Max de Regt – 21000268

Gerdtinus Netten - 2129861

# Korte beschrijving

Wij hebben de eerste opdracht gekozen. Wij hebben gekozen om zelf een hele architectuur te bedenken en te gebruiken (dus niet hoe Paul de Mast het heeft gedaan met tuples etc). Wij hebben alles een aparte klasse gegeven, maar gebruiken dezelfde Node klasse voor NDFA/DFA.

# Gerealiseerde conversie/methodes

* Hardcoded (N)DFA/regulier expressies
  + **Program.cs regel 152-191 (NDFA)**
  + **Program.cs regel 193-218 (Regex)**
  + **Program.cs regel 219-309(Gegenereerde DFA’s)**
* Parser voor reguliere expressies
  + **RegexParser.cs regel 24 – 92**
  + **RegexTester.cs regel 19 – 47**
* Woorden genereren die wel/niet in de reguliere expressie zit
  + **RegexTester.cs regel 55 – 92**
* DFA test of een woord wel/niet in de taal zit
  + **DFA.cs regel 70 – 102**
* NDFA test of een woord wel/niet in de taal zit
  + **NDFA.cs regel 150 – 183**
* DFA constructor
  + Begint met xyz
    - **GenerateDFA.cs regel 19 – 60**
  + Eindigt op xyz
    - **GenerateDFA.cs regel 68 – 143**
  + Bevat xyz
    - **GenerateDFA.cs regel 151 – 213**
* Thompson constructie
  + **Thompson.cs hele klasse vanaf regel 19**
* NDFA 🡪 DFA met Epsilon
  + **NDFAtoDFA.cs begint op regel 49**
* Reverse
  + **DFAReverse.cs regel 31 – 74**
* TestApplicatie (console based)
  + **Program.cs regel 604 – 776**
* Graphviz naar pdf
  + **Program.cs regel 777 – 782**