CIS 770: Formal Language Theory

Pavithra Prabhakar

Kansas State University

Spring 2015

Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language
- Very useful in studying the properties of one language by relating it to other (better understood) languages
- Most useful when the operations are sophisticated, yet are guaranteed to preserve interesting properties of the language.
- Today: A variety of operations which preserve regularity
  - i.e., the universe of regular languages is closed under these operations

Closure Properties

Definition

Regular Languages are closed under an operation op on languages if

$$L_1, L_2, \ldots L_n \text{ regular } \implies L = \mathrm{op}(L_1, L_2, \ldots L_n) \text{ is regular}$$

Example

Regular languages are closed under

- "halving", i.e., L regular $\implies \frac{1}{2}L$ regular.
- "reversing", i.e., L regular $\implies L^{\mathrm{rev}}$ regular.

Operations from Regular Expressions

Proposition

Regular Languages are closed under $\cup$, $\circ$ *and* *.

Proof.

(Summarizing previous arguments.)

- L_1, L_2 regular $\implies$ $\exists$ regexes R_1, R_2 s.t. $L_1 = L(R_1)$ and $L_2 = L(R_2)$.
  - $\implies L_1 \cup L_2 = L(R_1 \cup R_2) \implies L_1 \cup L_2$ regular.
  - $\implies L_1 \circ L_2 = L(R_1 \circ R_2) \implies L_1 \circ L_2$ regular.
  - $\implies L_1^* = L(R_1^*) \implies L_1^*$ regular. $\square$

Closure Under Complementation

Proposition

Regular Languages are closed under complementation, i.e., if L is regular then $\overline{L} = \Sigma^* \setminus L$ is also regular.

Proof.

- If L is regular, then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L = L(M)$.
- Then, $\overline{M} = (Q, \Sigma, \delta, q_0, Q \setminus F)$ (i.e., switch accept and non-accept states) accepts $\overline{L}$. $\square$

What happens if M (above) was an NFA?

Closure under $\cap$

Proposition

Regular Languages are closed under intersection, i.e., if L_1 and L_2 are regular then $L_1 \cap L_2$ is also regular.

Proof.

Observe that $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. Since regular languages are closed under union and complementation, we have

- $\overline{L_1}$ and $\overline{L_2}$ are regular
- $\overline{L_1} \cup \overline{L_2}$ is regular
- Hence, $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ is regular. $\square$

Is there a direct proof for intersection (yielding a smaller DFA)?

Cross-Product Construction

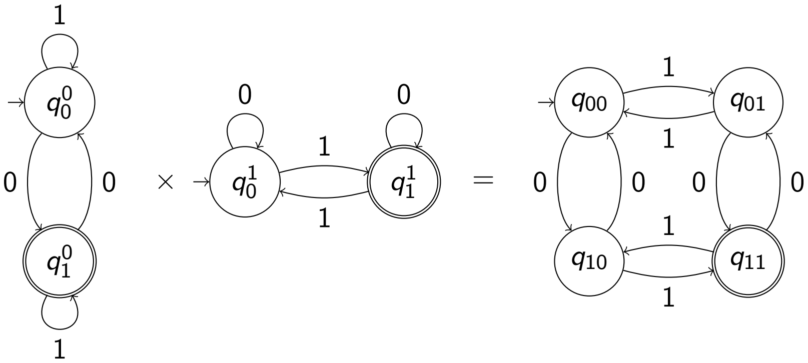
Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing L_1 and L_2, respectively.
Idea: Run M_1 and M_2 in parallel on the same input and accept if both M_1 and M_2 accept.

Consider $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $\delta(\langle p_1, p_2 \rangle, a) = \langle \delta_1(p_1, a), \delta_2(p_2, a) \rangle$
- $F = F_1 \times F_2$

M accepts $L_1 \cap L_2$ (exercise)
What happens if M_1 and M_2 where NFAs? Still works! Set $\delta(\langle p_1, p_2 \rangle, a) = \delta_1(p_1, a) \times \delta_2(p_2, a)$.

An Example



Homomorphism

Definition

A homomorphism is function $h : \Sigma^* \to \Delta^*$ defined as follows:

- $h(\epsilon) = \epsilon$ and for $a \in \Sigma$, $h(a)$ is any string in Δ^*
- For $a = a_1 a_2 \ldots a_n \in \Sigma^*$ $(n \geq 2)$, $h(a) = h(a_1)h(a_2)\ldots h(a_n)$.

- A homomorphism h maps a string $a \in \Sigma^*$ to a string in Δ^* by mapping each character of a to a string $h(a) \in \Delta^*$
- A homomorphism is a function from strings to strings that "respects" concatenation: for any $x, y \in \Sigma^*$, $h(xy) = h(x)h(y)$. (Any such function is a homomorphism.)

Example

$h : \{0, 1\} \to \{a, b\}^*$ where $h(0) = ab$ and $h(1) = ba$. Then $h(0011) = ababbaba$

Homomorphism as an Operation on Languages

Definition

Given a homomorphism $h : \Sigma^* \to \Delta^*$ and a language $L \subseteq \Sigma^*$, define $h(L) = \{h(w) \mid w \in L\} \subseteq \Delta^*$.

Example

Let $L = \{0^n 1^n \mid n \geq 0\}$ and $h(0) = ab$ and $h(1) = ba$. Then $h(L) = \{(ab)^n (ba)^n \mid n \geq 0\}$

Exercise: $h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$. $h(L_1 \circ L_2) = h(L_1) \circ h(L_2)$, and $h(L^*) = h(L)^*$.

Closure under Homomorphism

Proposition

Regular languages are closed under homomorphism, i.e., if L is a regular language and h is a homomorphism, then $h(L)$ is also regular.

Proof.

We will use the representation of regular languages in terms of regular expressions to argue this.

- Define homomorphism as an operation on regular expressions
- Show that $L(h(R)) = h(L(R))$
- Let R be such that $L = L(R)$. Let $R' = h(R)$. Then $h(L) = L(R')$.

Homomorphism as an Operation on Regular Expressions

Definition

For a regular expression R, let $h(R)$ be the regular expression obtained by replacing each occurence of $a \in \Sigma$ in R by the string $h(a)$.

Example

If $R = (0 \cup 1)^* 001 (0 \cup 1)^*$ and $h(0) = ab$ and $h(1) = bc$ then $h(R) = (ab \cup bc)^* ababbc (ab \cup bc)^*$

Formally $h(R)$ is defined inductively as follows.

$$\begin{array}{ll} h(\emptyset) = \emptyset & h(R_1 R_2) = h(R_1) h(R_2) \\ h(\epsilon) = \epsilon & h(R_1 \cup R_2) = h(R_2) \cup h(R_2) \\ h(a) = h(a) & h(R^*) = (h(R))^* \end{array}$$

Proof of Claim

Claim

For any regular expression R, $L(h(R)) = h(L(R))$.

Proof.

By induction on the number of operations in R

- Base Cases: For $R = \epsilon$ or $\emptyset$, $h(R) = R$ and $h(L(R)) = L(R)$. For $R = a$, $L(R) = \{a\}$ and $h(L(R)) = \{h(a)\} = L(h(a)) = L(h(R))$. So claim holds.
- Induction Step: For $R = R_1 \cup R_2$, observe that $h(R) = h(R_1) \cup h(R_2)$ and $h(L(R)) = h(L(R_1) \cup L(R_2)) = h(L(R_1)) \cup h(L(R_2))$. By induction hypothesis, $h(L(R_i)) = L(h(R_i))$ and so $h(L(R)) = L(h(R_1) \cup h(R_2))$
  Other cases ($R = R_1R_2$ and $R = R_1^*$) similar. □