NYU TANDON SCHOOL OF ENGINEERING

CS-UY3943 BK06/CS-GY6413: Compiler Design & Construction Prof. Boris Aronov
Department of Computer Science and Engineering Spring 2019
Class: 2MTC 812 Wed 3:20-5:50pm boris.aronov@nyu.edu
Office: 2MTC 10.008 +1-646-997-3092 Office hours: Mon 3–5, Wed 2–3/6–7 & by appt

Homework 1

Due Friday, February 15, 11:59pm. Notice the unusual due date.

By handing in the homework, you agree to the Homework Rules posted on Piazza.

"Extra thinking" means another related version of the problem, maybe harder maybe not. You do not have to answer it. You will not get extra credit for it. But you might learn something interesting. Feel free to discuss these on Piazza, after the due date. Or stop by and talk to me about them.

1. In language Bleh, comments look like this:

   `XYX I am a comment ... XYX`

   To be more precise, a comment starts with the three characters `XYX`, in this order, with nothing in between, cannot contain `XYX` as a substring, can contain any other sequence of characters inside (let's not worry about newlines, though in practice you have to), and ends with `XYX`.

   (*Food for thought*: What do you do with "`XYXY YXYX`"? What about "`XYXYX`"?)

   (a) Write a regular expression that matches one comment in this language and does not match anything that is not a single comment. It is easiest to grade if everyone uses standard regex notation I described in class and no exotic extensions. If you *are* using notation specific to a programming language (Perl/Python/whatever), please indicate what notation you are using.

   (b) Construct a DFA *by hand* that accepts a single comment and nothing else.

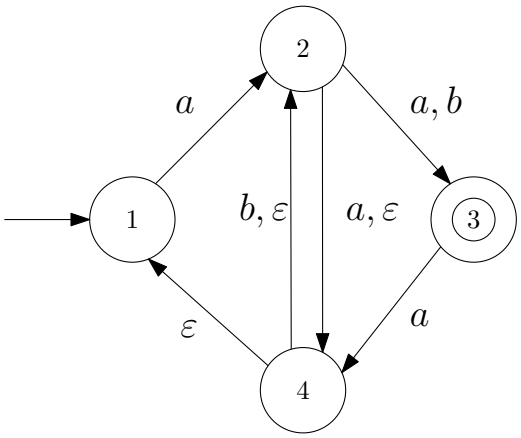
   The textbook has an extensive discussion of regular expressions. There are many websites describing regular expressions, such as

   `https://www.regular-expressions.info/index.html`.

   I am not endorsing it. It's just one of the first I found.

   Warning: There are many slightly incompatible conventions for writing regular expressions (examples: is "`(`" a character in the string and "`\(`" used for grouping or vice versa? is a `newline` treated as a regular character or not? etc etc). It's a problem. You have to live with it, if you program in Java/Perl/Python/etc. Sorry. That's why you need to tell us which flavor you are using in (a) above.

2. Consider the following NFA with alphabet $\{a, b\}$:



Construct an equivalent DFA. List its states (for example $A = \{1, 2, 5\}$ could mean that state A in the DFA corresponds to the set consisting of states 1, 2, and 5 of the NFA). Give the transition table of the DFA *and* draw the picture of the DFA. Do not forget marking the starting and ending states of the resulting DFA.

3. (a) First convert the regex `(01*|10*)*` to an NFA using algorithm 3.20 from the book. (Note that there are precedence rules among the operators in regex notation, so what I really meant is of course `((0(1*))|(1(0*)))*`. Wasn't this easier to read? :-)
(b) Then convert the NFA you obtained to a DFA using algorithm 3.23.

Note: I am *not* asking for a hand-constructed NFA/DFA for this regular expression. That's a completely different question.

Extra thinking: If you like finite automata (and there are people who do, even if it's hard to believe this! :-), there is also an algorithm for *minimizing* a DFA—finding a DFA that accepts precisely the same set of strings as the original one. It is useful for optimizing space in scanners and other software/hardware that simulates DFAs. What would this algorithm return on this regular expression? You can try thinking about the language that this expression defines, or you can actually run the algorithm and see what it does.