Proving Non-regularity

Lecture 6

Friday, February 7, 2020

LaTeXed: January 19, 2020 04:14

Regular Languages, DFAs, NFAs

Theorem

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

Regular Languages, DFAs, NFAs

Theorem

Languages accepted by DFA*s,* NFA*s, and regular expressions are the same.*

Question: Is every language a regular language?

Regular Languages, DFAs, NFAs

Theorem

Languages accepted by DFA*s,* NFA*s, 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

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
- Hence number of regular languages is *countably infinite*

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
- Hence number of regular languages is *countably infinite*
- Number of languages is *uncountably infinite*

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
- Hence number of regular languages is *countably infinite*
- Number of languages is *uncountably infinite*
- Hence there must be a non-regular language!

How to prove non-regularity?

Claim: Language L is not regular.

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$: DFA *for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$*: DFA for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Assume for the sake of contradiction that $\delta^*(s, x) = \delta^*(s, y)$.

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$: DFA *for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Assume for the sake of contradiction that $\delta^*(s, x) = \delta^*(s, y)$.
$\implies A \ni \delta^*(s, xw) = \delta^*(\delta^*(s, x), w)$

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$*:* DFA *for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Assume for the sake of contradiction that $\delta^*(s, x) = \delta^*(s, y)$.
$\implies A \ni \delta^*(s, xw) = \delta^*(\delta^*(s, x), w) = \delta^*(\delta^*(s, y), w)$

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$*: DFA for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Assume for the sake of contradiction that $\delta^*(s, x) = \delta^*(s, y)$.
$\implies A \ni \delta^*(s, xw) = \delta^*(\delta^*(s, x), w) = \delta^*(\delta^*(s, y), w)$
$= \delta^*(s, yw) \notin A$

How to prove non-regularity?

Claim: Language L is not regular.
Idea: Show # states in any DFA M for language L has infinite number of states.

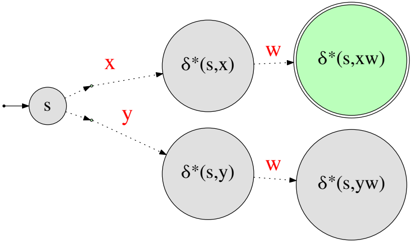
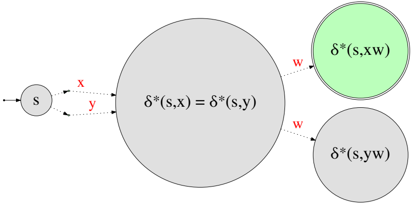
Lemma

Consider three strings $x, y, w \in \Sigma^*$.
$M = (Q, \Sigma, \delta, s, A)$: DFA *for language* $L \subseteq \Sigma^*$.
If $\delta^*(s, xw) \in A$ *and* $\delta^*(s, yw) \notin A$ *then* $\delta^*(s, x) \neq \delta^*(s, y)$.

Proof.

Assume for the sake of contradiction that $\delta^*(s, x) = \delta^*(s, y)$.
$\implies A \ni \delta^*(s, xw) = \delta^*(\delta^*(s, x), w) = \delta^*(\delta^*(s, y), w)$
$= \delta^*(s, yw) \notin A$
$\implies A \ni \delta^*(s, xw) \notin A$. Impossible! □

Proof by figures

Possible	Not possible
	

A Simple and Canonical Non-regular Language

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

A Simple and Canonical Non-regular Language

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

Theorem

L is not regular.

A Simple and Canonical Non-regular Language

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

Theorem

L is not regular.

Question: Proof?

A Simple and Canonical Non-regular Language

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

Theorem

L is not regular.

Question: Proof?

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

A Simple and Canonical Non-regular Language

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

Theorem

L is not regular.

Question: Proof?

Intuition: 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 states does M reach on the above strings? Let $q_i = \delta^*(s, 0^i)$.

By pigeon hole principle $q_i = q_j$ for some $0 \le i < j \le n$.
That is, 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 states does M reach on the above strings? Let $q_i = \delta^*(s, 0^i)$.

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

M should accept 0^i1^i but then it will also accept 0^j1^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 states does M reach on the above strings? Let $q_i = \delta^*(s, 0^i)$.

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

M should accept 0^i1^i but then it will also accept 0^j1^i where $i \neq j$. This contradicts the fact that M accepts L. Thus, there is no DFA for L.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$, 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.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$, 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.

Generalizing the argument

Definition

For a language L over Σ and two strings $x, y \in \Sigma^*$, 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.

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 $x, y \in \Sigma^*$, 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.

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 $L = \{w \mid w \text{ has } 00 \text{ as a substring}\}$

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 two 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 two distinct strings $x, y \in F$ are distinguishable.

Example: $F = \{0^i \mid i \geq 0\}$ 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 two distinct strings $x, y \in F$ are distinguishable.

Example: $F = \{0^i \mid i \geq 0\}$ is a fooling set for the language $L = \{0^k 1^k \mid k \geq 0\}$.

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 $|Q| = n$.

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 $|Q| = n$.
If $n < |F|$ then 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 $|Q| = n$.
If $n < |F|$ then 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 exactly one of xw, yw is in L.
However, M's behavior on xw and yw is exactly 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 $|F'| > n$. By preceding theorem, we obtain a contradiction. □

Examples

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

Examples

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

Examples

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

Examples

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

Examples

- $\{ww^R \mid w \in \Sigma^*\}$

Examples

- $\{ww^R \mid w \in \Sigma^*\}$
- $\{www \mid w \in \Sigma^*\}$

Exponential gap between NFA and DFA size

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

Exponential gap between NFA and DFA size

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

Exponential gap between NFA and DFA size

$L_k = \{w \in \{0,1\}^* \mid w$ has a 1 k 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 NFA and DFA size

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

Exponential gap between NFA and DFA size

$L_k = \{w \in \{0,1\}^* \mid w \text{ has a } 1 \; k \text{ positions from the end}\}$
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 first index where $a_i \neq b_i$
- $y = 0^{k-i-1}$ is a distinguishing suffix for the two strings

How do pick a fooling set

How do we pick a fooling set F?

- 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?

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 from scratch?

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 from scratch?

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

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

Non-regularity via closure properties

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

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

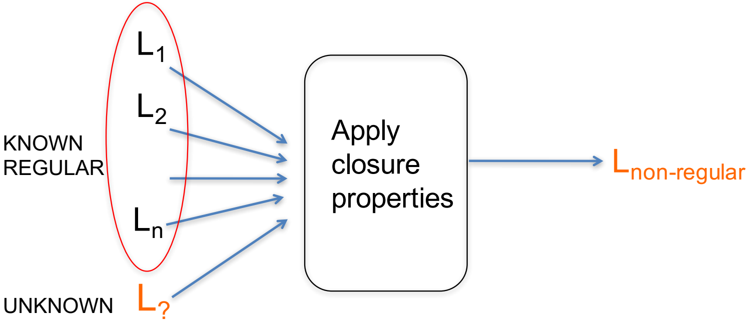
Suppose we have already shown that L' is non-regular. Can we show that L is non-regular without using the fooling set argument from scratch?

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

- 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

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

$\equiv_L$ 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 regular $\iff$ $\equiv_L$ 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.