Lecture 19 - Deterministic Finite-State Automaton

May 2, 2016

Reading: Chapter 12.2

A deterministic finite-state (DFA) automaton is one of the simplest models of computation.

A machine that behaves according to an DFA isn't very powerful but can be approporate for certain problems.

A DFA has a finite number of *states*. Computation begins in the DFA at a start state. The DFA repeatedly transitions from state to state using the input available and a state transition function. It continues until an *accepting state* is reached, at which point the computations stop.

A DFA, A, consists of five objects:

1. A finite set I, called the input alphabet , of input symbols
2. A finite set S of states that the automaton can be in
3. A designated state $s_0 \in S$ called the initial state
4. A designated set of states $T \subseteq S$ called the accepting states
5. A function $N : S \times I \to S$ that specifies state transitions. These can be represented as 3-tuples (state, input, next state).

0.1 How to Draw a DFA as a Directed Graph

Suppose we have a DFA defined by the following:

1. $I = \{0, 1\}$
2. $S = \{s_0, s_1, s_2\}$
3. start state: s_0
4. $T = \{s_2\}$
5. $N = \{(s_0, 0, s_0), (s_0, 1, s_1), (s_1, 0, s_1), (s_1, 1, s_2), (s_2, 0, s_2), (s_2, 1, s_2)\}$

The DFA's transition function can be represented in a table.

State transition table

		0	1
$\rightarrow$	s_0	s_0	s_1
◎	s_1	s_1	s_2
	s_2	s_2	s_2

Construction Steps

1. Draw circles with state inside.
2. Draw directed edge $\rightarrow$ to start state.
3. Draw double circle around accepting states, ◎
4. Draw transitions from state to state as directed edges according to transition table. Label edges with input. Separate input (if mulitple) with commas.

0.2 The Language Accepted by a DFA

Suppose a string of input symbols is fed into a DFA in a sequence and the DFA reaches an accept state. That sequence of symbols or *word* is said to be in the *language accepted by the DFA*.

Formally,

Let A be a DFA with the set of input symbols I. Let I^* be the set of all strings over I, and let $w \in I^*$. Then w is accepted by A if, and only if, A transitions to an accepting state when the symbols of w are input to A in sequence from left to right, starting when A is in its initial state.

The language accepted by A, denoted $L(A)$, is the set of all strings that are accepted by A.

We can sometimes describe the language accepted by a DFA. In the previous example, the langauge accepted by the DFA is the *set of all strings of 0s and 1s that contain exactly one 1.*

0.3 Kleene's Theorem

Given any language that is accepted by a finitie-state automaton, there is a regular expression that defines the same language. Given any language defined by a regular expression there is a DFA that accepts that language.

The regular expression for the previous DFA is 0^*10^*.

1 Regular and Non-regular Languages

An language accepted by a DFA is called a *regular language.*

Of course not all languages are regular languages. Consider the following language.

Let $\Sigma = \{a, b\}$ and let $L = \{s \in \Sigma^* | s = a^k b^k$ where k is a positive integer$\}$.

L is not a regular language.

Proof: Suppose L is a regular language. Then there is a DFA that accepts L. Let A be the DFA that accepts L. Since A is a DFA, A has n, a finite number, of states.

Consider the set, C, of prefixes of the strings in L that contain just a's: a, aa, aaa, aaaa, etc. There are infinitely many.

By the Pigeon Hole principle, a mapping f from C to the states in A, maps at least two strings, say a^p and a^q, in C to the same state, say s_m.

So the DFA, on input $a^p b^p$ transitions from the start state to s_m when processing the a's, then when processing p b's, eventually transitions to an accept state, since $a^p b^p$ is accepted by A.

Since a^q also reaches s_m, then when processing p b's, it too would reach an accept state. Therefore $a^q b^p$ is accepted by A. But $a^q b^p$ is not accepted by A, since A only accepts string sthat have an equal number of a's and b's. Hence a contradiction.

Thus L is not a regular language. ■