CS310 Automata Theory – 2016-2017

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Lecture 7: Finite state automata
January 16, 2017

Last class

Regular expressions

The language defined by any regular expression is regular.

For any regular language L there is a regular expression, say R, such that $L(R)$ is L.

Regular expressions

Various expressions formed by $\cup, \circ, *$ *operators on* Σ.

Definition (Regular expression)

The following are regular expressions:

1. ϵ,
2. $a, \forall a \in \Sigma$,
3. $\varnothing$,
4. $R_1 \cup R_2$,
5. $R_1 \circ R_2$,
6. R_1^*,

where, R_1, R_2 are regular expressions.

Example

$$\Sigma^* a \Sigma^* = \{w \mid w \text{ contains at least one } a\}$$

$$(\Sigma\Sigma)^* = w \mid |w| \equiv 0 (mod 2)$$

Language defined by a regular expression

Definition (Language defined by regular expression)

The language defined by a regular expression is:

1. $L(\epsilon) = \epsilon$,
2. $L(a) = \{a\}, \forall a \in \Sigma$,
3. $L(\varnothing) = \varnothing$,
4. $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
5. $L(R_1 \circ R_2) = L(R_1) \circ L(R_2)$,
6. $L(R_1^*) = (L(R_1))^*$,

where, R_1, R_2 are regular expressions.

Lemma

The language defined by any regular expression is regular.

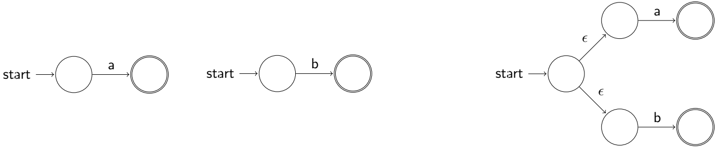
Language defined by regular expression

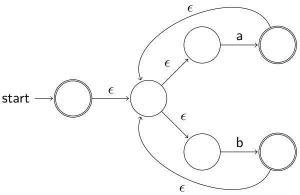
Lemma

The language defined by any regular expression is regular.

Example

$$(a \cup b)^*$$





Language defined by regular expression

Lemma

The language defined by any regular expression is regular.

Proof idea

It is easy to construct NFAs for 1.,2.,3.

If we inductively have NFAs for $L(R_1), L(R_2)$ then we can create an NFA for $L(R_1 \cup R_2)$ and $L(R_1 \circ R_2)$.

Similarly, if we inductively have NFAs for $L(R_1)$ then we can create an NFA for $(L(R_1))^*$

NFA to regular expression

Lemma

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

Examples in class

Limitations of NFA

Lemma

The number of regular languages is countable.

Proof.

By counting.

Every regular language is recognized by a DFA.

Every DFA has a finite description.

All DFAs can therefore be enumerated, i.e. there is a one-to-one mapping (bijection) from all DFAs to $\mathbb{N}$.

$\square$

This implies that there exist languages which are not accepted by any DFA.

Limitations of NFA

What are examples of languages not accepted by NFAs?

$$PAL = \{w \cdot w^R \mid w \in \Sigma^*\}.$$

$$EQ = \{w \cdot w \mid w \in \Sigma^*\}.$$

$$L_{a,b} = \{a^n \cdot b^n \mid n \geq 0\}.$$

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.

□