Formal Language Theory - Definitions

Finite-state technology

The obtained model is that of finite-state automata.

In this case it is better to turn the tree into a graph.

A tree can be augmented to store also a morphological dictionary specifying constituentive affixes, especially suffixes.

The solution: the word tree.

The technique: storing in a data structure a representation of the word.

Infinite-state technology

Implementing morphology and phonology
Formal language theory - definitions

Two special cases of substrings are prefix and suffix:

\[ \langle x^{i} \cdots x^{1} \rangle \text{ is a substring of } m \text{ if and only if } x \leq i \leq 1 \text{ such that } \langle u^{1} \cdots u^{i} \rangle \text{ occur in } m \text{. If } x \text{ then } \langle u^{1} \cdots u^{i} \rangle = m \text{ in the order in which they taking contiguous symbols of } m \text{ in the order in which they occur in } m \text{. A substring of } m \text{ is a sequence formed by substituting } x \text{ in the reverse order. Thus, if } m \text{ is a substring of } m \text{ then } m = m \cdot x \text{ and is obtained by concatenation of } x \text{ to the end of } m \text{. }\]

Example: Concatenation

Let \( \mathcal{A} = \{ a, b, c, d, e \} \) be an alphabet. Then:

\[ |m| = 3 \text{ and } |m \cdot x| = |m| + |x| = |\langle x \cdot x \cdots x \rangle | \text{ is the concatenation of } m \text{ and } x \text{. Denoted } |m \cdot x| \text{ and is denoted } |m| \text{ is the number of letters in the string } m \text{. The length of a string } m \text{, denoted } |m|, \text{ is the number of letters in } m \text{.} \]

For every string } m, m \cdot e = e = e \cdot m\]
Formal language theory – definitions

A formal language over an alphabet $\Sigma$ is a subset of $\Sigma^*$. Given an alphabet $\Sigma$, the set of all strings over $\Sigma$ is denoted $\Sigma^*$. By $\Sigma^*$, the set of all strings of length $\leq n$ is denoted $\Sigma^n$.

Formal language theory – definitions

Note that the first five languages are infinite while the

Example: Languages

Let $\Sigma = \{a, b, \ldots, z\}$. Then $\Sigma^*$ is the set of all

Languages include distinguishable, code and

are suffixes of $w$. Substrings that are neither prefixes nor

indistinguishable and indistinguishable are prefixes of $\Sigma^n$. Let $\Sigma = \{a, b, c, \ldots, z\}$ be an alphabet and $w$ is

Example: Substrings


Regular expressions are a formalism for defining (formal) languages. A regular expression is a set of strings in some formal simple "semantics" is set of strings: the denotation languages. Their "syntax" is formally defined and is relatively simple. Their "semantics" is set of strings: the denotation languages. The Kleene closure is defined as

The Kleene closure of $I$ and is denoted $I^*$ and is defined as

Formal Language Theory – Definitions

Formal Language Theory – Definitions
Example: Regular expressions and their denotations

Regular expressions

Let \( \Sigma \) be the alphabet \{ a, b, c \}. Some regular expressions over this alphabet are:

\[ \{ \epsilon \} \]
\[ \{ \sigma \} \]
\[ \emptyset \]

Strings defined as follows:

For every regular expression \( r \) its denotation \( [r] \) is a set of strings over \( \Sigma \). If nothing else is a regular expression over \( \Sigma \):

- If \( \epsilon \) is a regular expression then so is \( \{ \epsilon \} \).
- If \( \sigma \) is a regular expression then so are \( \{ \sigma \} \) and \( \emptyset \).
- If \( \sigma \in \Sigma \) is a letter then \( \sigma \) is a regular expression.
- If \( \emptyset \) is a regular expression then \( \emptyset \) is a regular expression.

Regular expressions are defined recursively as follows:

\[ [\epsilon] = \{ \epsilon \} \]
\[ \{ \sigma \} = [\sigma] \]
\[ \emptyset = [\emptyset] \]

If \( \epsilon \) and \( \sigma \) are regular expressions whose denotations are:

\[ \{ \epsilon \} \] and \( [\sigma] \) respectively then:

\[ \{ \epsilon \} \cap [\sigma] = [\sigma + \epsilon] \]
Propertes of Regular Languages

Closed under:

- Substitution and homomorphism
- Kleene-star
- Concatenation
- Difference
- Complementation
- Intersections
- Union

Regular Languages are closed under:

Properties of Regular Languages

Note that all these languages are infinite.

- The set of strings that end in either "tron" or "sion" is denoted by \( L_1 \).
- The set of all strings that begin in "un" is denoted by \( L_2 \).
- The set of all strings which contain a vowel is denoted by \( L_3 \).
- Given the alphabet of all English letters, \( \Sigma = \{ a,b,c,d,e,f, \ldots \} \), the language \( X \) is denoted by the regular expression expression \( \Sigma^* \).

Example: Regular expressions