Distinguishing strings

Let $L$ be a language over $\Sigma$, and let $x \in \Sigma^*$. Let

$$L/x = \{ z \in \Sigma^* \mid xz \in L \}$$

Two strings $x, y \in \Sigma^*$ are **distinguishable with respect to** $L$ if $L/x \neq L/y$.

Suppose we have an FA $M=(Q, \Sigma, q_0, A, \delta)$ and let $x, y \in \Sigma^*$ be such that

$$\delta^*(q_0, x) = \delta^*(q_0, y)$$

Then,

$$\delta^*(q_0, xz) = \delta^*(q_0, yz) \quad \forall z \in \Sigma^*$$

(i.e. once $x$ and $y$ "meet" in the same state, they go together from then on).

We say that $x, y$ are **indistinguishable**.

The above def. defines distinguishability wrt a language $L$, not wrt an automaton.
Distinguishing strings

Thm:

Let $L$ be a language over $\Sigma$. Suppose that there are $n$ strings that are pairwise distinguishable wrt $L$. Then every FA for $L$ needs at least $n$ states.

Suppose we have an FA $M = (Q, \Sigma, q_0, A, \delta)$ for $L$ and let $x_1, \ldots, x_n \in \Sigma^*$ be the distinguishable strings. Then for every $i \neq j$ $\delta^*(q_0, x_i) \neq \delta^*(q_0, x_j)$ (WHY ?)

Thus we need at least as many states as distinguishable strings.
Myhill-Nerode

Let \( L \) be a language over \( \Sigma \). We define the **indistinguishability relation** \( I_L \) on \( \Sigma^* \) as follows:

For every \( x, y \in \Sigma^* \): \( x \ I_L \ y \) iff \( L/x = L/y \)

**Claim**: \( I_L \) is an equivalence relation.

**Thm** (Myhill-Nerode):

\( L \) is regular iff the set of equivalence classes of \( I_L \) is finite.

\( \text{Pf: } \Rightarrow : \quad \text{"if } L \text{ has a corresponding FA then } \# \text{eq. classes } \leq \# \text{states"} \)

(same argument as thm from previous slide)
Using Myhill-Nerode

**Thm (Myhill-Nerode):**

L is regular iff the set of equivalence classes of $I_L$ is finite.

Is $abpal = \{ w \in \{a,b\}^* \mid w \text{ is a palindrome} \}$ regular?

Find 2 distinguishable strings: $x = aa$, $y = a$, $z = baa$ \Rightarrow xz$ is a palindrome, $yz$ is not.

**CLAIM:** $a^i$ and $a^j$ for $i \neq j$ are distinguishable ($L/a^i \neq L/a^j$)

**PF:** take $z = ba^i$

$a^i$, $i \geq 0$ are all pairwise disting. \Rightarrow \#equivalence classes infinite \Rightarrow L nonregular.
Using Myhill-Nerode

**Thm (Myhill-Nerode):**

$L$ is regular iff the set of equivalence classes of $I_L$ is finite.

Is $\{ w \in \{a,b\}^* \mid \text{number of } a\text{'s in } w \text{ is divisible by } 3 \}$ regular?

**CLAIM:** $x, y$ are indistinguishable iff $\#a's \text{ in } x \equiv \#a's \text{ in } y \pmod{3}$

**PF:** $L/x = L/y \iff \#z \in \Sigma^*: xz \in L \iff yz \in L$

**Equiv. classes:**

- $\{ x \mid \#a's \text{ in } x \equiv 0 \pmod{3} \}$
- $\{ x \mid 1 \}$
- $\{ x \mid 2 \}$

- $\forall m \in \{0, 1, 2\}$
  
  $x \in L$ iff $yz \in L$
  
  $\#a's \text{ in } x \equiv m \pmod{3}$

  $\equiv \#a's \text{ in } y \equiv m \pmod{3}$ for $yz \in L$

\[\square\]
**Claim**: If the set of equivalence classes of $I_L$ is finite, we can construct a finite automaton recognizing $L$.

If: Let $n$ be the finite # of equivalence classes of $I_L$. Let $x_1,\ldots,x_n$ be strings from each of these classes (respectively). $[y]$ - means the equivalence class containing $y$.

$$M = (\{[x_i] | i=1,\ldots,n\}, \Sigma, [\Lambda], \{[x_i] | x_i \in L\}, \delta)$$

$$\delta([x_i], \sigma) = [x_i \sigma] \quad \forall i \in \{1,\ldots,n\} \quad \forall \sigma \in \Sigma$$

**book**: $M = (\{[x] | x \in \Sigma^*\}, \Sigma, [\Lambda], \{[x] | x \in L\}, \delta)$

$$\delta([x], \sigma) = [x \sigma] \quad \forall x \in \Sigma^*, \forall \sigma \in \Sigma$$

Notice: This FA is “minimal” for $L$, i.e. its number of states is the smallest possible.
We know that we can find a minimal FA for a language $L$ from the equivalence classes of $I_L$.

What if $L$ is given by an FA?

1) remove unreachable states
Algorithm for identifying equivalent states:

1. List all pairs of different states.
2. Cross out all pairs with exactly one accepting state. (1)
3. Cross out every pair \((p,q)\) such that there exists \(\sigma \in \Sigma\) such that the pair \((\delta(p,\sigma),\delta(q,\sigma))\) is crossed out.
4. Repeat step 3 until no more new pairs are crossed out.
5. Remaining pairs form equivalent states.
Pumping Lemma for Regular Lang.

Let $M$ be an FA with $n$ states. Consider the computation of $M$ on a string $x$ with $|x| > n$.

If $|x| > n$ then a state must be repeated within the first $n$ steps of the computation.

$X = \sigma_1 \sigma_2 \ldots \sigma_k$

$X$ is accepted

$\sigma_1 \sigma_3 \sigma_9 \sigma_{10}$ is also accepted

$\sigma_1 \sigma_3 (\sigma_4 \ldots \sigma_8)^m \sigma_9 \sigma_{10}$ is accepted for $m \geq 0$
Thm (The Pumping Lemma for Regular Languages):

Let $L$ be a regular language. Then there exists an integer $n$ such that for every $x \in L$ with $|x| \geq n$ there are strings $u,v,w$ such that
1. $x = uvw,$
2. $|uv| \leq n,$
3. $|v| > 0,$ and
4. for every $m \geq 0,$ $uv^mw \in L.$

Answer these questions:

Can we use the pumping lemma to prove that a language is nonregular? **YES**
Can we use the pumping lemma to prove that a language is regular? **NO**
Using the Pumping Lemma

Is \( L = \{ a^k b^{2k} \mid k \geq 0 \} \) regular? \( \text{No} \)

Suppose \( L \) is regular. Then the Pump. L. holds for \( L \).

\( \exists n \) s.t. \( x = a^n b^{2n} \) satisfies the conditions 1-4)

\[ |uv| \leq n \Rightarrow u, v \text{ contain only } a \text{'s} \]  
(2)

\[ v \text{ contains } \geq 1 \ a \]  
(3)

Consider (4) with \( m = 0 \)

\[ uv^m w = uw \]

\( \text{has } 2n \ \text{b's} \)  
\( \text{has } \leq n-1 \ a \text{'s} \)  
\& \( L \)  
\( \downarrow \text{ thus } L \text{ nonreg.} \) \( \Box \)
Is \( L = \{ a^k \mid k \text{ is a square} \} \) regular? \( \text{NO.} \)

Suppose \( L \) is regular, then by the Pumping Lemma \( \exists \) \( n \) s.t.

for string \( x = a^{n^2} \in L \) (and \( |x| \geq n \)) the conditions (1)-(4) hold.

\[
\begin{align*}
\text{x:} & \quad a^{n^2} \\
\text{u:} & \quad a^n \\
\text{v:} & \quad a^m \\
\text{w:} & \quad a^{n^2-n} \\
\text{uNW:} & \quad a^n \\
\text{v:} & \quad a \\
\text{w:} & \quad a^{n^2-n} \\
\end{align*}
\]

\( m = 0: \)

\( uv^m w = uw \)

\( n^2 - n \leq |uw| \leq n^2 - 1 \)

which string \( \in L \) is the longest string shorter than \( x \)?

\( |a^{(n-1)^2}| = n^2 - 2n + 1 \)
Decision Problems for Regular Lang.

A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-Λ) M, is \( L(M) = \emptyset \)?

Reachable from \( q_0 \):

1) \( q_0 \) is reachable
2) for a reachable \( q \in Q \) \( \forall \sigma \in \Sigma \), let \( \delta(q, \sigma) \) be reachable
3) n.e.

ALGO: check if \( A \cap \) reachable states = \( \emptyset \)
- if YES, answer \( \emptyset \)
- else nonempty
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-$\Lambda$) $M$, is $L(M) = \emptyset$?
- Given an FA $M$, is $L(M)$ finite?

Consider all strings $x$ with length $\leq 2(\#\text{states})$

Check the computation of $M$ on $x$

- if $x$ is accepted and goes through a loop, answer NOT FINITE
- if no $x$ satisfies the above, answer FINITE

[Section 5.4]
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-Λ) $M$, is $L(M) = \emptyset$?
- Given an FA $M$, is $L(M)$ finite?
- Given two FA's $M_1$ and $M_2$, is $L(M_1) \cap L(M_2) = \emptyset$?

\begin{itemize}
  \item Construct $M_3$ such that $L(M_3) = L(M_1) \cap L(M_2)$
  \item Check $L(M_3) = \emptyset$
\end{itemize}
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-Λ) M, is $L(M) = \emptyset$?
- Given an FA M, is $L(M)$ finite?
- Given two FA’s $M_1$ and $M_2$, is $L(M_1) \cap L(M_2) = \emptyset$?
- Given two FA’s $M_1$ and $M_2$, is $L(M_1) = L(M_2)$?

**Solution 1:**

If $L(M_1) - L(M_2) = \emptyset$ and $L(M_2) - L(M_1) = \emptyset$, then if both answers are YES, answer YES, otherwise answer different.

**Solution 2:**

Minimize $M_1$, $M_2$ and see if you get the same FA.
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-Λ) $M$, is $L(M) = \emptyset$?
- Given an FA $M$, is $L(M)$ finite?
- Given two FA’s $M_1$ and $M_2$, is $L(M_1) \cap L(M_2) = \emptyset$?
- Given two FA’s $M_1$ and $M_2$, is $L(M_1) = L(M_2)$?
- Given two FA’s $M_1$ and $M_2$, is $L(M_1) \subseteq L(M_2)$?

check if $L(M_1) - L(M_2) = \emptyset$
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-[Λ]) M, is \( L(M) = \emptyset \) ?
- Given an FA M, is \( L(M) \) finite ?
- Given two FA's \( M_1 \) and \( M_2 \), is \( L(M_1) \cap L(M_2) = \emptyset \) ?
- Given two FA's \( M_1 \) and \( M_2 \), is \( L(M_1) = L(M_2) \) ?
- Given two FA's \( M_1 \) and \( M_2 \), is \( L(M_1) \subseteq L(M_2) \) ?
- Given two regular expressions, do they correspond to the same language?
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:

- Given an FA (NFA, NFA-Λ) \( M \), is \( L(M) = \emptyset \)?
- Given an FA \( M \), is \( L(M) \) finite?
- Given two FA’s \( M_1 \) and \( M_2 \), is \( L(M_1) \cap L(M_2) = \emptyset \)?
- Given two FA’s \( M_1 \) and \( M_2 \), is \( L(M_1) = L(M_2) \)?
- Given two FA’s \( M_1 \) and \( M_2 \), is \( L(M_1) \subseteq L(M_2) \)?
- Given two regular expressions, do they correspond to the same language?
- Given an FA \( M \), is it minimal? Run minimization algo & see