Assignment 2

CSCI 3136: Principles of Programming Languages

Due Feb 25, 2019

Assignments are due on the due date before 23:59. All assignments must be submitted electronically via the course SVN server. Plagiarism in assignment answers will not be tolerated. By submitting your answers to this assignment, you declare that your answers are your original work and that you did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

Submission instructions: Create a directory `a2` at the top level of your SVN repository. Place a single PDF file named `a2.pdf` containing your answers into this directory and submit it via `svn commit`.

Do the following for each of the languages below:

- State whether or not it is regular.
- If you claim the language is not regular, prove your claim using the Pumping Lemma.
- If you claim the language is regular,
  - Give a regular expression that matches exactly the strings in the language,
  - Give a graphical representation of an NFA that recognizes the language,
  - Convert the NFA into a DFA that recognizes the language, and
  - Minimize the DFA, that is, convert it into an equivalent DFA with as few states as possible and present it in graphical form.

  You do not need to construct the NFA using the method for translating a regular expression into an NFA discussed in class because this method tends to construct huge automata, which you do not want to manipulate manually. However, for the conversion from the NFA to an equivalent DFA and for minimizing the resulting DFA, you should use the methods discussed in class.

The languages you are asked to consider are the following:

(a) All binary strings with the same number of `0`s and `1`s.

(b) All binary strings with at most one `1` in each substring of length four. For example, the string `0001000100` belongs to this language: its substrings of length four are `0001`, `0010`, `0100`, `1000`, `0001`, `0010`, and `0100`, none of which contains more than one 1. The string `100010100` does not belong to this language because the underlined substring contains two 1s.

(c) All strings over the alphabet {`a`, `b`, `d`, `l`, `r`} that contain at least one of the words `bad`, `bald`, `ball` or `bard`.