Formal Languages and Linguistics

Pascal Amsili

Sorbonne Nouvelle, Lattice (CNRS/ENS-PSL/USN)

PSL Master's degree in Cognitive Science, February 2025

General introduction

1. Mathematicians (incl. Chomsky) have formalized the notion of **language**

   oversimplification ?
   maybe...
2. It buys us:
   2.1 Tools to think about theoretical issues about language/s (expressiveness, complexity, comparability...)
   2.2 Tools to manipulate concretely language (e.g. with computers)
   2.3 A research programme:
      - Represent the syntax of natural language in a fully unambiguously specified way

Now let's get familiar with the mathematical notion of language

Alphabet, word

Def. 1 (Alphabet)

An *alphabet* Σ is a finite set of symbols (letters).
The *size* of the alphabet is the cardinal of the set.

Def. 2 (Word)

A *word* on the alphabet Σ is a finite sequence of letters from Σ.
Formally, let $[p] = (1, 2, 3, 4, ..., p)$ (ordered integer sequence).
Then a word is a *mapping*

$$u : [p] \longrightarrow \Sigma$$

p, the length of u, is noted $|u|$.

Examples I

Alphabet {·, —}

Words ———·

·

··—·

...

Alphabet {·— , —··· , —·—· , —·· , · , ... }

Words ··· ——— ···

—·· ·· —·· · ·—· ——— —

·— —··· —··· ·— ·— —··· —··· ·—

...

Examples II

Alphabet	{0,1,2,3,4,5,6,7,8,9, · }
Words	235 · 29
	007 · 12
	·1 · 1 · 00 · ·
	~~3 · 1415962 . . .~~ (π)
	. . .
Alphabet	{a, woman, loves, man }
Words	a
	a woman loves a woman
	man man a loves woman loves a
	. . .

Monoid

Def. 3 (Σ^*)

Let Σ be an alphabet.
The set of all the words that can be formed with any number of letters from Σ is noted Σ^*

Σ^* includes a word with no letter, noted ε

Example: $\Sigma = \{a, b, c\}$
$\Sigma^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, \ldots, bbb, \ldots\}$

N.B.: Σ^* is always infinite, except. . .
if $\Sigma = \emptyset$. Then $\Sigma^* = \{\varepsilon\}$.

Structure of Σ^*

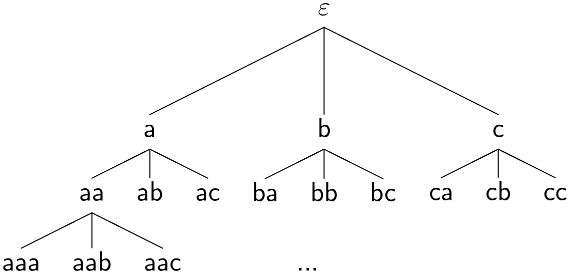
Let k be the size of the alphabet $k = |\Sigma|$.

Then Σ^* contains :

$k^0 = 1$	word(s) of 0 letters (ε)
$k^1 = k$	word(s) of 1 letters
k^2	word(s) of 2 letters
...	
k^n	words of n letters, $\forall n \geq 0$

Representation of Σ^*

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



- Words can be enumerated according to different orders
- Σ^* is a *countable* set

Concatenation

Σ^* can be equipped with a binary operation: *concatenation*

Def. 4 (Concatenation)

Let $[p] \xrightarrow{u} \Sigma$, $[q] \xrightarrow{w} \Sigma$. The concatenation of u and w, noted uw ($u.w$) is thus defined:

$$uw : \quad [p+q] \longrightarrow \Sigma$$
$$uw_i = \begin{cases} u_i & \text{for} \quad i \in [1,p] \\ w_{i-p} & \text{for} \quad i \in [p+1, p+q] \end{cases}$$

Example :	u	bacba
	v	cca
	uv	bacbacca

Factor

Def. 5 (Factor)

A *factor* w of u is a subset of adjascent letters in u.

–w is a factor of u $\Leftrightarrow$ $\exists u_1, u_2$ s.t. $u = u_1 w u_2$
–w is a left factor (*prefix*) of u $\Leftrightarrow$ $\exists u_2$ s.t. $u = w u_2$
–w is a right factor (*suffix*) of u $\Leftrightarrow$ $\exists u_1$ s.t. $u = u_1 w$

Def. 6 (Factorization)

We call *factorization* the decomposition of a word into factors.

Role of concatenation

1. Words have been defined on Σ.
   Given any two words, it's always possible to form a new word by concatenating them.
2. Any word can be factorised in many different ways:
   $a\,b\,a\,c\,c\,a\,b$
   $(a\,b\,a)(c\,c\,a\,b)(a\,b)(a\,c\,c)(a\,b)(a\,b\,a\,c\,c)(a\,b)(a)(b)(a)(c)(c)(a)(b)$
3. Since all letters of Σ form a word of length 1
   (this set of words is called the *base*),
4. Any word of Σ^* can be seen as a (unique) sequence of concatenations of length 1 words :
   $a\,b\,a\,c\,c\,a\,b$
   $((((((ab)a)c)c)a)b)$
   $((((((a.b).a).c).c).a).b)$

Properties of concatenation

1. Concatenation is non commutative
2. Concatenation is associative
3. Concatenation has an identity (neutral) element: ε

1. $uv.w \neq w.uv$
2. $(u.v).w = u.(v.w)$
3. $u.\varepsilon = \varepsilon.u = u$

Notation : $a.a.a = a^3$

Language

Def. 7 (Formal Language)

Let Σ be an alphabet.
A language on Σ is a set of words on Σ.
or, equivalently,
A language on Σ is a subset of Σ^*

Examples I

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

$L_1 = \{aa, ab, bac\}$	finite language
$L_2 = \{a, aa, aaa, aaaa \ldots\}$ or $L_2 = \{a^i \ / \ i \geq 1\}$	infinite language
$L_3 = \{\varepsilon\}$	finite language, reduced to a singleton
	$\neq$
$L_4 = \emptyset$	"empty" language
$L_5 = \Sigma^*$	

Sorbonne Nouvelle

Examples II

Let $\Sigma = \{\text{a, man, loves, woman}\}$.

$L = \{$ a man loves a woman, a woman loves a man $\}$

Let $\Sigma' = \{\text{a, man, who, saw, fell}\}$.

$$L' = \left\{ \begin{array}{l} \text{a man fell,} \\ \text{a man who saw a man fell,} \\ \text{a man who saw a man who saw a man fell,} \\ \dots \end{array} \right\}$$

Set operations

Since a language is a set, usual set operations can be defined:

- union
- intersection
- set difference

$\Rightarrow$ One may describe a (complex) language as the result of set operations on (~~simpler~~) languages:
$\{a^{2k} \; / \; k \geqslant 1\} = \{a, aa, aaa, aaaa, \ldots\} \cap \{ww \; / \; w \in \Sigma^*\}$

Additional operations

Def. 8 (product operation on languages)

One can define the *language product* and its closure *the Kleene star* operation:

- The *product* of languages is thus defined:

$$L_1.L_2 = \{uv \,/\, u \in L_1 \,\&\, v \in L_2\}$$

  Notation: $\overbrace{L.L.L\ldots L}^{k \text{ times}} = L^k \; ; \; L^0 = \{\varepsilon\}$

- The Kleene star of a language is thus defined:

$$L^* = \bigcup_{n \geqslant 0} L^n$$