**Lightweight integrated development environment**

**powered by finite automata**

**Jonas Caluwé -** [**Jonas.Caluwé@student.uantwerpen.be**](http://Jonas.Caluwé@student.uantwerpen.be)

**Flor Ronsmans De Vry -** [**Flor.RonsmansDeVry@student.uantwerpen.be**](mailto:Flor.RonsmansDeVry@student.uantwerpen.be)

**Achraf Yandouzi -** [**Achraf.Yandouzi@student.uantwerpen.be**](mailto:Achraf.Yandouzi@student.uantwerpen.be)

**Michel Dierckx -** [**Michel.Dierckx@student.uantwerpen.be**](mailto:Michel.Dierckx@student.uantwerpen.be)

Niets is erger dan het ontwikkelen van een applicatie in een code editor zonder focus op developer experience. Met een moderne integrated development environment kunnen softwareontwikkelaars pas echt hun maximale productiviteit bereiken. Deze complexe programma’s zijn uitgerust met, onder andere, een autocomplete functie en zogenaamde 'syntax highlighting', die het identificeren van stukken code makkelijker maakt. Deze hulpmiddelen zijn zo goed als onmisbaar voor de moderne softwareontwikkelaar. Langs de ene kant zal autocomplete ervoor zorgen dat de ontwikkelaar minder code zelf moet schrijven door frequent gebruikte constructies automatisch aan te vullen. Langs de andere kant is syntax highlighting een vaak onderschatte functie die we in veel IDE’s terugvinden. We weten namelijk pas echt dat we deze features nodig hebben, wanneer deze ontbreken. Syntax highlighting maakt het eenvoudiger om fouten op te sporen en te vermijden. Hiermee verhogen we de productiviteit van de ontwikkelaar en verlagen we dus ook de kostprijs van vele projecten.

Deze bouwstenen zijn dan ook wat wij wensen te ontwikkelen voor ons project. Deze bouwstenen zouden vervolgens samen kunnen komen in een interactieve editor voor Python. Het is een ambitieus idee om een volledige front-end in elkaar te steken die een autocomplete featured samen met een syntax highlighter. We zullen dus eerst ervoor zorgen dat de bouwstenen en fundering er zijn om deze front-end op te bouwen. We zorgen dat we suggesties kunnen genereren op basis van input en dat de al geschreven tekst correct wordt gehighlight in een statisch gegenereerde HTML. Hierna kunnen we eventueel aan een GUI werken die interactief in de browser draait.

We werken hiervoor dubbel verbredend. We implementeren een stochastische automaat met twee algoritmes, waarmee we platina hopen te behalen.

Deze stochastische automaat gebruiken we om een simpele autocomplete te implementeren. Op basis van de input van de gebruiker, bepaalt de stochastische automaat welk keyword de gebruiker waarschijnlijk wenst te typen. Om dit te realiseren hebben we natuurlijk data nodig. Deze data zullen we door middel van moderne web-scraping technieken uit populaire GitHub repositories verzamelen.

Een eerste algoritme dient om een stochastische automaat op te kunnen stellen vanuit een JSON-bestand. Daarnaast zullen we ook een algoritme nodig hebben om deze automaat te minimaliseren.

Verder zijn de algoritmes die we gebruiken: regex naar ε-NFA, deze naar een DFA d.m.v. de modified subset construction (MSSC), we construeren vervolgens van deze DFA’s de productautomaat, die we dan uiteindelijk minimaliseren m.b.v. het table-filling algoritme.

Dit project is een praktische toepassing van de theorie uit het vak Talen en Automaten. We demonstreren dat we de leerstof goed onder de knie hebben en er een reëel probleem mee kunnen oplossen.