



Stijn Janssens, Anthony Hermans, Evert Heylen, Pieter Coeck

Universiteit Antwerpen

België

Stijn.Janssens@student.uantwerpen.be, Anthony.Hermans@student.uantwerpen.be,

Evert.Heylen@student.uantwerpen.be, Pieter.Coeck@student.uantwerpen.be

Voor fase 1 hebben we de volgende algoritmes gekozen:

1. Anthony: DFA -> Regex
2. Pieter: Subsetconstruction (NFA->DFA)
3. Stijn: Productautomaat
4. Evert: ϵ -NFA -> DFA

Voor fase 2 hebben we gekozen om ons project te gaan zoeken in een biologische context. In deze context kwamen we op het idee om gebruik te maken van het concept DNA-strings.

DNA-strings zijn sequenties van DNA-bouwstenen (Adenine, Guanine, Thymine en Cytosine).

DNA-cellen liggen aan de basis van hoe onze cellen werken en zijn dus de kern van ons bestaan. Daarom vonden we het een goed plan om ons in deze materie te verdiepen en omtrent dit onderwerp ons project op te stellen.

Aan de hand van deze informatie hebben we beslist om een systeem implementeren dat nagaat of een kleine DNA-string voorkomt in een grotere DNA-string. Het efficiënt en snel opsporen van een bepaalde DNA-string kunnen we dan doen met behulp van regex. Deze regex kunnen we dan eventueel nog uitbreiden door gebruik te maken van suffixtrees wat onze performantie bevordert. Dit zal echter wel afhangen van het vlot verlopen van het project. Later kunnen we dit systeem ook uitbreiden door middel van meer gradatie op basis van DNA-sequenties waardoor het dus ook complexer wordt. De DNA-strings kunnen aanzienlijk groot worden waardoor we ook onze algemene performantie zo goed mogelijk moeten maken. De echte complexiteit van ons project schuilt dus in de snelheid en efficiëntie van het inlezen en vergelijken van deze DNA-strings.

Dit systeem hebben we de naam Helix gegeven. Deze naam is een verwijzing naar de dubbele helix van een DNA-molecule.

Helix is maatschappelijk relevant want het gebruikt kan worden in de medische sector zoals bv bij medicijnen, ziektes opsporen, DNA-matching etc. Het DNA-matchen houdt dan in dat we kunnen nagaan of een hele string gelijk is aan een gegeven string. Dit is dan later uit te breiden aan de hand van complementaire strings.

Helix leunt ook sterk aan bij de inhoud van de cursus Talen en Automaten aangezien we deze DNA-strings kunnen zien als regex. Bovendien gebruiken we ook NFA's bij het voorstellen van de strings. Naast het gebruik van NFA's en regex gaan we ook onze performantie zo goed mogelijk maken dankzij het "Table Filling Algorithm".

Om ons project zo vlot mogelijk te laten verlopen gaan we gebruik maken van Asana. Deze app laat ons toe om het werk beter te plannen en ons makkelijker te houden aan deadlines die we ons vooraf hebben opgelegd. We kunnen echter nog niet afspreken wie juist wat gaat implementeren omdat dit waarschijnlijk nog zal variëren.

Het basisconcept van ons project zal een studiebelasting hebben van ongeveer 70 werkuren per groepslid. Dit aantal kan echter nog zwaar veranderen aangezien we op voorhand niet kunnen voorspellen hoeveel tijd elk onderdeel juist in beslag zal nemen. Overigens zullen we met de nodige uitbreidingen de 90 werkuren per groepslid bereiken.