Programming Languages: Lecture 7
Regex

Rishabh Dhiman

15 January 2022

1 Regular Expressions Language

- Any set of strings built up from the symbols of A is called a language. A^* is the set of all finite strings buillt up form A.
- Each regex is a finite sequence of symbols made up of symbols from the alphabet and other symbols called operators.
- A regular expression may be used to describe an *infinite* collection of strings.

2 Language

Any collection of finite strings is a language.

3 Simple Language of Regular Expressions

We consider a simple language of regular expressions. Assume a (finite) alphabet A of symbols. Each regular expression r denotes a set of strings $\mathcal{L}(r)$. $\mathcal{L}(r)$ is also called the *language* spexified by the regular expression r.

- Symbol, for $a \in A$, $\{a\}$ refers to the single element a.
- Concatenation. $\mathcal{L}(rs) = \mathcal{L}(r)\mathcal{L}(s)$.
- Epsilon ε denotes the language with a single element the *empty* string, " ".

$$L(\varepsilon) = \{\varepsilon\}.$$

- Alternation. Given two regex r, s; $r \mid s$ is the set of union of the languages specified by r and s.

$$\mathcal{L}(r \mid s) = \mathcal{L}(r) \cup \mathcal{L}(s).$$

- Kleene Closure $r^* = r^0 \mid r^1 \mid \cdots$ denotes an infinite union of languages.

$$\mathcal{L}(r^*) = \bigcup_{n=0}^{\infty} \mathcal{L}(r^n).$$

- +-closure: $r^+ = r^1 \mid r^2 \mid \cdots$.
- Range specifications: $[a - c] = a \mid b \mid c$.

The set of regex over an alphabet A is a monoid under concatenation, also under alternation.

4 DFA/NFA

A regex expression can be turned into an NFA, which can be turned into a DFA.

For an NFA N, define $\mathcal{L}(N)$ as the set of languages that N accepts.

5 Regex to NFA Construction

We do so by structural induction.

Each regex operator adds at most 2 new states and at most 4 new transitions. So, for a regex r, N_r has at most $2\,|r|$ and $4\,|r|$ transitions.

6 Extensions

1. Show how to construct a NFA for ranges and multiple ranges of symbols.
2. Assuming N_r is an NFA for the regex r, how will you construct NFA N_{r^+}.
3. $\mathcal{L}(r\{k, n\}) = \bigcup_{k \le m \le n} \mathcal{L}(r^m)$.
4. $\mathcal{L}(\widehat{r}) = A^* - \mathcal{L}(r)$.