Theory of Formal Languages Introduction

Prof. Licia Sbattella

aa 2007-08

Translated and adapted by L. Breveglieri

BASICS / 1

ALPHABET: finite set of elements

- cardinality
- string, word, phrase

STRING: ordered set of atomic elements, possibly repeated.

LANGUAGE: finite or infinite set of strings.

The set-theoretic structure of a language has two levels.

LANGUAGE: unordered set of non-atomic elements that are in turn ordered sets of atomic elements:

- cardinality of L
- finite or infinite language

$$\sum = \{a_1, a_2, ..., a_k\}$$

$$\left|\sum\right| = k$$

$$\sum = \{a, b, c\}$$

$$abc, aabc, ac, bbb$$

$$\sum = \{a, b, c\}$$

$$L_1 = \{a\,b, a\,c\}$$

$$L_2 = \{bc, bbc\}$$

$$L_1 = \{a\,bc, aabbcc, abcabc, ...\}$$

$$|L_2| = |\{bc, bbc\}| = 2 \qquad |\varnothing| = 0 \qquad |bbc|_b = 2, |bbc|_a = 0$$

BASICS / 2

LENGTH OF A STRING x: |x|
is the number of elements (letters).

$$|bbc| = 3$$
$$|abbc| = 4$$

EQUALITY OF TWO STRINGS : two strings are equal if and only if (iff)

- they have the same length
- their elements orderly coincide, from left to right

$$x = a_1 a_2 \ldots a_h, \; y = b_1 b_2 \ldots b_k$$

$x = y$ if $h = k$

$a_i = b_i$ with $i = 1, \ldots h;$

$$bbc \neq bcb \neq bc$$

OPERATIONS ON STRINGS / 1

CONCATENATION (product of strings):

- is a basic opertion
- is associative
- changes the length

$$x = a_1 a_2 \dots a_h ,\ y = b_1 b_2 \dots b_k$$

$$x.y = a_1 a_2 \dots a_h b_1 b_2 \dots b_k = xy$$

$$(xy)z = x(yz)$$

$$|xyz| = |x| + |y| + |z|$$

$$x\varepsilon = \varepsilon x = x$$

$$|\varepsilon| = 0$$

EMPTY STRING (or NULL string):
ε is the neutral element with respect to concatenation: chaining ε on the left or right does not change the string

Pay attention: ε is NOT the same as Φ (the empty set) !

SUBSTRING: x = u y v

- y is a substring
- u is a prefix
- v is a suffix

Proper substring: y if u, v ≠ ε

Start $_k$ (x) = k : x

$x = abccbc$

p prefix $a, ab, abc, abcc, abccb, abccbc$

su suffix $c, bc, cbc, ccbc, bccbc, abccbc$

sc substring $\dots\dots, bc, cc, cb, \dots$

OPERATIONS ON STRINGS / 2

MIRRORING or REFLECTION

$$x = atri \qquad x^R = irta$$
$$x = bon \qquad y = ton$$
$$xy = bonton$$
$$(xy)^R = y^R x^R = notnob$$

$$x = a_1 a_2 \ldots a_h$$
$$x^R = a_h a_{h-1} \ldots a_2 a_1$$
$$(x^R)^R = x$$
$$(xy)^R = y^R x^R$$
$$\varepsilon^R = \varepsilon$$

REPETITION (or ITERATION): the *m*-th power of a string (where *m* is greater than or equal to 1) consists of concatenating the string to itself for *m* – 1 times.

$$x^m = \underbrace{xxx \ldots x}_{1\,2\,3\,\ldots\;m}$$
$$x^m = x^{m-1}x, \quad m > 0$$
$$x^0 = \varepsilon$$

$$x = ab \quad x^0 = \varepsilon \quad x^1 = x = ab \quad x^2 = (ab)^2 = abab$$
$$y = a^3 = aaa \quad y^3 = a^3a^3a^3 = a^9$$
$$\varepsilon^0 = \varepsilon \quad \varepsilon^2 = \varepsilon$$

PRECEDENCE AMONG OPERATORS:

- power precedes concatenation
- mirroring precedes concatenation

$$ab^2 = abb \qquad (ab)^2 = abab$$
$$ab^R = ab \qquad (ab)^R = ba$$

OPERATIONS ON LANGUAGES / 1

An operation defined on a language applies to each string in the language (and need be definable over any string).

$$L^R = \{x \mid x = y^R \wedge y \in L\}$$

characteristic predicate

$$\text{prefix}(L) = \{y \mid x = yz \wedge x \in L \wedge y, z \neq \varepsilon\}$$

PREFIX-FREE LANGUAGE: there is not any string in the language that is a prefix of another string in the language.

Equivalently, prefix(L) and L are disjoint sets (i.e. prefix(L) $\cap$ L $= \Phi$).

$$L_1 = \{x \mid x = a^n b^n \wedge n \geq 1\} \quad a^2b^2 \in L_1 \;\; a^2 b \notin L_1$$

L_1 is prefix free   prefixes are $a^n b^m$ where $n > m \geq 0$

$$L_2 = \{a^m b^n \mid m > n \geq 1\} \quad a^4 b^3 \in L_2 \;\; a^4 b^2 \in L_2$$

L_2 is not prefix-free

Caution: ε is prefix (or suffix, or substing) to any other string, including itself.

OPERATIONS ON LANGUAGES / 2
binary (two arguments) operations

CONCATENATION:

$$\boxed{L'L'' = \{xy \mid x \in L' \wedge y \in L''\}}$$

m-th POWER ($m \geq 0$)

$$L^m = L^{m-1}L, m > 0$$
$$L^0 = \{\varepsilon\}$$

Pay attention to the following consequences:

$$\varnothing^0 = \{\varepsilon\} \quad L.\varnothing = \varnothing.L = \varnothing \quad L.\{\varepsilon\} = \{\varepsilon\}.L = L$$

OPERATIONS ON LANGUAGES / 3

EXAMPLES:

$$L_1 = \{a^i \mid i \geq 0, \text{ even}\} = \{\varepsilon, a^2, a^4, a^6, \ldots\}$$

$$L_2 = \{b^j a \mid j \geq 1, \text{ odd}\} = \{ba, b^3a, b^5a, \ldots\}$$

$$L_1L_2 = \{a^i b^j a \mid (i \geq 0, \text{ even}) \wedge (j \geq 1, \text{ odd})\}$$

$$= \{\varepsilon ba, a^2ba, a^4ba, \ldots \varepsilon b^3a, a^2b^3a, \ldots\}$$

$$(L_1)^2 = \{\varepsilon, a^2, a^4, a^6, \ldots\}\{\varepsilon, a^2, a^4, a^6, \ldots\} =$$

$$= \{\varepsilon, \varepsilon a^2, \varepsilon a^4, \ldots, a^2\varepsilon, a^4, \ldots, a^4\varepsilon, a^6 \ldots\} = L_1$$

For every pair of even integers h and k, $h + k$ is even and a^{h+k} belongs to L_1.

CAUTION:

$$\{x \mid x = y^m \wedge y \in L\} \subset L^m$$

$$m = 2 \quad L_1 = \{a, b\}$$

$$\{a^2, b^2\} \subset L_1^2 = \{a^2, ab, ba, b^2\}$$

OPERATIONS ON LANGUAGES / 4

STRINGS OF FINITE LENGTH: the power operator allows to define in an expressive way the language of the strings having length not greater than (= less than or equal to) a given fixed integer K.

$$\boxed{\begin{array}{l} L = \{\varepsilon, a, b\}^3 \quad k = 3 \\ L = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots bbb\} \end{array}}$$

Notice the role of ε, that allows to obtain all the strings of length 0, 1, 2.

$$\boxed{\begin{array}{c} \{\varepsilon, a, b\} \\ \{\varepsilon, a, b\} \\ \{\varepsilon, a, b\} \end{array}}$$

And, in order to exclude the empty string ε, do as follows :

$$\boxed{L = \{a, b\}\{\varepsilon, a, b\}^2}$$

OPERATIONS ON LANGUAGES / 5

SET-THEORETIC OPERATIONS: these are the traditional operations of elementary set theory: union $\cup$, intersection $\cap$, complement $\neg$ (or overlining $\overline{\quad}$) and the traditional relational operators between sets: strict inclusion $\subset$, inclusion $\subseteq$, equality $=$, inequality $\neq$, etc

UNIVERSAL LANGUAGE: the set of ALL the strings defined over the alphabet Σ, of any length (including also length 0).
Also sometimes called the FREE MONOID.

$$L_{universal} = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$$
$$L_{universal} = \neg\varnothing$$

COMPLEMENT of a language L over the alphabet Σ: it si defined as the set-theoretic difference with respect to the universal language over Σ.
Equivalently, it is the set of all the strings over the alphabet Σ that do not belong to L.

$$\neg L = L_{universal} \setminus L$$

sometimes written
$L_{univ} - L$

OPERATIONS ON LANGUAGES / 6

EXAMPLES

The complement of a finite language is always an infinite language.

$$\neg\left(\{a,b\}^2\right) = \varepsilon \cup \{a,b\} \cup \{a,b\}^3 \cup \ldots$$

The complement of an infinite language may be infinite, but need not be always such (sometimes happens to be finite).

$$L = \left\{a^{2n} \mid n \geq 0\right\} \quad \neg \mathrm{L} = \left\{a^{2n+1} \mid n \geq 0\right\}$$

Set-theoretic difference:

$$\Sigma = \{a,b,c\}$$

$$L_1 = \left\{x \mid |x|_a = |x|_b = |x|_c \geq 0\right\}$$

$$L_2 = \left\{x \mid |x|_a = |x|_b \wedge |x|_c = 1\right\}$$

sometimes written $L_1 - L_2$

$$L_1 \setminus L_2 = \varepsilon \cup \left\{x \mid |x|_a = |x|_b = |x|_c \geq 2\right\}$$

$$L_2 \setminus L_1 = \left\{x \mid |x|_a = |x|_b \neq |x|_c = 1\right\}$$

OPERATIONS ON LANGUAGES / 7

In both natural and artificial languages, the phrases can be of any length.

But only formulae of finite length can be written to define a language.

It is necessary to introduce some operators to create infinitely many strings.

STAR OPERATOR (also called Kleene star or concatenation closure):
it is the limit of the power operator.

The union of all the powers of a language, for every positive or null exponent.

$$L^* = \bigcup_{h=0\ldots\infty} L^h = L^0 \cup L^1 \cup L^2 \ldots = \varepsilon \cup L^1 \cup L^2 \ldots$$

$$L = \{ab, ba\} \quad L^* = \{\varepsilon, ab, ba, abab, abba, baab, baba, \ldots\}$$

L is finite but L^* is infinite

Every string in the star language of L can be factored into substrings, each of which belongs to the language L.

Sometimes, the star language happens to be identical to the base language.

$$L = \{a^{2n} \mid n \geq 0\} \quad L^* = \{a^{2n} \mid n \geq 0\} \equiv L$$

OPERATIONS ON LANGUAGES / 8

If one takes the alphabet Σ as the base language, Σ^* contains all strings. (Σ^* is the universal language over the alphabet Σ). One may signify that L is a language over the alphabet Σ by writing as follows:

$$L \subseteq \Sigma^*$$

PROPERTIES OF THE STAR OPERATOR:

- monotonic
- closed w.r.t. concatenation
- idempotent
- commutes with mirroring

$$L \subseteq L^*$$

if $\left(x \in L^* \wedge y \in L^*\right)$ then $xy \in L^*$

$$\left(L^*\right)^* = L^*$$

$$\left(L^*\right)^R = \left(L^R\right)^*$$

Moreover:

$$\varnothing^* = \{\varepsilon\} \qquad \{\varepsilon\}^* = \{\varepsilon\}$$

Example (idempotence):

$$L_1 = \left\{a^{2n} \mid n \geq 0\right\} \qquad L_1^* = L_1$$

idempotence and $L^* = \{aa\}^*$

OPERATIONS ON LANGUAGES / 9

Example of star operator: an identifier, modeled as a string of letters and digits (alphanumeric), of arbitrary length (not null), but starting with a letter (not with a digit).

$$\Sigma_A = \{A, B, \ldots, Z\} \quad \Sigma_N = \{0, 1, 2, \ldots, 9\}$$

$$I = \Sigma_A (\Sigma_A \cup \Sigma_N)^*$$

if $\Sigma = \Sigma_A \cup \Sigma_N$

$$I_5 = \Sigma_A (\Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4)$$

$$I_5 = \Sigma_A (\Sigma \cup \varepsilon)^4$$

The C language would admit the underscore “_” as well, but not as the starting symbol. Extend the definition (do it yourself).

OPERATIONS ON LANGUAGES / 10

CROSS OPERATOR (also called Kleene cross or ε-free concatenation closure): is the non-reflexive closure with respect to concatenation (see below).

The unitory does not contain the null power.

Sometimes very useful, but not indispensable.

$$L^+ = \bigcup_{h=1\ldots\infty} L^h = L^1 \cup L^2 \cup \ldots$$

$$\{ab, bb\}^+ = \{ab, bb, ab^3, b^2ab, abab, b^4, \ldots\}$$

$$\{\varepsilon, aa\}^+ = \{\varepsilon, a^2, a^4, \ldots\} = \{a^{2n} \mid n \geq 0\}$$

The same language can be defined in different ways by different combinations of the same or other operators.

Example: the strings of length greater than or equal to 4:

$$\Sigma^4 \Sigma^*$$

$$(\Sigma^+)^4$$

OPERATIONS ON LANGUAGES / 11

QUOTIENT OPERATOR: it shortens the phrases of a language L’,
by stripping off a suffix out of another language L’’.

$$\boxed{L = L' / L'' = \left\{ y \mid (x = yz \in L') \wedge z \in L'' \right\}}$$

Example of quotienting:

$$L' = \left\{ a^{2n} b^{2n} \mid n > 0 \right\}, \quad L'' = \left\{ b^{2n+1} \mid n \geq 0 \right\}$$

$$L' / L'' = \left\{ a^r b^s \mid (r \geq 2 \ \ even) \wedge (1 \leq s < r, s \quad odd\,) \right\}$$

$$= \left\{ a^2 b, a^4 b, a^4 b^3, \ldots \right\}$$

$$L'' / L' = \emptyset$$

Question: what happens if x $\in$ L’ does not admit any suffix z $\in$ L’’ ?

Bibliography

– S. Crespi Reghizzi, *Linguaggi Formali e Compilazione*, Città Studi, UTET

– Hopcroft, Ullman, *Formal Languages and their Relation to Automata*, Addison Wesley, 1969

– A. Salomaa – *Formal Languages*, Academic Press, 1973

– D. Mandrioli, C. Ghezzi – *Theoretical Foundations of Computer Science*, John Wiley & Sons, 1987

– L. Breveglieri, S. Crespi Reghizzi, *Linguaggi Formali e Compilatori: Temi d'Esame Risolti,* web site