CS 241 — Lecture 10

Bartosz Antczak *Instructor: Kevin Lanctot* *February 2, 2017*

10.1 Finite Automata

This is also known as a deterministic *finite state machine* (FSM). It's comprised of:

- A finite set of states which includes a start state and at least one *accepting states*
- A finite set of input symbols
- A finite set of transitions

The *deterministic finite automata* (DFA) determines if input is accepted (e.g., is a word in a language) or rejected (e.g., the word is not in the language).

Example 10.1.1. *A deterministic finite automata of a sample person's daily routine*

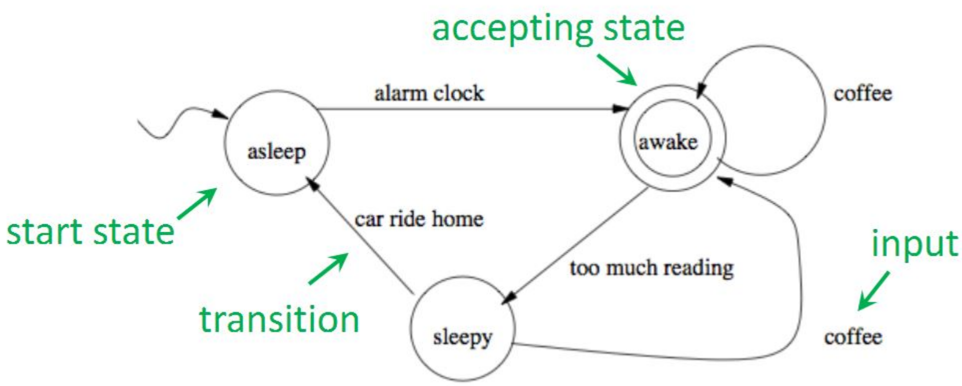


Figure 10.1: Courtesy from Prof. Lanctot's slides

A DFA contains:

- **States** (drawn as circles): there are two types
  - *start state:* denoted with a curved line
  - *accepting state(s):* denoted as two concentric circles

  We can label these states, but that's optional

- **Transitions:** an edge that moves from one state to another

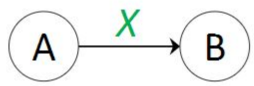


Figure 10.2: This means: *on input X, move from state A to state B*. If the input does not match any of the possible transitions, an error is thrown. Courtesy from Prof. Lanctot's slides

Comparison to Programming Languages

The components of finite automata can be linked to what you would see in a program:

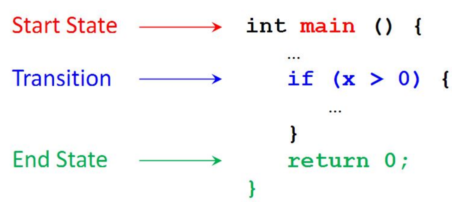


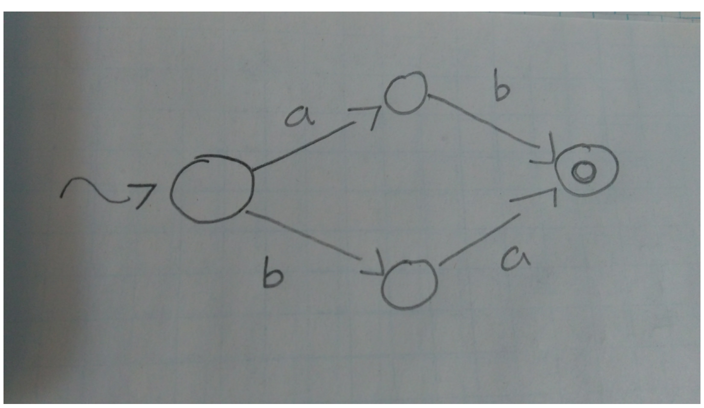
Figure 10.3: Courtesy from Prof. Lanctot's slides

10.1.1 Examples of DFAs

Let $\Sigma = \{a, b, c\}$

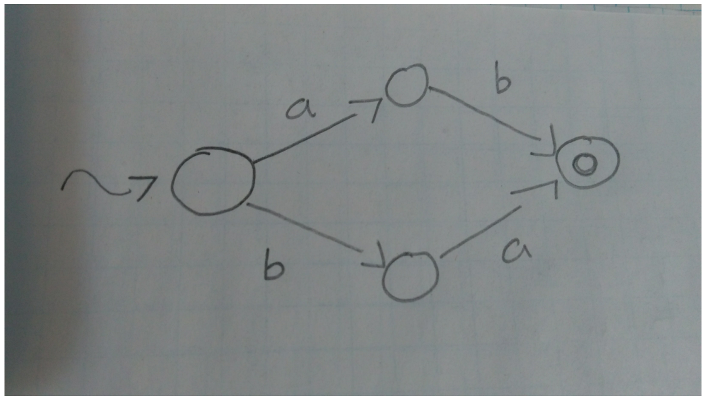
Exercise 1

Create a regular expression and a DFA that accepts the language of strings that contain exactly one a, one b, and no c's.
The language is $\{ab, ba\}$. The DFA is:



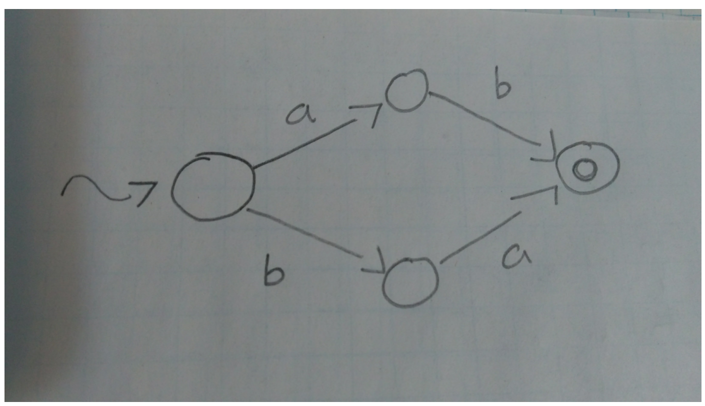
Exercise 2

Create a regular expression and a DFA that accepts the language of strings that contain at least one a. The regular expression is $(b|c)^*a(a|b|c)^*$ (the string starts with b or c, then contains one a, then is followed by any letter in the language). The DFA is:



Exercise 3

Create a DFA that accepts the language of strings that contain an even number of a's (including 0 a's). The DFA is: .



10.1.2 Formal Definition of a DFA

A DFA is a 5-tuple $(\Sigma, Q, q_0, A, \delta)$, where

- Σ is a finite alphabet, e.g., $\Sigma = \{b, e, n, q\}$
- Q is a finite set of states, e.g., $Q = \{S, b, be, bn, beq, bne\}$
- q_0 is the start state, e.g., $q_0 = \{S\}$
- A is the set of accepting states, e.g., $A = \{beq, bne\}$
- δ is the transition function that maps the pair of state and symbol to state. Denoted as $\delta(Q, \Sigma) = \Sigma'$.