Formal Languages

San Skulrattanakulchai

Feb 9, 2016

Terminology

Formal languages are our models for the data manipulated by computers.

- A *symbol* (or *letter*) is an undefined term.
- An *alphabet* is a nonempty, finite set of symbols, e.g., if $\Sigma = \{a, b\}$ then Σ is the alphabet while $\mathtt{a}$ and $\mathtt{b}$ are symbols.
- A *string* (*word*, *sentence*) is a finite list of symbols chosen from an alphabet, e.g., $\langle \mathtt{a}, \mathtt{b}, \mathtt{a} \rangle$, usually written $\mathtt{aba}$.

Terminology

- The length of string w, denoted $|w|$, is the length of the list.
- The *empty string* ε has length 0.
- Formal language theory allows infinite-length strings; we don't.
- If $|w| = n$, we write w as $w_1 w_2 \ldots w_n$. E.g., letting $w = \texttt{aba}$, we have $w_1 = w_3 = \texttt{a}$, and $w_2 = \texttt{b}$.
- For any string w and symbol a, we write $|w|_a$ to denote the number of times the symbol a occurs in string w. E.g.,
  - $|\texttt{aba}|_{\texttt{a}} = 2$
  - $|\texttt{aba}|_{\texttt{b}} = 1$
  - $|\texttt{aba}|_{\texttt{c}} = 0$

Terminology

- The length of string w, denoted $|w|$, is the length of the list.
- The *empty string* ε has length 0.
- Formal language theory allows infinite-length strings; we don't.
- If $|w| = n$, we write w as $w_1 w_2 \ldots w_n$. E.g., letting $w = \mathtt{aba}$, we have $w_1 = w_3 = \mathtt{a}$, and $w_2 = \mathtt{b}$.
- For any string w and symbol a, we write $|w|_a$ to denote the number of times the symbol a occurs in string w. E.g.,
    - $|\mathtt{aba}|_{\mathtt{a}} = 2$
    - $|\mathtt{aba}|_{\mathtt{b}} = 1$
    - $|\mathtt{aba}|_{\mathtt{c}} = 0$
- Notice how we use names (symbols) like w and a to talk about things made up of other symbols (like $\mathtt{a}$ and $\mathtt{b}$)? Keep them separate in your mind!

Terminology

- Let $x = x_1 x_2 \dots x_m$ and $y = y_1 y_2 \dots y_n$ be strings. The *concatenation* of x and y, written xy, is the string $x_1 x_2 \dots x_m y_1 y_2 \dots y_n$ of length $m + n$ that results from appending y to the end of x, e.g., concatenating `back` and `bone` gives `backbone`.
- String concatenation operation is *associative*, and ε is the *identity element*, i.e., $\varepsilon w = w\varepsilon = w$ for any string w. $\therefore$ the set of all strings over an alphabet is a *monoid* under concatenation.

Terminology

- If w is a string and n is a positive integer, we write w^n to mean the concatenation of n copies of w. The notation w^0 is defined to be ε.
- A string y is a *substring* (or *subword*) of string w if there exist strings x, z such that $w = xyz$.
- A string x is a *prefix* of string w if there exists a string y such that $w = xy$.
- A string y is a *suffix* of string w if there exists a string x such that $w = xy$.
- By definition,
  - an empty string is a substring, prefix, and suffix of any string
  - any string is a substring, prefix, and suffix of itself

Terminology

- String x is a *subsequence* of string y if x is obtained by striking out 0 or more symbols from y. E.g., `bat` is a subsequence of `habitat`.
- Let $w = w_1 w_2 \ldots w_n$ be a string of length n. By the *reverse of* w, notated w^R, we mean the string $w_n w_{n-1} \ldots w_1$. For example, $\texttt{star}^R = \texttt{rats}$.
- A string w is a `palindrome` if $w^R = w$. Examples of palindromes are `eve`, `madam`, `racecar`, `deified`, `rotator`.
- Given alphabet Σ, define Σ^* to be the set of all strings over Σ. E.g., if $\Sigma = \{a, b\}$ then $\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots\}$.
- The listing of strings above is in *shortlex order* (*string order*, *radix order*), i.e., ordered like in a dictionary, except that a shorter string always precedes a longer one.

Exercises

- Define precisely the less than relation $<$ for dictionary order (lexicographic order).
- Define precisely the less than relation $<$ for shortlex order (string order, radix order).
- What is the position of the string `ab`, when the strings of $\{a, b\}^*$ are arranged in dictionary order? in shortlex order?

Languages

- A *language over the alphabet* Σ is any subset of Σ^*.
- Some example languages:
  1. The set of all strings with an odd number of $\mathtt{a}$.
  2. The set of all palindromes.
  3. The set of all strings of "balanced" left and right parentheses.
  4. The set of all strings with equal numbers of $\mathtt{a}$, $\mathtt{b}$, and $\mathtt{c}$.
  5. The set of all binary strings that represent prime numbers.
  6. The set of all graphs with a Hamiltonian cycle, where the graph is encoded as a string.
  7. $\emptyset$ and $\{\varepsilon\}$ are different languages.

Remarks

- The subject matter of this course is languages and machines that recognize/compute them!
- Finite languages are trivial.
- A lone letter like a is ambiguous. It either represents a symbol or a string of length 1. Context decides which meaning is intended.
- The concepts of “string”, “concatenation”, “string length”, “string reversal”, etc., can be defined inductively.

Language Operations

- *Set Operations*: $\cup$, $\cap$, $\setminus$, $\triangle$, complement $\bar{A}$ of language A
- *Concatenation*: The concatenation of two languages A and B is AB, i.e., the set of all strings xy where $x \in A$ and $y \in B$. When precision is desired, concatenation is denoted by $\circ$, e.g., $x \circ y$, $A \circ B$.
- Let O $=$ {all strings of odd length}, E $=$ {all strings of even length}, and N $= \{\mathtt{a}\}$. Find ON, OE, and EE.

Language Operations

- *Set Operations*: $\cup$, $\cap$, $\setminus$, $\triangle$, complement $\bar{A}$ of language A
- *Concatenation*: The concatenation of two languages A and B is AB, i.e., the set of all strings xy where $x \in A$ and $y \in B$. When precision is desired, concatenation is denoted by $\circ$, e.g., $x \circ y$, $A \circ B$.
- Let O $=$ {all strings of odd length}, E $=$ {all strings of even length}, and N $=$ {a}. Find ON, OE, and EE.

```
Answer:
    ON = { all strings of even length ending in a }
    OE = O
    EE = E
```

Language Operations

- *Power*: For any language A, language A^0 denotes $\{\varepsilon\}$; languages A^i denotes AA^{i-1} whenever $i > 0$.
- *Kleene Closure*: $A^* = \bigcup_{i=0}^{\infty} A^i$. E.g., $\emptyset^* = \{\varepsilon\}$. Note how this definition of * agrees nicely with our previous definition of * in Σ^* if we identify a string of length one with the symbol contained in it.
- *Positive Closure*: $A^+ = \bigcup_{i=1}^{\infty} A^i$.

Exercises

- Is it true that $A^+ = A^* \setminus \{\varepsilon\}$ for every language A?.
- Which ones of the seven example languages satisfy $A = A^*$?
- Characterize languages A that satisfy $A^* = A^+$?
- Describe these languages: $A\emptyset$, $A\{\varepsilon\}$, $A \cup \emptyset$, $A \cup \{\varepsilon\}$.
- The $\cup$ and the $\circ$ operators for languages are comparable to the $+$ and the $\times$ operators for numbers, respectively.
  - What is the identity element for $\cup$? for $\circ$?
  - What rules governing $+$ and $\times$ are also obeyed by $\cup$ and $\circ$?