I am listening

By: Anas Stitou, Anas.Stitou@student.uantwerpen.be , Emil Lambert Emil.Lambert@student.uantwerpen.be , Tibo Verreycken tibo.verreycken@student.uantwerpen.be, Kars Christian van Velzen kars.vanvelzen@student.uantwerpen.be (UA: INF-BA1)

• Wat zijn onze projectplannen?

We willen graag een programma schrijven, inclusief universele documentatie, waarin we Songs* (audio files: .mid) kunnen vergelijken en categoriseren. Het project moet binnen de Tog-eisen vallen. Om onze vergelijkingen uit te voeren gebruiken we toepassingen op automaten.

• Waarom hebben we hiervoor gekozen?

In onze ogen voldoet ons project aan alle vereisten voor de toepassingsopdracht. Daarbij valt dit perfect binnen ons interessegebied; We zijn gefascineerd door het, doch bekende, Spotify-algoritme dat o.a. aanbevolen liedjes geeft om te luisteren.

• Waarom is ons project geschikt voor de conferentie?

De workflow en algoritmes die we willen gebruiken hebben mogelijke toepassingen vanuit de Automaten-theorie. Het is een uniek project (naar ons weten).

• WorkFlow

De eerste stap is altijd de verkregen data te verwerken en in te delen in data waarmee wij kunnen werken. We parsen het .mid bestand en zetten dit om in ons data formaat. Vervolgens zullen we een aantal functies hierop toepassen om een Regex te construeren. De regex valt volgens de gekende algoritmes om te zetten naar een DFA. Om de efficiëntste workflow te behouden, construeren we een e-NFA en zetten deze uiteindelijk om in de meeste minimale DFA. Hiermee kunnen we iedere mogelijke bewerking uitvoeren binnen onze vergelijkingen. Vervolgens voeren we de vergelijkingen uit en outen we ons resultaat. Het idee is dat er verschillende vergelijkingen gemaakt worden, op verschillende manieren worden uitgevoerd. De gebruiker kan de parameters hiervan selecteren, als wel de aanbevolen check te runnen. De aanbevolen check is een verzameling van alle checks waarbij de resultaten, volgens ons, het meest correct worden verrekend tot 1 vergelijking percentage.

Om extra zekerheid in te bouwen in de vergelijking percentages, zullen we ook vergelijkingen en controles opbouwen met andere automaten, zoals de GJDFA. Dit is een automaat die ons uitermate geschikt lijkt om een specifieke Use Case te implementeren, om het aantal noten van een liedje op te tellen op een efficiënte manier. De zogeheten, “Note Counter”, een doch essentiële component in ons vergelijkingssysteem. Hiermee zullen we vanuit een Regex/Song, graag alle noten optellen. Hiervoor zijn ook aansluitende algoritmes nodig om de GJDFA te kunnen omzetten naar een DFA.

Zie ook • FlowChart

• Contract & Taakverdeling

Zie Contract Contract_v0

• In welke categorie komt ons voorstel?

Categorie	Noden	Use	Usecase
Goud	Verbreding/Verdieping	Gebruik van [GJDFA] (General One-Way-Jumping Finite Automata) en de aansluitende algoritmes om deze om te zetten in Regex, e-NFA, DFA. De GJDFA is een automaat die aansluit bij de cursus maar niet besproken is in de cursus.	12
Zilver	Gebruik van extra algoritme uit de cursus (Niet uit Toi)	[Reverse DFA], [Complement DFA]. [Product Automaten] , [MSSC], [TFA], [State Elimination]	4,5,7
Brons	Gebruik van structuren Toi	[Product Automaten] , [MSSC], [TFA], [State Elimination]	4,5,7

• FlowChart

Het initiële idee van ons programma stellen we u graag voor in een versimpeld diagram.