Max de Regt – 21000268  
Gerdtinus Netten - 2129861

# Vereiste Nuget Packages

* DotNetGraph (word gebruikt om GraphViz dot diagrams te maken)

# Vereiste programma’s

* GraphViz (en toegevoegd in path)

# 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 151-190 (NDFA)**
  + **Program.cs regel 192-217 (Regex)**
  + **Program.cs regel 218-308(Gegenereerde DFA’s)**
* Parser voor reguliere expressies
  + **RegexParser.cs regel 23 – 91**
  + **RegexTester.cs regel 18 – 46**
* Woorden genereren die wel/niet in de reguliere expressie zit
  + **RegexTester.cs regel 54 – 91**
* DFA test of een woord wel/niet in de taal zit
  + **DFA.cs regel 45 – 67 (voor 1 specifiek woord)**
  + **DFA.cs regel 69 – 101 (voor alle woorden die wel/niet in de taal zittten)**
* NDFA test of een woord wel/niet in de taal zit
  + **NDFA.cs regel 21 – 75 (voor 1 specifiek woord)**
  + **NDFA.cs regel 149 – 182 (voor alle woorden die wel/niet in de taal zittten)**
* DFA constructor
  + Begint met xyz
    - **GenerateDFA.cs regel 18 – 59**
  + Eindigt op xyz
    - **GenerateDFA.cs regel 67 – 142**
  + Bevat xyz
    - **GenerateDFA.cs regel 150 – 212**
* Thompson constructie
  + **Thompson.cs hele klasse vanaf regel 18**
* NDFA 🡪 DFA met Epsilon
  + **NDFAtoDFA.cs begint op regel 46**
* Reverse
  + **DFAReverse.cs regel 27 – 70**
* TestApplicatie (console based)
  + **Program.cs regel 603 – 775**
* Graphviz naar pdf
  + **Program.cs regel 776 – 781**