Proving Non-regularity

Lecture 6
Thursday, September 10, 2020
6.1

Not all languages are regular
Theorem

Languages accepted by DFA\(^s\), NFA\(^s\), 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 \(\Sigma\) 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!
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.

- Each DFA $M$ can be represented as a string over a finite alphabet $\Sigma$ 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!
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 $\Sigma$ 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!
Theorem

Languages accepted by **DFA**s, **NFA**s, 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 $\Sigma$ 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!
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 $\Sigma$ 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!
Theorem

Languages accepted by DFA, NFA, 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 $\Sigma$ 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!
A direct proof

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

Theorem

\[ L \text{ is not regular.} \]
A Simple and Canonical Non-regular Language

\[ L = \{0^k1^k \mid i \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots\} \]

**Theorem**

\[ L \text{ is not regular.} \]

**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?
A Simple and Canonical Non-regular Language

\[ L = \{0^i1^i \mid i \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots\} \]

**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?
A Simple and Canonical Non-regular Language

\[ L = \{0^i 1^i \mid i \geq 0\} = \{\epsilon, 01, 0011, 000111, \ldots, \} \]

**Theorem**

\textit{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?
$L = \{0^k1^k \mid i \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots\}$

**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?
A Simple and Canonical Non-regular Language

$L = \{0^k1^k \mid i \geq 0\} = \{\epsilon, 01, 0011, 000111, \cdots\}$

**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 $|Q| = n$.

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 $i \neq j$. This contradicts the fact that $M$ accepts $L$. Thus, there is no DFA for $L$. 
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 $0^j1^j$ where $i \neq j$.
This contradicts the fact that $M$ accepts $L$. Thus, there is no DFA for $L$. 
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 \( i \neq j \).
This contradicts the fact that \( M \) accepts \( L \). Thus, there is no DFA for \( L \).
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 $i \neq j$.
This contradicts the fact that $M$ accepts $L$. Thus, there is no DFA for $L$. 
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 \( \epsilon, 0, 00, 000, \ldots, 0^n \) 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 \( 0^i 1^i \) but then it will also accept \( 0^j 1^i \) where \( i \neq j \).
This contradicts the fact that \( M \) accepts \( L \). Thus, there is no DFA for \( L \).
THE END

... 

(for now)