NFAs and DFAs

Sipser 1.2 (pages 47–63)

Last time...
NFA

- A nondeterministic finite automaton (NFA) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where
  - \(Q\) is a finite set called the states
  - \(\Sigma\) is a finite set called the alphabet
  - \(\delta: Q \times \Sigma \rightarrow P(Q)\) is the transition function
  - \(q_0 \in Q\) is the start state
  - \(F \subseteq Q\) is a set of accept states

- In-class exercise:

NFA computation

- Let \(N=(Q, \Sigma, \delta, q_0, F)\) be a NFA and let \(w\) be a string over the alphabet \(\Sigma\)
- Then \(N\) accepts \(w\) if
  - \(w\) can be written as \(w_1w_2w_3...w_m\) with each \(w_i\) in \(\Sigma\)
  - There exists a sequence of states \(s_0, s_1, s_2, ..., s_m\) exists in \(Q\) with the following conditions:
    1. \(s_0 = q_0\)
    2. \(s_{i+1} \in \delta(s_i, w_{i+1})\) for \(i = 0, ..., m-1\)
    3. \(s_m \in F\)
One last operation

Kleene star operation

• Let A be a language.
• The Kleene star operation is
  \[ A^* = \{x_1x_2...x_k \mid k \geq 0 \text{ and each } x_i \in A\} \]

Exercise
• \( A = \{w \mid w \text{ is a string of 0s and 1s containing an odd number of 1s}\} \)
• \( B = \{w \mid w \text{ is a string of 0s and 1s containing an even number of 1s}\} \)
• \( C = \{0, 1\} \)

• What are \( A^*, B^*, \) and \( C^*? \)
Clay Knee?

Kleene pronounced his last name klay'nee. His son, Ken Kleene, wrote: "As far as I am aware this pronunciation is incorrect in all known languages. I believe that this novel pronunciation was invented by my father." Dr. "Clay Knee" must have been a geek of the first water, to be sure!

- From http://visualbasic.about.com/od/usevb6/a/RegExVB6.htm

Kleene star

- Theorem: The class of languages recognized by NFAs is closed under the Kleene star operation.
If only…

... somebody would prove that the class of languages recognized by NFAs and the class of languages recognized by DFAs were equal...

Then...

• We’d have:
  – The class of regular languages is closed under:
    • Concatenation
    • Kleene star
And…

a cute proof for closure under union!

We can be that somebody!

• Theorem: A language is regular if and only if there exists an NFA that recognizes it.

• Proof:
  \( \Rightarrow \)
  Let \( A \) be a regular language…
Then there exists a DFA $M$

- $M$ is a 5-tuple $(Q, \Sigma, \delta_M, q_0, F)$, where
  - $Q$ is a finite set called the states
  - $\Sigma$ is a finite set called the alphabet
  - $\delta: Q \times \Sigma \rightarrow Q$ is the transition function
  - $q_0 \subseteq Q$ is the start state
  - $F \subseteq Q$ is a set of accept states

We need to show there exists an NFA $N$

- $N$ is a 5-tuple $(Q, \Sigma, \delta_N, q_0, F)$, where
  - $Q$ is a finite set called the states
  - $\Sigma$ is a finite set called the alphabet
  - $\delta: Q \times \Sigma \epsilon \rightarrow P(Q)$ is the transition function
  - $q_0 \subseteq Q$ is the start state
  - $F \subseteq Q$ is a set of accept states

- Can we define $N$ from $M$?
Now the other way!

• Theorem: A language is regular if and only if there exists an NFA that recognizes it.

• Proof:
  \( \Rightarrow \)
  Let \( A \) be a language accepted by NFA
  \( N = (Q, \Sigma, \delta, q_0, F) \)

Equivalent machines

• Definition: Two machines are equivalent if they recognize the same language

• Let’s prove:
  Theorem: Every NFA has an equivalent DFA.
Remember...

A simpler example
Removing choice

Proof:
• Let $A$ be a language accepted by NFA $N = (Q, \Sigma, \delta, q_0, F)$
• We construct a DFA $M = (Q', \Sigma, \delta', q_0', F')$ recognizing $A$
  – $Q' = P(Q)$
  – For $R \in Q'$ and $a \in \Sigma$, define $\delta'(R,a) = \bigcup_{r \in R} \delta(r,a)$
  – $q_0' = \{q_0\}$
  – $F' = \{ R \in Q' | R \text{ contains an accept state from } F \}$

Okay, but what about $\varepsilon$ arrows?

\begin{tikzpicture}
  \node[state, initial] (q1) at (0,0) {$q_1$};
  \node[state] (q2) at (1.5,0) {$q_2$};
  \node[state] (q3) at (3,0) {$q_3$};
  \node[state, accepting] (q4) at (4.5,0) {$q_4$};

  \draw[->] (q1) edge[loop above] node {$0,1$} (q1)
            edge node[above] {$1$} (q2)
            edge node[below] {$0,\varepsilon$} (q3);
  \draw[->] (q2) edge node[above] {$0,\varepsilon$} (q3)
            edge node[below] {$1$} (q4);
  \draw[->] (q3) edge node[above] {$0,1$} (q4);
\end{tikzpicture}
Modifying our construction

Proof:

• Let $A$ be a language accepted by NFA $N = (Q, \Sigma, \delta, q_0, F)$

• We construct a DFA $M = (Q', \Sigma, \delta', q_0', F')$ recognizing $A$
  - $Q' = P(Q)$
  - For $R \in Q'$ and $a \in \Sigma$,
    define $\delta'(R, a) = \bigcup_{r \in R} E(\delta(r, a))$
    where $E(R) = \{ q \mid q$ can be reached from $R$ along 0 or more $\varepsilon$ arrows $\}$
  - $q_0' = E(\{ q_0 \})$
  - $F' = \{ R \in Q' \mid R$ contains an accept state from $F \}$