

Alphabets and strings

Strings are a simpler version of lists, in which all list elements come from a finite set of symbols, called an *alphabet*.

Because of this simpler structure, there is no need to use " \langle ", " \rangle " and " $;$ " when representing strings.

For example, if the alphabet is $\{0, 1\}$, we write

010

to represent the string of length 3 that corresponds to the list $\langle 0, 1, 0 \rangle$.

But this simplified notation for strings imposes a slight cost. We need a notation for the *empty string* — the string of length 0. It is written

Λ .

Eventually we will work with strings a lot, and so we also introduce a convenient notation for the length of a string s :

$|s|$.

Languages, string concatenation

Given an alphabet A , a *language* over A is a set of strings over A .

For example, \emptyset , $\{\Lambda\}$, $\{a\}$, $\{\Lambda, a, aa\}$ are all languages over the alphabet $\{a\}$.

We write A^* to denote the set of all strings over alphabet A .

Notice that A^* itself is a language over A .

If x and y are strings, we denote their concatenation by writing

xy .

For example, if $x = 01$ and $y = 10$, then

$$xy = 0110, yx = 1001, xx = 0101, yy = 1010, xyx = 011001.$$

Notice that, for all strings s ,

$$\Lambda s = s = s\Lambda.$$

For any string s and $n \in \mathcal{N}$, s^n denotes the concatenation of s with itself n times:

$$s^0 = \Lambda, s^1 = s, s^2 = ss, s^3 = sss, \dots$$

Here are some examples of the use of exponent notation for string concatenation:

$$\{a^n \mid n \in \mathcal{N}\} =$$

$$\{ab^n \mid n \in \mathcal{N}\} =$$

$$\{a^n b^n \mid n \in \mathcal{N}\} =$$

$$\{(ab)^n \mid n \in \mathcal{N}\} =$$

$$\{xx^n \mid n \in \mathcal{N}, x \in \{a, b\}^*\} = \{a, b\}^* ?$$

Products of languages

The *product* of languages L and M is the language

$$LM = \{xy \mid x \in L, y \in M\}.$$

For example, if $L = \{0, 1\}$ and $M = \{\Lambda, 0\}$, then

$$LM = \quad \text{and} \quad ML =$$

Notice: For all languages L ,

$$L\{\Lambda\} = \{\Lambda\}L =$$

and

$$L\emptyset = \emptyset L =$$

We also have exponent notation for language products:

$$L^n = \{x_1 \cdots x_n \mid \text{for all } i (1 \leq i \leq n), x_i \in L\}.$$

The special case when $n = 0$ is given by

$$L^0 = \{\Lambda\}.$$

What is L^1 ?

Notice that $L^m L^n = L^{m+n}$.

(Kleene) closure of a language

The *closure* of a language L , written L^* , is defined as follows.

$$L^* = \bigcup_{i \in \mathcal{N}} L^i.$$

Consider some examples:

$$\{0\}^* =$$

$$\{00\}^* =$$

$$\{0, 1\}^* =$$

$$\{0\}^* \{1\}^* =$$

$$\{\wedge\}^* =$$

$$\emptyset^* =$$

Positive closure of a language

The *positive closure* of a language L , written L^+ , is defined as follows.

$$L^+ = \bigcup_{i \in \mathcal{N}} L^{i+1}.$$

For all languages L , $L^+ \cup \{\wedge\} =$

If $\wedge \in L$, then $L^+ = L^*$?

$$L^* L^* =$$

$$(L^*)^* =$$

$$L^+ L^+ =$$

$$(L^+)^+ =$$

$$(L^*)^+ =$$

Counting strings

How many strings of length k over alphabet A ?

How many strings of length 5 over $\{a, b, c, d\}$ that end with a or b ?

How many strings of length 5 over $\{a, b, c, d\}$ that end with a or b and contain at least one c ?

How many strings of length 5 over $\{a, b, c, d\}$ contain at least one c and at least one d ?

We can begin by subtracting from $\{a, b, c, d\}^5$ the strings that either lack c or lack d .

$$\begin{aligned} & |\{a, b, c, d\}^5 - (\{a, b, d\}^5 \cup \{a, b, c\}^5)| \\ &= |\{a, b, c, d\}^5| - |\{a, b, d\}^5 \cup \{a, b, c\}^5| \\ &= 4^5 - |\{a, b, d\}^5 \cup \{a, b, c\}^5| \\ &\quad \text{(and because } |A \cup B| = |A| + |B| - |A \cap B|) \\ &= 4^5 - (|\{a, b, d\}^5| + |\{a, b, c\}^5| - |\{a, b\}^5|) \\ &= 4^5 - (3^5 + 3^5 - 2^5) \\ &= 4^5 - 2(3^5) + 2^5 \end{aligned}$$