CSC 320 – Lecture 05

#regular #languages #closure #expressions

Regular Languages: Closure Properties

Remember. A language L is called a **regular language** if there exists a deterministic finite automaton that recognizes L.

Regular Operations

Regular languages are closed under: union, intersection, concatenation, (Kleene) star.

Regular Languages – Practice

Let L_1, L_2 be regular languages. Is $L = ((L_1 \cup L_2)L_1) \cap L_2$ a regular language?

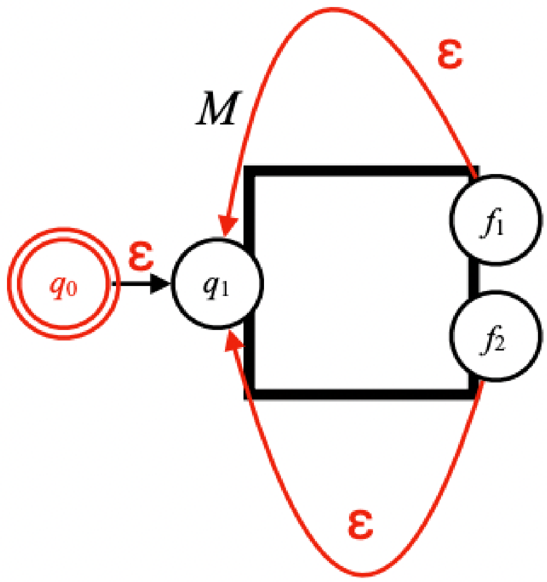
Yes! (Union, Concatenation, and Intersection).

Consider regular languages L_1, L_2, L_3, and L_4. Does there exist a deterministic finite automaton for language $L = ((L_1 L_2) \cup L_3)^* L_4$?

Closure Properties (4)

Theorem. If L is a regular language then L^* is a regular language also.

Proof. Sketch: Regular Languages are closed under star operation.



Regular Expressions

Wouldn't it be nice if we could describe regular languages shorter? (E.g., $(0 \cup 1)^*0$ for the language of all strings over the binary alphabet that end with 0).

Examples where regular languages are used in practice:

- Unix: awk, grep
- Perl
- Texteditors
- Design of compilers (lexical analyzer)

Definition

R is a **regular expression** if R is equal to: (one of the following)

- a, for some $a \in \Sigma$
- ϵ
- $\emptyset$
- $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions.
- $(R_1 R_2)$ where R_1 and R_2 are regular expression.
- (R_1^*) where R_1 is a regular expression.

Defined inductively. Regular expressions are:

- $a (a \in \Sigma)$, ε, $\emptyset$

- $(R_1 \cup R_2)$, $(R_1 R_2)$, and (R_1^*), where R_1 and R_2 are regular expressions.

Note. Inductive definition uses intersection, concatenation, star only!

Conventions

- Parentheses can be omitted.
  - If no parentheses, order to evaluate is: star, concatenation, union.
- $R^+ := RR^*$

Identities

- $R^+ := RR^*$

- $R \cup \emptyset = R$
- $R_\varepsilon = R$

Definition: Language Recognized by a Regular Expression

Assume R_1 and R_2 are regular expressions. The **language** $L(R)$ for **regular expression** R is defined as:

- If $R = a$, for some $a \in \Sigma$, the $L(R) = \{a\}$
- If $R = \varepsilon$ then $L(R) = \{\varepsilon\}$
- If $R = \emptyset$ then $L(R) = \emptyset$
- If $R = (R_1 \cup R_2)$ then $L(R) = L(R_1) \cup L(R_2)$
- If R = $(R_1 R_2)$ then $L(R) = L(R_1)L(R_2)$
- If $R = (R_1^*)$ then $L(R) = L(R_1)^*$

Equivalence of Regular Languages and the Set of Languages of Regular Expressions

Theorem. A language is regular if and only if there exists some regular expression that describes it.

Proof.

The proof consists of two parts:

1. If a language is described by a regular expression, then it is regular.
2. If a language is regular, then it can be described by a regular expression.

Claim 1. If a language is described by a regular expression, then it is regular.

Given a regular expression R, we show that there exists a finite automaton M with $L(M) = L(R)$.

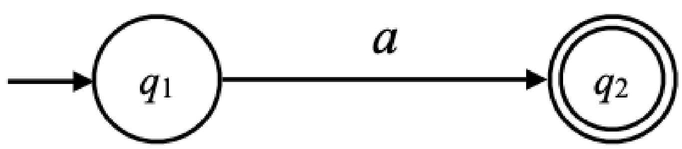
We distinguish the following cases: (Inductive Proof)

- $R = a$, $a \in \Sigma$
- $R = \varepsilon$
- $R = \emptyset$
- $R = (R_1 \cup R_2)$, R_1, R_2 regular expressions.
- $(R_1 R_2)$, R_1, R_2 regular expressions.
- (R_1^*) where R_1 is a regular expression.

Case 1: $R = a$, $a \in \Sigma$

NFA $N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$ with $\delta(q_1, a) = \{q_2\}$ and $\delta(r, b) = \emptyset$ for $(r, b) \neq (q_1, a)$

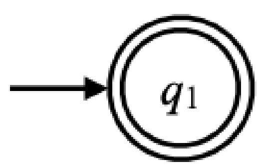
N



Case 2: $R = \varepsilon$

NFA $N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$ with $\delta(q_1, b) = \emptyset$ for any $b \in \Sigma \cup \{\varepsilon\}$

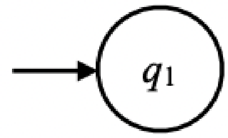
N



Case 3: $R = \emptyset$

NFA $N = (\{q_1\}, \Sigma, \delta, q_1, \emptyset)$ with $\delta(q_1, b) = \emptyset$ for any $b \in \Sigma \cup \{\varepsilon\}$

N



Case 4: $R = (R_1 \cup R_2)$, R_1, R_2 Regular Expressions

To show. Given regular expressions R_1 and R_2 where $L(R_1)$ **and** $L(R_2)$ **are regular languages** (Inductive Proof), else $L(R_1 \cup R_2)$ is regular.

We know. the set of regular languages closed under union. Therefore, $L(R_1) \cup L(R_2)$ is regular.

From definition of languages recognized by regular expressions:
$L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$.

Therefore, $L(R_1 \cup R_2)$ is regular.

Case 5: $(R_1 R_2)$, R_1, R_2 Regular Expressions

To show. Given regular expressions R_1 and R_2 where $L(R_1)$ **&** $L(R_2)$ (Inductive Proof) are regular languages, also $L(R_1 R_2)$ is regular.

We know. The set of regular languages is closed under concatenation. Therefore, $L(R_1)L(R_2)$ is regular.

From definition of languages recognized by regular expressions: $L(R_1 R_2) = L(R_1)L(R_2)$
.

Therefore, $L(R_1 R_2)$ is regular.

Case 6: (R_1^*) Where R_1 is a Regular Expression

To show. Given regular expression R_1 where $L(R_1)$ (Inductive Proof) is a regular language, also $L(R_1^*)$ is regular.

We know. the set of regular languages closed under star. Therefore, $L(R_1^*)$ is regular.

From definition of languages recognized by regular expressions: $L(R_1)^* = L(R_1^*)$.

Thus, language$L(R_1)^*$ described by regular expression R_1^* is regular.

Note. This is on Assignment 02 - We will do the proof.

Previous Lecture

Lecture04

Next Lecture

Lecture06