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 $x, y$ “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.

$x_1, \ldots, x_n$ be the strings

$n$ different states \( \text{(one of them might be equal to } q_0) \)
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.

PF: $\Rightarrow$ is (in contraposition) equiv. to Thm from previous slide

$\Leftarrow$ will prove shortly
Using Myhill-Nerode

Thm (Myhill-Nerode):
L is regular iff the set of equivalence classes of $I_L$ is finite.

Is $abpal = \{ \text{w} \in \{a,b\}^* \mid \text{w is a palindrome} \}$ regular? NO.

2 distinct strings:
- $x = ab$
- $y = b$
- $z = ab$

$xz = abab \in abpal$
$y \neq \epsilon$
$y \in abpal$

Consider $x = a^i b^i$ for $i \geq 0$, $i \neq j$ take $z = a^j$ therefore $x, y$ are distinct.

Thus, $a^i b^i$ are all pairwise distinct strings $\Rightarrow \# \text{equiv. classes of } I_L \text{ is } \infty$

$\Rightarrow L$ not regular
Using Myhill-Nerode

**Thm (Myhill-Nerode):**

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

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

Yes.

$[a] = \{ u \in \{a,b\}^* \mid \text{#a's in } u \equiv 1 \pmod{3} \}$

$[aa] = \{ 2 \}$

$[\Lambda] = \{ 0 \}$

number of eq. classes = 3 - finite $\Rightarrow L$ reg. \( \Box \)
Proving Myhill-Nerode

**Claim**: If the set of equivalence classes of $I_L$ is finite, we can construct a finite automaton recognizing $L \subseteq \Sigma^*$.

Suppose $n$ equivalence classes: $[x_1], [x_2], \ldots, [x_n]

M = (Q, \Sigma, \delta, \lambda, q_0)\text{ where }

\[\delta(q, \sigma) = [x_i, \sigma]\text{ for all } q \in Q, \sigma \in \Sigma\]

M_{\text{book}} = (\{[x] | x \in \Sigma^*\}, \Sigma, \lambda, \{[x] | x \in L\}, \delta_{\text{book}})\text{ where }

\[\delta_{\text{book}}([x], \sigma) = [x, \sigma]\]

Notice: This FA is “minimal” for $L$, i.e. its number of states is the smallest possible.
Minimizing FA

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?

**STEP 1:**

*cross out unreachable states*
Minimizing FA

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. (2)
4. Repeat step 3 until no more new pairs are crossed out.
5. Remaining pairs form equivalent states.

[Diagram showing the process of minimizing a finite automaton]
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 its computation will look like this:

(Will repeat a state in the first $n+1$ steps of the computation)
Pumping Lemma for Regular Lang.

**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 \( m \geq 0 \), \( uv^m w \in L \).

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?

Suppose \( L \) regular. Therefore the pumping lemma holds. Thus \( \exists n \).

Consider \( x = a^n b^{2n} \)

We have \( |x| \geq n \)

Since \( |uv| \leq n \)

we have:

- \( u, v \) contain only \( a \)'s

Consider \( m = 2 \) then \( uv^2w \) has \( 2n \) \( b \)'s

\( \geq n+1 \) \( a \)'s

\( \Rightarrow uv^2w \in L \)
Using the Pumping Lemma

Is \( L = \{ a^k \mid k \text{ is a square} \} \) regular?

Assume \( L \) regular.

Then the Pumping Lemma holds for \( L \Rightarrow \exists n \ s.t. \forall x \in L \ s.t. |x| \geq n \ \exists u, v, w \ \text{for which} \ \text{conditions 1-4 hold.} \)

Consider

\( x = a^{n^2} \in L \) and \( |x| = n^2 \geq n \)

by 2) \( |uv| \leq n \)

3) \( |v| \geq 1 \)

Thus \( 1 \leq |v| \leq n \) consider \( m=2: \ uv^m w = uvvw \ n^2 + 1 \leq |uvvw| \leq n^2 + n \)

What is the shortest string in \( L \) which is longer than \( x \)? \( a^{(n+1)^2} \)

somewhere here \( uvvw \in L \)

\( n^2 \in L \) nobody here in \( L \)

\( (n+1)^2 = n^2 + 2n + 1 \)

\( \in L \)
A decision problem: answer is YES or NO.

Give algorithms for the following decision problems:
- Given an FA (NFA, NFA-Λ) M, is \( L(M) = \emptyset \) ?

**Approach #1:** MINIMIZE AND SEE IF
GET

**Approach #2:**
Set of reachable states from \( q_0 \)
1) \( q_0 \) is reachable
2) if \( q \) is reachable then \( \forall \sigma \in \Sigma : s(q, \sigma) \) is also reachable
3) nothing else reachable
check if this set contains an accepting state
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?
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?
- Given two FA’s \(M_1\) and \(M_2\), is \(L(M_1) \cap L(M_2) = \emptyset\)?

Construct \(M_3\) s.t. \(L(M_3) = L(M_1) \cap L(M_2)\) and then
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?
- 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 $L(M_1) = L(M_2)$?

**Approach #1:**

Check if

\[
\begin{align*}
L(M_1) - L(M_2) &= \emptyset \\
L(M_2) - L(M_1) &= \emptyset
\end{align*}
\]

If both YES then say YES (equal)

Otherwise say NO (≠)

**Approach #2:**

Minimize both and check if equal
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-$\Lambda$) $M$, 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?

Convert them to FA’s and
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? 

minimize & see if you get the same FA.