CSC236 Week 05: Languages: Definitions

Hisbaan Noorani

October 7 – October 13, 2021

Contents

1 Some definitions

- **Alphabet:** Finite, non-empty set of symbols, e.g. $\{a, b\}$ or $\{0, 1, -1\}$. Conventionally denotes Σ.
- **String:** Finite (including empty) sequence of symbols over an alphabet: abba is a string over $\{a, b\}$. Convention: ε is the empty string, never an allowed symbol, Σ^* is set of all strings over Σ.
- **Language:** Subset of Σ^* for some alphabet Σ. Possible empty, possibly empty, possibly infinite subset. E.e. $\{\}, \{aa, aaa, aaaa, \ldots\}$.

N.B.: $\{\} \neq \{\varepsilon\}$. *Proof:* $|\{\}| = 0 \neq 1 = |\{\varepsilon\}| \implies \{\} \neq \{\varepsilon\}$ ■

Many problems can be reduced to languages: logical formulas, identifiers from compilation, natural language processing. The key question is recognition:

Given language L and string s, is $s \in L$?

2 More notation — string operations

- **String length:** denotes $|s|$, is the number of symbols in s, e.g. $|bba| = 3$.
- $s = t$**:** if and only if $|s| = |t|$, and $s_i = s_t$, for $0 \leq i < |s|$.
- s^R**:** reversal of s is obtained by reversing symbols of s, e.g. $1011^R = 1101$.

- **st or $s \circ t$:** concatenation of s and t — all characters of s followed by all those of t, e.g. $bba \circ bb = bbabb$.
- **s^k:** denotes s concatenated with itself k times, e.g. $ab^3 = ababab, 101^0 = \varepsilon$.
- **Σ^n:** all strings of length n over Σ, Σ^* denotes all strings over Σ.

3 Language operations

- **$\overline{L}$:** Complement of L, i.e. $\Sigma^* - L$. If L is a language of strings over $\{0, 1\}$ that start with 0, then $\overline{L}$ is the language of strings that begin with 1 plus the empty string.
- **$L \cup L'$:** Union.
- **$L \cap L'$:** Intersection.
- **$L - L'$:** Difference.
- **$\mathrm{Rev}(L)$:** $= \left\{s^R : s \in L\right\}$.
- **concatenation:** LL' or $L \circ L' = \{rt : r \in L, r \in L'\}$.

  There are some special cases, $L\{\varepsilon\} = L = \{\varepsilon\} L,$ and $L\{\} = \{\} = \{\} L$.
- **exponentiation:** L^k is concatenation of L, k times.

  There is a special case, $L^0 = \{\varepsilon\}$, including $L = \{\}$.
- **Kleene star:** $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

4 Another way to define languages

In addition to the set description $L = \{\ldots\}$.

Definition: The regular expressions (regexps or REs) over alphabet Σ is the smallest set such that

- $\emptyset, \varepsilon,$ and x, for every $x \in \Sigma$ are REs over Σ.
- If T and S are REs are over Σ, then so are:
  - $(T + S)$ (union) — lowest precedence operator
  - (TS) (concatenation) — middle precedence operator
  - T^* (star) — highest precedence

5 Regular expression to languages:

The $L(R)$, the language denoted (or described) by R is defined by structural induction.

- Basis; If R is a regular expression by the basis of the definition of regular expressions, then define $L(R)$:
  - $L(\emptyset) = \emptyset$ (the empty language – no strings!)
  - $L(\varepsilon) = \{\varepsilon\}$ (the language consisting of just the empty string)
  - $L(x) = \{x\}$ (the language consisting of the one-symbol string)

- Induction step: If R is a regular expression by the induction step of the definition, then define $L(R)$:
  - $L((T+S)) = L(S) \cup L(T)$
  - $L((TS)) = L(S)L(T)$
  - $L(T^*) = L(T)*$

We are assuming about $(S+T)$ and (ST) above?

6 Regexp examples

- $L(0+1) = L(0) \cup L(1) = \{0, 1\}$
- $L((0+1)^*)$ All binary strings over $\{0, 1\}$
- $L((01)^*) = \{\varepsilon, 01, 0101, 010101, \ldots\}$
- $L(0^*1^*)$ 0 or more 0s followed by 0 or more 1s
- $L(0^* + 1^*)$ 0 or more 0s or 0 or more 1s
- $L((0+1)\,(0+1)^*)$ Non-empty binary strings over $\{0, 1\}$