University of New York Tirana
Automata Theory
Fall 2013

Assignment 2 – Decidability

Out: 18/12/2013
Due: 12/01/2014

Introduction

In this assignment you will work with decidability problems regarding finite automata. More specifilly you will implement a problem that reads the description of a Non-Deterministic Finite Automaton (NFA), asks for input strings, and decides whether these strings are accepted by the NFA. You are free to use any representation of the NFA that you want to. However, you are required to

- design a data structure for NFA
- design a data structure for Deterministic Finite Automata
- read the actual NFA from a text file (format given below)
- convert the NFA to its equivalent DFA
- determine whether the input string is accepted by the equivalent DFA

File Structure

Remember the formal definition of NFA is given as a five-tuple $(Q, \Sigma, \delta, q_0, F)$ where the transition function is given as $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$. A transition rule has the form $\delta(q, x) = R$ where $R \subseteq Q$. The input file describing an actual NFA will represent each of these components as follows:

- Line 1 contains Q and q_0: it is a space-separated list of all states, where the first state is the start state. We will represent a state by its label, e.g. q0, two-zeroes, even etc.
- Line 2 represents the alphabet a string: each character is a symbol in the alphabet
- Line 3 represents F: it is a space separated list of accepting states
- Remaining lines represent δ, one line per rule. Each line has the syntax: state symbol $state_1$ $state_2$ ... $state_n$. We will use character e to represent the empty string. You may

assume e will not appear in the alphabet. We will not list the transition rules which map to empty set Φ, so each line would correspond to n actual "arrows" in the schematic representation.

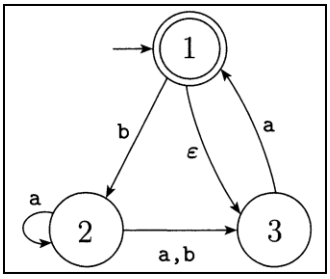

Figure 1: Sample NFA

Sample Input

Given the NFA in Figure 1, the corresponding input file is given Table 2 below :

```
1 2 3
ab
1
1 b 2
1 e e
2 a 2 3
2 b 3
3 a 1
```

Table 2: Input file for NFA in Figure 1

Notice and keep in mind

- Start state will be the first token of the first line
- Symbol e stands for empty string ε.
- Line 5 represents the transition rule $\delta(1, a) = \{2,3\}$, which in turn corresponds to two arrows in the diagram.
- State labels in general might be multiple letters. Do not make the assumption that each state is one letter long. Instead tokenize with respect to space.
- Transition rules that map to empty set will not be represented in the file, just like they are not represented in the digram. For example $\delta(1, b) = \Phi$ and no such line will appear on the input file.
- Input string(s) to be tested for acceptance will not be represented in the input file, instead you will read them (in a loop)during execution.

Sample Execution

```
>> Please enter the path to input file
D:\algorithms\assign02\input.txt
>> Reading input ... done
>> NFA structure
... some representation of the NFA
>> Converting NFA to DFA ... done
... some representation of the DFA
>>> Enter a string to test for acceptance (or Ctrl-C to exit)
01101
>>> Accepted
>>> Enter a string to test for acceptance (or Ctrl-C to exit)
110
>>> Rejected
>>> Enter a string to test for acceptance (or Ctrl-C to exit)
Ctrl^C
```

Submission Requirements

(via Turnitin)

- The source code
- A report outlining your work
- Outline of algorithms you have implemented (and time complexity discussion)

(via email)

- The IDE-generate project (Netbeans, Eclipse, .NET, etc.) for your whole work

As a final note, any formulas that you will need to use in your report **must** be prepared with Microsoft Equation editor, or you might think that now is a good time for you to learn LaTeX and prepare your documents with LaTeX from now on. Feel free to ask me or other instructors in Computer Science department about the use of LaTeX.