COL 352 Introduction to Automata and Theory of Computation

Nikhil Balaji

Bharti 420
Indian Institute of Technology, Delhi
nbalaji@cse.iitd.ac.in

January 29, 2023

Lecture 8: Regular Expressions

Recap

Definition (Pattern)

A pattern α is a string of symbols of a certain form representing a (possibly infinite) set of strings in Σ^*.

$$L(\alpha) = \{x \in \Sigma^* \mid x \text{ matches } \alpha\}$$

Recap: Atomic and Compound Patterns

1. $a \in \Sigma$, $L(a) = \{a\}$
2. ε, $L(\varepsilon) = \{\varepsilon\}$
3. $\varnothing$, $L(\varnothing) = \varnothing$
4. Σ, matching any alphabet
5. Σ^*, matching any finite string
6. x matches $\alpha + \beta$ if $L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
7. x matches $\alpha \cap \beta$ if $L(\alpha \cap \beta) = L(\alpha) \cap L(\beta)$
8. x matches $\alpha\beta$ if $x = yz$ where $L(\alpha\beta) = L(\alpha)L(\beta)$
9. x matches $\overline{\alpha}$ if $L(\overline{\alpha}) = \overline{L(\alpha)} = \Sigma^* \setminus L(\alpha)$
10. x matches α^* if x can be expressed as zero or more of strings that match α, i.e., $L(\alpha^*) = L(\alpha)^*$
11. x matches α^+ if x can be expressed as one or more of strings that match α, i.e., $L(\alpha^+) = L(\alpha)^+$

Recap: DFA to regular expression

Lemma

Any regular language can be specified by a regular expression

Recap: DFA to regular expression

Lemma

Any regular language can be specified by a regular expression

Want: Given any DFA, convert it into a regular expression.

Lemma

Given any DFA A, we can obtain a regular expression, say R_A, such that $L(A) = L(R_A)$.

Recap: Computing with labelled graphs

Lemma

Any regular language can be specified by a regular expression

Want: Given any DFA, convert it into a regular expression.

Lemma

Given any DFA A, we can obtain a regular expression, say R_A, such that $L(A) = L(R_A)$.

Regular expressions

For a regular expression E we write $L(E)$ for its language. The set of valid regular expressions RegEx can be defined recursively as the following:

	Syntax	Semantics
Empty String	ϵ	$L(\epsilon) = \{\epsilon\}$
Empty Set	$\varnothing$	$L(\varnothing) = \varnothing$
Single Letter	a	$L(a) = \{a\}$
Union	$E + F$	$L(E + F) = L(E) \cup L(F)$
Concatenation	$E.F$	$L(E.F) = L(E) \circ L(F)$
Kleene Star	E^*	$L(E)^*$

NFA to regular expressions

NFA to regular expressions

Theorem

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, then there is an RE R such that $L(R) = L(A)$.

NFA to regular expressions

Theorem

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, then there is an RE R such that $L(R) = L(A)$.

NFA to regular expressions

Theorem

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, then there is an RE R such that $L(R) = L(A)$.

Proof.

Let us assign states in $Q = \{q_1, \ldots, q_n\}$ an arbitrary order, where $q_0 = q_1$ and $q_1 < \cdots < q_n$.

NFA to regular expressions

Theorem

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, then there is an RE R such that $L(R) = L(A)$.

Proof.

Let us assign states in $Q = \{q_1, \dots, q_n\}$ an arbitrary order, where $q_0 = q_1$ and $q_1 < \cdots < q_n$.

Each path on a DFA corresponds to a word.

NFA to regular expressions

Theorem

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, then there is an RE R such that $L(R) = L(A)$.

Proof.

Let us assign states in $Q = \{q_1, \ldots, q_n\}$ an arbitrary order, where $q_0 = q_1$ and $q_1 < \cdots < q_n$.

Each path on a DFA corresponds to a word.

We will incrementally consider longer and longer paths.

NFA to regular expressions (contd)

Proof. (contd.)

NFA to regular expressions (contd)

Proof. (contd.) Let us define the following set of paths.

NFA to regular expressions (contd)

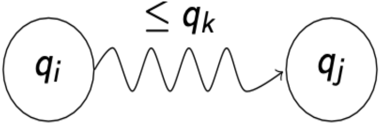
Proof. (contd.) Let us define the following set of paths.

$p(i, j, k)$:= the set of paths from q_i to q_j that do not have intermediate states that are greater than q_k.

NFA to regular expressions (contd)

Proof. (contd.) Let us define the following set of paths.

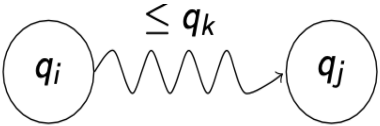
$p(i, j, k) :=$ the set of paths from q_i to q_j that do not have intermediate states that are greater than q_k.



NFA to regular expressions (contd)

Proof. (contd.) Let us define the following set of paths.

$p(i, j, k)$:= the set of paths from q_i to q_j that do not have intermediate states that are greater than q_k.

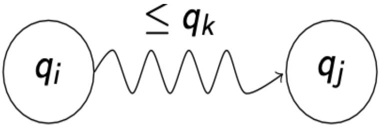


Note that q_i and q_j need not be smaller than q_k.

NFA to regular expressions (contd)

Proof. (contd.) Let us define the following set of paths.

$p(i,j,k) :=$ the set of paths from q_i to q_j that do not have intermediate states that are greater than q_k.



Note that q_i and q_j need not be smaller than q_k.

Let $R(i,j,k)$ be the regular expression that defines the set of words along the paths in $p(i,j,k)$.

NFA to regular expressions (contd)

Proof. (contd.)

NFA to regular expressions (contd)

Proof. (contd.)

Let us define $R(i, j, k)$ by induction over k.

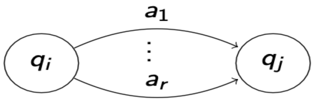
NFA to regular expressions (contd)

Proof. (contd.)

Let us define $R(i,j,k)$ by induction over k.

Base case: Let $\{a_1,\ldots,a_r\} = \{a \mid \delta(q_i,a) = q_j\}$, i. e., letters that take q_i to q_j.

$$R(i,j,0) := \begin{cases} a_1 + \cdots + a_r, i \neq j \\ a_1 + \cdots + a_r + \varepsilon \quad \text{Otherwise} \end{cases}$$



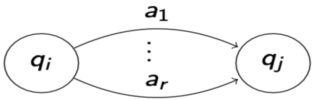
NFA to regular expressions (contd)

Proof. (contd.)

Let us define $R(i, j, k)$ by induction over k.

Base case: Let $\{a_1, \ldots, a_r\} = \{a \mid \delta(q_i, a) = q_j\}$, i. e., letters that take q_i to q_j.

$$R(i, j, 0) := \begin{cases} a_1 + \cdots + a_r, i \neq j \\ a_1 + \cdots + a_r + \varepsilon \quad \text{Otherwise} \end{cases}$$

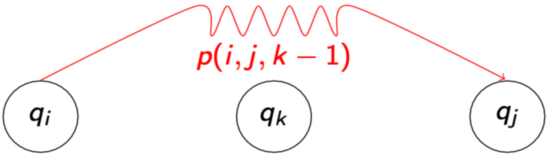


NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.

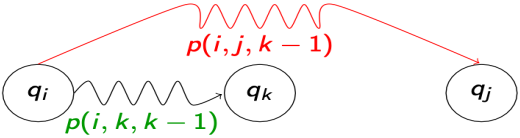
NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.



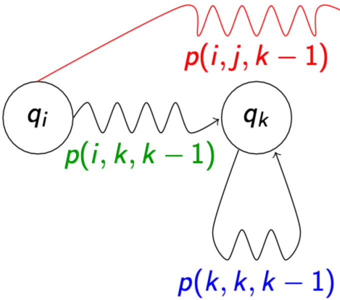
NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.



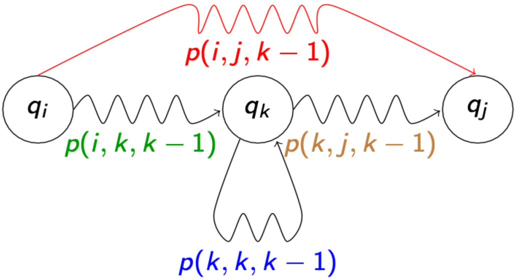
NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.



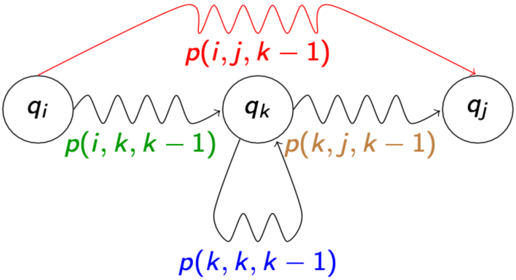
NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.



NFA to regular expressions (contd)

Induction step: By induction hypothesis, we have regular expressions for the paths upto $k-1$. Let us consider the paths that also go via state q_k.



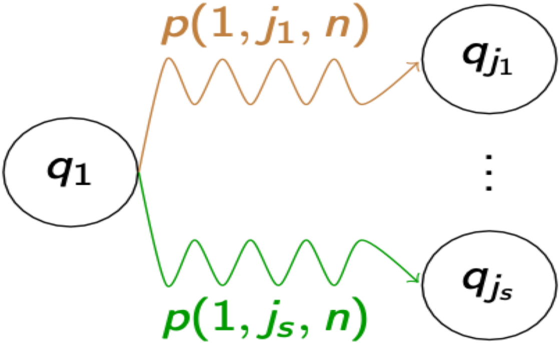
$$R(i, j, k) := R(i, j, k-1) + R(i, k, k-1)R(k, k, k-1)^* R(k, j, k-1)$$

NFA to regular expressions (contd)

Let $F = \{q_{j1}, \ldots, q_{js}\}$

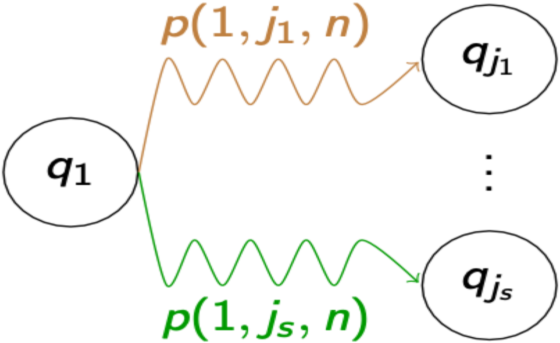
NFA to regular expressions (contd)

Let $F = \{q_{j1}, \ldots, q_{js}\}$



NFA to regular expressions (contd)

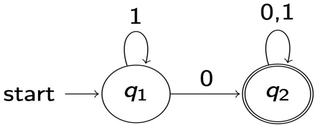
Let $F = \{q_{j1}, \ldots, q_{js}\}$



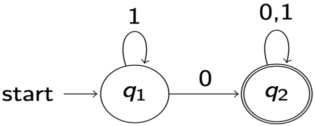
The following regular expression will recognize $L(A)$

$$R(1, j_1, n) + \cdots + R(1, j_s, n)$$

Examples



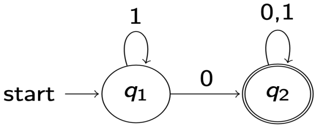
Examples



Base Cases:

$$R(1,1,0) = \varepsilon + 1$$

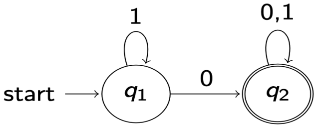
Examples



Base Cases:

$$R(1,1,0) = \varepsilon + 1$$
$$R(2,2,0) = \varepsilon + 0 + 1$$

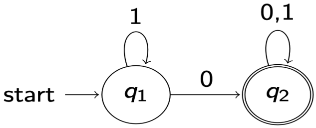
Examples



Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0
\end{aligned}
$$

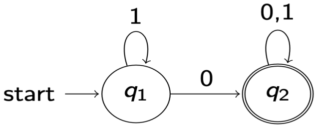
Examples



Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

Examples

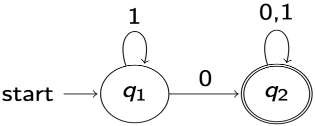


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$R(1,2,1) = R(1,2,0) +$$

Examples

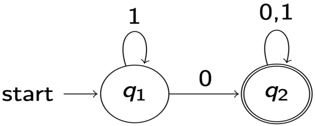


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$R(1,2,1) = R(1,2,0) + R(1,1,0)R(1,1,0)^* R(1,2,0)$$

Examples

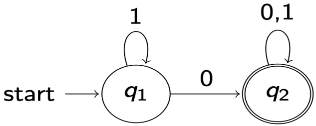


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^* R(1,2,0) \\
&= 0 +
\end{aligned}
$$

Examples

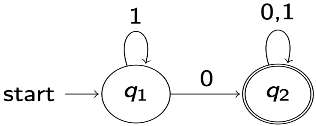


Base Cases:

$$\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}$$

$$\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^{*}R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^{*}0
\end{aligned}$$

Examples

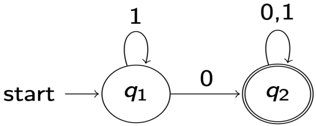


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^* R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^* 0 = 1 * 0
\end{aligned}
$$

Examples

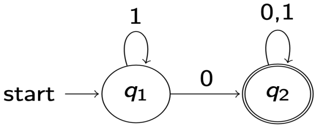


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^* R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^* 0 = 1 * 0 \\
R(2,2,1) &= R(2,2,0) + R(2,1,0)R(1,1,0)^* R(1,2,0)
\end{aligned}
$$

Examples

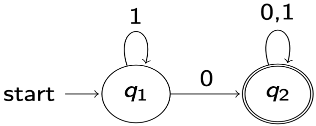


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^* R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^* 0 = 1 * 0 \\
R(2,2,1) &= R(2,2,0) + R(2,1,0)R(1,1,0)^* R(1,2,0) \\
&= \varepsilon + 0 + 1 +
\end{aligned}
$$

Examples

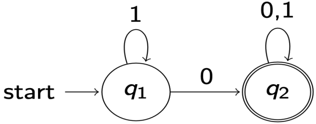


Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^*R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^*0 = 1 * 0 \\
R(2,2,1) &= R(2,2,0) + R(2,1,0)R(1,1,0)^*R(1,2,0) \\
&= \varepsilon + 0 + 1 + \varnothing(\varepsilon + 1)^*0
\end{aligned}
$$

Examples



Base Cases:

$$
\begin{aligned}
R(1,1,0) &= \varepsilon + 1 \\
R(2,2,0) &= \varepsilon + 0 + 1 \\
R(1,2,0) &= 0 \\
R(2,1,0) &= \varnothing
\end{aligned}
$$

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^*R(1,2,0) \\
&= 0 + (\varepsilon + 1)(\varepsilon + 1)^*0 = 1 \ast 0 \\
R(2,2,1) &= R(2,2,0) + R(2,1,0)R(1,1,0)^*R(1,2,0) \\
&= \varepsilon + 0 + 1 + \varnothing(\varepsilon + 1)^*0 = \varepsilon + 0 + 1
\end{aligned}
$$

Examples (Contd.)

$$
\begin{aligned}
L(A) &= R(1,2,2) \\
&= R(1,2,1) + R(1,2,1)R(2,2,1)^* R(2,2,1) \\
&= 1^*0 + 1^*0(\varepsilon + 0 + 1)^*(\varepsilon + 0 + 1) \\
&= 1^*0(0+1)^*
\end{aligned}
$$

Limitations of Finite Automata

Limitations of Finite Automata

$$L_{0,1} = \{0^n 1^n \mid n \geq 0\}$$

00000000000000000000000000000111111111111111111111111111111

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$.

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$. By Pigeon Hole Principle, $\exists i, j \in \mathbb{N}$ such that $i \neq j$

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$. By Pigeon Hole Principle, $\exists i, j \in \mathbb{N}$ such that $i \neq j$, automaton reaches the same state after reading both $0^i, 0^j$.

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$. By Pigeon Hole Principle, $\exists i, j \in \mathbb{N}$ such that $i \neq j$, automaton reaches the same state after reading both $0^i, 0^j$.
- Then $0^i \cdot 1^j$ and $0^j \cdot 1^j$ are both accepted or both rejected

Proving that $L_{0,1}$ is not a regular language

Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$. By Pigeon Hole Principle, $\exists i, j \in \mathbb{N}$ such that $i \neq j$, automaton reaches the same state after reading both $0^i, 0^j$.
- Then $0^i \cdot 1^j$ and $0^j \cdot 1^j$ are both accepted or both rejected, which is a contradiction.

Proving that $L_{0,1}$ is not a regular language

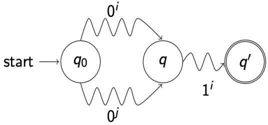
Lemma

There is no finite state automaton accepting $L_{0,1}$

Proof.

- Suppose there was a DFA accepting $L_{0,1}$. By Pigeon Hole Principle, $\exists i, j \in \mathbb{N}$ such that $i \neq j$, automaton reaches the same state after reading both $0^i, 0^j$.
- Then $0^i \cdot 1^j$ and $0^j \cdot 1^j$ are both accepted or both rejected, which is a contradiction.

$\square$



Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Suppose $\exists x, y \in \Sigma^n$ such that $x \neq y$

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Suppose $\exists x, y \in \Sigma^n$ such that $x \neq y$,

automaton reaches the same state after reading both x, y.

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Suppose $\exists x, y \in \Sigma^n$ such that $x \neq y$,

automaton reaches the same state after reading both x, y.

Then $x \cdot x^R$ and $y \cdot x^R$ are both accepted or both rejected

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Suppose $\exists x, y \in \Sigma^n$ such that $x \neq y$,

automaton reaches the same state after reading both x, y.

Then $x \cdot x^R$ and $y \cdot x^R$ are both accepted or both rejected,

which is a contradiction.

Proving that PAL is not a regular language

Lemma

$\forall n \in \mathbb{N}$ let $PAL_n = \{w \cdot w^R \mid w \in \Sigma^*, |w| = n\}$. Any automaton accepting PAL_n must have $|\Sigma|^n$ states.

Proof.

By Pigeon Hole Principle.

Suppose $\exists x, y \in \Sigma^n$ such that $x \neq y$,

automaton reaches the same state after reading both x, y.

Then $x \cdot x^R$ and $y \cdot x^R$ are both accepted or both rejected,

which is a contradiction.

□

Corollary

Let $PAL = \cup_{n \geq 0} PAL_n$. PAL is not regular.