

Poglavje 8

Celularni avtomati

Celularni ali *celični avtomati* (angl. *cellular automata* - CA) predstavljajo diskretne modele sistemov, s katerimi proučujemo področji računske teorije in dinamike kompleksnih sistemov, ki temeljijo na distribuiranem odločanju. Značilnost *diskretnosti* glasi tako na *časovnost* (dinamika se modelira v diskretnih časovnih korakih), kot tudi na *zalogo stanj*, ki jih lahko zasedejo posamezne entitete modela. Pred samo definicijo CA si njihovo uporabno vrednost oglejmo na poznanem vzorčnem primeru Conwayeve igre življenja.

8.1 Conwayeva igra življenja

J. H. Conway igro življenja predstavi l.1970. Gre za diskretni model dinamike ali *evolucije* hipotetične umetne populacije entitet, čigar simulacijo sprožimo na osnovi začetnega stanja CA strukture, ki je definirana s stanjem posameznih entitet, v nadaljevanju pa analiziramo dinamiko, do katere privede začetno stanje strukture.

Conway igro definira v neskončnem 2-D prostoru, v katerem so regularno ali v obeh dimenzijah enakomerno razporejene kvadratne celice. Posamezno celico CA strukture si intepretiramo kot avtomat, ki v diskretnih časovnih korakih menja svoja stanja v odvisnosti od svojega predhodnega stanja in stanja sosednjih celic na predhodnem koraku. Po Conwayu vsaka od celic lahko zasede eno od dveh možnih stanj in sicer je celica lahko „živa“ ali „mrtva“. Ob predpostavki, da je 2-D prostor celic indeksiran in s tem posledično naslovljiv, kar pomeni, da se na posamezne celice lahko slicujemo preko njihovih naslovov, formacijo novega stanja $c_{x,y}$ celice z lokacije (x, y) formalno zapišemo z izrazom

$$c_{x,y}(t+1) = f(c_{x,y}(t), n(c_{x,y}(t))), \quad (8.1)$$

pri čemer $c_{x,y}(t)$ predstavlja stanje opazovane celice na lokaciji (x, y) v času t , $n(c_{x,y}(t))$ stanja sosednjih celic opazovane celice v času t , f pa preslikovalno funkcijo. Conway sosedstvo ali stanja sosednjih celic definira z množico celic, ki neposredno mejijo na opazovano celico. Formalno sosedstvo opazovane celice z

lokacije (x, y) zapišemo z izrazom

$$n(c_{x,y}) = \{c_{x-1,y+1}, c_{x,y+1}, c_{x+1,y+1}, c_{x-1,y}, c_{x+1,y}, c_{x-1,y-1}, c_{x,y-1}, c_{x+1,y-1}\}. \quad (8.2)$$

Na sliki 8.1 je predstavljena opazovana celica na lokaciji (x, y) in njene sosede. Preslikovalno funkcijo f Conway definira na osnovi sledečih *jezikovnih* ali *ling-*

$c_{x-1,y+1}$	$c_{x,y+1}$	$c_{x+1,y+1}$
$c_{x-1,y}$	$c_{x,y}$	$c_{x+1,y}$
$c_{x-1,y-1}$	$c_{x,y-1}$	$c_{x+1,y-1}$

Slika 8.1: Conwayeva definicija sosedstva celice uporabljena v igri življenja.

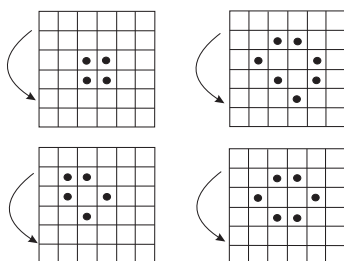
vističnih pravil:

- vsaka živa celica bo na naslednjem koraku mrtva, če ima v sosedstvu na tekočem koraku manj kot dve živi celici,
- vsaka živa celica bo tudi na naslednjem koraku ostala živa, če ima na tekočem koraku v sosedstvu dve ali tri žive celice,
- vsaka živa celica bo na naslednjem koraku mrtva, če ima na tekočem koraku v sosedstvu več kot tri žive celice,
- vsaka mrtva celica bo na naslednjem koraku oživila, če ima na tekočem koraku v sosedstvu natanko tri žive celice.

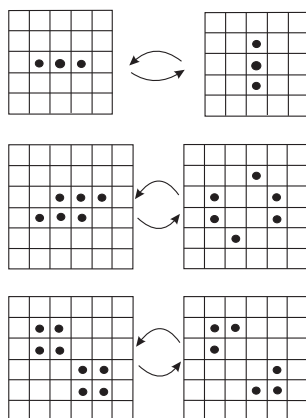
Conway lingvistična pravila formira po hipotetičnem principu razpoložljivih virov za življenje celic. Če je tako živih celic v sosedstvu preveč ali premalo, zmanjka virov za preživljanje celic, če pa so pogoji idealni, mrtve celice preidejo v žive [84]. Avtor začetne konfiguracije stanj CA strukture in njim ustrezne dinamike na osnovi pravil razdeli v naslednje skupine:

- konfiguracije stanj, ki se skozi čas ohranjajo (angl. *still lifes*),
- konfiguracije stanj, ki skozi čas oscilirajo (angl. *oscillators*),
- konfiguracije stanj, ki privedejo do potujočih vzorcev v 2-D prostoru (angl. *space ships*) in
- konfiguracije stanj, ki vodijo do kaotične dinamike, v kateri ni zaznati nekega globalnega časovnega vzorca.

Na sliki 8.2 so predstavljena začetna stanja CA struktur Conway-eve igre življenja, ki se skozi diskretne korake simulacije ohranjajo, na sliki 8.3 pa začetna stanja struktur, ki oscilirajo s periodo 1, pri čemer so na obeh slikah celice



Slika 8.2: Statične CA strukture igre življenja [84]. Vzorci stanj se skozi čas ohranjajo (novo stanje vzorca je enako staremu).

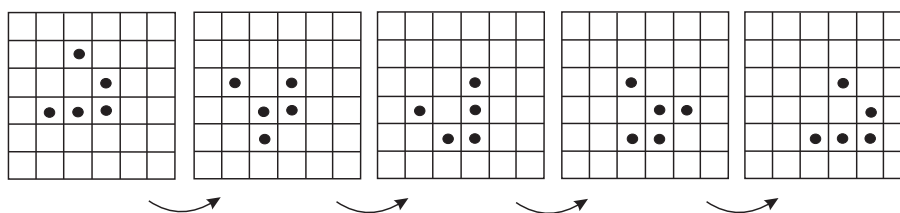


Slika 8.3: Oscilirajoče CA strukture igre življenja s periodo oscilacij 1 [84].

označene s črno piko žive, celice brez označitve s črno piko pa mrtve. Obstajajo tudi začetne konfiguracije, ki vodijo do daljših period oscilacij [84]. Še zanimivejše so začetne konfiguracije, ki vodijo do „potujočih“ vzorcev po površini homogenega 2-D prostora celic. Zgled tovrstnega primera „drsalca“ (angl. *glider*) je prikazan na sliki 8.4. Oblikovno se vzorec obnovi po štirih korakih dinamike, pri čemer se premakne za eno mesto po diagonali desno navzdol v koordinatnem sistemu. Tako se koordinate *globalnega vzorca* živih celic po vsakih štirih korakih dinamike spremenijo po pravilu

$$x(t + 4) = x(t) + 1, y(t + 4) = y(t) - 1. \quad (8.3)$$

Izkaže se, da z omenjeno igro lahko zgradimo Turingov stroj, s čimer se model izkaže kot univerzalen koncept za vzpostavitev modela poljubnega diskretnega dinamičnega procesa.



Slika 8.4: Potujoča CA struktura „drsalca“ v igri življenja v 2-D prostoru [84].

8.2 Enodimensionalni celularni avtomati

Za začetek zapišimo najpreprostejšo definicijo celularnih avtomatov v enodimensionalnem (1-D) prostoru, povzeto po [85].

Definicija 9 1-D celularni avtomat je sestavljen iz zaporedja celic. Zaloga njihovih stanj je diskretna in končna - posamezna celica se lahko nahaja v enem od k možnih stanj. Ob predpostavki, da je prostor celic indeksiran (vsaka celica ima svoj indeks i ($i = 1, \dots, n$), preko katerega se lahko nanjo sklicujemo), se stanje vsake celice c_i spreminja skozi diskretne časovne korake na osnovi pravila, ki je enako za vse celice. Velja izraz

$$c_i(t+1) = f(c_{i-r}(t), c_{i-r+1}(t), \dots, c_{i+r-1}(t), c_{i+r}(t)), \quad (8.4)$$

pri čemer f predstavlja preslikovalno funkcijo, $c_i(t)$ stanje i -te celice v času t , r pa oddaljenost vplivajočih celic na formacijo stanja celice c_i v času $(t+1)$.

Na osnovi končne in diskretne zaloge stanj posamezne celice c_i pridemo do ugotovitve, da ima preslikovalna funkcija f kar k^{2r+1} vhodnih vektorjev in da obstaja kar $k^{k^{2r+1}}$ različnih takšnih preslikovalnih funkcij.

Definicija 10 V primeru, da ima 1-D celularni avtomat zalogo stanj $\{0, 1\}$ ($k = 2$), da je $r = 1$ in da je preslikovalna funkcija f Boolean-ova funkcija, takšen 1-D celularni avtomat imenujemo za elementaren.

Elementarnost 1-D celularne strukture tako izhaja iz majhnega neposrednega vplivnega sosedstva ($r = 1$), majhne dvovrednostne zaloge vrednosti stanj ($k = 2$) in enostavne trovhodne dvovrednostne preslikovalne funkcije. V primeru predhodno navedene definicije sosedstvo i -te celice predstavljata zgolj njena leva in desna soseda v zaporedju.

$c_{i-1}(t)$	$c_i(t)$	$c_{i+1}(t)$	$c_i(t+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Tabela 8.1: Primer preslikovalne funkcije - pravila „90“, $r = 1$.

Podano definicijo 1-D celularnih struktur dodatno pojasnimo z njihovimi sledečimi značilnostmi [85]:

- prostorska diskretnost (angl. *discrete in space, discrete lattice of cells*);
- časovna diskretnost (angl. *discrete in time*): na vsakem diskretnem časovnem koraku vsaka celica ažurira svoje stanje glede na stanja sosednjih celic;
- diskretnost in končnost zaloge stanj posamezne celice (*discrete in states*),
- homogenost celic v kontekstu njihove identičnosti (vse celice imajo enako preslikovalno funkcijo ali enak program);
- sinhronsko ažuriranje vrednosti stanj vseh celic;
- determinističnost preslikovalne funkcije;
- prostorska lokalnost preslikovalne funkcije: na novo stanje opazovane celice vplivajo le stanja sosednjih celice, iz česar izhaja, da imamo opravka z lokalnimi interakcijami;
- temporalnost preslikovalne funkcije - deluje na osnovi časovno opredeljenih vrednosti stanj s časovno omejitvijo „vpogleda v zgodovinska stanja“ posameznih celic;

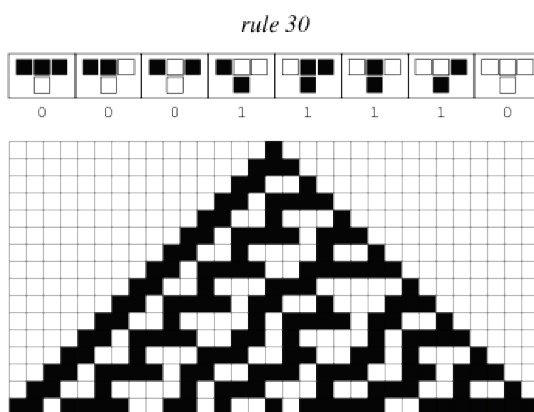
Predpostavimo, da definiramo preslikovalno funkcijo, kot je prikazano v tabeli 8.1. Na osnovi tabele ugotovimo, da imamo pri enostavnem sosedstvu leve in desne celice možnost določiti $2^8 = 256$ različnih preslikovalnih funkcij ali pravil. Pravilo, predstavljeno v tabeli 8.1, deluje po logiki modula po vsoti stanj sosedov ($c_{i-1}(t) \oplus c_{i+1}(t)$). Imenujemo ga tudi pravilo „90“ po principu polinomskega zapisa izhodnih vrednosti preslikovalne funkcije

$$0 * 2^7 + 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 90, \quad (8.5)$$

pri čemer glede na polinomski zapis, definirano sosedstvo celic in njihovo zalogo vrednosti stanj poznamo pravila z označbami od „0“ do „255“.

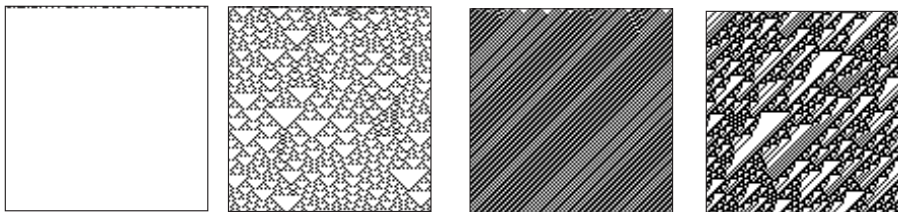
8.2.1 Primeri evolucij 1-D avtomatov

Primer časovno krajše evolucije pravila „30“ je prikazan na sliki 8.5, povzeti po viru [86], pri čemer so potemnjene celice v stanju 1, nepotemnjene pa v stanju 0. Gornji del slike grafično predstavi delovanje pravila, spodnji mrežni del slike pa njegovo *evolucijo*, ali spreminjanje stanj celic skozi čas. Pri tem prva gornja vrstica v mreži predstavlja stanja celic v času t , druga vrstica v mreži stanja celic v času $t + 1$ itd. Začetno stanje sistema predstavljeno v prvi vrstici je definirano z le eno celico v stanju 1. Primeri časovno daljših evolucij pravil „0“,



Slika 8.5: Primer časovno krajše evolucije pravila „30“ [86].

„18“, „45“ in „106“ so predstavljeni na sliki 8.6, povzeti po viru [87]. Tudi v tem primeru se evolucije pričnejo z gornjimi vrsticami in tečejo po vrsticah navzdol v skladu z diskretnimi koraki. Začetno stanje sistema je definirano z naključno določenimi stanji celic.



Slika 8.6: Časovno daljše evolucije pravil „0“, „18“, „45“ in „106“ [87].

8.2.2 Klasifikacija in interpretacija evolucij 1-D avtomatov

V predhodnem razdelku smo prikazali evolucije posameznih 1-D elementarnih celularnih avtomatov. Glede na dinamiko, ki jo povzroči posamezno pravilo, lahko pravila razdelimo v sledeče *kvalitativne razrede* [85]:

- evolucija skozi čas vodi v homogenost stanj (vsa stanja v CA strukturi postanejo enaka); primeri tovrstnih pravil so pravila 0, 4, 16, 32, 48, 54, 60 in 62;
- evolucija skozi čas vodi v stabilne dele podstruktur ali ciklične dele podstruktur; primeri tovrstnih pravil so pravila 8, 24, 40, 56 in 58;
- evolucija skozi čas vodi v kaotične vzorce; primeri tovrstnih pravil so 2, 6, 10, 12, 14, 18, 22, 26, 28, 30, 34, 38, 42, 44, 46 in 50;
- evolucija skozi čas vodi v kompleksne lokalne strukture z dolgotrajnim razvojnim potekom; primeri tovrstnih pravil sta 20 in 52;

Pregled delovanja vseh pravil je bralcu dosegljiv na spletni povezavi [88].

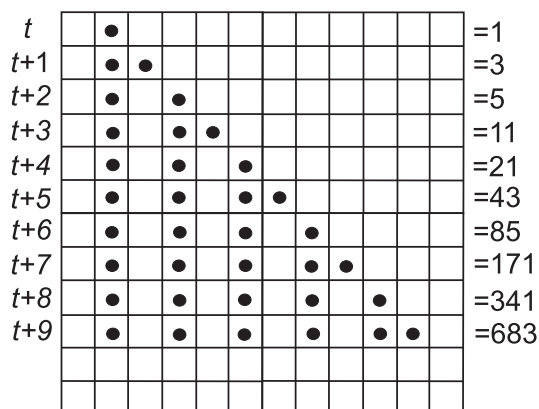
Na tem mestu se lahko vprašamo, kako izkoristiti dinamiko vzorcev, ki se nam na takšen procesno nezahteven način tvorijo. V strokovni literaturi najdemo več izhodišč za izkoriščanje evolucijskih rezultatov. Le ti so sledeči:

- izkoriščanje dinamike vzorcev za potrebe hitrega grafičnega procesiranja,
- izkoriščanje dinamike vzorcev za potrebe pomenskega grafičnega procesiranja,
- izkoriščanje dinamike vzorcev kot funkcijskih generatorjev itd.

Na tem mestu osvetlimo uporabo 1-D CA struktur kot funkcijskih generatorjev. Predpostavimo, da imamo opravka s pravilom „28“ in da imamo v začetnem stanju CA strukture le eno celico s stanjem 1, ostale celice pa so v stanju 0, kot je prikazano v gornji vrstici slike 8.7. Iz tega začetnega stanja sprožimo evolucijo. Ob predpostavki, da skrajno desno celico z vrednostjo 1 na posameznem evolucijskem koraku (v posamezni vrstici) smatramo kot pozicijo, ki označuje 0-to pozicijo bitnega zapisa (2^0), ostale celice pa z velikostjo odmika i proti levi utežene s polinomskim nastavkom 2^i , pridemo do tvorbe števil 1, 3, 5, 11, 21, 43, 85, 171, 341, 683 itd. Omenjeno zaporedje lahko ponazorimo z matematično funkcijo iz izraza

$$f(k) = \frac{4 * 2^k - (-1)^k}{3}, \quad (8.6)$$

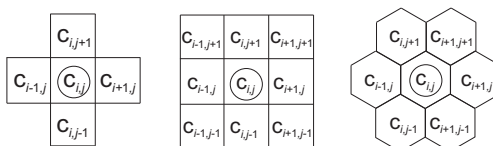
kar pomeni, da pravilo „28“ predstavlja generator funkcije $f(k)$, pri čemer k predstavlja indeks koraka evolucije.



Slika 8.7: Pravilo „28“ kot funkcijski generator [87].

8.3 Dvodimenzionalni celularni avtomati

Dvodimenzionalni (2-D) celularni avtomati se od 1-D CA struktur razlikujejo le v dimenziji prostora, v katerem so regularno razporejene celice. Zopet imajo vse celice enako in enostavno prehajalno funkcijo (pravilo ali program), zaloge stanj celic temeljijo na diskretni končni množici stanj, več alternativ pa se nam ponuja z vidika sosednosti. Na sliki 8.8 so tako predstavljene Von Neumannova (levo), Mooreova (na sredini) in heksagonalna sosednost (desno) kot tri najpogostejše definicije sosednosti v 2-D CA strukturah.

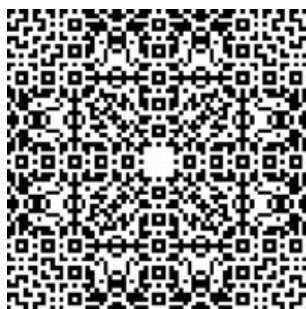


Slika 8.8: Von Neumannova (levo), Mooreova (sredina) in heksagonalna (desno) definicija sosednosti v 2-D CA strukturah.

Kot zgled 2-D CA strukture predstavimo primer *paritetnega pravila* ob upoštevanju Von Neumannove sosednosti. V tem primeru se z vidika opazovane celice na stanja celic sklicujemo z indeksi c (angl. *center*), ki predstavlja stanje opazovane celice ter indeksi n (angl. *north*), e (angl. *east*), s (angl. *south*) in w (angl. *west*), ki predstavljajo stanja sosednjih celic. Zgled je povzet po viru [89], pravila pa so predstavljena v tabeli 8.2. Na sliki 8.9 je prikazana evolucija paritetnega pravila po 120 korakih.

$cnesw(t)$	$c(t+1)$	$cnesw(t)$	$c(t+1)$	$cnesw(t)$	$c(t+1)$	$cnesw(t)$	$c(t+1)$
00000	0	01000	1	10000	1	11000	0
00001	1	01001	0	10001	0	11001	1
00010	1	01010	0	10010	0	11010	1
00011	0	01011	1	10011	1	11011	0
00100	1	01100	0	10100	0	11100	1
00101	0	01101	1	10101	1	11101	0
00110	0	01110	1	10110	1	11110	0
00111	1	01111	0	10111	0	11111	1

Tabela 8.2: Paritetno pravilo.



Slika 8.9: Evolucija paritetnega pravila v 2-D CA strukturah po 120 korakih.

8.4 Simulacijska programska orodja

Za potrebe modeliranja in simulacij evolucije struktur celularnih avtomatov je na razpolago veliko število prosto dostopnih, kot tudi plačljivih programskih orodij. Pregled razpoložljivih orodij je dosegljiv v viru [90].

8.5 Povzetek poglavja

V pričujočem poglavju smo predstavili osnove celularnih avtomatov. Njihove osnovne značilnosti so sledeče:

- opravka imamo s paralelizmom proženja dinamike celic - avtomatov,
- vse celice (avtomati) imajo enak enostaven program,
- doseči želimo željeni globalni vzorec dinamike na osnovi lokalnih interakcij in
- opravka imamo z regularnostjo prostora celic.

Celularni avtomati predstavljajo že dolgo prisotno paradigmo procesiranja, ki se manifestira v rešitvah kot so paralelizacija reševanja težjih računskih problemov, senzorska omrežja, QCA strukture itd. Zgodovinsko gledano predstavljajo CA strukture enega od prvih konceptualnih odmikov od klasične Von Neumannove arhitekture računalnikov z njenim „zloglasnim“ ozkim grlom.

Literatura

- [1] “The scale of things.” <http://science.energy.gov/bes/community-resources/scale-of-things-chart/>, September 2016.
- [2] M. Hak, *The MEMS Handbook*. CRC Press, 2002.
- [3] M. Mack, “The multiple lives of Moore’s law,” *IEEE Spectrum*, vol. 4, pp. 29–35, 2015.
- [4] D. Kodek, *Arhitektura in organizacija računalniških sistemov*. Bi-Tim, Slovenija, 2008.
- [5] *From editors of Scientific American: Understanding nanotechnology*. Warner Books, ZDA, 2002.
- [6] “2001: A Space Odyssey,” 1968.
- [7] A. Adamatzky, B. Costello, and T. Asai, *Reaction diffusion computers*. Elsevier, 2005.
- [8] B. Hayes, “Third base,” *American Scientist*, vol. 89, no. 6, 2001.
- [9] W. Aspray, *John Von Neumann and The Origins Of Modern Computing*. The MIT Press, England, 1990.
- [10] “There’s Plenty of Room at the Bottom.” https://en.wikipedia.org/wiki/There%27s_Plenty_of_Room_at_the_Bottom/, September 2016.
- [11] E. Regis, *Nano – the emerging science of nanotechnology*. BackBay Books, 1995.
- [12] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, “Quantum cellular automata,” *Nanotechnology*, vol. 4, 1993.
- [13] C. Lent and P. Tougaw, “Lines of interacting quantum-dot cells: a binary wire,” *Journal of Applied Physics*, vol. 74, 1993.
- [14] I. L. Bajec and M. Mraz, “Večstanjsko procesiranje v strukturah kvantnih celičnih avtomatov,” *Elektrotehniški vestnik*, vol. 73, no. 2-3, 2006.

- [15] P. Pečar, "Uporaba adiabatnega pristopa pri realizaciji trojiškega procesiranja na osnovi kvantnih celičnih avtomatov," Master's thesis, Faculty of computer and Information science, University of Ljubljana, 2007.
- [16] G. Snider, A. Orlov, I. Amlani, and G. Bernstein, "Quantum-dot cellular automata: line and majority logic gate," *Japanese Journal of Applied Physics*, vol. 38, 1999.
- [17] K. Walus, T. Dysart, G. Jullien, and R. Budiman, "Qcadesigner: a rapid design and simulation tool for quantum dots cellular automata," *IEEE Transactions on Nanotechnology*, vol. 3, 2004.
- [18] T. Cole and J. Lusth, "Quantum-dot cellular automata," *Progress in Quantum Electronics*, vol. 25, 2001.
- [19] Z. Kohavi, *Switching and finite automata theory*. McGraw-Hill Inc., USA, 1978.
- [20] M. Niemier and P. Kogge, *Nano, Quantum and Molecular Computing - Implications to High Level Design and Validation*, ch. Origins and Motivations for Design Rules in QCA. Kluwer Academic Publishers, Boston, 2004.
- [21] "Reverzibilnost procesiranja." <http://strangepaths.com/reversible-computation/2008/01/20/en/>, Oktober 2016.
- [22] B. Hayes, "Reverse engineering," *American Scientist*, vol. 94, 2006.
- [23] P. R. Yelekar and S. S. C. Hanson, "Introduction to reversible logic gates and its application," *International journal of computer applications*, 2011.
- [24] S. M. R. Taha, *Reversible logic synthesis methodologies with application to quantum computing*. Springer, Switzerland, 2016.
- [25] P. J. Denning and T. G. Lewis, "Computers that can run backwards," *American Scientist*, vol. 105, no. 5, 2017.
- [26] "Reversible." <http://www.eng.fsu.edu/~symbol{126}mpf/pubs.htm>, Maj 2015.
- [27] "Reversible." <http://www.cise.ufl.edu/research/revcomp/>, Maj 2015.
- [28] K. Perumalla, *Introduction to Reversible Computing*. Chapman & Hall/CRC Press, 2014.
- [29] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, UK, 2009.
- [30] C. Calude and G. Păun, *Computing with cells and atoms, An introduction to quantum, DNA and membrane computing*. Taylor and Francis, London, 2001.

-
- [31] "Single particle interference." <http://player.slideplayer.com/26/8574538/#>, December 2017.
- [32] "Quantum computing." http://en.wikipedia.org/wiki/Quantum_computing, Maj 2015.
- [33] M. Nagy and S. G. Akl, "Quantum computation and quantum information," tech. rep., 2005. Technical report, 2005-496.
- [34] "Elementary quantum notation." <http://www-users.cs.york.ac.uk/schmuel/comp/node6.html>, Maj 2015.
- [35] J. Virant, *Načrtovanje nanoračunalniških struktur*. Didakta, Slovenija, 2007.
- [36] M. Hirvensalo, *Quantum Computing*. Springer Verlag, 2001.
- [37] H. Stöcker, *Matematični priročnik z osnovami računalništva*. Tehniška založba, Slovenija, 2006.
- [38] O. Bronštejn, K. Semendjajev, G. Musiol, and H. Mühlig, *Matematični priročnik*. Tehniška založba, Slovenija, 1997.
- [39] A. Pittenger, *An Introduction to Quantum Computing Algorithms*. Birkhäuser, ZDA, 2000.
- [40] D. Deutsch and R. Jozsa, "Rapid solutions of problems by quantum computation," *Proceedings of the Royal Society of London*, vol. 439, no. 1907, pp. 553–558, 1992.
- [41] "Quantum programming languages." <http://www.dcs.gla.ac.uk/~simon/quantum/>, November 2015.
- [42] "GUI environment for Quantum Computer Simulator." <http://qcad.osdn.jp/>, November 2015.
- [43] "Quantiki homepage." <http://www.quantiki.org/wiki/list-qc-simulators>, November 2015.
- [44] E. Dahl, "Programming with D-Wave: Map coloring problem," tech. rep., D-Wave Systems, 2013.
- [45] "Radix economy." https://en.wikipedia.org/wiki/Radix_economy, October 2017.
- [46] D. Miller and M. Thornton, *Multiple Valued Logic, Concepts and Representations*. Morgan and Claypool Publishers, USA, 2008.
- [47] "Balanced ternary - Wikipedia." https://en.wikipedia.org/wiki/Balanced_ternary/, September 2016.

- [48] J. Connelly, C. Patel, and A. Chavez, "Ternary Computing Testbed 3-Trit Computer Architecture," tech. rep., California Polytechnic State University of San Luis Obispo advised by Professor Phillip Nico, 2008.
- [49] E. Katiyar, "A Treatise on the Fundamentals of Ternary Arithmetic and Logic," *International Journal of Computer Science Engineering*, vol. 6, no. 8, 2017.
- [50] G. Trishala and K. Ragini, "Design and Implementation of Ternary Logic Circuits for VLSI Applications," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, 2020.
- [51] "Ternary computer - Wikipedia." https://en.wikipedia.org/wiki/Ternary_computer/, September 2016.
- [52] "Sistemska in sintezna biologija." <http://web.bf.uni-lj.si/bi/biokemija/SBD/Docs/sinbiol.pdf>, Maj 2015.
- [53] "Sistemska biologija." <http://www.zrss.si/bzid/geni/pdf/baebler-clanek.pdf>, November 2017.
- [54] "Deoksiribonukleinska kiselina." https://hr.wikipedia.org/wiki/Deoksiribonukleinska_kiselina, November 2017.
- [55] M. Ridley, *Genom: biografija človeške vrste*. Učila International, 2002.
- [56] J. D. Watson and A. Berry, *DNK - Skrivnost življenja*. Modrijan založba d.o.o., 2007.
- [57] "Human Genome." https://en.wikipedia.org/wiki/Human_genome, November 2017.
- [58] N. Cristianini and M.W.Hahn, *Introduction to Computational Genomics - A Case Study Approach*. Cambridge University Press, UK, 2007.
- [59] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.
- [60] L. M. Adleman, "Computing with DNA," *Scientific American*, vol. 279, pp. 54–62, 1998.
- [61] C. S. Calude and G. Paun, *Computing with cells and atoms*. Taylor and Francis Inc., 2001.
- [62] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, and et al., "The Sequence of the Human Genome," *Science*, vol. 291, pp. 1304–1351, Feb. 2001.
- [63] J. C. Venter, *Genom mojega življenja*. Modrijan založba d.o.o., 2009.

- [64] “Cost per genome.” https://www.genome.gov/images/content/costpergenome_2017.jpg, November 2017.
- [65] “DNA sequencing.” https://en.wikipedia.org/wiki/DNA_sequencing, November 2017.
- [66] “Cost to sequence human genome.” https://en.wikipedia.org/wiki/DNA_sequencing, November 2018.
- [67] M. Moškon, N. Zimic, and M. Mraz, “Realizacija dvojiškega pomnjenja v preprostih bioloških sistemih,” *Elektrotehniški vestnik*, vol. 83, no. 4, pp. 194–200, 2016.
- [68] G. M. Church, Y. Gao, and S. Kosuri, “Next-Generation Digital Information Storage in DNA,” *Science*, vol. 337, p. 1628, 2012.
- [69] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA,” *Nature*, vol. 494, pp. 77–80, 2013.
- [70] C. Bancroft, T. Bowler, B. Bloom, and C. Clelland, “Long-term storage of information in DNA,” *Science*, vol. 293, no. 5536, pp. 1763–5, 2001.
- [71] G. M. Church and E. Regis, *Regenesis: How Synthetic Biology Will Reinvent Nature and Ourselves*. Perseus Books Group, USA, 2012.
- [72] M. Moškon, *Modeli in metrike dinamike preklopa v enostavnih bioloških sistemih za potrebe računalniških struktur prihodnosti*. PhD thesis, Univerza v Ljubljani, 2012.
- [73] R. Gaber, T. Lebar, A. Majerle, B. Šter, A. Dobnikar, M. Benčina, and R. Jerala, “Designable dna-binding domains enable construction of logic circuits in mammalian cells,” *Nature Chemical Biology*, vol. 10, no. 3, pp. 203–208, 2014.
- [74] “Registry of standard biological parts.” http://parts.igem.org/Main_Page, November 2017.
- [75] M. Stražar, M. Mraz, N. Zimic, and M. Moškon, “An adaptive genetic algorithm for parameter estimation of biological oscillator models to achieve target quantitative system response,” *Natural Computing*, vol. 13, pp. 119–127, 2014.
- [76] “SBML Software Matrix.” http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix, November 2017.
- [77] “SBML Software Summary.” http://sbml.org/SBML_Software_Guide/SBML_Software_Summary, November 2017.
- [78] L. Borkowski, *Jan Łukasiewicz Selected works*. North-Holland Publishing Company, 1970.

- [79] I. L. Bajec, N. Zimic, and M. Mraz, "Towards the bottom-up concept: extended quantum-dot cellular automata," *Microelectronic engineering*, vol. 83, no. 4/9, 2006.
- [80] I. L. Bajec, N. Zimic, and M. Mraz, "The ternary quantum-dot cell and ternary logic," *Nanotechnology*, vol. 17, no. 8, 2006.
- [81] P. Pečar, "Uporaba adiabatnega pristopa pri realizaciji trojiškega procesiranja na osnovi kvantnih celičnih avtomatov," Master's thesis, Faculty of computer and Information science, University of Ljubljana, 2007.
- [82] P. Pečar, M. Mraz, N. Zimic, and I. L. Bajec, "Solving the ternary quantum-dot cellular automata logic gate problem by means of adiabatic switching," *Japanese journal of applied physics*, vol. 47, no. 6, 2008.
- [83] P. Pečar, A. Ramšak, N. Zimic, M. Mraz, and I. L. Bajec, "Adiabatic pipelining: A key to ternary computing with quantum dots," *Nanotechnology*, vol. 19, no. 49, 2008.
- [84] "Conway's Game of Life." https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life, September 2016.
- [85] S. Wolfram, *Theory and Applications of Cellular Automata*. World Scientific, Singapore, 1986.
- [86] "Mathworld: Rule 30." <http://mathworld.wolfram.com/CellularAutomaton.html>, Oktober 2016.
- [87] "Wikipedia - different rules evolution." https://en.wikipedia.org/wiki/Elementary_cellular_automaton#Rule_28, Oktober 2016.
- [88] "Elementary cellular automata." <http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>, Oktober 2018.
- [89] M. Sipper, *Evolution of Parallel Cellular Machines*. Springer Inc., UK, 1997.
- [90] "Orodja za simulacije evolucije CA." http://uncomp.uwe.ac.uk/genaro/Cellular_Automata_Repository/Software.html, Oktober 2016.