

Virus Spreading Simulator

Frederic Hamelink en Bert Livens

Universiteit Antwerpen

België

frederic.hamelink@student.uantwerpen.be

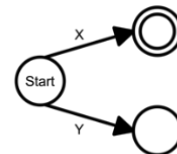
bert.livens@student.uantwerpen.be

Covid-19 is tot nu toe de overheersende factor in de actualiteit voor 2020 en 2021. Dit is niets nieuw, virussen en bacteriële infecties zijn al eeuwenlang een plaag voor de mensheid. Om virussen te voorkomen en in te dammen zijn er al verschillende tools en technieken ontwikkeld, zoals: social distancing, contact tracing, etc. Een van de vroegste ontwikkelingen was door John Snow[1]; een Britse wetenschapper die voor het eerst data science op volksgezondheid toegepast heeft.

Om op deze technologieën uit te breiden hebben wij besloten om een virus spreading simulator te ontwikkelen. Deze virus spreading simulator zal werken met cellulaire automaten[2]. De toestand van een cellulaire automaat wordt bepaald door zijn huidige toestand en de toestand van zijn rechtstreekse burens. Dit is goed want virussen verspreiden per definitie naar rechtstreekse burens en zo kunnen we het verspreiden van een virus simuleren met verschillende inputs.

Niet enkel het virus is een variabele, maar ook de omstandigheden. Denk maar aan bevolkingsdichtheid, populatie, gezondheid etc. Deze variabelen gaan toegediend worden m.b.v. input files, deze input files worden gecontroleerd op correctheid door reguliere expressies. Ook verschillende eigenschappen van het virus gaan worden verklaard in de input files, denk maar aan kans op mutatie, besmettingsgraad, sterftegraad etc.

Eens de input files correct verklaard zijn start de constructie van de bevolking grid en het virus. Elke eigenschap van het virus is een NFA met 3 states, waarbij de transities kansen zijn op “slagen” en “falen”. Een eigenschap ziet er als volgt uit:



Een virus is dus de verzameling van deze eigenschappen, om een virus te verkrijgen gebruiken we dus de subset constructie om de eigenschappen (NFA's) om te zetten naar DFA's. Vervolgens creëren we een product automaat, het resultaat is het volledige virus. Om te controleren of een virus gemuteerd is kunnen we gebruikmaken van het table filling algoritme. Als de twee DFA's volgens het TFA niet equivalent zijn weten we dat er een mutatie is gebeurd.

Eens er een mutatie gedetecteerd is moet deze benoemd worden. A.d.h.v. reguliere expressies creëren we een “name generator”, deze “name generator” is een soort encryptie van het virus zijn eigenschappen. We starten met de productautomaat om te zetten naar een reguliere expressie. Vervolgens zetten we alle cijfers (kansen) om naar zijn bijhorende letter (bv. 0,15 = het 15^{de} letter in het alfabet). Wat er overblijft is de benaming van het gemuteerde virus.

De bevolkings-grid wordt voorgesteld door een aangepaste epsilon nfa, de transities zijn de afstand tussen states (deze stellen personen voor). Een epsilon transitie is een instant transmissie van het virus. Deze bevolkings-grid wordt verkregen door een gegeven reguliere expressie om te zetten in een epsilon nfa. Tenslotte is er een optionele optie om de bevolkings-grid om te zetten in een DFA voor consistentie.

Referenties

[1] John Snow

https://nl.wikipedia.org/wiki/John_Snow

[2] Cellulaire automaten

https://nl.wikipedia.org/wiki/Cellulaire_automaat