CSE 460: Computability and Formal Languages

Finite State Automata (FA) and Regular Languages
Finite State Automata (FA)

1. FA: $M = \langle Q, \Sigma, \delta, q_0, A \rangle$
   - $Q$: Finite set of states
   - $\Sigma$: finite input set
   - $\delta$: transition function $Q \times \Sigma \to Q$
   - $q_0 \in Q$ initial state
   - $A$: set of acceptance states $\subseteq Q$

2. Example:

   ![Figure 1](image)

   Figure 1:

   $Q = \{q_0, q_1, \emptyset\}, \Sigma = \{a, b\}, A = \{q_1\}$

   $\delta(q_0, a) = q_0, \delta(q_0, b) = q_1, \delta(\emptyset, b) = q_1, \delta(q_1, a) = \emptyset,$

   $\delta(\emptyset, a) = \emptyset, \delta(\emptyset, b) = \emptyset$
Definition of FA Continued

1. Acceptance Conditions:
   
   (a) must be in final state  
   **and at the same time**  
   (b) reach the end of the string.

2. For the example FA:
   
   - Transitions for the string $x=aab$:
     $q_0 \rightarrow q_0 \rightarrow q_0 \rightarrow q_1$
     accepted?

   - What are the transitions for $x=abb$?
     accepted?

   - What are the transitions for $x=abba$?
     accepted?

   - Language (all accepted strings) for the FA:
     $L = a^*b^+$
Transition Function on a string ($\delta^*$)

1. Interested in the final state reached for a string.

2. Transition function $\delta^*$ defined for strings:
   \[ \delta^*(q, w) = \text{state reached after reading the input string } w, \text{ starting from state } q. \]

   **Note:**
   - $\delta$ function makes transition on an input symbol.
   - $\delta^*$ is a function defined in terms of $\delta$ on an input string.

3. $\delta^*$ is defined in terms of $\delta$ as follows:
   \[ \delta^*(q, ax) = \delta^*(q_1, x), \text{ where } q_1 = \delta(q, a) \]

   Write the above notation in English.

4. $\delta^*(q, \Lambda) = q$ needed?
   yes, for the base case of the recursive definition and this allows $\Lambda$ in a language.

   For the above FA: $\delta^*(q_0, ab) = q_1$ why?
   \[ \delta^*(\delta(q_0, a), b) = \delta^*(q_1, b) = \delta^*(\delta(q_1, b), \Lambda) = \delta^*(q_1, \Lambda) = q_1 \]

   Difference between $\delta(q_0, a) =$ and $\delta^*(q_0, a) =$?
Recursive Definitions of $\delta^*$

1. Explicitly showing the transition on the first input symbol of the string:

   $\delta^*(q, ax) = \delta^*(\delta(q, a), x)$

   $\delta^*(q, \Lambda) = q$

   This is same as described in previous page.

2. Explicitly showing the transition on the last input symbol of the string:

   $\delta^*(q, xa) = \delta(\delta^*(q, x), a)$

   $\delta^*(q, \Lambda) = q$

   Example: $\delta^*(q_0, aab) = \delta(\delta^*(q_0, aa), b) = \delta(\delta^*(q_0, a), a, b) = \delta(\delta(\delta^*(q_0, \Lambda), a), a, b) = \delta(\delta(\delta(q_0, a), a), b) =$
Definition of the Language Accepted by an FA

1. The Language recognized by an FA:
   If $M$ is an FA then
   $$T(M) = \{ x \in \Sigma^* | \delta^*(q_0, x) \in A \}$$

2. Another Definition of a Regular Language:

3. A set $L \subseteq \Sigma^*$ is regular if $\exists$ fa: $L = T(M)$
Pairwise Distinguishable Strings

1. Definition:
Two strings \(x\) and \(y\) in \(\Sigma^*\) are distinguishable with respect to a language \(L\) if there is a string \(z \in \Sigma^*\) so that one and only one of \(xz\) or \(yz\) is in \(L\).

\(z\) is said to distinguish \(x\) and \(y\) with respect to \(L\).

2. Example:
Let \(L\) be the language \(\{x \in \{0, 1\}^* | x \text{ ends with } 10\}\)
The strings 01011 and 100 are distinguishable with respect to \(L\) because for \(z = 0, 01011z \in L\) but \(100z \notin L\).

3. A set of pairwise distinguishable strings:
A set of strings is pairwise distinguishable with respect to a language \(L\) if every pair of strings in the set is distinguishable with respect to \(L\).

Example: \(L= (1 + 0)0^*\)
A set of pairwise distinguishable strings with respect to \(L\):
\(\{\Lambda, 1, 11\}\) because
\((\Lambda, 1): z = \Lambda, (\Lambda, 11): z = 1, (1, 11): z = \Lambda\)
Distinguishable Strings and Nonregular Languages

1. Theorem: If for some positive integer \( n \), there are \( n \) strings in \( \Sigma^* \), any two of which are distinguishable with respect to a language \( L \) (i.e., a set of \( n \) pairwise distinguishable strings), then there can be no FA recognizing \( L \) with fewer than \( n \) states.

Proof:
Suppose \( x_1, x_2, \ldots, x_n \) are \( n \) strings, any two of which are distinguishable with respect to \( L \). If \( M \) is any FA with fewer than \( n \) states, then by pigeonhole principle, the states \( \delta^*(q_0, x_1), \delta^*(q_0, x_2), \ldots \delta^*(q_0, x_n) \) cannot all be distinct, and so for some \( i \neq j \),
\[
\delta^*(q_0, x_i) = \delta^*(q_0, x_j). \tag{A}
\]
Since \( x_i \) and \( x_j \) are distinguishable with respect to \( L \), \( \delta^*(q_0, x_iz) \neq \delta^*(q_0, x_jz) \) for some \( z \in \Sigma^* \). Because,
\[
\delta^*(q_0, x_iz) = \delta^*(\delta^*(q_0, x_i), z) \quad \text{and} \quad \delta^*(q_0, x_jz) = \delta^*(\delta^*(q_0, x_j), z)
\]
we get
\[
\delta^*(q_0, x_i) \neq \delta^*(q_0, x_j).
\]
This is a contradiction with (A) above.

2. \( L = \{a^ib^i \mid i \geq 0\} \)
• has a set of infinite number of pairwise distinguishable strings:
  \( \{a^2b, a^3b, a^4b, \ldots \} \)

• Needs infinite number of states.

• Not a regular language.
Indistinguishable Strings and Minimum State FA

1. Lemma:

If $M = \{Q, \Sigma, q_0, A, \delta\}$ recognizes a language $L \subseteq \Sigma^*$, and $x$ and $y$ are two strings in $\Sigma^*$ for which $\delta^*(q_0, x) = \delta^*(q_0, y)$, then $x$ and $y$ are indistinguishable with respect to $L$.

2. (a) FA1: $L = (1 + 0)0^*$
   FA2: $L = (1 + 0)0^*$

(b) Set of strings corresponding to a state:

$L_q = \{x \mid \delta^*(q_0, x) = q\}$

For FA1: $L_{q_0} = \Lambda$, $L_{q_1} = (0 + 1)0^*$,
$L_{q_2} = (0 + 1)0^*1(0 + 1)^*$
A string from $L_{q0}$, a string from $L_{q1}$, and a string from $L_\emptyset$ are pairwise distinguishable.

Thus, for FA1:
If $x \in L_{q0}$, $y \in L_{q1}$, $z \in L_\emptyset$
Then $\{x, y, z\}$ is pairwise distinguishable $\{\Lambda, 1, 11\}$.

For FA2: If $x \in L_{q0}$, $y \in L_{q1}$, $z \in L_\emptyset$, $w \in L_{q2}$
Then $y$ and $w$ are not distinguishable with respect to $L$. $1$, $00$ are not distinguishable strings with respect to $(1 + 0)0^*$.

(c) $q_1, q_2$ could be merged.
Composite FA

1. Closure properties of Regular Languages:
   By definition of regular languages we can say:
   If $L_1$ and $L_2$ are two regular languages then
   $L_1 \cup L_2$, $L_1L_2$, $L_1^*$ are regular.

2. How about $L_1 \cap L_2$, $L_1 - L_2$ ? also regular.

3. Regular languages are closed under union, concatenation, Kleene star, set intersection, set difference, etc.

4. Given two FAs $M_1$ and $M_2$, can we construct new FAs to accept the the languages $L(M_1) \cup L(M_2)$, $L(M_1)L(M_2)$, $L(M_1)^*$, $L_1 \cap L_2$ and $L_1 - L_2$?

5. There are construction algorithms for creating composite FAs given two FAs.

6. We will, for now, focus on a construction algorithm for composite FA for $L(M_1) \cup L(M_2)$, $L_1 \cap L_2$ and $L_1 - L_2$
Construction of Composite FA for $L(M_1) \cup L(M_2), L_1 \cap L_2$ and $L_1 - L_2$

- Given two FAs, $M_1$ and $M_2$, construct FAs accepting $L(M_1) \cup L(M_2), L_1 \cap L_2$ and $L_1 - L_2$?
- States of new machine remember transitions in both machines.
- Simulate transitions of both machines by a composite state in new machine.
- $\delta_1(q_1, a) = p_1$ for $M_1$ and $\delta_2(q_2, a) = p_2$ for $M_2$ then $\delta_c((q_1, q_2), a) = (p_1, p_2)$ for the composite machine.
- State $(p_1, p_2)$ in new machine remembers transition in $M_1$ and transition in $M_2$ for the same input $a$ from $q_1$ to $p_1$ and from $q_2$ to $P_2$. 


\begin{itemize}
  \item $\delta^*(q_1, 1) = p_1$
  \item $\delta^*(q_2, 1) = p_2$
  \item $\delta^*((q_1, q_2), 1) = (p_1, p_2)$
  \item $\delta^*(q_1, 10) = p_1$
  \item $\delta^*(q_2, 10) = r_2$
  \item $\delta^*((q_1, q_2), 10) = (p_1, r_2)$
  \item $\delta^*(q_1, x) = m_1$
  \item $\delta^*(q_2, x) = n_2$
  \item $\delta^*((q_1, q_2), x) = (m_1, n_2)$
\end{itemize}
Theorem 3.4:
Suppose that $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ and $M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2)$ accept languages $L_1$, $L_2$, respectively. Let $M$ be a composite FA defined by
$M = (Q, \Sigma, q_0, A, \delta)$ where
$Q = Q_1 \times Q_2$ and $q_0 = (q_1, q_2)$
and the transition function $\delta$ is defined by
$\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$
for any $p \in Q_1, q \in Q_2,$ and $a \in \Sigma$ then

1. If $A = \{(p, q) \mid p \in A_1 \text{ or } q \in A_2\}$, $M$ accepts the language $L_1 \cup L_2$.
2. If $A = \{(p, q) \mid p \in A_1 \text{ and } q \in A_2\}$, $M$ accepts the language $L_1 \cap L_2$.
3. If $A = \{(p, q) \mid p \in A_1 \text{ and } q \notin A_2\}$, $M$ accepts the language $L_1 - L_2$.

Proof:
By Mathematical induction show
$\delta^*((p, q), x) = (\delta_1^*(p, x), \delta_2^*(q, x)) \forall x \in \Sigma^*$
Base case: $\delta^*((p, q), \Lambda) = (\delta_1^*(p, \Lambda), \delta_2^*(q, \Lambda))$
$\delta^*((p, q), \Lambda) = (p, q)$, $\delta_1^*(p, \Lambda) = p$ and $\delta_2^*(q, \Lambda)$ by definition of $\delta^*$
True for $|x| = n$ then true for $|x| = n + 1$
\[ \delta^*((p, q), xa) = \delta(\delta^*((p, q), x), a) = \delta((\delta_1^*(p, x), \delta_2^*(q, x)), a) = \\
(\delta_1(\delta_1^*(p, x), a), \delta_2(\delta_2^*(q, x), a)) = (\delta_1^*(p, xa), \delta_2^*(q, xa)) \]

A string is accepted by the composite machine if 

\[ (\delta_1^*(p, x), \delta_2^*(q, x)) \in A \]

If the set is defined as in case 1 this is the same as

saying \( (\delta_1^*(p, x) \in A_1 \) or \( \delta_2^*(q, x) \in A_2 \). In other words, that \( x \in L_1 \cup L_2 \) Cases 2 and 3 are similar.
Example

• $L_1 = 10^* = \{1, 10, 100, 1000, \ldots\}$
  $L_2 = 101^* = \{10, 101, 1011, \ldots\}$

• $L_1 \cap L_2 = 10$
  $L_1 \cup L_2 = 1 + 10 + 1000^* + 1011^*$
  $L_1 - L_2 = 1 + 1000^*$

Figure 3:

Composite FA: