CS310 : Automata Theory 2019

Lecture 9: Properties of regular languages II

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-01-22

More Closure properties of regular languages

We have already seen a few closure properties of regular languages.

- Union
- Intersection
- Complementation
- Difference

We will see a few more such properties.

- Closure
- Concatenation
- Reversal
- Homomoprphism
- Inverse homomoprphism

Exercises : closure properties

Exercise 9.1

Is a subset of a regular language also regular?

Exercise 9.2

Is a superset of a regular language also regular?

Exercise 9.3

If L is a regular language, is L^* also regular?

Exercise 9.4

If L and M are regular languages, is LM also regular?

Reversal

For word w, let w^R be the reverse word of w.

Let $L^R \triangleq \{w^R \mid w \in L\}$.

Theorem 9.1

If L is a regular language. L^R is also regular.

Proof.

We use regular expressions(REs) to prove this theorem.

We inductively show that for each RE E there is an RE recognizing $L(E)^R$.

base case:

All three base cases ϵ, $\emptyset$, and a are the reverse language of themselves. ...

Reversal

Proof (contd.).

induction step:
Due to induction hypothesis, we assume that E_1^R and E_2^R are the REs that recognize $L(E_1)^R$ and $L(E_2)^R$ respectively.

- $E = E_1 E_2$

  claim: $E^R = E_2^R E_1^R$
  Let $w_1 w_2 \in E$ such that $w_1 \in E_1$ and $w_2 \in E_2$.
  - $E^R \ni (w_1 w_2)^R = w_2^R w_1^R = \underbrace{w_2^R}_{\in E_2^R} \underbrace{w_1^R}_{\in E_1^R} \in E_2^R E_1^R$

- $E = E_1 + E_2$, clearly $E^R = E_1^R + E_2^R$.(why?)

...

Reversal

Proof (contd.).

- $E = E_1^*$,
  claim: $E^R = (E_1^R)^*$
  Let $w_1...w_n \in E$ such that $w_i \in E_1$ for each $i \in 1..n$.
  - $E^R \ni (w_1...w_n)^R = w_n^R...w_1^R = \underbrace{w_n^R}_{\in E_1^R} ... \underbrace{w_1^R}_{\in E_1^R} \in (E_1^R)^*$

□

Exercise 9.5

Write the regular expression for the following reverse languages.

- $(01 + 11)^R =$
- $(1(1 + 01)^*0)^R =$

Homomorphisms

Definition 9.1

A word homomorphism h is a function of type $\Sigma \rightarrow \Sigma'^*$, *where* Σ *and* Σ' *are alphabets.*

Definition 9.2

For a given word $a_1, ...a_n \in \Sigma^*$, *let us define*

$$h(a_1, ...a_n) \triangleq h(a_1)...h(a_n)$$

Example 9.1

Consider $h = \{0 \mapsto ab, 1 \mapsto \epsilon\}$.

- $h(0011) = abab$
- $h(101) = ab$

Homomoprphism of languages

Definition 9.3

For a given language L over Σ and word homomoprphism h, let us define

$$h(L) \triangleq \{h(w) | w \in L\}.$$

Example 9.2

Consider $h = \{0 \mapsto ab, 1 \mapsto \epsilon\}$.

- $h(01^*0) = abab$
- $h(10^*1) = (ab)^*$

Exercise 9.6

Consider $h = \{0 \mapsto a, 1 \mapsto ab\}$.

- $h(01^* + 0) =$
- $h(10^*1) =$

The definition can be naturally extended to the maps from REs to REs.

Closure under Homomorphisms

Theorem 9.2

Let L be a regular language and h be a word homomorphism, $h(L)$ is a regular language.

Proof.

Let R be the RE such that $L = L(R)$.

Let $h(R)$ be a regular expression obtained by replacing occurrences of each letter a in R by $h(a)$.

We will inductively show that $h(L(R)) = L(h(R))$.

base case:

Three cases

- $R = \epsilon$, $h(L(\epsilon)) = h(\{\epsilon\}) = \{\epsilon\} = L(\epsilon) = L(h(\epsilon))$
- $R = \emptyset$, $h(L(\emptyset)) = h(\emptyset) = \emptyset = L(\emptyset) = L(h(\emptyset))$
- $R = a$, $h(L(a)) = h(\{a\}) = \{h(a)\} = L(h(a)) = L(h(a))$

...

Closure under Homomorphisms II

Proof(contd.).

induction step:

Due to induction hypothesis we assume $h(L(E_1)) = L(h(E_1))$ and $h(L(E_2)) = L(h(E_2))$

- $R = E_1 + E_2$,

$$\begin{aligned} h(L(R)) &= h(L(E_1 + E_2)) \\ &= h(L(E_1) + L(E_2)) \\ &= h(L(E_1)) + h(L(E_2)) \\ &= L(h(E_1)) + L(h(E_2)) \\ &= L(h(E_1) + h(E_2)) \\ &= L(h(E_1 + E_2)) = L(h(R)) \end{aligned}$$

- $R = E_1E_2$ and $R = E_1^*$ are proven similarly.

□

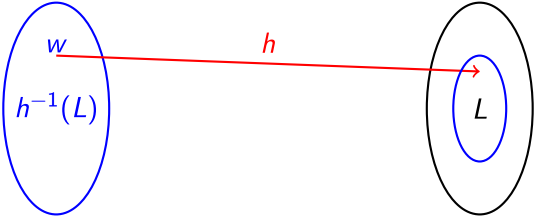
Exercise 9.7

Complete the above proof.

Inverse homomorphisms

Definition 9.4

Let L be a language. Let $h^{-1}(L) \triangleq \{w | h(w) \in L\}$.



Not all elements of L are mapped from $h^{-1}(L)$.

Example 9.3

Consider $L = (00 + 1)^*$ and $h = \{a \mapsto 01, b \mapsto 10\}$.
In L, 0s occur in even numbers.
For any w, $h(w)$ cannot have 0s longer than two. *(why?)*
Therefore, only words that are mapped to L by h are from $(ba)^*$.

Exercise 9.8 *Give a word w in L such that for each w', $h(w') \neq w$.*

Colusre under inverse homomorphisms

Theorem 9.3

If L is a regular language and word homomorphism $h : \Sigma' \to \Sigma^*$, $h^{-1}(L)$ is also a regular language.

Proof.

Let DFA $A = (Q, \Sigma, \delta, q_0, F)$ recognize L.

Let DFA $A' = (Q, \Sigma', \delta', q_0, F)$, where $\delta'(q, a) \triangleq \hat{\delta}(q, h(a))$.

claim: $L(A') = h^{-1}(A)$

Using induction on $|w|$, we can show $\hat{\delta}'(q_0, w) = \hat{\delta}(q_0, h(w))$.

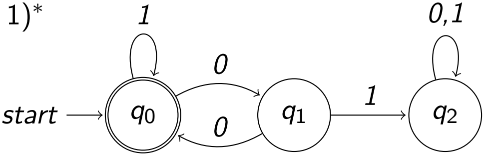
Since acceptance condition has not changed. A' accepts w iff A accepts $h(w)$. □

Example: inverse homomoprphism

Example 9.4

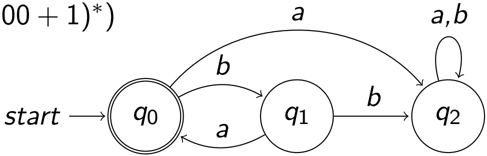
Consider again $L = (00+1)^*$ *and* $h = \{a \mapsto 01, b \mapsto 10\}$.

The DFA for $(00+1)^*$



$$\hat{\delta}(q_0, h(a)) = \hat{\delta}(q_0, 01) = q_2 \qquad \hat{\delta}(q_0, h(b)) = \hat{\delta}(q_0, 10) = q_1$$
$$\hat{\delta}(q_1, h(a)) = \hat{\delta}(q_1, 01) = q_0 \qquad \hat{\delta}(q_1, h(b)) = \hat{\delta}(q_1, 10) = q_2$$
$$\hat{\delta}(q_2, h(a)) = \hat{\delta}(q_2, 01) = q_2 \qquad \hat{\delta}(q_2, h(b)) = \hat{\delta}(q_2, 10) = q_2$$

The DFA for $h^{-1}((00+1)^*)$



Topic 9.1

Regular languages decision problems

Decision problems

- Is a language empty?
- Is a word belongs to a language?
- Does a language contain another language?

Difficulty of the question depends on the representation of the language.

Is L empty?

L is given as

- *DFA* or *NFA*
  - Check if reachable states from the initial state include any accepting state
  - Cost of finding reachable states $O(n^2)$, where n is the number of states

- *RE*

  Traverse RE bottom up and apply the following rules.
  - $\emptyset$ is empty.
  - ϵ and a are not empty.
  - $L_1 + L_2$ is empty if both L_1 and L_2 are empty.
  - $L_1 L_2$ is empty if either L_1 or L_2 are empty.
  - L_1^* is not empty.(why?)

  Cost of deciding the emptiness $O(n)$, n is the size of RE.

Exercise 9.9

Why is there a the gap in the computational cost?

Is $w \in L$?

Language given as

- DFA
  - run it on w and check if an accepting state is reached
  - Cost or running DFA $O(|w|)$
- *NFA*
  - compute reachable states on w
  - Cost or running NFA $O(|w|n^2)$, where n is the number of states

Exercise 9.10

What is cost of checking membership if L is given as RE?

End of Lecture 9