# Hospital Support System (HSS)

## Koen Pauwels, Ajaya Adhikari, Rafael De Smet

Universiteit Antwerpen

België

*koen.pauwels@student.uantwerpen.be*

### Fase 1

De algoritmes die we zullen implementeren zijn:

* Omzetting ε-NFA naar DFA (Rafael De Smet)
* Omzetting DFA naar regex (Ajaya Adhikari)
* DFA minimalisatie door table-filling algoritme (Koen Pauwels)

### Fase 2

Als project hebben we gekozen om de werking van een ziekenhuis te simuleren.

Het nut hiervan is het ondersteunen van de vlotte werking van een ziekenhuis. We doen dit door een aantal basisactoren te simuleren aan de hand van automaten. We hebben als basisactoren gekozen voor een patiënt, een balie, een aantal dokters. Het project is geschikt voor de conferentie omdat het een internationaal toepasbaar systeem gebaseerd op de principes van Talen & Automaten.

De actoren ondergaan transities als gevolg van hun eigen handelingen, de handelingen van anderen of externe gebeurtenissen.

De toestanden en transities van een patiënt:

* Starttoestand: 1
* Toestanden:

1. Ongeregistreerd
2. Geregistreerd en wachtend
3. In behandeling
4. Genezen
5. Niet genezen
6. Observatie
7. Ontslagen
8. Dood

* Accepttoestanden: 7 en 8

De toestanden en transities van de balie:

* Starttoestand: toestand 1
* Toestanden:

1. Wachten op een patiënt
2. Bezig zijn met binnenkomende patiënten

* Accepttoestand: Geen
* Transities, deze passen de eigen toestand niet aan, maar triggeren wel andere automaten. Zo veranderen de toestanden van de andere automaten.

1. Patiënt meldt zich aan
2. Patiënt naar wachtzaal sturen, agenda juiste dokters aanpassen

De toestanden en transities van een dokter:

* Starttoestand: 1
* Toestanden:

1. Wachten
2. Werken
3. Thuis

* Accepttoestand: 3
* Transities, deze passen de eigen toestand niet aan, maar triggeren wel andere automaten. Zo veranderen de toestanden van de andere automaten.

1. Agenda controleren
2. Patiënt gaan halen en behandelen
3. Patiënt ontslaan (patiënt is genezen)
4. Patiënt binnenhouden (patiënt is niet genezen)
5. Patiënt doodverklaren
6. Naar huis gaan

Mogelijke uitbreidingen:

Om het systeem uit te breiden, kunnen we nog een paar extensies toevoegen.

Fase 1:

* Omzetting van regex naar DFA

Fase 2:

1. Een wachtrij simuleren aan de balie
2. Dokter kan zelf plannen om een patiënt terug te bezoeken
3. Spoed en niet-spoed onderscheiden: Spoed werkt aan de hand van een priority queue en niet-spoed werkt aan de hand van een gewone queue.
4. Onderscheid maken tussen een patiënt met afspraak en één zonder afspraak (in de niet-spoed afdeling)
5. Beperkt aantal kamers voor patiënten
6. Ruimtelijke voorstelling van het ziekenhuis

**Tijdsindeling Fase 1**

|  |  |  |
| --- | --- | --- |
| Implementatie alfabet, via associatieve containers. | 1 uur | Koen Pauwels |
| Implementatie DFA . | 15 uur | Ajaya Adhikari |
| Implementatie ε-NFA. | 15 uur | Rafael De Smet |
| Omzetting ε-NFA naar DFA | 20 uur | Rafael De Smet |
| Omzetting DFA naar regex | 20 uur | Ajaya Adhikari |
| DFA minimalisatie door table-filling algoritme | 20 uur | Koen Pauwels |

**Tijdsindeling Fase 2**

|  |  |  |
| --- | --- | --- |
| Automaat voor een patiënt | 3 uur | Rafael De Smet |
| Automaat voor een dokter | 3 uur | Ajaya Adhikari |
| Automaat voor de balie | 4 uur | Koen Pauwels |
| Interactie tussen de verschillende automaten | 15 uur per persoon | Allen |
| Weergave huidige toestand ziekenhuis | 15 uur | Rafael De Smet, Ajaya Adhikari |
| Input systeem | 20 uur | Koen Pauwels |
| Implementatie gegevensstructuren | 10 uur per persoon | Allen |
| Eventuele uitbreidingen | TBA | Reserve tijd |