Languages Not Regular

Lin Chen

Email: Lin.Chen@ttu.edu

Grader: zulfi.khan@ttu.edu

TEXAS TECH UNIVERSITY

Non-regular languages

- Can you create a DFA/NFA that accepts the following:
  $\mathrm{L} = \{0^n 1^n : n > 0\}$

Non-regular languages

- Can you create a DFA/NFA that accepts the following:

  $\mathrm{L} = \{0^n 1^n : n > 0\}$

  Oblivious!

  - a DFA/NFA for $\mathrm{L}' = \{(01)^n : n > 0\}$ is easy, but $L \cdots$

  - we need to keep track of the total number of $\mathbf{0}$, then check whether there is the same number of $\mathbf{1}$s

Non-regular languages

- Can you create a DFA/NFA that accepts the following:

  $\mathrm{L} = \{0^n 1^n : n > 0\}$

  - a DFA/NFA for $\mathrm{L}' = \{(01)^n : n > 0\}$ is easy, but $L\cdots$

  - we need to keep track of the total number of $\mathbf{0}$, then check whether there is the same number of $\mathbf{1}$s

  - The information (number) has to be stored using states

  - Finite states and infinite different numbers⋯

Non-regular languages

- Can you create a DFA/NFA that accepts the following:

  $\mathrm{L} = \{0^n 1^n : n > 0\}$

  - a DFA/NFA for $\mathrm{L}' = \{(01)^n : n > 0\}$ is easy, but $L\cdots$

  - we need to keep track of the total number of $\mathbf{0}$, then check whether there is the same number of $\mathbf{1}$s

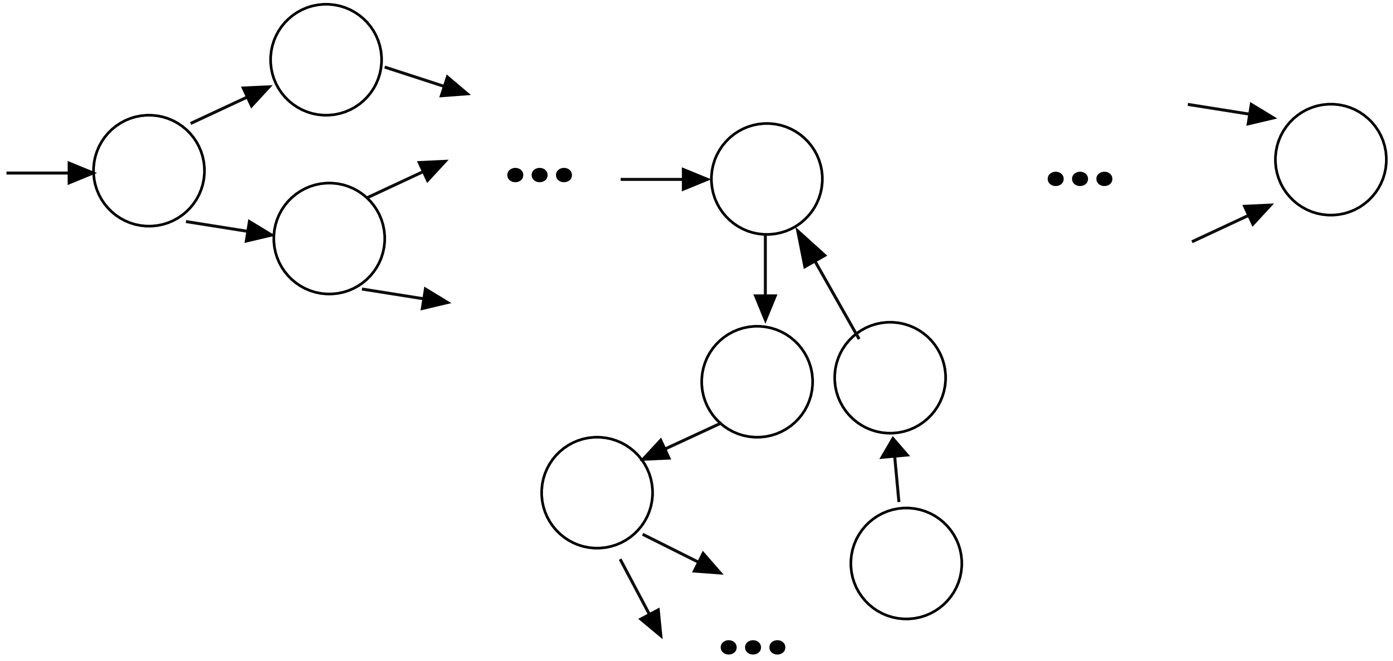
  - The information (number) has to be stored using states

  - Finite states and infinite different numbers⋯

- Probably the answer is "no", but how can we show it?

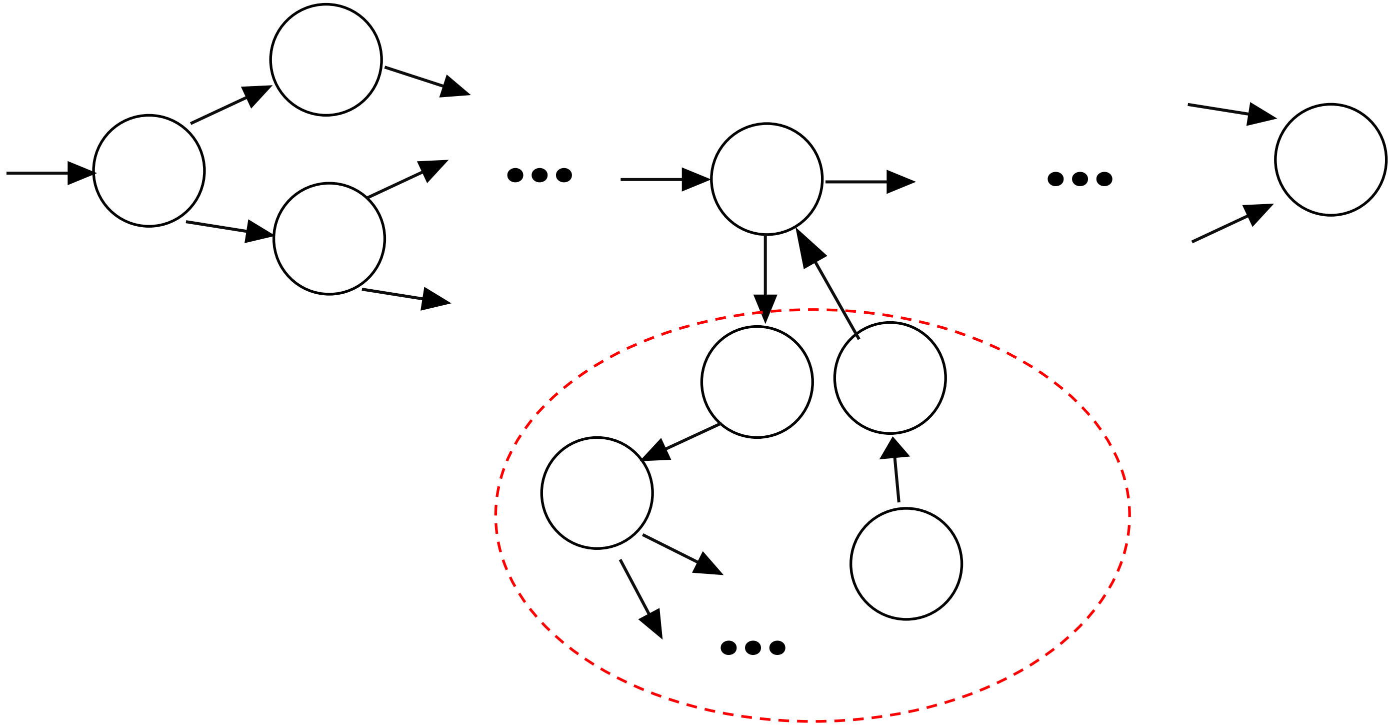
Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$



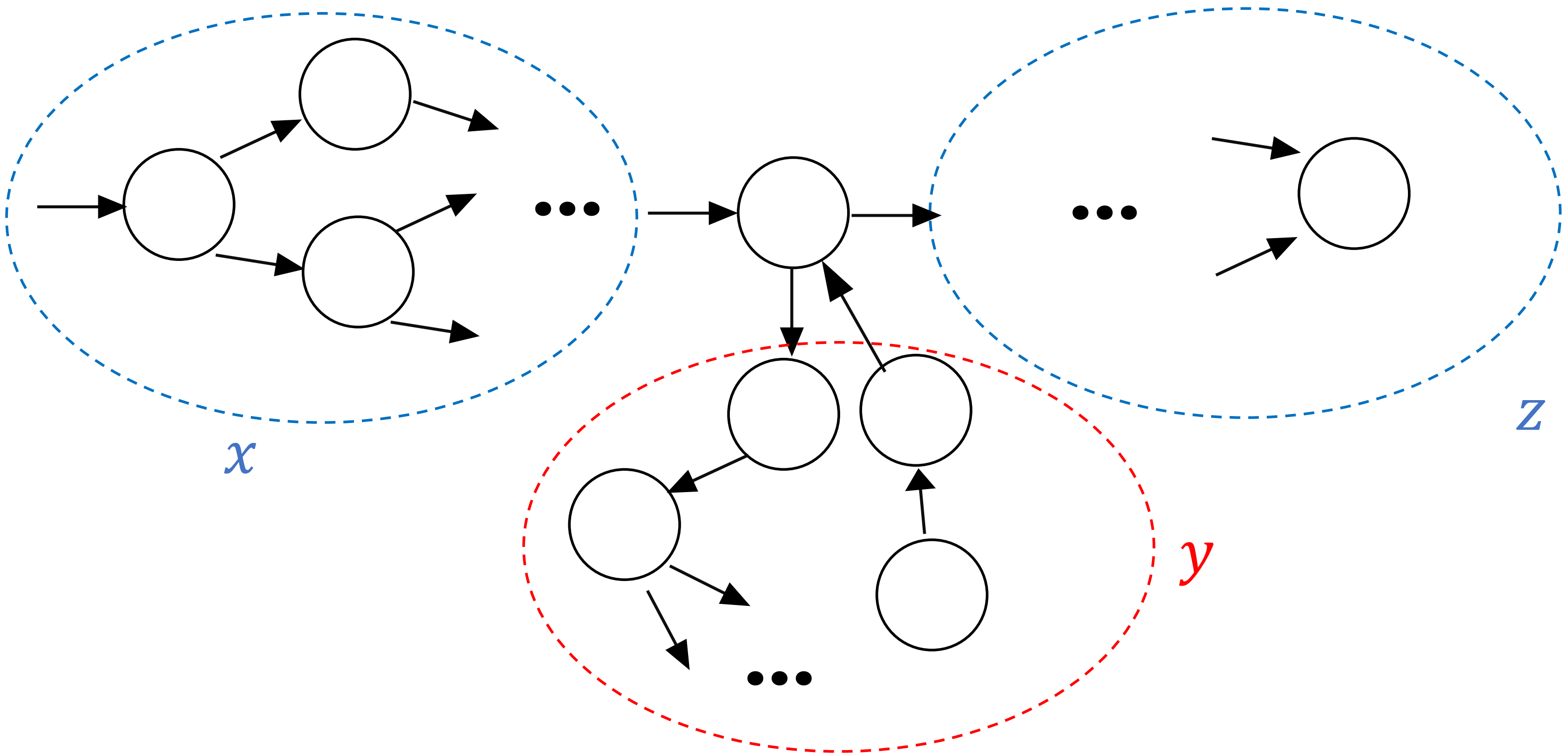
Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- The machine only has finite states, so if n is sufficiently large, there must be some loop (pigeonhole principle).



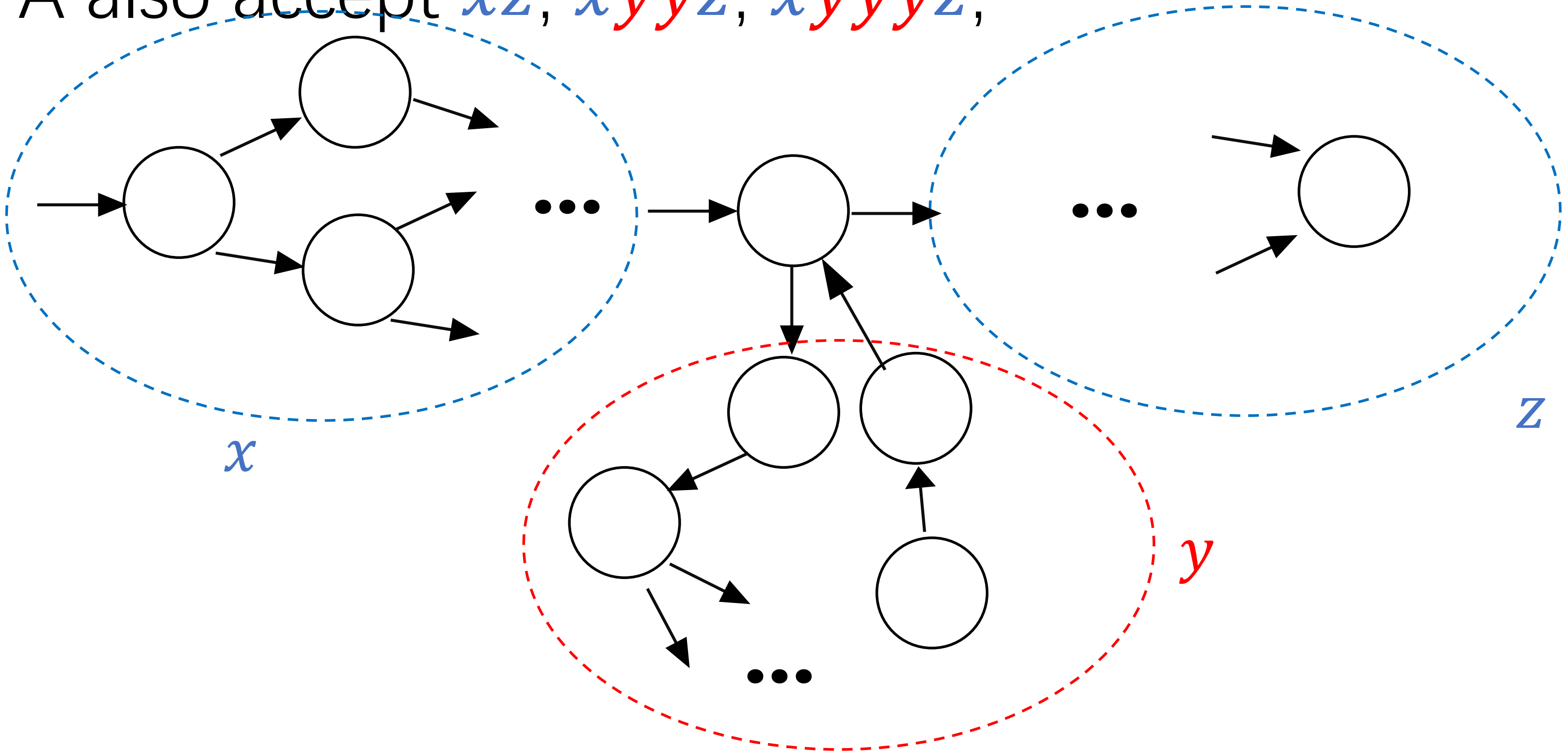
Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$



Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$,⋯



Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$, $\cdots$
- What can be the possible y?

Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$, $\cdots$
- What can be the possible y?
  - $y = 000 \cdots 0$
  - $y = 111 \cdots 1$
  - $y = 000 \cdots 0111 \cdots 1$

Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$, $\cdots$
- What can be the possible y?
  - $y = 000 \cdots 0$
  - $y = 111 \cdots 1$
  - $y = 000 \cdots 0111 \cdots 1$

Impossible, because then xz will have different numbers of 0 and 1

Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$, $\cdots$
- What can be the possible y?
  - $y = 000 \cdots 0$
  - $y = 111 \cdots 1$
  - $y = 000 \cdots 0111 \cdots 1$

Impossible, because then $xyyz$ will have 10 as substring

Non-regular languages

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- For sufficiently large n, $0^n 1^n = xyz$
- The DFA also accept xz, $xyyz$, $xyyyz$,⋯
- What can be the possible y?
  - $y = 000 \cdots 0$
  - $y = 111 \cdots 1$
  - $y = 000 \cdots 0111 \cdots 1$
- Impossible to have such a DFA, $\mathrm{L} = \{0^n 1^n : n > 0\}$ is non-regular

Pumping lemma

- If a language L is regular, then there exists some $n \geq 1$ such that
  - for any string $w \in L$, $|w| \geq n$, we have $w = xyz$ such that
    - $y \neq e$
    - $|xy| < n$
    - $xy^iz \in L$ for any $i \geq 0$

Pumping lemma

- If a language L is regular, then there exists some $n \geq 1$ such that
  - for any string $w \in L$, $|w| \geq n$, we have $w = xyz$ such that
    - $y \neq e$
    - $|xy| < n$
    - $xy^i z \in L$ for any $i \geq 0$

Repeating started once the length exceeds the total number of states

Pumping lemma

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- Exists some n, such that if $2m \geq n$, then $0^m 1^m = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^i z \in L$ for any $i \geq 0$

Pumping lemma

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- Exists some n, such that if $2m \geq n$, then $0^m 1^m = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^i z \in L$ for any $i \geq 0$
- Choose $m = n$
- $y = 00 \cdots 0$

Pumping lemma

- Suppose you have a magic DFA to accept $\mathrm{L} = \{0^n 1^n : n > 0\}$
- Exists some n, such that if $2m \geq n$, then $0^m 1^m = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^i z \in L$ for any $i \geq 0$
- Choose $m = n$
- $y = 00 \cdots 0$
- The DFA should also accept xz, contradiction.

Pumping lemma

- Example: prove that $L = \{ww \colon w \in (a+b)^*\}$ is not regular

Pumping lemma

- Suppose $L = \{ww: w \in (a+b)^*\}$ is regular
- Exists some n, such that if $v \in L$, $|v| \geq n$, then $v = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^iz \in L$ for any $i \geq 0$

Pumping lemma

- Suppose $L = \{ww : w \in (a+b)^*\}$ is regular
- Exists some n, such that if $v \in L$, $|v| \geq n$, then $v = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^iz \in L$ for any $i \geq 0$
- Choose $v = a^nba^nb$
  - $x = a^i$, $y = a^k$ for some i, k
  - $xy^2z = a^{n+k}ba^nb \notin L$

The way to prove- apply Pumping lemma

- You have an adversary who thinks L is regular. You need to prove that your adversary is wrong.
  - you: Language L is not regular!
  - adv: Yes it is! I have a DFA to prove it!
  - you: Oh really? How many states are in your DFA?
  - adv: n
  - you: OK, here's a string $w \in L$ with $|w| > n$. Your machine must accept w – but since $|w| > n$, there must be a loop in your computation. Where's the loop?
  - adv: Right here! (breaks w into xyz, where y is the part of the string that goes through the loop)
  - you: Ah hah! If we go through the loop 2 times instead of 1, we get a string not in L that your machine will accept!
  - adv: ⋯

The way to prove- apply Pumping lemma

- Your adversary picks an n
  - because you do not know this n
- You pick a $w \in L$ (such that $|w| > n$)
- Your adversary breaks w into xyz (subject to $|xy| \leq n, |y| > 0$)
  - because you do not know how exactly it is split
- You pick an i such that $xy^iz \notin L$

The way to prove- apply Pumping lemma

- Example: prove that $L = \{w: w = a^p, p \textbf{ is prime}\}$ is not regular

The way to prove- apply Pumping lemma

- Suppose $L = \{w: w = a^p, p \text{ is prime}\}$ is regular
- Exists some n, such that if $w \in L$, $|w| \geq n$, then $w = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^iz \in L$ for any $i \geq 0$

The way to prove- apply Pumping lemma

- Suppose $L = \{w: w = a^p, p \textbf{ is prime}\}$ is regular
- Exists some n, such that if $w \in L$, $|w| \geq n$, then $w = xyz$ such that
  - $y \neq e$
  - $|xy| < n$
  - $xy^iz \in L$ for any $i \geq 0$
- Pick $|w| = n + 2$
  - $x = a^\alpha, y = a^\beta, z = a^\gamma$ such that
    - $\beta \geq 1$, $\alpha + \beta < n$, $\alpha + i\beta + \gamma$ is always prime for any $i \geq 0$
    - choose $i = \alpha + \gamma$, then $\alpha + i\beta + \gamma = (\alpha + \gamma)(\beta + 1)$

The way to prove- apply Pumping lemma

- What happens if we try to apply Pumping lemma to a regular language

The way to prove- apply Pumping lemma

- What happens if we try to apply Pumping lemma to a regular language
- Consider $L = \{w : w = a^p, p \text{ is even}\}$
  - Exists some n, such that if $w \in L$, $|w| \geq n$, then $w = xyz$ such that
    - $y \neq e$
    - $|xy| < n$
    - $xy^iz \in L$ for any $i \geq 0$
  - Choose some $w = a^{2k}$ for $2k \geq n$
  - $x = a^\alpha, y = a^\beta, z = a^\gamma$ such that
    - $\beta \geq 1$, $\alpha + \beta + \gamma = 2k$, $\alpha + i\beta + \gamma$ is always even for any $i \geq 0$

The way to prove- apply Pumping lemma

- What happens if we try to apply Pumping lemma to a regular language
- Consider $L = \{w: w = a^p, p \text{ is even}\}$
  - Exists some n, such that if $w \in L$, $|w| \geq n$, then $w = xyz$ such that
    - $y \neq e$
    - $|xy| < n$
    - $xy^iz \in L$ for any $i \geq 0$
  - Choose some $w = a^{2k}$ for $2k \geq n$
  - $x = a^\alpha, y = a^\beta, z = a^\gamma$ such that
    - $\beta \geq 1$, $\alpha + \beta + \gamma = 2k$, $\alpha + i\beta + \gamma$ is always even for any $i \geq 0$

This can be correct if β is even – we have no control on the values of α, β, γ, so no contradiction

The way to prove- apply Pumping lemma

- Not able to apply pumping lemma to show a language is non-regular does not necessarily mean this language is regular

  - To show it is regular, create a regular expression or DFA/NFA

The way to prove- apply Pumping lemma

- Not able to apply pumping lemma to show a language is non-regular does not necessarily mean this language is regular

  - To show it is regular, create a regular expression or DFA/NFA

- Not able to create a DFA/NFA for a language does not necessarily mean this language is non-regular

  - To show it is non-regular, apply pumping lemma suitably

Closure Properties

- Besides creating DFA/NFA or using pumping lemma, is there any other way to assert a language is regular or non-regular

Closure Properties

- Besides creating DFA/NFA or using pumping lemma, is there any other way to assert a language is regular or non-regular
- We may utilize the closure properties
  - Is L_{REG} closed under union?

    $L1 = L[r1], L2 = L[r2]$, then $L1 \cup L2 = L[(r1 + r2)]$

Closure Properties

- Besides creating DFA/NFA or using pumping lemma, is there any other way to assert a language is regular or non-regular
- We may utilize the closure properties
  - Is L_{REG} closed under union?

    $L1 = L[r1], L2 = L[r2]$, then $L1 \cup L2 = L[(r1 + r2)]$
  - Is L_{REG} closed under complementation?

    Given a DFA $M = (K, \Sigma, \delta, s, F)$ for L_{REG}, we create $M' = (K, \Sigma, \delta, s, K - F)$

Closure Properties

- Besides creating DFA/NFA or using pumping lemma, is there any other way to assert a language is regular or non-regular
- We may utilize the closure properties
  - Is L_{REG} closed under union?

    $L1 = L[r1], L2 = L[r2]$, then $L1 \cup L2 = L[(r1 + r2)]$
  - Is L_{REG} closed under complementation?

    Given a DFA $M = (K, \Sigma, \delta, s, F)$ for L_{REG}, we create $M' = (K, \Sigma, \delta, s, K - F)$
  - Is L_{REG} closed under intersection?

    $A \cap B = \overline{\bar{A} \cup \bar{B}}$

Closure Properties

- Prove that $L = \{w \in \{a, b\}^*: w \text{ has an equal number of } a \text{ and } b\}$

Closure Properties

- Prove that $L = \{w \in \{a, b\}^* : w \text{ has an equal number of } a \text{ and } b\}$
  - $a^* b^*$ is regular
  - $L \cap a^* b^* = \{a^n b^n : n \geq 0\}$

Closure Properties

- Prove that $L = \{w \in \{a, b\}^* : w \text{ has an equal number of } a \text{ and } b\}$
  - a^*b^* is regular
  - $L \cap a^*b^* = \{a^n b^n : n \geq 0\}$
  - we have shown $\{a^n b^n : n \geq 0\}$ is not regular
  - if L is regular, then by closure property $L \cap a^*b^*$ is regular
  - so L is not regular