CSE211 – Formal Languages and Automata Theory

U1L12 – Regular Expressions

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Recap of previous class
- Operations on Strings
- Operations on Languages
- What is Regular Language?
- Definition of Regular Language
- Examples and Exercise

Note: NFA to DFA Conversion

- After conversion, the number of states in the resulting DFA may or may not be same as NFA
- The maximum number of states that may be present in the DFA are $2^{\text{Number of states in the NFA}}$
- If m is the number of states in the NFA, then the relationship exists between the number of states in the NFA and DFA-

$$1 <= m <= 2^m$$

- In the resulting DFA, all those states that contain the final state(s) of NFA are treated as final states

Operations on strings

- Given two strings $s = a_1 \ldots a_n$ and $t = b_1 \ldots b_m$, we define their concatenation $st = a_1 \ldots a_n b_1 \ldots b_m$

  s = abb, t = cba

  st = abbcba

- We define s^n as the concatenation $ss \ldots s$ n times

  s = 011 **s^3 = 011011011**

Operations on languages

- The concatenation of languages L_1 and L_2 is

$$L_1L_2 = \{st: s \in L_1, t \in L_2\}$$

- Similarly, we write L^n for $LL\ldots L$ (n times)
- The union of languages $L_1 \cup L_2$ is the set of all strings that are in L_1 or in L_2

- Example: $L_1 = \{01, 0\}, L_2 = \{\varepsilon, 1, 11, 111, \ldots\}$. What is L_1L_2 and $L_1 \cup L_2$?

Operations on languages

- The star (Kleene closure) of L are all strings made up of zero or more chunks from L:

$$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$$

  - This is always infinite, and always contains ε

- $\phi^* = \{\varepsilon\}$ because $\phi^0 = \{\varepsilon\}$.

Operators of RE

- Review of three operations on two languages L and M:
- *Union:*
  - $L \cup M = \{x \mid x \in L \text{ or } x \in M\}$
- *Concatenation*:
  - $LM = \{xy \mid x \in L, y \in M\}$
  - Example --- $L^0 = \{\varepsilon\}$, $L^1 = L$, $L^2 = LL$, ...
- *Closure* (or *star*, or *Kleene closure*) ---
  - $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

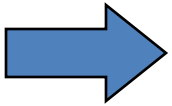
Operators of RE

- If L = {0, 1}, then
  - L^0 = {ε},
  - L^1 = L,
  - L^2 = {00, 01, 10, 11}, …
- If L is the set of all strings of 0's, then it can be proved that L^* is L itself
- Example: $L_1 = \{01, 0\}, L_2 = \{\varepsilon, 1, 11, 111, \ldots\}$. What is ${L_1}^*$ and ${L_2}^*$?

Constructing languages with operations

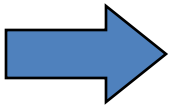
- Let's fix an alphabet, say $\Sigma = \{0, 1\}$
- We can construct languages by starting with simple ones, like $\{0\}$, $\{1\}$ and combining them
- Examples:

$\{0\}(\{0\} \cup \{1\})^*$
all strings that start with 0



$0(0+1)^*$

$(\{0\}\{1\}^*) \cup (\{1\}\{0\}^*)$



01^*+10^*

Regular Expressions

- The regular expression is a kind of generator for languages.
- It offers a “declarative” way of expressing strings of symbols
- It defines all and only regular languages
- Ex: a(b/c)*

Regular expressions

- A regular expression over Σ is an expression formed using the following rules:
  - The symbol $\emptyset$ is a regular expression
  - The symbol ε is a regular expression
  - For every $a \in \Sigma$, the symbol a is a regular expression
  - If R and S are regular expressions, so are RS, $R+S$ and R^*.
- Definition of regular language

A language is **regular** if it is represented by a regular expression

Examples

1. $01^* = \{0, 01, 011, 0111, \ldots..\}$
2. $(01^*)(01) = \{001, 0101, 01101, 011101, \ldots..\}$
3. $(0+1)^*$
4. $(0+1)^*01(0+1)^*$
5. $((0+1)(0+1)+(0+1)(0+1)(0+1))^*$
6. $((0+1)(0+1))^*+((0+1)(0+1)(0+1))^*$
7. $(1+01+001)^*(\varepsilon+0+00)$

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

Finite Automata and Regular Expression

THANK YOU.