimptotski polinomsko vrijeme, iako se smatralo da je izvrsenje 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 specificnih 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 specificno.

3.1 Generalne oblasti primjene celijskih automata

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

Pregledajmo prvo kljucne osobine koje predstavljaju potencijalno plodno tlo za realne primjene celijskih automata. Za pocetak, celijski automati, bar ono sto se klasicno smatra njima, baziraju se na paralelnosti i homogenosti, to jest sinhronoj promjeni po identicnim pravilima stanja svih komponenti vodjeno globalnim diskretnim satom. Nesto slicno vidimo i kod modernih racunara gdje se elektronska logika ponasa na slican nacin. Nadalje, celijski automati bazirani su na principu lokalnosti, sto znaci da pojedina celija ne zna nista o cjelokupnom stanju sistema, vec samo stanja svojih najbilizih susjeda, sto omogucava skladistenje jako male kolicine informacije i jednostavne komponente. Kao sto je vidjeno, celijski automati iako jednostavni predstavljaju univerzalan izracunljiv sistem, tako da jednostavnost ne znaci nuzno i beskorisnost. Takodjer, kako spadaju u klasu nelinearnih dinamickih sistema, pojedine instance imaju haoticno i random ponasanje, osobina koja takodjer moze biti od koristi u pojedinim oblastima.

Iz ovih osobina i njihovih kombinacija mozemo zakljuciti oblasti u kojima su te osobine korisne, te tu pokusati primijeniti celijske automate.

Iz osobina paralelnosti i univerzalnosti izracunavanja, moze se pretpostaviti da bi bilo moguce iskoristiti celijske automate u svrhu paralelnog racunanja koje sve vise postaje paradigma na koju se pokusava prebaciti moderno racunarstvo. 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 celija predstavljala zaseban jednostavni procesor u visokoparalelnom racunanju. Takodjer, moguce je i implementirati celijskim automatima neke visoko paralelizabilne zadatke poput obrade digitalnih slika gdje bi se za svaki pixel pojedinacno uspostavljao celijski automat koji bi ga mijenjao u zavisnosti od susjednih pixela. Istrazivanja u ovoj oblasti vec 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 ponasanje. Upravo ovo cini celijske automate pogodnim za razne vrste simulacija, od kojih ce neke biti obradje i u specificnim implementacijama primjena.

Ako dodamo mogucnost izmjene, heterogenosti i napredovanja pravila prirodnom selekcijom, tada bi celijski auomati mogli biti iskoristeni za primjene na polju vjestacke inteligencije. Na samom zacetku oblasti sam Wolfram je izucavao potencijal celijskih autoamta u primjenama simulacija neuronskih mreza koje su veoma koristan i napredan koncept vjestacke inteligencije danas siroko koristen u raznim primjenama.

Takodjer nepredvidivost nekih odredjenih instanci celijskih automata cini ih korisnim u primjenema gdje je potrebno da je tesko moguce pogoditi pocetno 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 pravljeni 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 najvecim dijelom obradjivana 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 poprilicno 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.

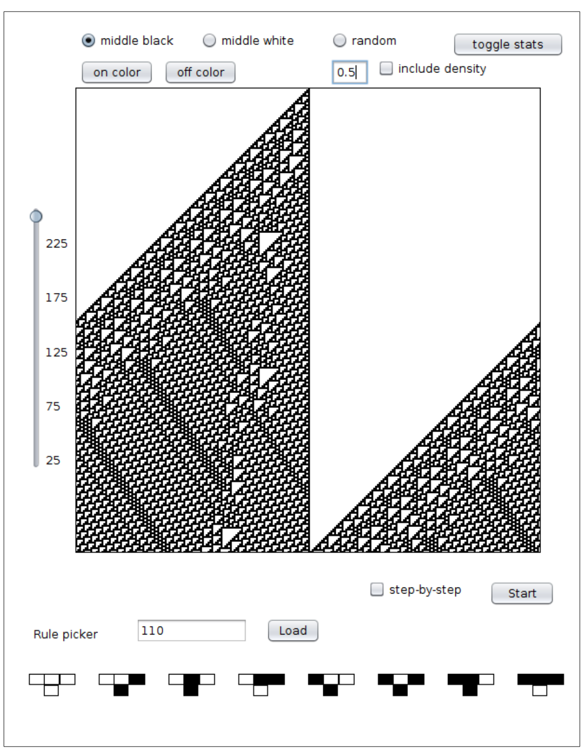
Generalna metodologija izucavanja bilo koje vrste sistema nalaze da se prvo izuce specificne instance da bi se dobila neka osnovna ideja o generalnim osobinama koje se prostiru kroz sve njih, te da se kasnije postulati dobijeni ovim postupkom pokusaju dokazati. Ovo smo mogli vidjeti na primjerima klasifikacije 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 mogucnost izbora pocetnog stanja, pravila evolucije graficki i numericki, te broj koraka simulacije koje je potrebno izvrsiti.

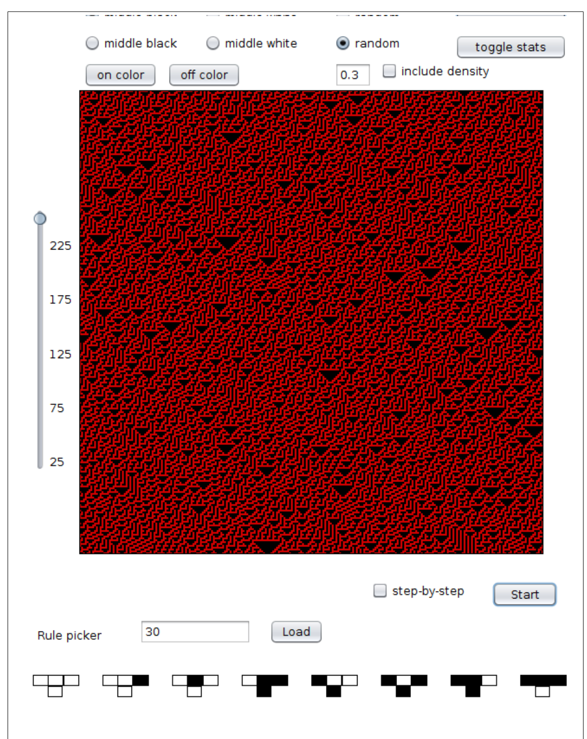
Nadalje, kako smo vidjeli celijski automati tesko se obradjuju konrektnim formulama i modelima s obzirom na njihovu nepredvidivu prirodu koja proizilazi iz njihove nelinearnosti, pa je tako statistika jedan od najmocnih alata koji moze da pomogne. Tako da bi software koji bi vrsio simulaciju trebao da sadrzi i neki vid statistike o izvrsenoj simulaciji da bi se mogli donijeti neki zakljucci.

Takodjer, ovako opisan software imao bi i edukacionu vrijednost gdje bi se mogle na prakticnim primjerima kroz graficko okruzenje pokazati osobine celijskih automata koje bi mozda zaintrigirale studente za daljnja izucavanja ovakvih sistema.

A.2 Prikaz software-a i kljucni featuri

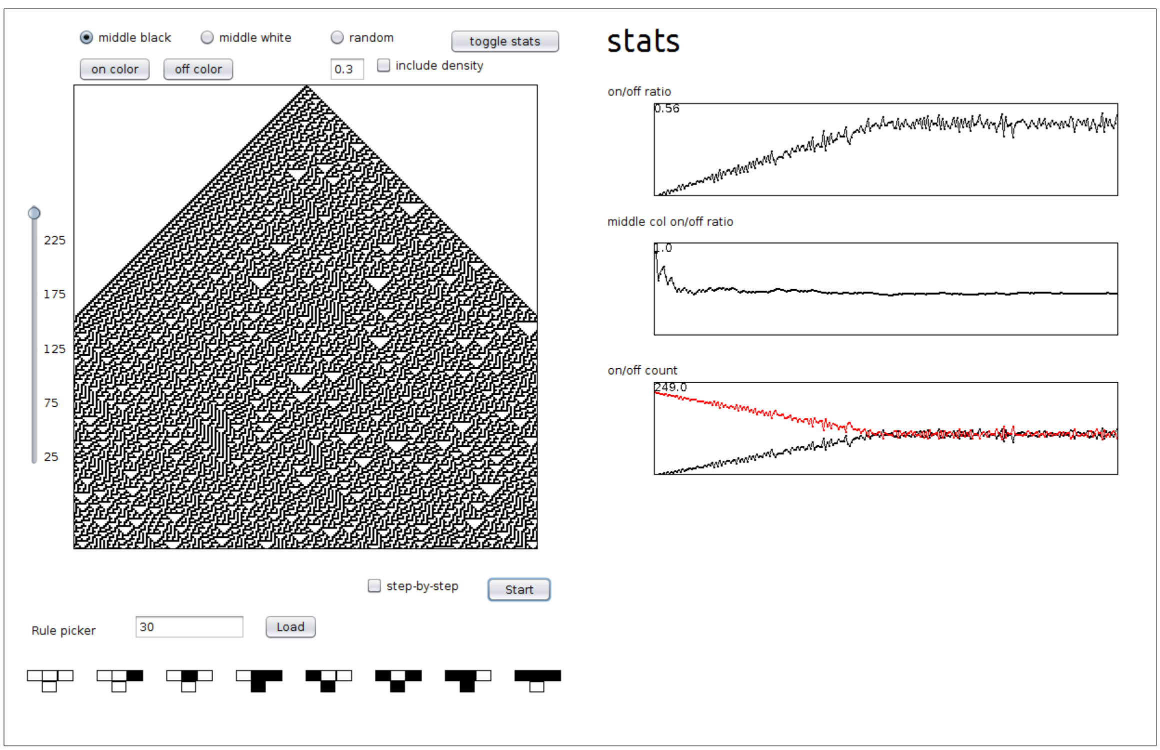
Pokusacemo kroz graficki interface software-a proci kroz njegove osnovne elemente i slucajeve upotrebe.





__Slika 21.__ Osnovni graficki interfejs softvera pisanog u sklopu rada.

Ako pogledamo sliku 21, vidimo osnovni prozor grafickog interface-a software-a, te je moguce uociti elemente prozora za prikaz simulacije, prostora za izbor pravila na numericki nacin cistim ukucavanjem njegovog Wolfram koda, ili izborom prelaza stanja za svaku od mogucih kombinacija susjedstva graficki. Takodjer u gornjem dijelu prozora moguce je izabrati nacin formiranja pocetne konfiguracije gdje je moguce izabrati srednju on ili off celiju, te nasumicnu pocetnu konfiguraciju koja zadovoljava odredjenu distribuciju. Takodjer postoje i kontrole koje omogucavaju mijenanje boja on i off stanja radi boljeg grafickog prikaza u nekim slucajevima. Vidimo da korisnik ima veliku slobodu izbora parametara simulacije. Na lijevoj strani slike 21 prikazan je software sa defaultnim pocetnim parametrima, dok na desnoj strani mozemo vidjeti kako izgleda simulacija sa izmijenjenim parametrima boje pravila i pocetne konfiguracije.

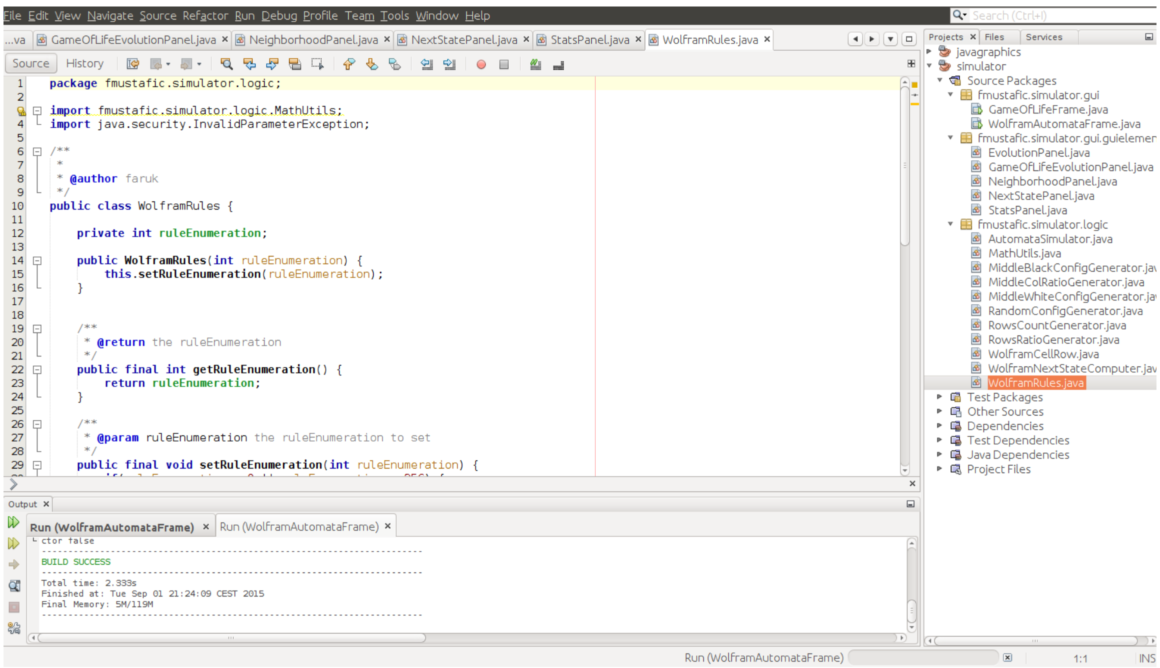


__Slika 22.__ Prosireni graficki interfejs softvera pisanog u sklopu rada sa prikazanim dijelom za generisanje statistike.

Na slici 22 prikazan je graficki interface software-a sa prosirenim prozorom za statistiku koji se otvara/zatvara na odgovarajuce toggle dugme. Na istoj slici takodjer mozemo vidjeti i primjer prakticnog koristenja software-a, gdje statistika za Wolframovo pravilo 30 daje naznake da je srednja kolona nasumicna s obzirom da broj on i off celija u njoj konvergira u odnos 0.5 kako sistem dalje evoluira, sto je raspravljeno detaljnije u poglavlju 2 gdje se formalno tretira nasumicnost elementarnih celijskih automata.

A.3 Tehnoloske implementacione odluke

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



Slika 23. *Prikaz razvojnog okruzenja u kojem je vrsen razvoj softvera.*

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 omogucava ubrzan razvoj software-a, te tako Java kao programski jezik koji je poznat kao jedan od kljucnih u ovoj oblasti omogucava dijeljenje koda sa ostatkom open-source zajednice u svrhu njegovog razvijanja. Nadalje, Java kao jezik i tehnologija uz Java virtualnu masinu omogucava cross-platform development, sto je takodjer bio jedan od ciljeva, tako da software moze da se pokrene na mnostvu operativnih sistema koji podrzavaju Java platformu. Java takodjer kao jezik visokog nivoa ima jednu od najrazvijenijih grafickih biblioteka, sto je omogucilo olaksan razvoj grafickog interfejsa programa. Takodjer, s obzirom na osobine paralelnosti celijskih automata kao sistema, uzeta je u obzir i Javina podrska paralelizaciji zadataka koja bi mogla biti korisna u ovoj oblasti za ubrzanje i optimizaciju izracunavanja. Nedostatak Jave je nemogucnost 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 ubrzava proces razvoja s obzirom da se ne mora paziti na detalje niskog nivoa.

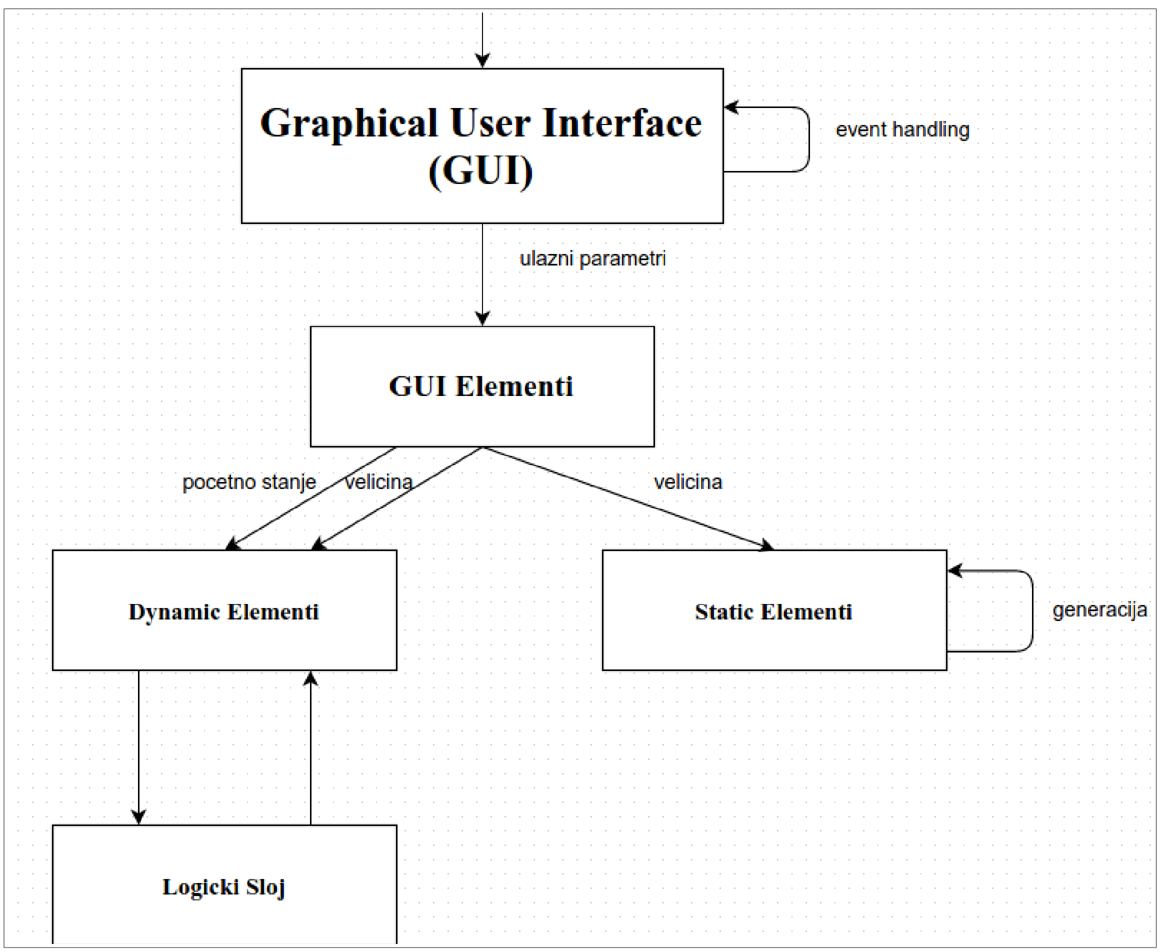
Netbeans razvojno okruzenje prikazano na slici 23 izabrano je iz razloga jednostavnosti integrisanog GUI dizajnera, te vece stabilnosti na platformi Linux od najveceg konkurenta Eclipse IDE-a.

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 izvrsiti i pregled arhitekture software-a da bi se detaljije shvatile specificnosti istog, s obzirom da je uvijek dobro imati neku vrstu pregleda visokog nivoa. Kroz ovaj pregled, u ovom slucaju generalna arhitektura sistem, daje mogucnost stvaranja sire slike sistema na visem nivou, gdje se pojedini dijelovi posebno obradjuju i detaljnije popunjavaju implementacionim specificnostima.



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 Intreface), 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 – dogadjaja na grafickom interfejsu, te poziva odgovarajucih nizih slojeva u zavisnosti od akcije zahtijevane dogadjajem. Ovo je prikazano na dijagramu sa desne strane, gdje cemo uspostaviti konvenciju da strelice oznacavaju prenos informacija, dok strelice koje se vracaju u objekat definisu vrstu logickog procesiranja samog sloja/elementa. Dogadjaji su generisani od strane korisnika njegovom interakcijom sa grafickim elementima.

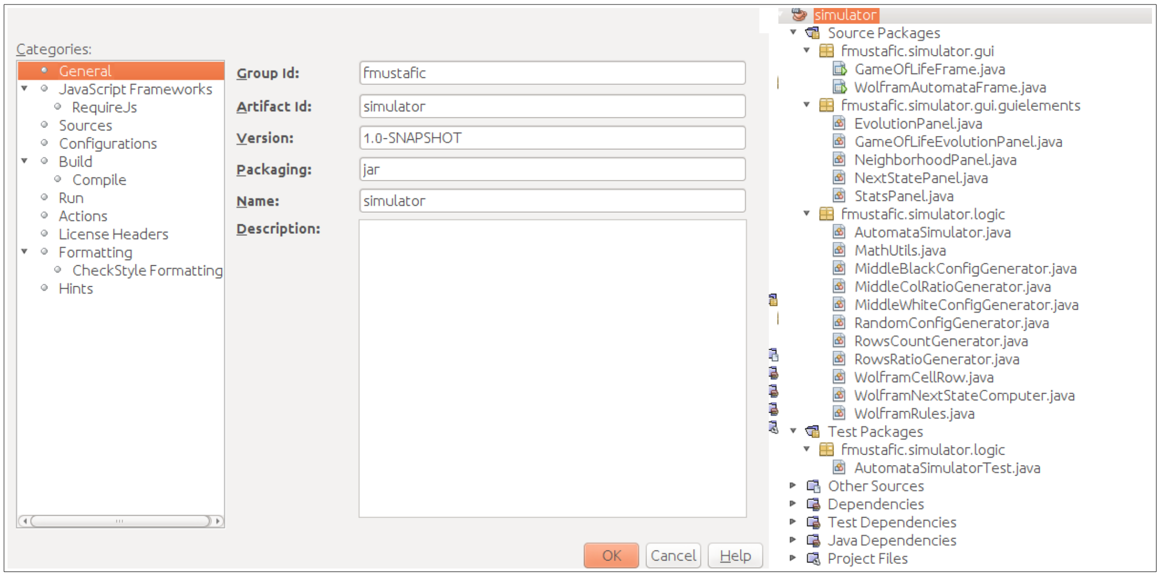
Graficki interfejs sastoji se od standardnih grafickih elemenata poput dugmadi i slidera, medjutim javlja se potreba i za specificnim custom elementima kao sto je u nasem slucaju panel za prikaz evolucije celijskih automata. Upravo to je iduci sloj arhitekture na slici prikazan kao GUI Elementi blok. Na ovom sloju postoje specificne implementacije grafickih elemenata koji nisu dio standardnog toolkita nekog jezika, te se potreba za njima javlja pretezno 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 pretezno 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 specificno.

A.4.2 Specificna implementacija arhitekture

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



Slika 25. *Specificna implementacija arhitekture.*

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

Java paketi predstavljaju nacin grupisanja logicki 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 ugradjene Java containere za prikaz ostalih elemenata. Takodjer na njima se nalaze osnovni GUI elementi poput buttona, te logika u ovim file-ovima predstavlja vecinski event handling.

U sloju grafickih elemenata, potrebno je bilo custom definisati nekoliko istih. Vidimo da paket fmustafic.simulator.guielements sadrzi elemente potrebne za prikaz evolucije automata, graficki 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 takodjer predstavlja vid containerskog elementa u Javi.

Na kraju paket fmustafic.simulator.logic koji odgovara najnizem sloju arhitekture sadrzi klase odgovorne za logicko procesiranje evolucije celijskih automata, te generisanje slucajnih pocetnih uslova za elementarni celijski automat.

A.4.3. Implementacione odluke po slojevima

Razmotrimo jos i specificne 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 pojedinacno. Na prva dva sloja nije bilo previse razmatranja s obzirom da postoji poprilicno 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 podrske 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 Phenomena10, 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