Languages Accepted by DFAs

• Take DFA \( M \)

• Definition:
  – The language \( L(M) \) contains all input strings accepted by \( M \)

  \[ L(M) = \{ \text{strings that drive } M \text{ to a final state} \} \]

Example

\[ L(M) = \{abba\} \]

\[ \ \]

\[ M \]

\[ q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_5 \]

\[ a, b \rightarrow a, b \]

\[ \text{accept} \]
Another Example

$L(M) = \{\lambda, ab, abba\}$

Formally

• For a DFA $M = (Q, \Sigma, \delta, q_0, F)$

• Language accepted by $M$:

$L(M) = \{w \in \Sigma^*: \delta^{*}(q_0, w) \in F\}$

Observation

• Language rejected by $M$:

$L(M) = \{w \in \Sigma^*: \delta^{*}(q_0, w) \notin F\}$

More Examples

$L(M) = \{a^n b : n \neq 0\}$
Regular Languages

- A language $L$ is regular if there is a DFA $M$ such that $L = L(M)$.
- All regular languages form a language family.

Examples of regular languages:

- $\{abba\}$
- $\{\lambda, ab, abba\}$
- $\{a^n b : n \geq 0\}$
- $\{\text{all strings with prefix } ab\}$
- $\{\text{all strings without substring } 001\}$

There exist automata that accept these languages.
Another Example

$L = \{awa : w \in \{a, b\}^*\}$

There exist languages which are not Regular:

Example: $L = \{a^n b^n : n \geq 0\}$

There is no DFA that accepts such a language

(we will prove this later in the semester)

Nondeterministic Finite Accepter (NFA)

Alphabet = \{a\}

Nondeterministic Finite Accepter (NFA)

Alphabet = \{a\}

Two choices
Nondeterministic Finite Accepter (NFA)

Alphabet = \{a\}

Two choices

\[ q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \] No transition

\[ q_0 \xrightarrow{a} q_3 \] No transition

First Choice

\[ a \]

First Choice

\[ a \]

First Choice

\[ a \]
First Choice

All input is consumed

Second Choice

No transition: the automaton hangs
An NFA accepts a string: when there is a computation of the NFA that accepts the string

AND

all the input is consumed and the automaton is in a final state

Example

*aa* is accepted by the NFA:

```
q0  a   q1
  a ---^- is accepted
  -  q2
    "reject??" But this only tells us that choice didn't work...
```

because this computation accepts *aa*

Rejection example

```
q0  a   q1
    a ---^- is rejected
    -  q2
```

"accept"
First Choice

Second Choice

"reject??"
An NFA rejects a string:
when there is no computation of the NFA
that accepts the string:
  • All the input is consumed and the
    automaton is in a non final state
  OR
  • The input cannot be consumed

Example

a is rejected by the NFA:

All possible computations lead to rejection
First Choice

No transition: the automaton hangs

Second Choice

Input cannot be consumed
Second Choice

No transition: the automaton hangs

Second Choice

"reject??"

aaa is rejected by the NFA:

"reject??"

All possible computations lead to rejection
Language accepted: \[ L = \{aa\} \]

Lambda Transitions

\[
\begin{align*}
q_0 \xrightarrow{a} q_1 & \xrightarrow{\lambda} q_2 \xrightarrow{a} q_3 \\
q_0 \xrightarrow{a} q_1 & \xrightarrow{\lambda} q_2 \xrightarrow{a} q_3 \\
\end{align*}
\]
(read head does not move)

String is accepted

all input is consumed

String aa is accepted

Rejection Example
No transition: the automaton hangs

String aaa is rejected
Language accepted:  \( L = \{aa\} \)

Another NFA Example
Language accepted

\[ L = \{ab, abab, ababab, \ldots\} = \{ab\}^* \]

Another NFA Example

\[ q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{\lambda} q_3 \]

\[ q_0 \xrightarrow{0} q_1 \xrightarrow{0, 1} q_2 \]
Language accepted

\[ L(M) = \{\lambda, 10, 1010, 101010, \ldots\} = \{10\}^* \]

Remarks:
- The \( \lambda \) symbol never appears on the input tape
- Simple automata:

\[
\begin{align*}
M_1 & \quad \quad M_2 \\
q_0 & \quad \quad q_0 \\
L(M_1) = \{\} & \quad L(M_2) = \{\lambda\}
\end{align*}
\]

• NFAs are interesting because we can express languages easier than DFAs

Formal Definition of NFAs

\[ M = (Q, \Sigma, \delta, q_0, F) \]

- \( Q \) : Set of states, i.e. \( \{q_0, q_1, q_2\} \)
- \( \Sigma \) : Input alphabet, i.e. \( \{a, b\} \)
- \( \delta \) : Transition function
- \( q_0 \) : Initial state
- \( F \) : Final states
Transition Function $\delta$

$\delta(q_0, 1) = \{q_1\}$

$\delta(q_1, 0) = \{q_0, q_2\}$

$\delta(q_0, \lambda) = \{q_0, q_2\}$

$\delta(q_2, 1) = \emptyset$
Extended Transition Function

\[ \delta^*(q_0, a) = \{q_1\} \]

\[ \delta^*(q_0, aa) = \{q_4, q_5\} \]

Formally

\[ q_j \in \delta^*(q_i, w) : \text{there is a walk from } q_i \text{ to } q_j \text{ with label } w \]
The Language of an NFA

\[ F = \{ q_0, q_5 \} \]

\[ \delta^*(q_0, aa) = \{ q_4, q_5 \} \quad \text{if } aa \in L(M) \]

\[ F = \{ q_0, q_5 \} \]

\[ \delta^*(q_0, ab) = \{ q_2, q_3, q_0 \} \quad \text{if } ab \in L(M) \]

\[ F = \{ q_0, q_5 \} \]

\[ \delta^*(q_0, abaa) = \{ q_4, q_5 \} \quad \text{if } abaa \in L(M) \]

\[ F = \{ q_0, q_5 \} \]

\[ \delta^*(q_0, aba) = \{ q_1 \} \quad \text{if } aba \notin L(M) \]
Formally

- The language accepted by NFA $M$ is:
  \[ L(M) = \{w_1, w_2, w_3, \ldots \} \]
  where
  \[ \delta^*(q_0, w_m) = \{q_i, q_j, \ldots, q_k, \ldots\} \]
  and there is some $q_k \in F$ (final state)

What’s Next

- Read
  - Linz Chapter 1, 2.1-2.3 (skip 2.4), 3.1, 3.2
  - JFLAP Startup, Chapter 1, 2.1, (skip 2.3), 4
- Next Lecture Topics from Chapter 2.3, 3.1 and 3.2
  - Equivalence of NFA and DFA
  - Regular Expressions and Regular Languages
- Quiz 1 in Recitation on Wednesday 9/17
  - Covers Linz 1.1, 1.2, 2.1, 2.2, 2.3, and JFLAP 1, 2.1
  - Closed book, but you may bring one sheet of 8.5 x 11 inch paper with any notes you like.
  - Quiz will take the full hour
- Homework
  - Homework 1 Due Today
  - Homework 2 Available By Friday Morning,
  - Homework 2 Due Next Thursday