| | |
|---|---|
| **University of Virginia - cs3102: Theory of Computation** | Spring 2010 |
| DFA Notes | **26 January** |

**Upcoming Schedule:**

- **Thursday's office hours:** Sonali Parthasarathy will be covering my scheduled office hours Thursday (3:15-4:30). They will be held in Thornton Stacks (see the "cs3102 TA" sign) instead of my office.
- **Tuesday, 2 February:** Problem Set 1 is due at the beginning of class. Note that Problem 4 has been removed from PS1.
- Reading for next week: Finish Chapter 1

**Problems and Machines**
One of the main goals in this class is to understand the *power* of different abstract machines. For each of the abstract machine models we introduce, we will seek to define precisely the set of problems that can be solved by that machine model.

We focus on *decision problems*. A decision problem is a problem where the output is a single bit: "yes" or "no". We will see later that many search problems with arbitrarily long outputs can be converted to a sequence of decision problems.

One type of decision problem is the language recognition problem: *is string $s$ part of the language $L$?* Recall that a language is a *set of strings*.

We can characterize the power of a given type of abstract machine by the set of languages it can recognize. A machine recognizes a language $L$ if it accepts (outputs "yes") for every string that is a member of $L$ and rejects (outputs "no") for every string that is not a member of $L$.

What is a *problem*?

What does it mean for a machine to *solve a problem*?

Why do all (theoretically) interesting problems has infinitely many inputs?

**Definition:** A *finite automaton* (DFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$:

1. $Q$ — a *finite* set (the states)
2. $\Sigma$ — a *finite* set (the alphabet)
3. $\delta: Q \times \Sigma \rightarrow Q$ —- a function from a state and alphabet symbol to a state (the transition function)
4. $q_0 \in Q$ — the start state
5. $F \subseteq Q$ — the set of accept states

How many start states can a DFA have?

How many accept states can a DFA have?

How big a table would be needed to fully describe the $\delta$ function?

What language does the DFA defined below recognize:

$Q = \{A, B\}$
$\Sigma = \{0, 1\}$
$\delta$ is described by:

| | 0 | 1 |
|---|---|---|
| $A$ | $A$ | $B$ |
| $B$ | $B$ | $A$ |

$q_0 = A$
$F = \{B\}$

**Definition:** A *regular language* is a language that can be recognized by some finite automaton.

**Language Recognition:** $\mathcal{L}(A)$ represents the language recognized by DFA $A$.

**Computation Model for DFA**
Define $\delta^*$ as the *extended transition function*:

$$w \in \mathcal{L}(A) \Leftrightarrow \delta^*_A(q_0, w) \in F_A$$

$\delta^* : Q \times \Sigma^* \rightarrow Q$

We define $\delta^*$ inductively on *strings* (its second input):

- **Basis:**

- **Induction:**