Regular Expressions

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in

Former Professor & Head, Department of Computer Sc. & Engineering
MBM Engineering College, Jodhpur

Friday 22nd January, 2021

Regular Languages

- The set R on an alphabet Σ is regular language, defined as follows:
  $\phi \subseteq R$, $\{\varepsilon\} \subseteq R$, $\{a\} \subseteq R$ for all $a \in \Sigma$
  if $L_1, L_2 \subseteq R$, then $L_1 \cup L_2 \subseteq R, L_1 \circ L_2 \subseteq R$, and $L_1^* \subseteq R$.
- The language $\{0, 00, 01, 000, 001, 010, 011, 0000, \ldots\}$, which consists all binary strings is regular. It can be constructed using regular expression $\{0\} \circ (\{0\} \circ \{1\})^*$.
- A Regular language is represented by regular expression.

Definition

Following are regular expressions:

1. ϕ, ε, and $a \in \Sigma$ are regular expressions.
2. if α, β are regular expressions, the $\alpha \cup \beta$, $\alpha \circ \beta$, and α^* are also regular expressions.
3. Nothing else is regular expression.

- A language is regular *iff* there is regular expression to represent it.

Regular Languages

Definition

Every regular expression has functional mapping to a regular language, as follows:

- The regular languages corresponding to regular expressions ϕ, ε are: ϕ, and $\{\varepsilon\}$, respectively.
- if α, β are reg. expressions, then corresponding regular languages are: $\mathscr{L}\{\alpha\}$, and $\mathscr{L}\{\beta\}$.
- Also, $\mathscr{L}\{\alpha \circ \beta\} = \mathscr{L}\{\alpha\}\mathscr{L}\{\beta\}$, $\mathscr{L}\{\alpha \cup \beta\} = \mathscr{L}\{\alpha\} \cup \mathscr{L}\{\beta\}$, $\mathscr{L}\{\alpha^*\} = \mathscr{L}\{\alpha\}^*$

- If L is regular, then is $\bar{L}$ also regular (to be seen in the study of finite automata)
- Is the intersection of two regular languages also necessarily regular? (to be seen later)
- Are all the languages regular? Justify(to be seen later)

Regular Languages

- All the regular expressions can be listed of increasing length. Hence, they can be mapped with set $\mathbb{N}$.Thus regular expressions are **countable**. However, set of all the possible languages on set Σ^* is power set $2^{|\Sigma^*|}$, which is not countable. **Conclusion: There are not enough regular expressions to represent all languages.**
- Language over $\Sigma = \{a, b\}$ for *regex* $(a+b)^*$ is: $L = \mathscr{L}((a+b)^*)$ $= \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots\}$, which is countably infinite set. Thus, L is *bijection* to the set $\mathbb{N}$. Thus, L is called Recursively Enumerable language (RE).

Definition

(RE). A language L is RE if there exists an algorithm that, when given an input w, eventually halts *iff* $w \in L$. Equivalently, there is an algorithm that enumerates the members of L. If necessary, this algorithm may run forever. Thus, L is called *semidecidable language.*

Regular Languages

Consider the language $L = \{a, ab, bba, bca, abcd\}$, which has bijection with a set $n \in \mathbb{N}$, such that $|n| = |L|$. The mapping of bijection is $\{(0, a), (1, ab), (2, bba), (3, bca), (4, abcd)\}$. The language, L above is called **recursive.(R)**

Definition

(Recursive). A language L is recursive if there exists an algorithm that, given an input word w, will determine in a finite amount of time if $w \in L$ or not. A recursive language is **decidable**.

- A recursive (**R**) language is enumerable. Hence, R $\subseteq$ RE.
- A set of programs that do not terminate on some inputs are **not RE**.
- Since regular languages are countable, and for each countable set there are uncountable subsets of languages. Thus, there exists languages which are not regular.
- **Conclusion: All languages are not the regular languages.**