CS 121, Section 1

Week of September 10, 2013

Overview

This week we will focus on reviewing the core concepts involved with DFAs, NFAs, and the properties of regular languages.

Outline:

1. **Concept Review**
2. **Exercises.**

1 Concept Review

1.1 DFA

Intuitively a Deterministic Finite Automaton (DFA) represents a machine with limited memory which computes over arbitrary input to either "accept" or "reject" that input. We represent the input as a string of symbols in a given alphabet and the DFA's memory as a set of possible internal states. Informally then a DFA can be described by a state diagram giving transitions between states according to the symbols it reads, as seen in Sipser.

More formally, a DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of final (accepting) states. We say that $(Q, \Sigma, \delta, q_0, F)$ *accepts* a string $w = \sigma_1 \cdots \sigma_n$ if and only if, for some sequence of states $q_0, q_1, \cdots, q_n \subseteq Q$, we have $q_n \in F$ and for each $i \in \{0 \cdots n-1\}$, $\delta(q_i, \sigma_{i+1}) = q_{i+1}$.

1.2 NFA

A Nondeterministic Finite Automaton (NFA) gives us more freedom in it's construction than a DFA. It is similar except a state can have two outgoing arrows with the same label (or arrows with the label ε). Then for a given input as long as there exists some series of transitions to follow that lead to an "accept" state, the string is accepted. Formally then the transition function becomes $\delta : Q \cup \{\varepsilon\} \times \Sigma \to \mathcal{P}(Q)$ and, for each $i \in \{0 \cdots n-1\}$, $\delta(q_i, \sigma_{i+1}) \ni q_{i+1}$.

NFAs are equivalent to DFAs. We can use the subset construction to convert a NFA to a DFA.

1.3 Regular Languages

A regular language is a set of strings recognised by a DFA (or, equivalently, a NFA) i.e. L is a regular language if and only if there exists a DFA (or NFA) M such that

$$L = \{x \in \Sigma^* : M \text{ accepts } x\}.$$

Some examples of simple regular languages are as follows.

- Finite languages.
- Strings containing a fixed substring.
- String containing an even number of occurances of a character.
- Strings containing a bounded number of occurances of a character.

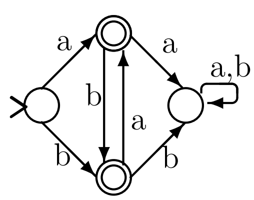
1.4 Closure properties

Let L_1 and L_2 be regular languages over Σ. Then the following are also regular languages.

- $\Sigma^* - L_1$, the complement of L_1.
- L_1^*, the concatenation of any number of strings from L_1.
- $L_1 \cap L_2$, the set of strings in both languages.
- L_1^{R}, the reversal of all strings in the language (see below).

2 Exercises

Exercise 2.1. *Describe informally the language recognized by the following DFA:*



Exercise 2.2. *Draw an NFA that recognizes the language of all strings with the* subsequence *abba. (A subsequence is like a substring, except it doesn't have to be consecutive characters. For example, ababa would be in the language.)*

Exercise 2.3. *Draw an NFA that recognizes the language of all strings with the substring abba.*

Exercise 2.4. *Convert your NFA from the previous exercise to a DFA using the subset construction.*

Exercise 2.5. *For any language L, let $L^R = \{w^R : w \in L\}$. Show that if L is regular, then so is L^R*

Exercise 2.6. *The following proof is incorrect. Explain why.*

1. $L_1 = \{x : x \in \Sigma^*\}$ is regular

2. Regular languages are closed under reversal (see pset 1), so: $L_2 = \{x^R : x \in \Sigma^*\}$ is regular

3. Regular languages are closed under concatenation, so: $L_3 = \{xx^R : x \in \Sigma^*\}$ is regular

4. L_3 is the language of even palindromes, so the language of even palindromes is regular

(While we can now only show that this proof is incorrect, we will later prove that the language of even palindromes is not regular)

Exercise 2.7. *For any language L, let NOREPEATB(L) be the language of strings in L, but with any b's that are immediately preceded by another b removed. So, for example, if babbaaababbb $\in L$, then babaaabab $\in$NOREPEATB(L). Show that if L is regular, then so is NOREPEATB(L).*