**SVEUČILIŠTE U SPLITU**

**FAKULTET ELEKTROTEHNIKE, STROJARSTVA I**

**BRODOGRADNJE**

**ZAVRŠNI RAD**

**MODELIRANJE ŠIRENJA ZARAZE POMOĆU STANIČNIH AUTOMATA**

**Jere Zambarlin**

**Split, rujan 2021**

**Sadržaj**

[1. Uvod 2](#_Toc82731686)

[1.1 Općenito o staničnim automatima 2](#_Toc82731687)

[1.2 Igra života 3](#_Toc82731688)

[1.3 SIR Model 3](#_Toc82731689)

[2. Programsko okruženje 5](#_Toc82731690)

[2.1 Julia 5](#_Toc82731691)

[2.2 Pluto Bilježnice 5](#_Toc82731692)

[3. Modeli i kod 6](#_Toc82731693)

[3.1 Test Funkcija 6](#_Toc82731694)

[3.2 Model 100 7](#_Toc82731695)

[3.3 Dinamički model 13](#_Toc82731696)

[4. Zaključak 16](#_Toc82731697)

[5. Sažetak 17](#_Toc82731698)

[6. Summary 18](#_Toc82731699)

[7. Popis literature: 19](#_Toc82731700)

# 

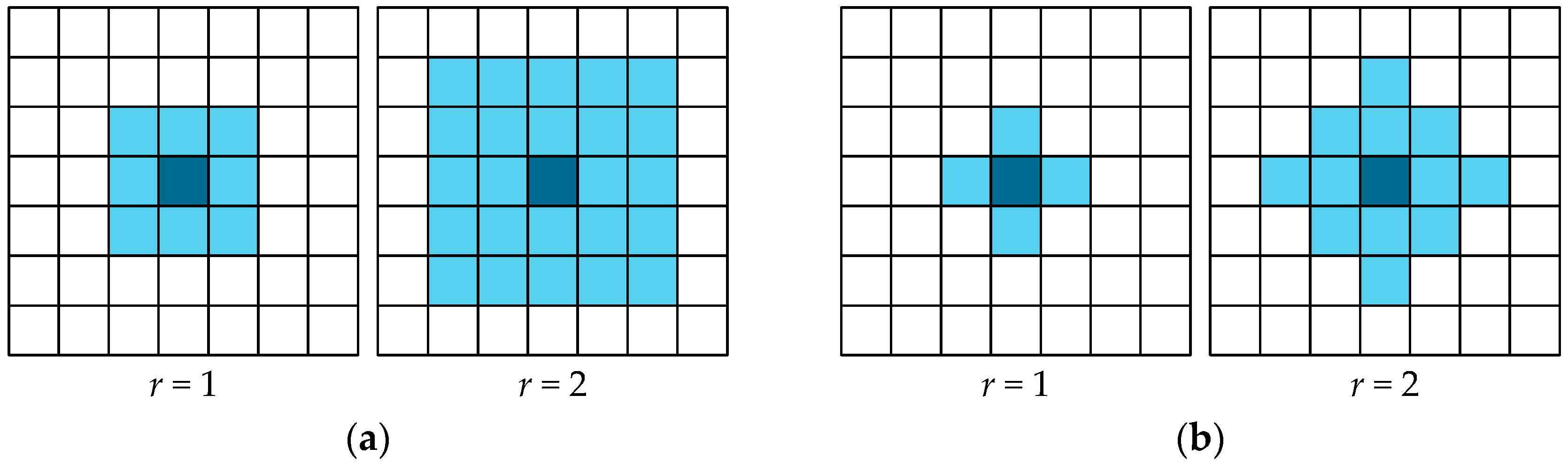
# **1. Uvod**

## 

## **1.1 Općenito o staničnim automatima**

Stanični ili ćelijski automati su način matematičkog ili računalnog modeliranja. Za modeliranje, koristi se dvodimenzionalna mreža podijeljena u ćelije gdje svaka ćelija ima određene susjede [1][2]. Ako je ploha veličine r x c, tada ju možemo zapisati kao ravninu C na sljedeći način



Postoje dvije metode određivanja susjeda, to su Mooreova i von Neumannova. U Mooreovoj metodi, svakoj su ćeliji susjedne one koje je direktno diraju, odnosno, sveukupno ih je 9, uključujući i samu sebe. U von Neumannovoj metodi, susjedne su ćelije one koje su iznad, ispod, lijevo i desno od početne ćelije, sveukupno 5, uključujući i početnu. U proširenom susjedstvu, možemo reći da su početnoj ćeliji susjedne i one koje su za r=2 udaljene od nje.

Na slikama je početna ćelija označena tamno plavom bojom, a njene susjedne su svijetlom. Na lijevoj slici (a) se radi o Mooreovom susjedstvu, a na desnoj (b) o von Neumannovom susjedstvu. Vidimo primjer kako to izgleda za slučaj da gledamo samo prve susjede, kao i prošireno susjedstvo. Možemo to zapisati i vektorski. Za Mooreovu metodu vrijedi da je

V = {(0,0),(0,1),(0,-1),(-1,0),(-1,-1),(-1,1),(1,-1),(1,0),(1,1)}

Za von Neumannovu metodu vrijedi da je

V = {(0,0)(-1,0)(1,0)(0,-1)(0,1)}

U staničnim automatima, svaka ćelija utječe na sve svoje susjede, ali i susjedi na nju. Da bi model imao smisla, moramo ga gledati kao konačnu plohu, a onda se postavlja pitanje, što s ćelijama na rubovima. Oni imaju manje susjeda od onih u sredini. Ukoliko bismo zamislili da plohu savijemo, te zalijepimo lijevi za desni rub, te gornji i donji, dobili bismo torusni oblik naše mreže. Drugim riječima, u konačnoj mreži, ukoliko se promatrana ćelija nalazi na dnu mreže, njen donji susjed će bit ona ćelija u istom stupcu, ali na vrhu mreže. Analogno vrijedi za sve ćelije na rubovima.

Ovakvi su automati diskretni i konačni. U trenutku t=0, svaka od ćelija ima svoje početno stanje, te se u idućem diskretnom trenutku, t=1, može, ali i ne mora promijeniti vrijednost svake od njih u ovisnosti našem modelu, odnosno gledajući način na spregu svake ćelije s njenim susjedima.

Većina matematičkih modela koristi diferencijalne jednadžbe za proračun. Međutim, to dolazi uz određene nedostatke jer se određene stvari poput ponašanja osoba, miješanja ljudi i sličnih stvari, ne uzimaju u obzir. Zbog tog razloga je došlo do potrebe za staničnim automatima, te oni rješavaju te probleme.

## **1.2 Igra života**

Igra života je jedan od poznatijih primjera kako funkcioniraju stanični automati. [3]

Britanski matematičar John Horton Conway je 1970. godine kreirao automat pod nazivom „Igra života“. U njegovom primjeru, svaka ćelija može imati 2 stanja, „živo“ i „mrtvo“. I prije toga se govorilo o staničnim automatima, ali ovo ih je populariziralo. Naime, Coway je kreirao automat s 4 jednostavna pravila koristeći Mooreovo susjedstvo:

1. Svaka živa ćelija sa manje od 2 živa susjeda nakon diskretnog vremena umire
2. Svaka živa ćelija s 2 ili 3 živa susjeda živi i dalje
3. Svaka živa ćelija s više od 3 živa susjeda umire
4. Svaka mrtva ćelija sa točno 3 živa susjeda postaje živa

Ovaj model je lako razumljiv jer ga možemo promatrati kao život ljudi. Ćelije umiru zbog prenapučenosti ili zbog nedostatka populacije, a žive zbog reprodukcije ili generacije. U ovom modelu, ćelije se promatraju binarno, odnosno mogu imati samo 2 stanja. Mogli smo ih označavati i kao crne i bijele ili 0 i 1 na primjer.

Na ovaj način se može modelirati širenje zaraze po nekim pravilima. Što se zaraza lakše prenosi, to će biti potreban manji broj zaraženih susjeda zdravoj ćeliji da se razboli i što zaraza duže traje, to će više diskretnih trenutaka biti potrebno ćeliji da ozdravi, ako su u tim trenucima ispunjeni uvjeti za to

U ovom radu će se širenje zaraze modelirati s 2 modela. Model 100 i Dinamički model ne koriste ni Mooreovo ni von Neumannovo susjedstvo, već slučajnost uz određenu statističku vjerojatnost. Taj način prikaza je intuitivniji za razumijevanje i pregledniji.

## **1.3 SIR Model**

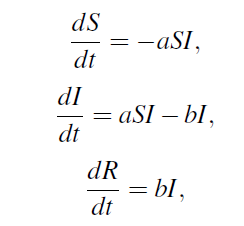
SIR model je model širenja zaraze koristeći stanične automate, te će on biti osnova u ovome radu. [1] Ime dolazi iz engleskog jezika i opisuje 3 moguća stanja ljudi.

S (eng. Susceptible) – Osoba koja je zdrava, ali je podložna zarazi

I (eng. Infected) – Zaražena osoba koja širi zarazu

R (eng. Recovered) – Imuna osoba koja je zdrava te se ne može zaraziti niti širiti zarazu

Ovaj se model nekada proširuje u SEIR model, gdje su E osobe koje su zaražene, ali ne prenose zarazu. Izbor modela će ovisiti o bolesti čije se širenje promatra. Kod realnih, novih zaraza, sve počinje od toga da je jedna osoba zaražena (I), te se onda sve ostale osobe potencijalno mijenjaju kroz ove 3 kategorije. Te promjene se lako prikažu s 3 diferencijalne jednadžbe



Gdje je a = vjerojatnost zaraze; b = vjerojatnost oporavka. U ovakvim modelima, pretpostavka je da nema mrtvih, ili da je njihov broj zanemarivo mali u odnosu na ukupan broj ostalih, tako da u svakom trenutku vrijedi S(t)+I(t)+R(t) = S(t+T)+I(t+T)+R(t+T). U staničnim automatima se vjerojatnost za zarazu najčešće modelira eksponencijalnom funkcijom gdje je x neka konstanta, a n broj zaraženih susjeda.

Za inspiraciju za ovaj rad, poslužio je projekt sa sveučilišta u Massachusettsu, SAD

# **2. Programsko okruženje**

## **2.1 Julia**

Julia je programski jezik koji se pojavio 2012. godine. Glavni osnivači su Jeff Bezanson, [Stefan Karpinski](https://en.wikipedia.org/wiki/Stefan_Karpinski), [Viral B. Shah](https://en.wikipedia.org/wiki/Viral_B._Shah), and [Alan Edelman](https://en.wikipedia.org/wiki/Alan_Edelman). Oni su počeli raditi na projektu 2009. godine s ciljem da stvore jedan veoma popularan i brz programski jezik. Do 2020. godine, koristilo ga je više od 10000 kompanija diljem svijeta i više od 1500 sveučilišta.

U 2015 godini, banka sustava federalnih rezervi u New Yorku je u Juliji napravila model ekonomije SAD-a koji je bio 10 puta brži nego njegova prethodna verzija napravljena u MATLAB-u.

Danas, poznato američko sveučilište MIT polaže sva prava na Juliju, te rade mnogobrojne projekte. [4]

## **2.2 Pluto Bilježnice**

Pluto bilježnice su jedan od načina za programiranje u Juliji [5]. To je interaktivno programsko okruženje te se svaka funkcija programira u svojoj ćeliji. Instalacija je jednostavna, a kodiranje se u potpunosti odvija u Juliji. Nema nikakvih skrivenih dijelova, sve je vidljivo korisniku, tzv. „User friendly“

Kod Pluto bilježnica, velika je prednost u odnosu na ostala programska okruženja ta interaktivnost. Kada se u bilo kojem dijelu koda promijeni neka varijabla, gdjegod je ta varijabla uključena, tu se također promijeni te se rezultati odmah vide.

Kodiranje se ne mora odvijati po redu. Ako neka funkcija sadrži varijablu koja nije definirana prije same funkcije, ona će provjeriti je li definirana nakon. Moguće je korištenje vanjskih biblioteka za kodiranje, kao što ćemo vidjeti u ovome radu.

# **3. Modeli i kod**

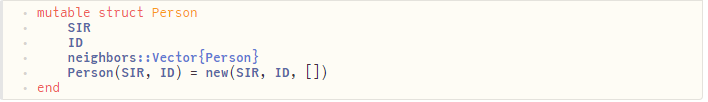
## 

## **3.1 Test Funkcija**

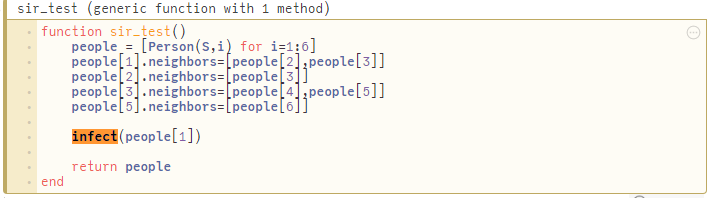
[6]



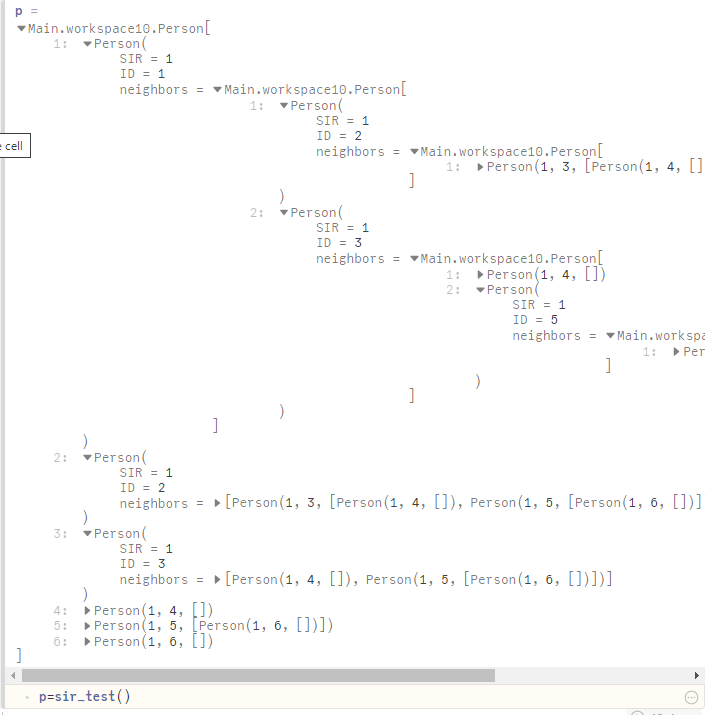
Ovo su biblioteke koje program koristi. One su potrebne da bi kod normalno funkcionirao jer se u njima sadrži sintaksa određenih funkcija koje Pluto kao takav ne prepoznaje



Definira se struktura „Person“ odnosno osoba koja sadrži identifikacijske podatke. Svakoj je osobi pridružena jedna od varijabli (S,I ili R), te svaka osoba ima svoj ID (od 1 do 100)



Testna funkcija sir\_test za definiranje susjeda. U primjeru su susjedi ručno definirani, sve su osobe postavljene u stanje S, te na kraju osoba 1 postaje zaražena

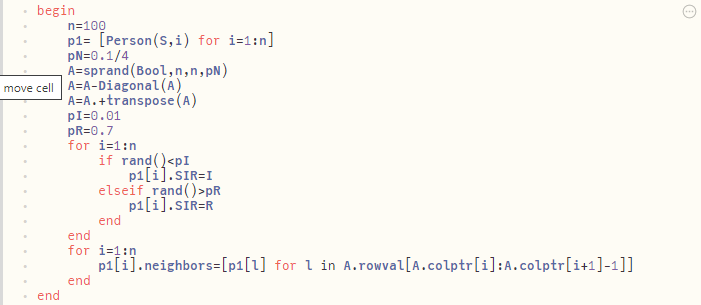


Pozivanje funkcije sir\_test(). Kako je osoba broj 1 zaražena a ona je susjed sa svima (direktno i indirektno), dok se iterira kroz sve članove, na kraju će svih 6 osoba biti zaraženo. To vidimo u prvom članu gdje svi imaju broj „1“ koji definira zaraženu osobu. Također, na slici je detaljnije prikazano susjedstvo za osobu broj 1 kroz iteracije dok se ne dođe do kraja.

## **3.2 Model 100**

Model 100 je model sa 100 ljudi. Grafički su prikazani u krugu sa slučajnim susjedstvom te promatramo što se događa s njima

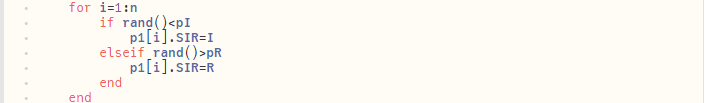
Oslanjajući se na test funkciju, napravljen je „Model 100“. Da bi program uspješno radio, trebalo ga je automatizirati, te je to napravljeno na način da koristi određene vjerojatnosti. Na taj način, svakim novim pokretanjem programa bismo dobili drukčiji ishod iz razloga jer i u prirodi postoje slučajnosti, posebno kada je u pitanju širenje zaraze. Zaražena osoba ako je došla u kontakt sa zdravom osobom, potencijalno je može, ali i ne mora zaraziti



Ovo je glavna funkcija odgovorna za rad ovog programa koja će sada biti objašnjena po djelovima

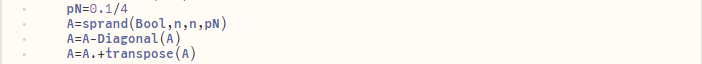


Na početku se definira broj osoba, koji je 100 za ovaj model, te se svaku osobu postavlja u stanje S, odnosno da je zdrava, ali podložna zarazi

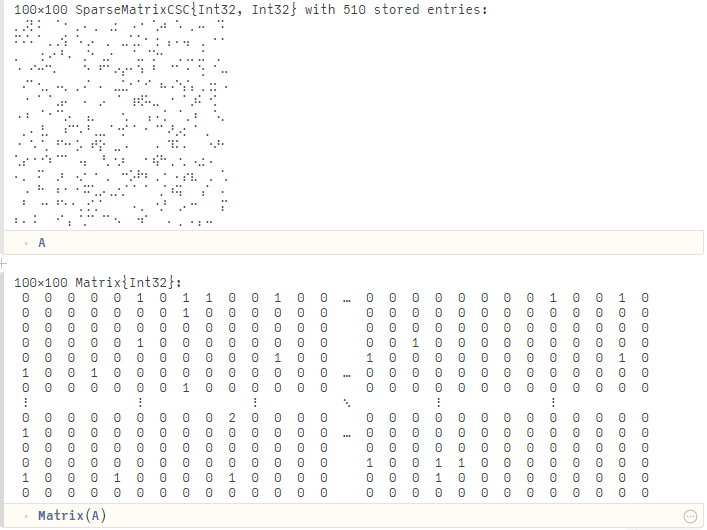


U „for“ petlji se izabire nasumičan broj između 0 i 1 i tako 100 puta, za svaku osobu po jednom. Ako je dobiveni broj manji od pI(0.01), tada osoba koja je prvotno definirana kao S, prelazi u stanje I. Dok, ako je taj broj veći od pR(0.7), osoba iz prethodno definiranog stanja S, prelazi u stanje R. Na taj način se dobije u prosjeku 1% zaraženih, 30% imunih, te ostalih 69% zdravih, ali podložnih bolesti.

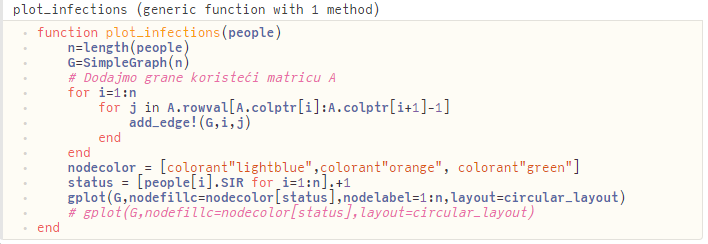
Ovo stanje opisuje početak, prije negoli se zaraza počela širiti.



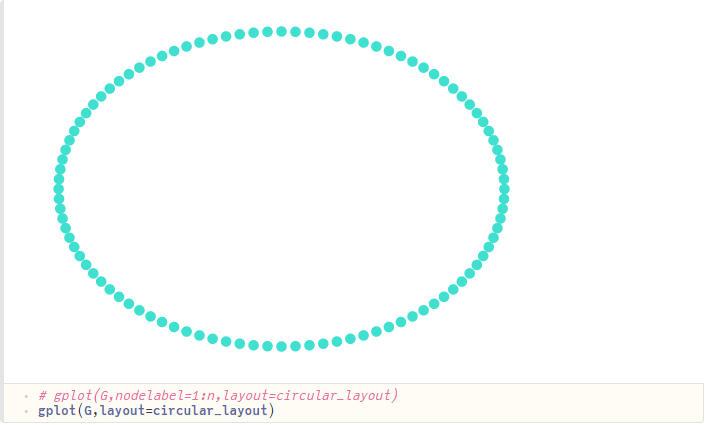
Definira se matrica dimenzija n\*n koja će biti odgovorna za određivanje susjeda. Ukoliko je element prvog stupca i trećeg retka jedinica, to znači da su prva i treća osoba susjedi, i tako vrijedi za sve kombinacije svih 100 ljudi. Također je jasno da matrica ne smije imati dijagonalu jer osoba ne može sama sebi biti susjed, te treba biti simetrična.

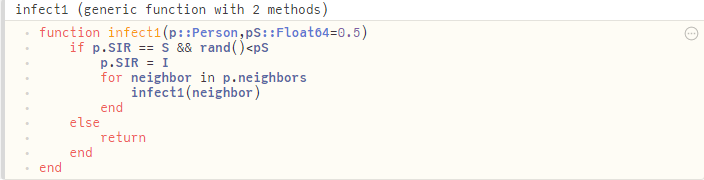


Izgled matrice 100x100

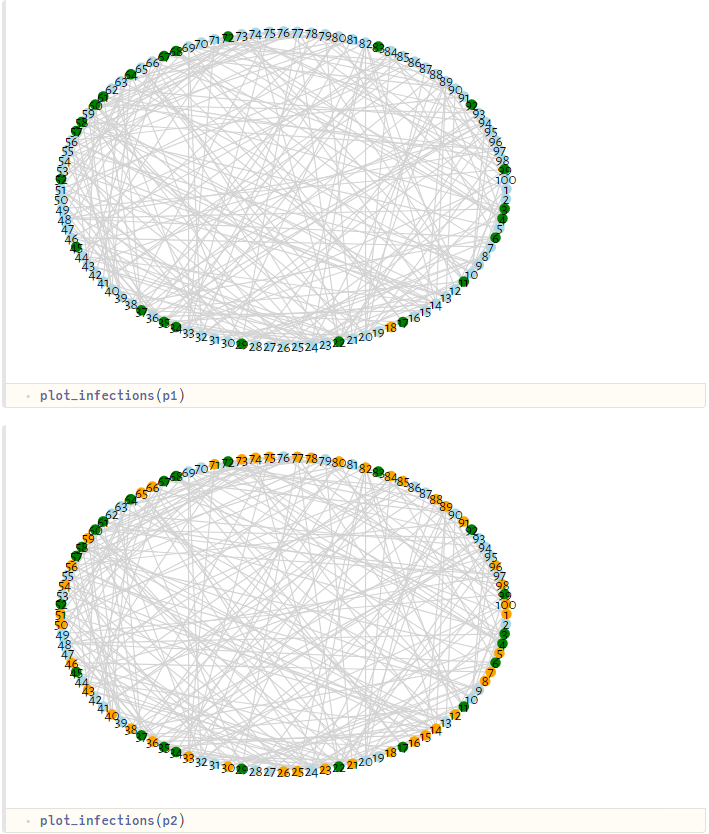


Crtanje grafa i dodavanja linija koje grafički prikazuju vezu među susjedima.



Općeniti izgled grafa

Funkcija za širenje zaraze. U ovoj funkciji postoji također nasumičnost, odnosno vjerojatnost od 50% da zaražena osoba zarazi svog susjeda ukoliko je on zdrav.

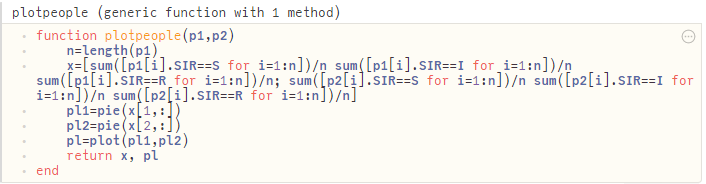


Grafovi na početku, te nakon što se zaraza proširila.

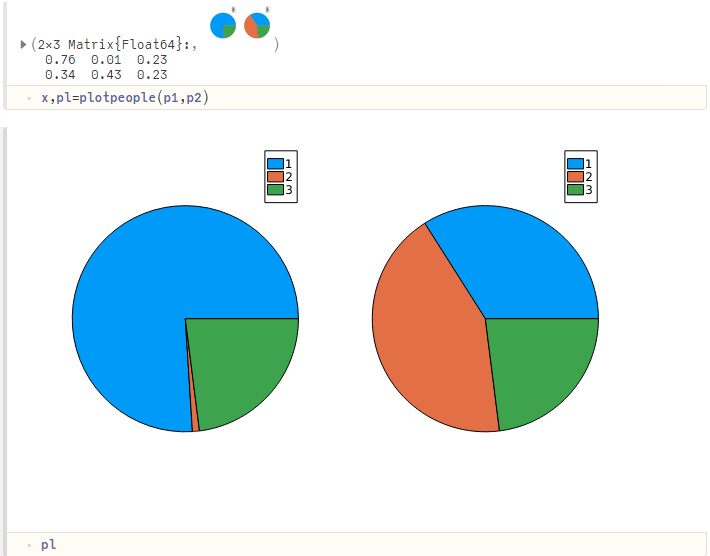
Svijetlo plavo – Zdrave osobe podložne zarazi (S)

Narančasto – Zaražene osobe (I)

Zeleno – Imune osobe (R)



Funkcija koja crta grafove u obliku kružnog dijagrama s udjelima tipova osoba prije i nakon širenja zaraze



Grafovi prije i nakon uz pripadajuće vjerojatnosti

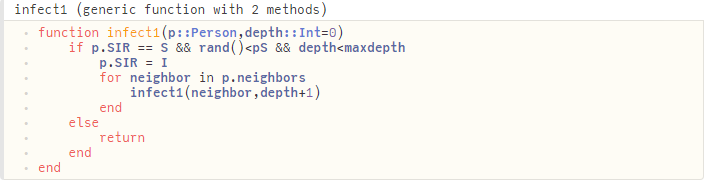
1. (S) osobe
2. (I) osobe
3. (R) osobe

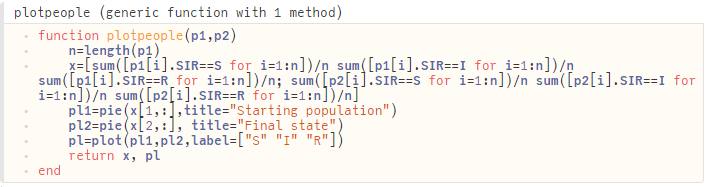
## **3.3 Dinamički model**

Dinamički model je model koji je programiran na način da korisnik može mijenjati parametre po želji. Veći je od Modela 100, i iz tog razloga prikaz sa kružnim grafom ne bi bio pregledan pa su rezultati prikazani kružnim dijagramom

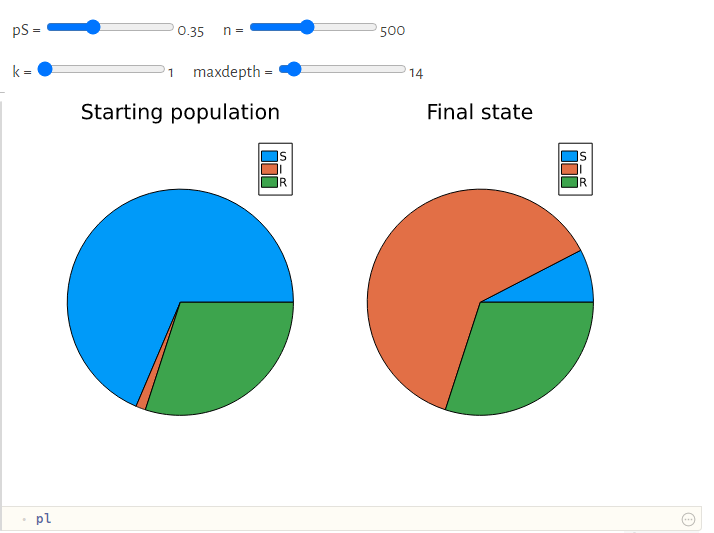


Model je definiran na isti način kao i model 100. Razlika je što je u modelu 100 „n“=100, a u ovom modelu je „n“ promjenjiva varijabla

Osim što se koristi ista funkcija za širenje zaraze kao i u prethodnom modelu, uvodi se nova varijabla „depth“. Ona kaže kroz koliko iteracija potencijalnog širenja zaraze prolazi svaka osoba od postojećih „n“. Da toga nema, funkcija bi iterirala sve dok se sve osobe ne zaraze i onda se ne bi dobio reprezentativan rezultat



Definiraju se 2 grafa, jedan za početno te jedan za konačno stanje. Ovo su također kružni dijagrami koji se crtaju u ovisnosti o postotku pojedinog tipa osoba



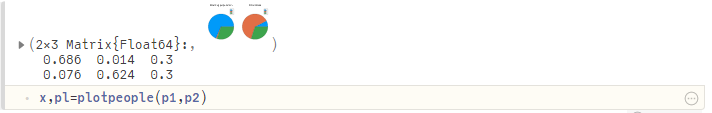
Grafovi početnog i konačnog stanja. Postoje 4 promjenjive varijable koje korisnik može samostalno mijenjati da vidi kako se mijenjaju rezultati u ovisnosti o početnom proizvoljnom stanju

pS – Vjerojatnost da se zaraza prenese sa zaražene na zdravu osobu (sa I na S)

n – Broj osoba

k – Indeks prve zaražene osobe

maxdepth – maksimalan broj iteracija funkcije za širenje zaraze za svaki čvor (osobu)



Matrični prikaz udjela pojedinog tipa osoba. Prvi redak definira početno stanje, drugi redak stanje na kraju. Stupci redom definiraju osobe tipa S, I te R, a brojevi su iskazani u udjelima.

# **4. Zaključak**

Širenje zaraze je teško za predvidjeti. Postoje razni matematički modeli koji su se konstruirali kroz povijest u pokušaju da se može zaraza zaustaviti prije negoli se krene nekontrolirano širiti. Postoje modeli sa diferencijalnim jednadžbama, postoje stanični automati. Međutim, kako se radi o ljudima i o realnoj situaciji, ljudski faktor nije uvijek lako uzeti u obzir. Neki su ljudi odgovorniji, neki su manje odgovorni. Dok jedni putuju više, drugi putuju manje. Dio ljudi je društven i u direktnom kontaktu s više ljudi čime doprinose širenju zaraze, a ostali su povučeniji.

Tehnologija jako brzo napreduje, pa bismo u budućnosti mogli vidjeti sve točnije algoritme te bi prevencija mogla biti puno efikasnija. Globalizacija svakako ne pomaže jer se zaraza lako seli po svijetu. U prošlosti smo imali uglavnom epidemije, a danas se nalazimo usred pandemije, i vjerojatno će sve ostale zaraze koje slijede biti pandemije. Razlika je ta što je epidemija zaraza na ograničenom području, a pandemija je na većoj površini, na kontinentu ili po cijelom svijetu

# **5. Sažetak**

Na početku ovog rada, opisani su općenito stanični automati i način na koji susjednost funkcionira kod njih. Za primjer je uzeta „Igra života“ koja pomoću staničnih automata opisuje ljudski život u ovisnosti u kakvom se okruženju nalaze.

Objašnjen je S(E)IR model i kako on funkcionira u smislu širenja zaraze. Detaljnije su opisani Julia i Pluto bilježnice, jezik i okruženje koji su korišteni za ovaj rad. U njima su kodirana 2 modela koji rade na sličan način uz različite parametre i razlikuju se u nekim sitnicama uz testni kod, koji je preuzet sa sveučilišta MIT u Sjedinjenim Američkim Državama.

Prikazani su Model 100 i Dinamički model kojima grafički vidimo kako se zaraza širi uz točan kod kojim je to postignuto. Korištene su vjerojatnosti za promjenu stanja pojedine osobe.

Ključne riječi: Stanični automati, SIR, Julia, Pluto, širenje zaraze, model

# **6. Summary**

At the beginning, cellular automata was described in general and the way neighboring is affecting the state of the cells. Conway’s “Game of Life” was chosen to represent an example of cellular automata which describes the human life depending on surrounding of each individual

S(E)IR model is explained and how it works when it comes to disease spread. Julia and Pluto Notebooks are explained in some details. They are programming language and environment used to program these 2 models. They work similarly with some slight differences. There is also a test code that was taken from the MIT university, USA.

With Model 100 and Dynamic model, we can see how does the disease spread with exact code that has been used. Statistical probability is taken into the consideration when coding them to get the state of every individual.

Key words: Cellular automata, SIR, Julia, Pluto, disease spread, model

# **7. Popis literature:**

[1] - A cellular Automata SIR Model for Landscape Epidemiology, [Erin L. Landguth](https://www.researchgate.net/profile/Erin-Landguth) https://www.researchgate.net/publication/255519512\_A\_Cellular\_Automata\_SIR\_Model\_for\_Landscape\_Epidemiology

[2] - Modeling epidemics using cellular automata, [S. Hoya White](https://www.ncbi.nlm.nih.gov/pubmed/?term=White%20SH%5BAuthor%5D&cauthor=true&cauthor_uid=32287494), [A. Martín del Rey](https://www.ncbi.nlm.nih.gov/pubmed/?term=del%20Rey%20AM%5BAuthor%5D&cauthor=true&cauthor_uid=32287494), and [G. Rodríguez Sánchez](https://www.ncbi.nlm.nih.gov/pubmed/?term=S%26%23x000e1%3Bnchez%20GR%5BAuthor%5D&cauthor=true&cauthor_uid=32287494) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7127728/>

[3] – Conway's game of life <https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life>

[4] - https://github.com/JuliaLang/julia/blob/master/LICENSE.md

[5] - <https://github.com/fonsp/Pluto.jl>

[6] - <https://computationalthinking.mit.edu/Fall20/lecture10/>