Proving Non-regularity

Lecture 6

February 6, 2025

Regular Languages, DFAs, NFAs

Theorem

Languages accepted by DFAs, NFAs, and regular expressions are the same.

Question: Is every language a regular language?

Regular Languages, DFAs, NFAs

Theorem

Languages accepted by DFAs, NFAs, and regular expressions are the same.

Question: Is every language a regular language? No.

Regular Languages, DFAs, NFAs

Theorem

Languages accepted by DFAs, NFAs, and regular expressions are the same.

Question: Is every language a regular language? No.

- Each DFA M can be represented as a string over a finite alphabet Σ by appropriate encoding. Or think of regular expressions which are easy to view as strings.
- Hence number of regular languages is *countably infinite*
- Number of languages is *uncountably infinite*
- Hence there must be a non-regular language!

A simple, canonical non-regular language

$L = \{0^k 1^k \mid k \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots, \}$

A simple, canonical non-regular language

$L = \{0^k 1^k \mid k \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots, \}$

Theorem

L is not regular.

A simple, canonical non-regular language

$L = \{0^k 1^k \mid k \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots, \}$

Theorem

L is not regular.

Question: Proof?

A simple, canonical non-regular language

$L = \{0^k 1^k \mid k \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots, \}$

Theorem

L is not regular.

Question: Proof?

Intution: Any program to recognize L seems to require counting number of zeros in input which cannot be done with fixed memory.

A simple, canonical non-regular language

$L = \{0^k 1^k \mid k \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots, \}$

Theorem

L is not regular.

Question: Proof?

Intution: Any program to recognize L seems to require counting number of zeros in input which cannot be done with fixed memory.

How do we formalize intuition and come up with a formal proof?

Proof by Contradiction

- Suppose L is regular. Then there is a DFA M such that $L(M) = L$.
- Let $M = (Q, \{0, 1\}, \delta, s, A)$ where $|Q| = n$.

Proof by Contradiction

- Suppose L is regular. Then there is a DFA M such that $L(M) = L$.
- Let $M = (Q, \{0, 1\}, \delta, s, A)$ where $|Q| = n$.

Consider strings $\epsilon, 0, 00, 000, \cdots, 0^n$ total of $n + 1$ strings.

Proof by Contradiction

- Suppose L is regular. Then there is a DFA M such that $L(M) = L$.
- Let $M = (Q, \{0, 1\}, \delta, s, A)$ where $|Q| = n$.

Consider strings $\epsilon, 0, 00, 000, \cdots, 0^n$ total of $n + 1$ strings.

What is the behavior of M on these strings? Let $q_i = \delta^*(s, 0^i)$.

By pigeon hole principle $q_i = q_j$ for some $0 \leq i < j \leq n$.
M **is in the same state** after reading 0^i and 0^j where $i \neq j$.

Proof by Contradiction

- Suppose L is regular. Then there is a DFA M such that $L(M) = L$.
- Let $M = (Q, \{0, 1\}, \delta, s, A)$ where $|Q| = n$.

Consider strings $\epsilon, 0, 00, 000, \cdots, 0^n$ total of $n + 1$ strings.

What is the behavior of M on these strings? Let $q_i = \delta^*(s, 0^i)$.

By pigeon hole principle $q_i = q_j$ for some $0 \leq i < j \leq n$.
M **is in the same state** after reading 0^i and 0^j where $i \neq j$.

M should accept $0^i 1^i$ but then it will also accept $0^j 1^i$ where $i \neq j$.

Proof by Contradiction

- Suppose L is regular. Then there is a DFA M such that $L(M) = L$.
- Let $M = (Q, \{0, 1\}, \delta, s, A)$ where $|Q| = n$.

Consider strings $\epsilon, 0, 00, 000, \cdots, 0^n$ total of $n + 1$ strings.

What is the behavior of M on these strings? Let $q_i = \delta^*(s, 0^i)$.

By pigeon hole principle $q_i = q_j$ for some $0 \leq i < j \leq n$.
M **is in the same state** after reading 0^i and 0^j where $i \neq j$.

M should accept $0^i 1^i$ but then it will also accept $0^j 1^i$ where $i \neq j$.

This contradicts the fact that M accepts L.
Since M was arbitrary, there is no DFA for L.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are suffix distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. In other words either $xw \in L, yw \notin L$ or $xw \notin L, yw \in L$.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are suffix distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. In other words either $xw \in L, yw \notin L$ or $xw \notin L, yw \in L$.

x, y are indistinguishable with respect to L if there is no such w.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are suffix distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. In other words either $xw \in L, yw \notin L$ or $xw \notin L, yw \in L$.

x, y are indistinguishable with respect to L if there is no such w.

Example: If $i \neq j$, 0^i and 0^j are distinguishable with respect to $L = \{0^k 1^k \mid k \geq 0\}$

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are suffix distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. In other words either $xw \in L, yw \notin L$ or $xw \notin L, yw \in L$.

x, y are indistinguishable with respect to L if there is no such w.

Example: If $i \neq j$, 0^i and 0^j are distinguishable with respect to $L = \{0^k 1^k \mid k \geq 0\}$

Example: 000 and 0000 are indistinguishable with respect to the language $L = \{w \mid w \text{ has } 00 \text{ as a substring}\}$

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are suffix distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. In other words either $xw \in L, yw \notin L$ or $xw \notin L, yw \in L$.

x, y are indistinguishable with respect to L if there is no such w.

Example: If $i \neq j$, 0^i and 0^j are distinguishable with respect to $L = \{0^k 1^k \mid k \geq 0\}$

Example: 000 and 0000 are indistinguishable with respect to the language $L = \{w \mid w \text{ has } 00 \text{ as a substring}\}$

Important: (In)distiguishability is with respect to a specific language

Wee Lemma

Lemma

Suppose $L = L(M)$ *for some DFA* $M = (Q, \Sigma, \delta, s, A)$ *and suppose* x, y *are distinguishable with respect to* L*. Then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Wee Lemma

Lemma

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, A)$ and suppose x, y are distinguishable with respect to L. Then $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Since x, y are distinguishable let w be the distinguishing suffix. If $\delta^*(s, x) = \delta^*(s, y)$ then M will either accept both the strings xw, yw, or reject both. But exactly one of them is in L, a contradiction. □

Fooling Sets

Definition

For a language L over Σ a set of strings F (could be infinite) is a fooling set or distinguishing set for L if **every pair** *of distinct strings* $x, y \in F$ are distinguishable.

Fooling Sets

Definition

For a language L over Σ a set of strings F (could be infinite) is a fooling set or distinguishing set for L if **every pair** *of distinct strings* $x, y \in F$ are distinguishable.

Examples:

- $F = \{\varepsilon, 0, 01\}$ is a fooling set for $L = \{w \in \{0,1\}^* \mid w \text{ ends in } 01\}$

Fooling Sets

Definition

For a language L over Σ a set of strings F (could be infinite) is a fooling set or distinguishing set for L if **every pair** *of distinct strings* $x, y \in F$ are distinguishable.

Examples:

- $F = \{\varepsilon, 0, 01\}$ is a fooling set for $L = \{w \in \{0,1\}^* \mid w \text{ ends in } 01\}$
- $F = \{0^i \mid i \geq 0\}$ is a fooling set for the language $L = \{0^k 1^k \mid k \geq 0\}$.

Fooling Sets

Definition

For a language L over Σ a set of strings F (could be infinite) is a fooling set or distinguishing set for L if **every pair** *of distinct strings* $x, y \in F$ are distinguishable.

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Proof of Theorem

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Proof.

Suppose there is a DFA $M = (Q, \Sigma, \delta, s, A)$ that accepts L. Let $|Q| = n$ and $n < |F|$.

Proof of Theorem

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Proof.

Suppose there is a DFA $M = (Q, \Sigma, \delta, s, A)$ that accepts L. Let $|Q| = n$ and $n < |F|$.

By pigeon hole principle there are two strings $x, y \in F$, $x \neq y$ such that $\delta^*(s, x) = \delta^*(s, y)$ but x, y are distinguishable.

Proof of Theorem

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Proof.

Suppose there is a DFA $M = (Q, \Sigma, \delta, s, A)$ that accepts L. Let $|Q| = n$ and $n < |F|$.

By pigeon hole principle there are two strings $x, y \in F$, $x \neq y$ such that $\delta^*(s, x) = \delta^*(s, y)$ but x, y are distinguishable.
Implies that there is w such that exaclty one of xw, yw is in L.
However, M's behaviour on xw and yw is exacly the same and hence M will accept both xw, yw or reject both. A contradiction. □

Infinite Fooling Sets

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Corollary

If L has an infinite fooling set F then L is not regular.

Infinite Fooling Sets

Theorem

Suppose F is a fooling set for L. If F is finite then there is no DFA M that accepts L with less than $|F|$ states.

Corollary

If L has an infinite fooling set F then L is not regular.

Proof.

Suppose for contradiction that $L = L(M)$ for some DFA M with n states.

Any subset F' of F is a fooling set. (Why?) Pick $F' \subseteq F$ arbitrarily such that $|F'| > n$; this is possible since F is infinite. By preceding theorem, we obtain a contradiction. □

Proving Non-regularity

Theorem

A language L is non-regular iff there is an infinite fooling set F for it.

Thus, finding an infinite fooling set is a *fool-proof* technique to prove non-regularity. We saw that if L has an infinite fooling set then it is non-regular. The other direction is not obvious but is a consequence of the Myhill-Nerode theorem.

Another formulation:

Theorem

A language L is non-regular iff the following is true: for very $n > 0$ there is a fooling set F_n with $|F_n| \geq n$.

Examples of non-regular languages

We will prove by describing infinite fooling sets.

- $\{0^k 1^k \mid k \geq 0\}$

Examples of non-regular languages

We will prove by describing infinite fooling sets.

- $\{0^k 1^k \mid k \geq 0\}$
- {bitstrings with equal number of 0s and 1s}

Examples of non-regular languages

We will prove by describing infinite fooling sets.

- $\{0^k 1^k \mid k \geq 0\}$
- {bitstrings with equal number of 0s and 1s}
- $\{0^k 1^\ell \mid k \neq \ell\}$

Examples of non-regular languages

We will prove by describing infinite fooling sets.

- $\{0^k 1^k \mid k \geq 0\}$
- {bitstrings with equal number of 0s and 1s}
- $\{0^k 1^\ell \mid k \neq \ell\}$
- $\{0^{k^2} \mid k \geq 0\}$

Examples of non-regular languages

We will prove by describing infinite fooling sets.

- $\{0^k 1^k \mid k \geq 0\}$
- {bitstrings with equal number of 0s and 1s}
- $\{0^k 1^\ell \mid k \neq \ell\}$
- $\{0^{k^2} \mid k \geq 0\}$
- $\{w \mid w$ is valid C program$\}$
- $\{w \mid w$ is valid regular expression over $\{0,1\}\}$

Exponential gap between NFAs and DFAs

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a } 1 \text{ } k \text{ positions from the end}\}$

Exponential gap between NFAs and DFAs

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a 1 } k \text{ positions from the end}\}$

Recall that L_k is accepted by a NFA N with $k+1$ states.

Exponential gap between NFAs and DFAs

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a } 1 \text{ } k \text{ positions from the end}\}$
Recall that L_k is accepted by a NFA N with $k+1$ states.

Theorem

Every DFA that accepts L_k has at least 2^k states.

Exponential gap between NFAs and DFAs

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a } 1 \text{ } k \text{ positions from the end}\}$
Recall that L_k is accepted by a NFA N with $k+1$ states.

Theorem

Every DFA that accepts L_k has at least 2^k states.

Claim

$F = \{w \in \{0,1\}^* : |w| = k\}$ is a fooling set of size 2^k for L_k.

Why?

Exponential gap between NFAs and DFAs

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a 1 } k \text{ positions from the end}\}$
Recall that L_k is accepted by a NFA N with $k+1$ states.

Theorem

Every DFA that accepts L_k has at least 2^k states.

Claim

$F = \{w \in \{0,1\}^* : |w| = k\}$ is a fooling set of size 2^k for L_k.

Why?

- Suppose $a_1 a_2 \ldots a_k$ and $b_1 b_2 \ldots b_k$ are two distinct bitstrings of length k
- Let i be smallest index where $a_i \neq b_i$.
- $y = 0^{i-1}$ is a distinguishing suffix for the two strings

How to pick a fooling set

- Most important: try to understand why recognizing strings in L requires memory. If you believe L is non-regular it means that any program recognizing L should have increasing memory.
- Play a game. Assume there is a DFA M for L with n states. How would you fool it if you believe L requires increasing memory? What set of strings will force M to make mistakes because it has only n states?
- If x, y are in F and $x \neq y$ they should be distinguishable! Of course. All strings in F except maybe one should be prefixes of strings in the language L.
  For example if $L = \{0^k 1^k \mid k \geq 0\}$ do not pick 1 and 10 (say). Why?
- Instead of picking an infinite fooling set one can also show that for every $n > 0$ there is a fooling set F_n such that $|F_n| \geq n$.

Part I

Non-regularity via closure properties

Non-regularity via closure properties

$L = \{$bitstrings with equal number of 0s and 1s$\}$

$L' = \{0^k 1^k \mid k \geq 0\}$

Suppose we have already shown that L' is non-regular. Can we show that L is non-regular without using the fooling set argument?

Non-regularity via closure properties

$L = \{\text{bitstrings with equal number of 0s and 1s}\}$

$L' = \{0^k 1^k \mid k \geq 0\}$

Suppose we have already shown that L' is non-regular. Can we show that L is non-regular without using the fooling set argument? Yes!

Non-regularity via closure properties

$L = \{\text{bitstrings with equal number of 0s and 1s}\}$

$L' = \{0^k 1^k \mid k \geq 0\}$

Suppose we have already shown that L' is non-regular. Can we show that L is non-regular without using the fooling set argument? Yes!

$L' = L \cap L(0^*1^*)$

Claim: The above and the fact that L' is non-regular implies L is non-regular. Why?

Non-regularity via closure properties

$L = \{\text{bitstrings with equal number of 0s and 1s}\}$

$L' = \{0^k 1^k \mid k \geq 0\}$

Suppose we have already shown that L' is non-regular. Can we show that L is non-regular without using the fooling set argument? Yes!

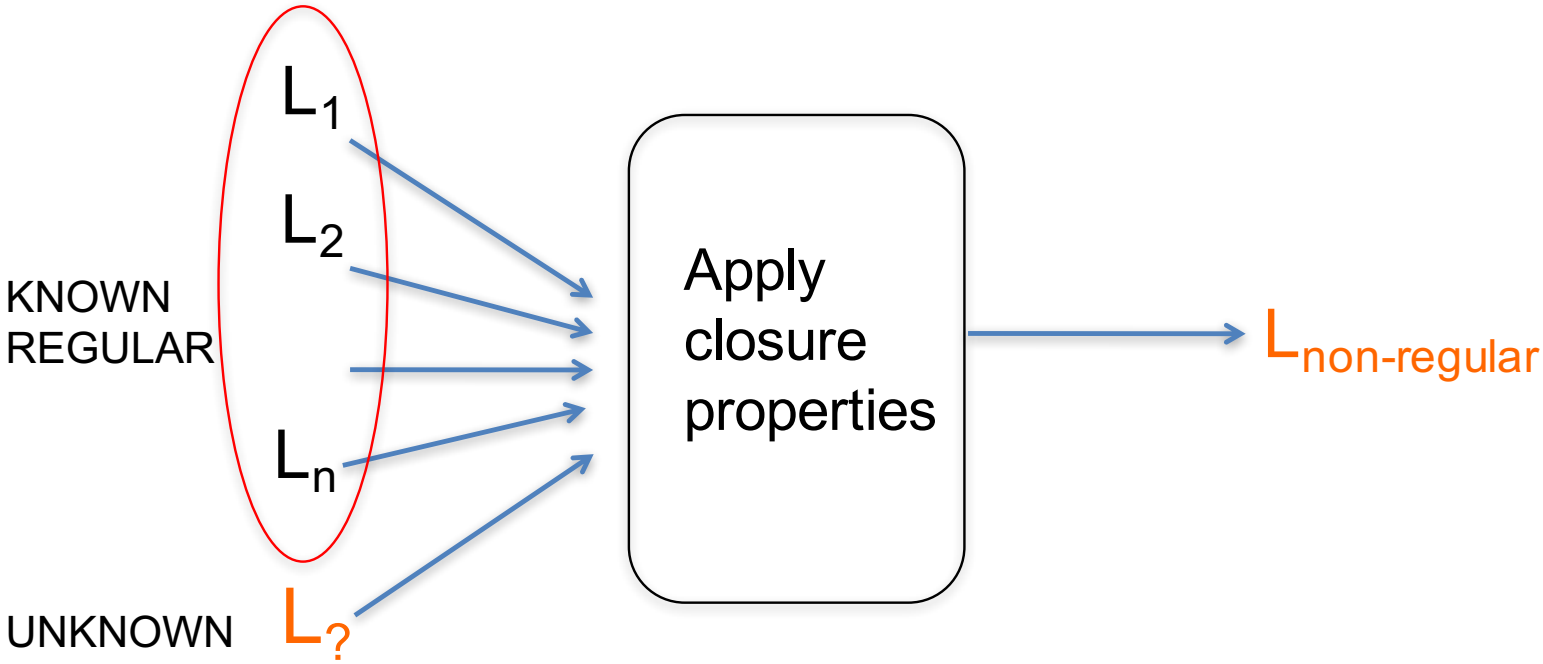
$L' = L \cap L(0^*1^*)$

Claim: The above and the fact that L' is non-regular implies L is non-regular. Why?

Suppose L is regular. Then since $L(0^*1^*)$ is regular, and regular languages are closed under intersection, L' also would be regular. But we know L' is not regular, a contradiction.

Non-regularity via closure properties

General recipe:



Proving non-regularity: Summary

- DFAs have fixed memory. Any language that requires memory that grows with input size is not regular. Not always easy to tell!
- Method of distinguishing suffixes. To prove that L is non-regular find an infinite fooling set.
- Closure properties. Use existing non-regular languages and regular languages to prove that some new language is non-regular.
- Pumping lemma. We did not cover it but it is sometimes an easier proof technique to apply, but not as general as the fooling set technique.

Part II

Myhill-Nerode Theorem (Optional)

Indistinguishability

Recall:

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. x, y are indistinguishable with respect to L if there is no such w.

Given language L over Σ define a relation $\equiv_L$ over strings in Σ^* as follows: $x \equiv_L y$ iff x and y are indistinguishable with respect to L.

Indistinguishability

Recall:

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$ we say that x and y are distinguishable with respect to L if there is a string $w \in \Sigma^*$ such that exactly one of xw, yw is in L. x, y are indistinguishable with respect to L if there is no such w.

Given language L over Σ define a relation $\equiv_L$ over strings in Σ^* as follows: $x \equiv_L y$ iff x and y are indistinguishable with respect to L.

Claim

For any L, $\equiv_L$ is an equivalence relation over Σ^*.

Therefore, $\equiv_L$ partitions Σ^* into a collection of equivalence classes $X_1, X_2, \ldots,$

Claim

$\equiv_L$ is an equivalence relation over Σ^*.

Therefore, $\equiv_L$ partitions Σ^* into a collection of equivalence classes.

Claim

Let x, y be two distinct strings. If x, y belong to the same equivalence class of $\equiv_L$ then x, y are indistinguishable. Otherwise they are distinguishable.

Corollary

If $\equiv_L$ is finite with n equivalence classes then there is a fooling set F of size n for L. If $\equiv_L$ is infinite then there is an infinite fooling set for L.

Myhill-Nerode Theorem

Theorem (Myhill-Nerode)

L is is regular if and only if $\equiv_L$ has a finite number of equivalence classes. If $\equiv_L$ is finite with n equivalence classes then there is a DFA M accepting L with exactly n states and this is the minimum possible.

Corollary

A language L is non-regular if and only if there is an infinite fooling set F for L.

Algorithmic implication: For every DFA M one can find in polynomial time a DFA M' such that $L(M) = L(M')$ and M' has the fewest possible states among all such DFAs.