**Eigen Markup Language en JSON**

Project door

*Liam Wouters – Siebe Wijnants – Axel De Leeuw – Michel Dierckx*

[*Liam.Wouters2@student.uantwerpen.be*](mailto:Liam.Wouters2@student.uantwerpen.be) *-* [*siebe.wijnants@student.uantwerpen.be*](mailto:siebe.wijnants@student.uantwerpen.be) *-* [*axel.deleeuw@student.uantwerpen.be*](mailto:axel.deleeuw@student.uantwerpen.be) *-* [*michel.dierckx@student.uantwerpen.be*](mailto:michel.dierckx@student.uantwerpen.be)

In de digitale wereld is het van belang om gegevens te ordenen en te structureren volgens een bepaald gegevensformaat. Zo kan data snel en efficïent opgeslagen, ontvangen en verstuurd worden. Markup languages en JSON zijn hier voorbeelden van.

Voor ons project zullen we ons toespitsen op JSON-bestanden en een eigen gemaakte markup language.

Voor JSON zullen we een validator maken waarmee de user kan nagaan of een bestand al dan niet een geldig JSON-bestand is. Dit zal gebeuren in een aantal stappen. Allereerst zal er een context free grammar opgesteld moeten worden die het JSON-formaat zo goed mogelijk omschrijft. Vervolgens zal de inputtekst moeten worden gesplitst in tokens. Via een parser naar keuze kan dan worden nagegaan of het bestand al dan niet een geldig JSON-bestand is. Ook zal het mogelijk gemaakt worden voor de gebruiker om een schema voor een specifiek JSON-formaat mee te geven. Hierin kunnen velden en waarden verder gespecifieerd worden.

We zullen ook een simpele eigen gemaakte markup language opstellen, gebaseerd op XML. Ook voor deze eigen markup language moet er een context free grammar opgesteld worden. Ook zal er een validator gemaakt worden, zodat de user kan nagaan of een opgegeven inputbestand voldoet aan de markup language. Hiervoor moet de tekst weer in tokens worden opgedeeld en vervolgens geparsed worden.

De user zal voor het parsen een parser naar keuze kunnen gebruiken. In totaal zullen er vier parsers geïmplementeerd worden: CYK, LL, LR en Earley. Ook zullen we de benodigde tijd voor het parsen van verschillende inputfiles vergelijken tussen de opties.

Om het voor de gebruiker gemakkelijker te maken om te herkennen waarom een bepaald bestand niet voldoet aan het JSON-formaat of het formaat van onze eigen markup language, zullen bepaalde elementen zoals tags en haakjes gehighlight worden.

Ook zal de gebruiker een geldig JSON-bestand kunnen omzetten naar een bestand volgens het formaat van onze eigen markup language, en omgekeerd.

Om de verschillende features samen te brengen tot één geheel zal er een command line interface geïmplementeerd worden.

De features die we implementeren zullen grondig getest worden. Indien blijkt dat sommige features in veel minder tijd dan verwacht geïmplementeerd werden, kunnen eventueel nog features zoals parse error melding of het visualiseren van de parse trees toegevoegd worden.