Formal Languages and Linguistics

Pascal Amsili

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

Cogmaster, september 2023

Overview

Formal Languages

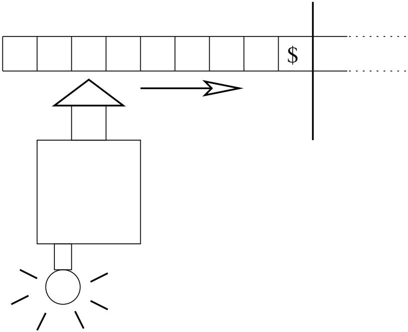
Regular Languages
- Automata
- Properties
- Regular expressions
- Definition

Formal Grammars

Formal complexity of Natural Languages

Sorbonne Nouvelle

Metaphoric definition



Formal definition

Def. 9 (Finite deterministic automaton (FDA))

A finite state deterministic automaton $\mathcal{A}$ is defined by :

$$\mathcal{A} = \langle Q, \Sigma, q_0, F, \delta \rangle$$

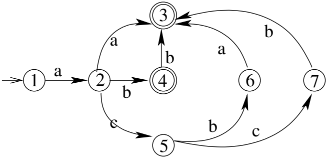
- Q is a finite set of states
- Σ is an alphabet
- q_0 is a distinguished state, the initial state,
- F is a subset of Q, whose members are called final/terminal states
- δ is a mapping **fonction** from $Q \times \Sigma$ to Q. Notation $\delta(q, a) = r$.

Sorbonne Nouvelle

Example

Let us consider the (finite) language $\{aa, ab, abb, acba, accb\}$.
The following automaton recognizes this langage: $\langle Q, \Sigma, q_0, F, \delta \rangle$, avec $Q = \{1, 2, 3, 4, 5, 6, 7\}$, $\Sigma = \{a, b, c\}$, $q_0 = 1$, $F = \{3, 4\}$, and δ is thus defined:

$$
\delta : \begin{array}{l}
(1,a) \mapsto 2 \\
(2,a) \mapsto 3 \\
(2,b) \mapsto 4 \\
(2,c) \mapsto 5 \\
(4,b) \mapsto 3 \\
(5,b) \mapsto 6 \\
(5,c) \mapsto 7 \\
(6,a) \mapsto 3 \\
(7,b) \mapsto 3
\end{array}
$$



	a	b	c
$\rightarrow$ 1	2		
2	3	4	5
$\leftarrow$ 3			
$\leftarrow$ 4		3	
5		6	7
6	3		
7		3	

Sorbonne Nouvelle

Recognition

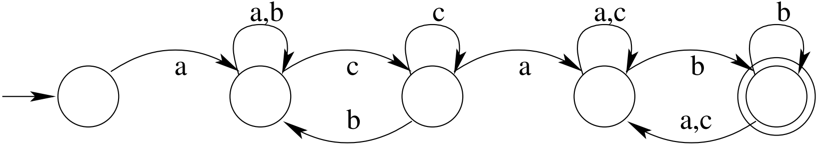
Recognition is defined as the existence of a sequence of states defined in the following way. Such a sequence is called a path in the automaton.

Def. 10 (Recognition)

A word $a_1a_2...a_n$ is **recognized/accepted** by an automaton iff there exists a sequence $k_0, k_1, ..., k_n$ of states such that:

$$k_0 = q_0$$
$$k_n \in F$$
$$\forall i \in [1, n], \;\; \delta(k_{i-1}, a_i) = k_i$$

Example

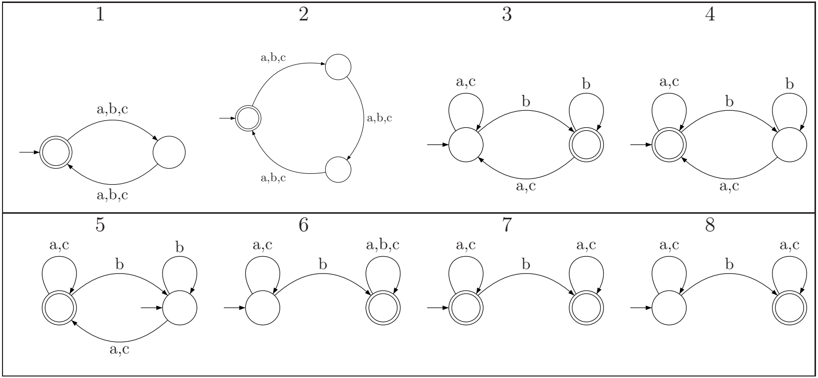


Exercices

Let $\Sigma = \{a, b, c\}$. Give deterministic finite state automata that accept the following languages:

1. The set of words with an even length.
2. The set of words where the number of occurrences of b is divisible by 3.
3. The set of words ending with a b.
4. The set of words not ending with a b.
5. The set of words non empty not ending with a b.
6. The set of words comprising at least a b.
7. The set of words comprising at most a b.
8. The set of words comprising exactly one b.

Answers


1
2
3
4
a,b,c
a,b,c
a,b,c
a,b,c
a,b,c
a,c
b
b
a,c
a,c
b
b
a,c
5
6
7
8
a,c
b
b
a,c
a,c
b
a,b,c
a,c
b
a,c
a,c
b
a,c

Overview

Formal Languages

Regular Languages

- Automata
- Properties
- Regular expressions
- Definition

Formal Grammars

Formal complexity of Natural Languages

Ways of non-determinism

A word is recognized if there exists a path in the automaton. It is not excluded however that there be several paths for one word: in that case, the automaton is non deterministic.
What are the sources of non determinism?

- $\delta(a, S_1) = \{S_2, S_3\}$
- "spontaneous transition" = ε-transition

Equivalence theorems

For any non-deterministic automaton, it is possible to design a complete deterministic automaton that recognizes the same language.
Proofs: algorithms (constructive proofs)
First “remove” ε-transitions, then “remove” multiple transitions.

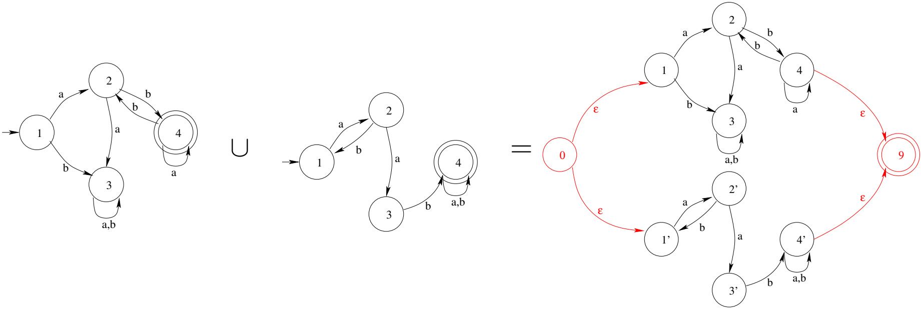
Closure (1)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

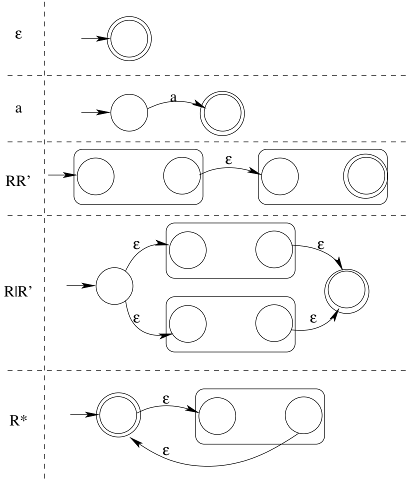
- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)

(rational operations)

Union of regular languages: an example



Rational operations



Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star) *(rational operations)*
  $\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L

Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)
  (rational operations)
  $\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L and vice-versa

Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)
  (rational operations)
  $\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L and vice-versa
- $L \cap L'$ (intersection); $\overline{L}$ (complement)
- ...

Sorbonne Nouvelle

Intersection of regular languages

Algorithmic proof
Deterministic complete automata

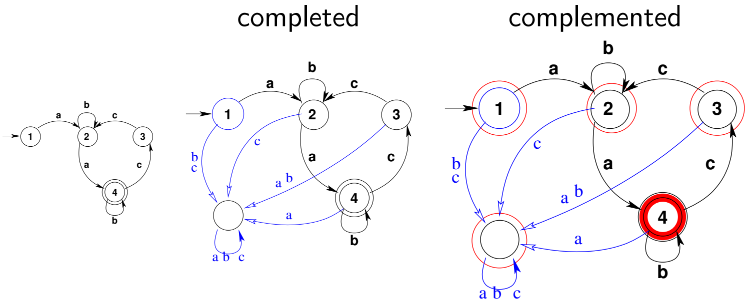
L_1	a	b
$\rightarrow$ 1	2	4
2	4	3
$\leftarrow$ 3	3	3
4	4	4

L_2	a	b
$\leftrightarrow$ 1	2	5
2	5	3
3	4	5
4	1	4
5	5	5

$L_1 \cap L_2$	a	b
$\rightarrow$ (1,1)	(2,2)	(4,5)
(2,2)	(4,5)	(3,3)
(4,5)	(4,5)	(4,5)
(3,3)	(3,4)	(3,5)
(3,4)	(3,1)	(3,4)
$\leftarrow$ (3,1)	(3,2)	(3,4)
(3,2)	(3,4)	(3,3)
(3,5)	(3,5)	(3,5)

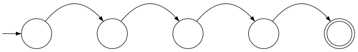
Complement of a regular language

Deterministic complete automata



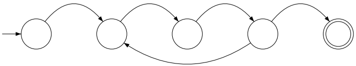
Pumping lemma (intuition)

Take an automaton A with k states.
If $\mathcal{L}(A)$ is infinite,
then $\exists w \in \mathcal{L}(A), |w| \geq k$.
Therefore, when accepting w, A goes through some state q at least twice.
That means that there is a loop $q \stackrel{w_{i:j}}{\rightarrow} q$.
Repeating the loop any number of times (even 0) always produces a word ($w_{1:i-1}\, {w_{i:j}}^n\, w_{j+1:|w|}$) in $\mathcal{L}(A)$.



Pumping lemma (intuition)

Take an automaton A with k states.
If $\mathcal{L}(A)$ is infinite,
then $\exists w \in \mathcal{L}(A), |w| \geq k$.
Therefore, when accepting w, A goes through some state q at least twice.
That means that there is a loop $q \overset{w_{i:j}}{\rightarrow} q$.
Repeating the loop any number of times (even 0) always produces a word ($w_{1:i-1}\, {w_{i:j}}^n\, w_{j+1:|w|}$) in $\mathcal{L}(A)$.



Pumping lemma (definition)

Pumping Lemma

Let L be a regular language.
$\exists k \in \mathbb{N}$ such that
$\forall w \in L$ such that $|w| \geq k$,
$\exists x, u, y$ such that $w = xuy$ and that

1. $|u| \geq 1$;
2. $|xu| \leq k$;
3. $\forall n \in \mathbb{N}$, $xu^n y \in L$.

$\rightarrow$ "L has the pumping property."

Is NL regular? Pumping lemma (example I)

a^*bc (i.e. $\{a^n bc \mid n \in \mathbb{N}\}$) is regular (there is a DFA).
So, it must have the pumping property.

It happens that $k = 3$ works.
For example, $w = abc \in L$ is long enough and can be decomposed:

$$\underbrace{\epsilon}_{x} \quad \underbrace{a}_{u} \quad \underbrace{b \quad c}_{y}$$

1. $|u| \geq 1$ ($u = a$);
2. $|xu| \leq k$ ($xu = a$);
3. $\forall n \in \mathbb{N}$, $xu^n y$ (i.e. $a^n bc$) belongs to the language.

Pumping lemma (consequences)

regular	$\Rightarrow$	pumping property satisfied
pumping property **NOT** satisfied	$\Rightarrow$	**NOT** regular
pumping property satisfied	$\not\Rightarrow$	regular

To prove that L is

regular provide a DFA;

not regular show that the pumping property is not satisfied.

Sorbonne Nouvelle

Pumping lemma (example II)

Let's show that $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular.

- Consider any $k \in \mathbb{N}$.
- Consider $w = a^k b^k \in L$ ($|w| \geq k$).
- If $w = xuy$ with $|u| \geq 1$ and $|xu| \leq k$, then u contains no b.
- But then, $xu^0y = xy \notin L$ (strictly less as than bs).
- So no $k \in \mathbb{N}$ works; L does not have the pumping property.

A similar reasoning applies to $\{xu^n yv^n z \mid x, y, z, u, v \in \Sigma^*\}$.

Overview

Formal Languages

Regular Languages
- Automata
- Properties
- Regular expressions
- Definition

Formal Grammars

Formal complexity of Natural Languages

Regular expressions

It is common to use the 3 *rational* operations:

- union
- product
- Kleene star

to characterize certain languages...

Regular expressions

It is common to use the 3 *rational* operations:

- union
- product
- Kleene star

to characterize certain languages...

$(\{a\} \cup \{b\})^*.\{c\} = \{c, ac, abc, bc, \ldots, baabaac, \ldots\}$
(simplified notation $(a|b)^*c$ — regular expressions)

Sorbonne Nouvelle

Regular expressions

It is common to use the 3 *rational* operations:

- union
- product
- Kleene star

to characterize certain languages...

$(\{a\} \cup \{b\})^*.\{c\} = \{c, ac, abc, bc, \ldots, baabaac, \ldots\}$
(simplified notation $(a|b)^*c$ — regular expressions)

... but not all languages can be thus characterized.

Def. 11 (Rational Language)

A rational language on Σ is a subset of Σ^* inductively defined thus:

- $\emptyset$ and $\{\varepsilon\}$ are rational languages ;
- for all $a \in X$, the singleton $\{a\}$ is a rational language ;
- for all g and h rational, the sets $g \cup h$, $g.h$ and g^* are rational languages.

Results: expressivity

- Any finite langage is regular
- $a^n b^m$ is regular
- $a^n b^n$ is not regular
- ww^R is not regular (^{R} : reverse word)

Decidable problems

- The "word problem" $w \overset{?}{\in} L(\mathcal{A})$ is decidable.

$\Rightarrow$ A computation on an automaton always stops.

Decidable problems

- The "word problem" $w \overset{?}{\in} L(\mathcal{A})$ is decidable.

$\Rightarrow$ A computation on an automaton always stops.

- The "emptiness problem" $L(\mathcal{A}) \overset{?}{=} \emptyset$ is decidable.

$\Rightarrow$ It's enough to test all possible words of length $\leq k$, where k is the number of states.

Decidable problems

- The “word problem” $w \overset{?}{\in} L(\mathcal{A})$ is decidable.

⇒ A computation on an automaton always stops.

- The “emptiness problem” $L(\mathcal{A}) \overset{?}{=} \emptyset$ is decidable.

⇒ It’s enough to test all possible words of length $\leq k$, where k is the number of states.

- The “finiteness problem” $L(\mathcal{A})$ $\overset{?}{\text{is}}$ *finite* is decidable.

⇒ Test all possible words whose length is between k and $2k$. If there exists u s.t. $k < |u| < 2k$ and $u \in L(\mathcal{A})$, then $L(\mathcal{A})$ is infinite.

Decidable problems

- The "word problem" $w \stackrel{?}{\in} L(\mathcal{A})$ is decidable.

$\Rightarrow$ A computation on an automaton always stops.

- The "emptiness problem" $L(\mathcal{A}) \stackrel{?}{=} \emptyset$ is decidable.

$\Rightarrow$ It's enough to test all possible words of length $\leq k$, where k is the number of states.

- The "finiteness problem" $L(\mathcal{A}) \stackrel{?}{\text{is}}$ *finite* is decidable.

$\Rightarrow$ Test all possible words whose length is between k and $2k$. If there exists u s.t. $k < |u| < 2k$ and $u \in L(\mathcal{A})$, then $L(\mathcal{A})$ is infinite.

- The "equivalence problem" $L(\mathcal{A}) \stackrel{?}{=} L(\mathcal{A}')$ is decidable.

$\Rightarrow$ it boils down to answering the question:
$\Big(L(\mathcal{A}) \cap \overline{L(\mathcal{A}')}\Big) \cup \Big(L(\mathcal{A}') \cap \overline{L(\mathcal{A})}\Big) = \emptyset$

À quoi ça sert?

Why would you want to define (formally) a language?

- to formulate a request to a search engine (`mang.*`)
- to associate actions to (classes of) words (e.g., transducers)
  - formal languages (math. expressions, programming languages...)
  - artificial (interface) languages
  - (subpart of) natural languages

Overview

Formal Languages

Regular Languages
- Automata
- Properties
- Regular expressions
- Definition

Formal Grammars

Formal complexity of Natural Languages

Definition

3 possible definitions

1. a regular language can be defined by rational/regular expressions
2. a regular language can be recognized by a finite automaton
3. a regular language can be generated by a regular grammar

Overview

Formal Languages

Regular Languages

Formal Grammars
- Definition
- Language classes

Formal complexity of Natural Languages

Formal grammar

Def. 12 ((Formal) Grammar)

A **formal grammar** is defined by $\langle \Sigma, N, S, P \rangle$ where

- Σ is an alphabet
- N is a disjoint alphabet (non-terminal vocabulary)
- $S \in V$ is a distinguished element of N, called the *axiom*
- P is a set of « *production rules* », namely a subset of the cartesian product $(\Sigma \cup N)^* N (\Sigma \cup N)^* \times (\Sigma \cup N)^*$.

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \Bigg\langle$$

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \left\langle \{joe, sam, sleeps\}, \right.$$

Sorbonne Nouvelle

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \left\langle \{joe, sam, sleeps\}, \{N, V, S\}, \right.$$

Sorbonne Nouvelle

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \left\langle \{joe, sam, sleeps\}, \{N, V, S\}, S, \right.$$

Sorbonne Nouvelle

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \left\langle \{joe, sam, sleeps\}, \{N, V, S\}, S, \left\{ \begin{array}{l} (N, joe) \\ (N, sam) \\ (V, sleeps) \\ (S, N\ V) \end{array} \right\} \right\rangle \}$$

Sorbonne Nouvelle

Examples

$$\langle \Sigma, N, S, P \rangle$$

$$\mathcal{G}_0 = \left\langle \{joe, sam, sleeps\}, \{N, V, S\}, S, \left\{ \begin{array}{l} N \rightarrow joe \\ N \rightarrow sam \\ V \rightarrow sleeps \\ S \rightarrow N\ V \end{array} \right\} \right\rangle \}$$

Examples (cont'd)

$$\mathcal{G}_1 = \left\langle \{jean, dort\}, \{Np, SN, SV, V, S\}, S, \left\{ \begin{array}{l} S \rightarrow SN\ SV \\ SN \rightarrow Np \\ SV \rightarrow V \\ Np \rightarrow jean \\ V \rightarrow dort \end{array} \right\} \right\rangle\}$$

$$\mathcal{G}_2 = \langle \{(,)\}, \{S\}, S, \{S \longrightarrow \varepsilon \,|\, (S)S\} \rangle$$

Notation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

Notation

$$
\begin{array}{llll}
\mathcal{G}_3 : & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

$$\mathcal{G}_3 = \langle \{+, \times, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{E, F\}, E, \{\ldots\} \rangle$$

Notation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0 \,|\, 1 \,|\, 2 \,|\, 3 \,|\, 4 \,|\, 5 \,|\, 6 \,|\, 7 \,|\, 8 \,|\, 9
\end{array}
$$

$$\mathcal{G}_3 = \langle\{+, \times, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{E, F\}, E, \{\ldots\}\rangle$$

$$G_4 = E \rightarrow E + T \mid T, T \rightarrow T \times F \mid F, F \rightarrow (E) \mid a$$

Sorbonne Nouvelle

Immediate Derivation

Def. 13 (Immediate derivation)

Let $\mathcal{G} = \langle X, V, S, P \rangle$ a grammar, $(f, g) \in (X \cup V)^*$ two "words", $r \in P$ a production rule, such that $r : A \longrightarrow u$ $(u \in (X \cup V)^*)$.

- f derives into g (immediate derivation) with the rule r (noted $f \xrightarrow{r} g$) iff
  $\exists v, w$ s.t. $f = vAw$ and $g = vuw$
- f derives into g (immediate derivation) in the grammar $\mathcal{G}$ (noted $f \xrightarrow{\mathcal{G}} g$) iff
  $\exists r \in P$ s.t. $f \xrightarrow{r} g$.

Derivation

Def. 14 (Derivation)

$$f \xrightarrow{\mathcal{G}*} g \text{ if } \begin{cases} f = g & \text{or} \\ \exists f_0, f_1, f_2, ..., f_n \text{ s.t. } & f_0 = f \\ & f_n = g \\ & \forall i \in [1, n] : f_{i-1} \xrightarrow{\mathcal{G}} f_i \end{cases}$$

An example with $\mathcal{G}_0$:

N V joe N

Derivation

Def. 14 (Derivation)

$f \xrightarrow{\mathcal{G}*} g$ if $f = g$ or

$\exists f_0, f_1, f_2, ..., f_n$ s.t. $f_0 = f$

$f_n = g$

$\forall i \in [1, n] : f_{i-1} \xrightarrow{\mathcal{G}} f_i$

An example with $\mathcal{G}_0$:

$N\ V\ joe\ N \longrightarrow sam\ V\ joe\ N$

Derivation

Def. 14 (Derivation)

$$f \xrightarrow{\mathcal{G}*} g \text{ if } \begin{cases} f = g & \text{or} \\ \exists f_0, f_1, f_2, ..., f_n \text{ s.t. } & f_0 = f \\ & f_n = g \\ & \forall i \in [1, n] : f_{i-1} \xrightarrow{\mathcal{G}} f_i \end{cases}$$

An example with $\mathcal{G}_0$:

$N\ V\ joe\ N \longrightarrow sam\ V\ joe\ N \longrightarrow \ sam\ V\ joe\ joe$ or

Sorbonne Nouvelle

Derivation

Def. 14 (Derivation)

$$f \xrightarrow{\mathcal{G}*} g \text{ if } \begin{cases} f = g & \text{or} \\ \exists f_0, f_1, f_2, ..., f_n \text{ s.t. } & f_0 = f \\ & f_n = g \\ & \forall i \in [1, n] : f_{i-1} \xrightarrow{\mathcal{G}} f_i \end{cases}$$

An example with $\mathcal{G}_0$:

$N\ V\ joe\ N \longrightarrow sam\ V\ joe\ N \longrightarrow$ $sam\ V\ joe\ joe$ or
$sam\ V\ joe\ sam$ or

Sorbonne Nouvelle

Derivation

Def. 14 (Derivation)

$f \xrightarrow{\mathcal{G}*} g$ if $f = g$ or

$\exists f_0, f_1, f_2, ..., f_n$ s.t. $f_0 = f$

$f_n = g$

$\forall i \in [1, n] : f_{i-1} \xrightarrow{\mathcal{G}} f_i$

An example with $\mathcal{G}_0$:

N V joe N $\longrightarrow$ *sam V joe N* $\longrightarrow$ *sam V joe joe* or

sam V joe sam or

sam sleeps joe N or

...

Sorbonne Nouvelle

Endpoint of a derivation

$$
\begin{array}{lrcl}
\mathcal{G}_3 : & E & \longrightarrow & E + E \\
& & | & E \times E \\
& & | & (\ E\ ) \\
& & | & F \\
& F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$E \times E$

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E$$

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E$$

Sorbonne Nouvelle

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (E) \\
 & & | & F \\
 & F & \longrightarrow & 0 \,|\, 1 \,|\, 2 \,|\, 3 \,|\, 4 \,|\, 5 \,|\, 6 \,|\, 7 \,|\, 8 \,|\, 9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E)$$

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E)$$

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3 : & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0 \,|\, 1 \,|\, 2 \,|\, 3 \,|\, 4 \,|\, 5 \,|\, 6 \,|\, 7 \,|\, 8 \,|\, 9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E) \longrightarrow 3 \times (E + F)$$

Endpoint of a derivation

$$\begin{array}{llll} \mathcal{G}_3: & E & \longrightarrow & E + E \\ & & | & E \times E \\ & & | & (\ E\ ) \\ & & | & F \\ & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9 \end{array}$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E) \longrightarrow 3 \times (E + F) \longrightarrow 3 \times (E + 4)$$

Endpoint of a derivation

$$
\begin{array}{llll}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$
E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E) \longrightarrow 3 \times (E + F) \longrightarrow 3 \times (E + 4) \longrightarrow 3 \times (F + 4)
$$

Endpoint of a derivation

$$
\begin{array}{rrcl}
\mathcal{G}_3: & E & \longrightarrow & E + E \\
 & & | & E \times E \\
 & & | & (\ E\ ) \\
 & & | & F \\
 & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9
\end{array}
$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E) \longrightarrow$$
$$3 \times (E + F) \longrightarrow 3 \times (E + 4) \longrightarrow 3 \times (F + 4) \longrightarrow 3 \times (5 + 4)$$

Endpoint of a derivation

$$\begin{array}{llll} \mathcal{G}_3: & E & \longrightarrow & E + E \\ & & | & E \times E \\ & & | & (\ E\ ) \\ & & | & F \\ & F & \longrightarrow & 0\,|\,1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,6\,|\,7\,|\,8\,|\,9 \end{array}$$

An example with $\mathcal{G}_3$:

$$E \times E \longrightarrow F \times E \longrightarrow 3 \times E \longrightarrow 3 \times (E) \longrightarrow 3 \times (E + E) \longrightarrow$$
$$3 \times (E + F) \longrightarrow 3 \times (E + 4) \longrightarrow 3 \times (F + 4) \longrightarrow 3 \times (5 + 4) \longrightarrow\!|$$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$:

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \to (S)S$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \to (S)S \to ()S \to ()$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \to (S)S \to ()S \to ()$

as well as $((()))$, $()()()$, $((()()()))$...

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$

as well as $((()))$, $()()()$, $((()()()))\ldots$

but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$
as well as $((()))$, $()()()$, $((()()()))\ldots$
but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :
$)S(\rightarrow$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$

as well as $((()))$, $()()()$, $((()()()))\ldots$

but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :

$)S(\rightarrow)(S)S(\rightarrow$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$

as well as $((()))$, $()()()$, $((()()()))$...

but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :

$)S(\rightarrow)(S)S(\rightarrow)()S(\rightarrow$

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$
as well as $((()))$, $()()()$, $((()()()))\ldots$
but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :
$)S(\rightarrow)(S)S(\rightarrow)()S(\rightarrow)()($

Sorbonne Nouvelle

Engendered language

Def. 15 (Language engendered by a word)

Let $f \in (\Sigma \cup N)^*$.

$L_{\mathcal{G}}(f) = \{g \in X^* / f \xrightarrow{\mathcal{G}*} g\}$

Def. 16 (Language engendered by a grammar)

The *language engendered by a grammar* $\mathcal{G}$ is the set of words of Σ^* derived from the axiom.

$L_{\mathcal{G}} = L_{\mathcal{G}}(S)$

For instance $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$

as well as $((()))$, $()()()$, $((()()()))\ldots$

but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :

$)S(\rightarrow)(S)S(\rightarrow)()S(\rightarrow)()($

for there is no way to arrive at $)S($ starting with S.

Example

$$G_4 = E \rightarrow E + T \mid T, T \rightarrow T \times F \mid F, F \rightarrow (E) \mid a$$

$a + a$, $a + (a \times a)$, ...

Sorbonne Nouvelle

Proto-word

Def. 17 (Proto-word)

A proto-word (or proto-sentence) is a word on $(\Sigma \cup N)^* N (\Sigma \cup N)^*$ (that is, a word containing at least one letter of N) produced by a derivation from the axiom.

$$E \to E + T \to E + T * F \to T + T * F \to T + F * F \to T + a * F \to F + a * F \to a + a * F \to \cancel{a + a * a}$$

Multiple derivations

A given word may have several derivations:
$E \rightarrow E + E \rightarrow F + E \rightarrow F + F \rightarrow 3 + F \rightarrow 3 + 4$

Multiple derivations

A given word may have several derivations:

$E \rightarrow E + E \rightarrow F + E \rightarrow F + F \rightarrow 3 + F \rightarrow 3 + 4$

$E \rightarrow E + E \rightarrow E + F \rightarrow E + 4 \rightarrow F + 4 \rightarrow 3 + 4$

Multiple derivations

A given word may have several derivations:
$E \rightarrow E + E \rightarrow F + E \rightarrow F + F \rightarrow 3 + F \rightarrow 3 + 4$
$E \rightarrow E + E \rightarrow E + F \rightarrow E + 4 \rightarrow F + 4 \rightarrow 3 + 4$
... but if the grammar is not ambiguous, there is only one **left** derivation:

Sorbonne Nouvelle

Multiple derivations

A given word may have several derivations:
$E \rightarrow E + E \rightarrow F + E \rightarrow F + F \rightarrow 3 + F \rightarrow 3 + 4$
$E \rightarrow E + E \rightarrow E + F \rightarrow E + 4 \rightarrow F + 4 \rightarrow 3 + 4$
... but if the grammar is not ambiguous, there is only one **left** derivation:
$\underline{E} \rightarrow \underline{E} + E \rightarrow \underline{F} + E \rightarrow 3 + \underline{E} \rightarrow 3 + \underline{F} \rightarrow 3 + 4$

Multiple derivations

A given word may have several derivations:

$E \to E + E \to F + E \to F + F \to 3 + F \to 3 + 4$

$E \to E + E \to E + F \to E + 4 \to F + 4 \to 3 + 4$

... but if the grammar is not ambiguous, there is only one **left** derivation:

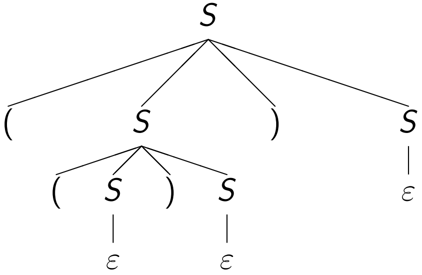
$\underline{E} \to \underline{E} + E \to \underline{F} + E \to 3 + \underline{E} \to 3 + \underline{F} \to 3 + 4$

parsing: trying to find the/a left derivation (resp. right)

Derivation tree

For context-free languages, there is a way to represent the set of equivalent derivations, via a derivation tree which shows all the derivation independantly of their order.

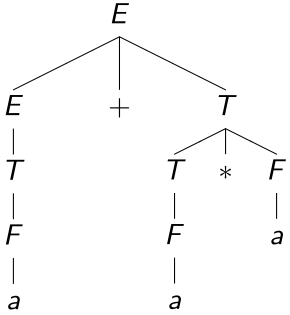
Grammar $\mathcal{G}_2$: $S \longrightarrow \varepsilon \mid (S)S$



$$S \rightarrow (S)S \rightarrow ((S)S)S \rightarrow ((S)S) \rightarrow ((S)) \rightarrow (())$$

Structural analysis

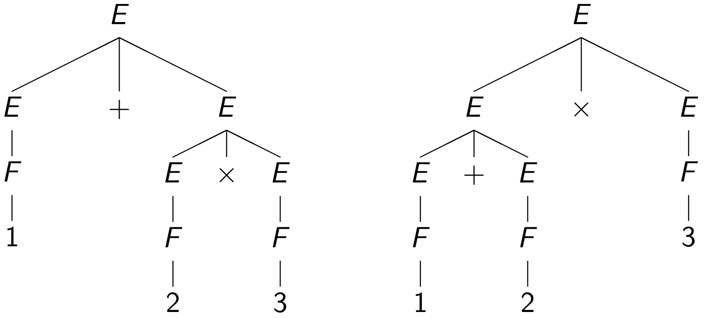
Syntactic trees are precious to give access to the semantics



Ambiguity

When a grammar can assign more than one derivation tree to a word $w \in L(G)$ (or more than one left derivation), the grammar is *ambiguous*.

For instance, $\mathcal{G}_3$ is ambiguous, since it can assign the two follwing trees to $1 + 2 \times 3$:



About ambiguity

- Ambiguity is not desirable for the semantics
- Useful artificial languages are rarely ambiguous
- There are context-free languages that are intrinsequely ambiguous (1)
- Natural languages are notoriously ambiguous...

(1) $\{a^n b a^m b a^p b a^q | (n \geqslant q \wedge m \geqslant p) \vee (n \geqslant m \wedge p \geqslant q)\}$

Comparison of grammars

- different languages generated ⇒ different grammars
- same language generated by $\mathcal{G}$ and $\mathcal{G}'$:
  ⇒ same weak generative power
- same language generated by $\mathcal{G}$ and $\mathcal{G}'$,
  and same structural decomposition :
  ⇒ same strong generative power

References I

Bar-Hillel, Yehoshua, Perles, Micha, & Shamir, Eliahu. 1961. On formal properties of simple phrase structure grammars. *STUF-Language Typology and Universals*, **14**(1-4), 143–172.

Chomsky, Noam. 1957. *Syntactic Structures*. Den Haag: Mouton & Co.

Chomsky, Noam. 1995. *The Minimalist Program*. Vol. 28. Cambridge, Mass.: MIT Press.

Gazdar, Gerald, & Pullum, Geoffrey K. 1985 (May). *Computationally Relevant Properties of Natural Languages and Their Grammars*. Tech. rept. Center for the Study of Language and Information, Leland Stanford Junior University.

Gibson, Edward, & Thomas, James. 1997. The Complexity of Nested Structures in English: Evidence for the Syntactic Prediction Locality Theory of Linguistic Complexity. *Unpublished manuscript, Massachusetts Institute of Technology*.

Joshi, Aravind K. 1985. *Tree Adjoining Grammars: How Much Context-Sensitivity is Required to Provide Reasonable Structural Descriptions?* Tech. rept. Department of Computer and Information Science, University of Pennsylvania.

Langendoen, D Terence, & Postal, Paul Martin. 1984. *The vastness of natural languages*. Basil Blackwell Oxford.

Mannell, Robert. 1999. *Infinite number of sentences*. part of a set of class notes on the Internet. http://clas.mq.edu.au/speech/infinite_sentences/.

Schieber, Stuart M. 1985. Evidence against the Context-Freeness of Natural Language. *Linguistics and Philosophy*, **8**(3), 333–343.

Stabler, Edward P. 2011. Computational perspectives on minimalism. *Oxford handbook of linguistic minimalism*, 617–643.

Steedman, Mark, *et al.* . 2012 (June). *Combinatory Categorial Grammars for Robust Natural Language Processing*. Slides for NASSLLI course http://homepages.inf.ed.ac.uk/steedman/papers/ccg/nasslli12.pdf.

Vijay-Shanker, K., & Weir, David J. 1994. The Equivalence of Four Extensions of Context–Free Grammars. *Mathematical Systems Theory*, **27**, 511–546.