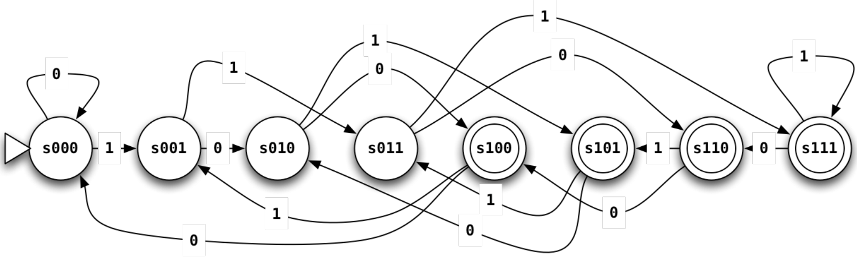
Regular Languages

November 5, 2012

CS 81: Computability and Logic

Components of a State Machine?



1. K, a set of states
2. Σ, the alphabet
3. $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$, the transition relation
   (or $\delta : Q \times \Sigma \rightarrow Q$, the transition function)
4. $s \in K$, the start state
5. $A \subseteq K$, the final/accepting states

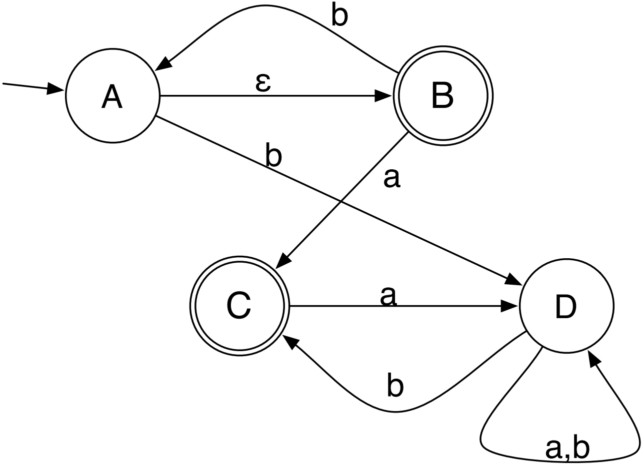
A Jewel of Theoretical Computer Science

The following are equivalent:

1. There is a DFA accepting the language L
2. [Rabin and Scott] There is an NFA accepting L
3. [Kleene] L is a regular set.

FROM NFA TO DFA: THE **SUBSET CONSTRUCTION**



Regular Languages

An *inductively-defined* collection of sets!

- ✓ $\emptyset$ is a regular language.
- ✓ $\{a\}$ is regular for any $a \in \Sigma$.
- ✓ If L and M are regular, then so is LM and $L \cup M$.
- ✓ If L is regular, then so is L^*.

True or False?

1. Σ^* is regular.
2. $\{\varepsilon\}$ is regular.
3. If $w \in \Sigma^*$, then $\{w\}$ is regular.
4. Every finite language is regular.
5. Every set is regular (since $\{w_1, w_2, \ldots\} = \{w_1\} \cup \{w_2\} \cup \cdots$).

Regular Expressions

An *inductively-defined* collection of expressions!

- ✓ $\emptyset$ is a regexp
- ✓ ε is a regexp
- ✓ a is a regexp for any $a \in \Sigma$.
- ✓ If r_1 and r_2 are regexps, then so is $(r_1 r_2)$ and $(r_1 | r_2)$.
- ✓ If r is a regexp, then so is (r^*).

Parenthesis Convention:

$$ab^*|c^* \quad = \quad (a(b^*)) \mid (c^*)$$

Regexp Interpretations

Regular expressions abbreviate regular languages.

- ✓ $L(\emptyset) = \emptyset$
- ✓ $L(\varepsilon) = \{\varepsilon\}$
- ✓ $L(a) = \{a\}$
- ✓ $L(r_1 r_2) = L(r_1)\, L(r_2)$
- ✓ $L(r_1 \mid r_2) = L(r_1) \cup L(r_2)$
- ✓ $L(r^*) = L(r)^*$

We say that "r matches w" if $w \in L(r)$.

True or False?

- ✓ $L(r_1) = L(r_2) \quad \longrightarrow \quad r_1 = r_2$
- ✓ There is a regular expression r with $L(r) = \Sigma^*$

Regular Expression Examples ($\Sigma = \{0, 1\}$)

Describe the Language

1. $0 \mid 1$
2. $(0|1)^*$
3. $(0|1)\, 0^*\, 1^*$
4. $0^*110^*|1^*001^*$

Find the regular expression

1. Strings where *every* 1 is followed by a 0.
2. Strings where *no* 1 is followed by a 0.
3. Strings where *every* 1 is preceded by and followed by 0.

From Regular Expression to NFA

Construct $\mathrm{NFA}(r)$ by structural induction (recursion) on the regular expression r.

- ✓ $\emptyset$ is a regexp
- ✓ ε is a regexp
- ✓ a is a regexp for any $a \in \Sigma$.
- ✓ If r_1 and r_2 are regexps, then so is $(r_1 r_2)$ and $(r_1 | r_2)$.
- ✓ If r is a regexp, then so is (r^*).