Assignment 3

The full assignment specification is available via resources tab here on Ed.

Task 1: NFA -> Epsilon closures

Task 2: NFA, epsilon closures -> Epsilon free NFA

Task 3: Epsilon free NFA -> DFA

Task 4: Decide strings using DFA

If you are using Python, main.py is the entry point to your application, parser.py is some helper code that you can use to read the input.

I recommend you create another .py file for your own code, and import it into main.py, but you could implement everything directly in main.py if you wanted to.

The first thing you should implement is some more reasonable way to represent the NFA than the dictionary output by the skeleton code's parser (a list of tuples is a terribly inefficient representation of the transition function (delta)).

If you are using another language, you will need to edit "run.sh" (and "build.sh" if using a compiled language like Java or C) so that Ed knows how to run your program.

If you want to test your program in the browser, and see its full output, click the 'terminal' button, then type

bash run.sh < tests/example1.in

Eight example tests are provided (the ones named appendix_* are the tests from appendix 6.3, which are worth 5 marks each for passing). I recommend that you create some more test cases to help you debug your own code.

Be aware that there are hidden test cases, and we may add more tests during marking. Passing the visible tests is necessary, but not necessarily sufficient for full marks.