

Compiler Construction

Hans de Nivelle

Due: 28 October 2009

1.
 - (a) Transform $(\epsilon|aa|ab|ac)^*$ into an N DFA.
 - (b) Construct the corresponding DFA, using the construction on the slides.
 - (c) Simplify the DFA, using the construction on the slides.
2.
 - (a) Write a regular expression of form $(\Sigma|N)^*$, where N is your first name, with non-ASCII (Polish) characters replaced by ASCII characters. If your name is too long, you can choose another word between 4 and 8 characters.
 - (b) Transform the previous regular expression into an N DFA, using the translation on the slides.
 - (c) Transform the N DFA into a DFA, using the algorithm on the slides.
 - (d) Simplify the DFA from the previous task, using the algorithm on the slides.
 - (e) At this point, everyone should have the same DFA, do you understand why?
3. Consider $\Sigma = \{(',')\}$, and the language \mathcal{L} defined by $\sigma \in \mathcal{L}$ if σ consists of a sequence of parentheses that could occur in a legal C -program. (So $"()$ " is OK, $"(())"$ also, but $)()$ " or $"())"$ is not)
Is there a regular expression that recognizes this language? Give an expression, or an explanation why such regular expression does not exist.
4. Download FLEX from the course homepage.
 - (a) Rewrite the scanner that you wrote in the first task list using FLEX. Add more tokens to it. (identifiers, reals, integers, strings, a couple of reserved words).