

CS 483 - Introduction to the Theory of Computation

Lecture 6

Non-regular Language Revisited

Why is the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$ not regular?

Non-regular Language Revisited

Why is the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$ not regular?

To compute L , the machine needs to **count** the number of a 's and b 's.

Consider the input $a^n b^m$ presented as a stream:

$a \quad a \quad a \quad \dots \quad a \quad a \quad b \quad b \quad \dots \quad b \quad b$



need to store n

Non-regular Language Revisited

To store a natural number up to n , the machine needs $\log_2 n$ bits of storage and therefore must **scale** with n .

The “storage” of a DFA is the set of states which does **NOT scale**.

The DFA cannot perfectly distinguish among the infinite set of prefixes

$$a^0, a^1, \dots, a^n, \dots$$

Non-regular Language Revisited

Alternative proof that $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular:

Assume L is regular and is computed by a DFA $M = (Q, \Sigma, \delta, q_0, F)$.

For $w \in \Sigma^*$, let $r(w) \in Q$ be the final state that M reaches on input w .

- Crucially, $r(w)$ is also the state that M reaches on input wv after the prefix w .

Non-regular Language Revisited

Alternative proof that $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular:

Assume L is regular and is computed by a DFA $M = (Q, \Sigma, \delta, q_0, F)$.

For $w \in \Sigma^*$, let $r(w) \in Q$ be the final state that M reaches on input w .

- Crucially, $r(w)$ is also the state that M reaches on input wv after the prefix w .

Claim: $r(a^i) \neq r(a^j)$ for every $i \neq j$, which contradicts Q being finite.

- If $r(a^i) = r(a^j)$, then $a^j b^i$ will also be accepted by M .

Myhill-Nerode Theorem

Definition. Given a language $L \subseteq \Sigma^*$, we say two strings $x, y \in \Sigma^*$ are **indistinguishable** under L if

$$\forall z \in \Sigma^*, \quad xz \in L \Leftrightarrow yz \in L.$$

In this case we write $x \sim_L y$.

Myhill-Nerode Theorem

Definition. Given a language $L \subseteq \Sigma^*$, we say two strings $x, y \in \Sigma^*$ are **indistinguishable** under L if

$$\forall z \in \Sigma^*, \quad xz \in L \Leftrightarrow yz \in L.$$

In this case we write $x \sim_L y$.

Notice that $\sim_L$ is an equivalence relation:

- Reflexive: $x \sim_L x$
- Symmetric: if $x \sim_L y$ then $y \sim_L x$
- Transitive: if $x \sim_L y$ and $y \sim_L z$ then $x \sim_L z$

Myhill-Nerode Theorem

Theorem. If L is a regular language, then $\sim_L$ has a finite number of equivalence classes.

Myhill-Nerode Theorem

Theorem. If L is a regular language, then $\sim_L$ has a finite number of equivalence classes.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that computes L .

For each $q \in Q$, define:

$$C_q = \{w \in \Sigma^* \mid M \text{ reaches } q \text{ on input (prefix) } w\}$$

Then the strings in the same C_q are indistinguishable.

Myhill-Nerode Theorem

Theorem. If L is a regular language, then $\sim_L$ has a finite number of equivalence classes.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that computes L .

For each $q \in Q$, define:

$$C_q = \{w \in \Sigma^* \mid M \text{ reaches } q \text{ on input (prefix) } w\}$$

Then the strings in the same C_q are indistinguishable.

- If $x, y \in C_q$ but $x \not\sim_L y$, assume $xz \in L$ and $yz \notin L$
- it means starting from q , the transition with z ends up both in F and $Q \setminus F$.

Myhill-Nerode Theorem

Theorem. If L is a regular language, then $\sim_L$ has a finite number of equivalence classes.

To apply the theorem to prove non-regularity of L , we need to find an **infinite** set of strings that are **pairwise distinguishable** under L .

Applications of Myhill-Nerode Theorem

Example 1.

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$

The strings $a^0, a^1, a^2, \dots$ are pairwise distinguishable:

When $i \neq j$, we have $a^i \not\sim_L a^j$ since

$$a^i b^i \in L, \quad a^j b^i \notin L$$

Applications of Myhill-Nerode Theorem

Example 2.

$$L = \{ww \mid w \in \{a, b\}^*\}$$

Applications of Myhill-Nerode Theorem

Example 2.

$$L = \{ww \mid w \in \{a, b\}^*\}$$

The strings $a^0b, a^1b, a^2b, \dots$ are pairwise distinguishable:

When $i \neq j$, we have $a^ib \not\sim_L a^jb$ since

$$a^ib a^ib \in L, \quad a^jb a^ib \notin L$$

Applications of Myhill-Nerode Theorem

Example 3.

$$L = \{c^m a^n b^n \mid m \geq 1, n \in \mathbb{N}\} \cup \{a^m b^n \mid m, n \in \mathbb{N}\}$$

Applications of Myhill-Nerode Theorem

Example 3.

$$L = \{c^m a^n b^n \mid m \geq 1, n \in \mathbb{N}\} \cup \{a^m b^n \mid m, n \in \mathbb{N}\}$$

The strings $ca^0, ca^1, ca^2, \dots$ are pairwise distinguishable:

When $i \neq j$, we have $ca^i \not\sim_L ca^j$ since

$$ca^i b^i \in L, \quad ca^j b^i \notin L$$

Applications of Myhill-Nerode Theorem

Example 4.

$$L = \{a^n \mid n \in \mathbb{N} \text{ is a prime number}\}$$

Applications of Myhill-Nerode Theorem

Example 4.

$$L = \{a^n \mid n \in \mathbb{N} \text{ is a prime number}\}$$

The strings in L are pairwise distinguishable:

For prime numbers $p < q$, we have $a^p \not\sim_L a^q$ since otherwise:

$$a^q \in L \Rightarrow a^{q+(q-p)} \in L \Rightarrow \dots \Rightarrow a^{q+q(q-p)} \in L$$

But $q + q(q - p) = q(1 + q - p)$ is composite.

Applications of Myhill-Nerode Theorem

Example 5.

$$L = \{a; b \mid a \text{ and } b \text{ are coprime binary integers}\}$$

Applications of Myhill-Nerode Theorem

Example 5.

$$L = \{a; b \mid a \text{ and } b \text{ are coprime binary integers}\}$$

The strings $p_1; , p_2; , p_3; , \dots$ are pairwise distinguishable, where each p_i is a different prime number.

For prime numbers $p \neq q$, we have $p; \not\sim_L q;$ since

$$p; q \in L, \quad q; q \notin L$$

Myhill-Nerode Theorem

Theorem. A language L is a regular **if and only if** $\sim_L$ has a finite number of equivalence classes.

Myhill-Nerode Theorem

Theorem. A language L is a regular **if and only if** $\sim_L$ has a finite number of equivalence classes.

To prove the converse, we need to construct a DFA M for L .

The previous proof of the forward direction shows that the states of M must correspond to $\sim_L$ or a refinement of $\sim_L$.

Myhill-Nerode Theorem

Example: $\sim_L$ has two equivalence classes:

$$C_0 = \{w \in \{0,1\}^* \mid w \text{ contains even number of } 1's\}$$

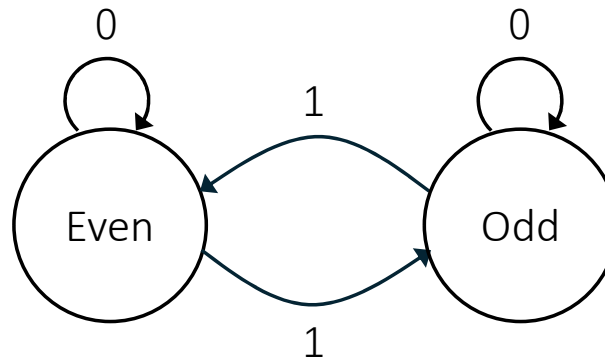
$$C_1 = \{w \in \{0,1\}^* \mid w \text{ contains odd number of } 1's\}$$

Myhill-Nerode Theorem

Example: $\sim_L$ has two equivalence classes:

$$C_0 = \{w \in \{0,1\}^* \mid w \text{ contains even number of } 1's\}$$

$$C_1 = \{w \in \{0,1\}^* \mid w \text{ contains odd number of } 1's\}$$



Myhill-Nerode Theorem

Suppose $\sim_L$ has equivalent classes $C_0, C_1, \dots, C_k$.

Construct DFA $M = (Q, \Sigma, \delta, q_0, F)$ where:

- $Q = \{q_0, q_1, \dots, q_k\}$
- $\delta(q_i, a) = q_j$ if $xa \in C_j$ for **some** $x \in C_i$

Myhill-Nerode Theorem

Suppose $\sim_L$ has equivalent classes $C_0, C_1, \dots, C_k$.

Construct DFA $M = (Q, \Sigma, \delta, q_0, F)$ where:

- $Q = \{q_0, q_1, \dots, q_k\}$
- $\delta(q_i, a) = q_j$ if $xa \in C_j$ for **some** $x \in C_i$
 - The choice of x **does not matter**: If $x \sim_L y$, then $xa \sim_L ya$.
- q_0 is chosen by $C_0 \ni \varepsilon$
- $F = \{q_i \mid C_i \subseteq L\}$.

Myhill-Nerode Theorem

Theorem. A language L is a regular if and only if $\sim_L$ has a finite number of equivalence classes.

Forward direction: For every M , $|Q| \geq \#$ of equivalence classes;

Converse direction: There exists M with $|Q| = \#$ of equivalence classes.

Myhill-Nerode Theorem

Theorem. A language L is a regular **if and only if** $\sim_L$ has a finite number of equivalence classes.

When L is regular, the number of $\sim_L$ equivalence classes is exactly the **minimum number of states** of a DFA computing L .

Forward direction: For every M , $|Q| \geq \#$ of equivalence classes;

Converse direction: There exists M with $|Q| = \#$ of equivalence classes.

Myhill-Nerode Theorem

Theorem. A language L is a regular **if and only if** $\sim_L$ has a finite number of equivalence classes.

When L is regular, the number of $\sim_L$ equivalence classes is exactly the **minimum number of states** of a DFA computing L .

Furthermore, the minimum DFA is **unique** (up to renaming the states).

DFA Minimization

Problem: Given a DFA $M = (Q, \Sigma, \delta, p_0, F)$, construct another DFA M' such that $L(M') = L(M)$ with the minimum number of states.

DFA Minimization

Problem: Given a DFA $M = (Q, \Sigma, \delta, p_0, F)$, construct another DFA M' such that $L(M') = L(M)$ with the minimum number of states.

For each $q \in Q$, define:

$$C_q = \{w \in \Sigma^* \mid M \text{ reaches } q \text{ on input (prefix) } w\}$$

By Myhill-Nerode Theorem, we only need to:

- Remove states q with $C_q = \emptyset$;
- Find out which pairs of states q and q' are distinguishable.

DFA Minimization

Algorithm:

- Starts with two class of states: F and $Q \setminus F$;
- While there exists a class S and a symbol a such that
$$\delta(q, a) \text{ is not in the same class for all } q \in S:$$
 - Split S into subclasses according to $\delta(q, a)$;
- Merge the states within the same class.

NFA Minimization?

There is no nice analogy of Myhill-Nerode Theorem for NFAs:

