Formal Languages and Linguistics

Pascal Amsili

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

Cogmaster, october 2021

Overview

Formal grammar

Def. 14 ((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

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

$$\mathcal{G}_0 = \Big\langle \{joe, sam, sleeps\},$$

Examples

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

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

Examples

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

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

Examples

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

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

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

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

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

Notation

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

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

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

Immediate Derivation

Def. 15 (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. 16 (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. 16 (Derivation)

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

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

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

Derivation

Def. 16 (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

Derivation

Def. 16 (Derivation)

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

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

$$N\ V\ joe\ N \longrightarrow sam\ V\ joe\ N \longrightarrow \begin{cases} sam\ V\ joe\ joe & \text{or} \\ sam\ V\ joe\ sam & \text{or} \end{cases}$$

Derivation

Def. 16 (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 $\mathcal{G}_0$:

$$N\ V\ joe\ N \longrightarrow sam\ V\ joe\ N \longrightarrow \begin{array}{ll} sam\ V\ joe\ joe & \text{or} \\ sam\ V\ joe\ sam & \text{or} \\ sam\ sleeps\ joe\ N & \text{or} \\ \ldots & \end{array}$$

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 $\mathcal{G}_3$:

$E \times E$

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 $\mathcal{G}_3$:

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

Endpoint of a derivation

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

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

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

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

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

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

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

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

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

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. 17 (Language engendered by a word)

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

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

Def. 18 (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. 17 (Language engendered by a word)

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

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

Def. 18 (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. 17 (Language engendered by a word)

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

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

Def. 18 (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. 17 (Language engendered by a word)

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

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

Def. 18 (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. 17 (Language engendered by a word)

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

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

Def. 18 (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 $() \in L_{\mathcal{G}_2}$: $S \rightarrow (S)S \rightarrow ()S \rightarrow ()$

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 $(()))$, $()()()$, $((()(())))$...

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 :

Sorbonne Nouvelle

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 :
$)S(\rightarrow$

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 $((()))$, $()()()$, $((()()()))\dots$
but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :
$)S(\rightarrow)(S)S(\rightarrow$

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 :
$)S(\rightarrow)(S)S(\rightarrow)()S(\rightarrow$

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 $(())), ()()(), ((()()())) \ldots$
but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :
$)S(\to)(S)S(\to)()S(\to)()($

Engendered language

Def. 17 (Language engendered by a word)

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

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

Def. 18 (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 $((()))$, $()()()$, $((()()()))\ldots$
but $)()(\notin L_{\mathcal{G}_2}$, even though the following is a licit derivation :
$)S(\to)(S)S(\to)()S(\to)()($
for there is no way to arrive at $)S($ starting with S.

Example

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

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

Proto-word

Def. 19 (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.

$$E \rightarrow E + T \rightarrow E + T * F \rightarrow T + T * F \rightarrow T + F * F \rightarrow T + a * F \rightarrow F + a * F \rightarrow a + a * F \rightarrow \not{a} \not{+} \not{a} \not{*} \not{a}$$

Multiple derivations

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

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:

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:

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

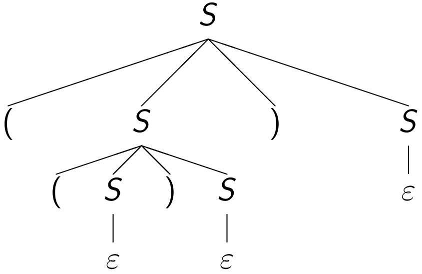
Multiple derivations

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:
$\underline{E} \rightarrow \underline{E} + E \rightarrow \underline{F} + E \rightarrow 3 + \underline{E} \rightarrow 3 + \underline{F} \rightarrow 3 + 4$

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.



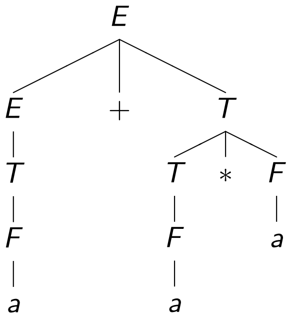
Grammar $\mathcal{G}_2$:

$$S \longrightarrow \varepsilon \mid (S)S$$

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

Structural analysis

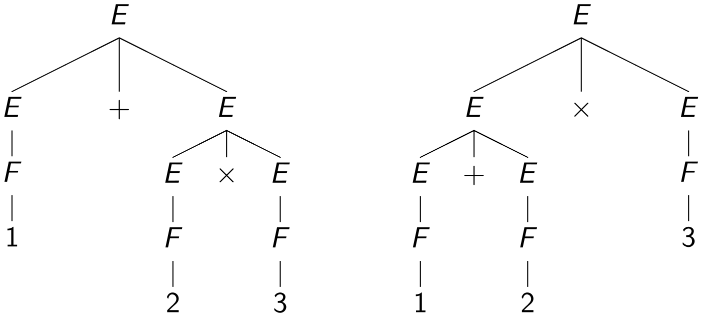
Syntactic trees are precious to give access to the semantics



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 (3)
- Natural languages are notoriously ambiguous...

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

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

Overview

Principle

Define language families on the basis of properties of the grammars that generate them :

1. Four classes are defined, they are included one in another
2. A language is of type k if it **can** be recognized by a type k grammar (and thus, by definition, by a type $k-1$ grammar) ; and **cannot** be recognized by a grammar of type $k+1$.

Chomsky's hierarchy

type 0 No restriction on $P \subset (X \cup V)^* V (X \cup V)^* \times (X \cup V)^*$.

type 1 (*context-sensitive* grammars) All rules of P are of the shape $(u_1 S u_2, u_1 m u_2)$, where u_1 and $u_2 \in (X \cup V)^*$, $S \in V$ and $m \in (X \cup V)^+$.

type 2 (*context-free* grammar) All rules of P are of the shape (S, m), where $S \in V$ and $m \in (X \cup V)^*$.

type 3 (*regular* grammars) All rules of P are of the shape (S, m), where $S \in V$ and $m \in X.V \cup X \cup \{\varepsilon\}$.

Examples

type 3:

$$
\begin{array}{lcl}
S & \rightarrow & aS \mid aB \mid bB \mid cA \\
B & \rightarrow & bB \mid b \\
A & \rightarrow & cS \mid bB
\end{array}
$$

Examples

type 3:

$$\begin{array}{lcl} S & \rightarrow & aS \mid aB \mid bB \mid cA \\ B & \rightarrow & bB \mid b \\ A & \rightarrow & cS \mid bB \end{array}$$

type 2:

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

Example 1 type 0

Type 0:

$$
\begin{array}{lll}
S \rightarrow SABC & AC \rightarrow CA & A \rightarrow a \\
S \rightarrow \varepsilon & CA \rightarrow AC & B \rightarrow b \\
AB \rightarrow BA & BC \rightarrow CB & C \rightarrow c \\
BA \rightarrow AB & CB \rightarrow BC &
\end{array}
$$

generated language :

Example 1 type 0

Type 0:

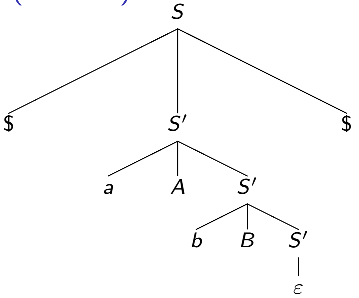
$$
\begin{array}{lll}
S \rightarrow SABC & AC \rightarrow CA & A \rightarrow a \\
S \rightarrow \varepsilon & CA \rightarrow AC & B \rightarrow b \\
AB \rightarrow BA & BC \rightarrow CB & C \rightarrow c \\
BA \rightarrow AB & CB \rightarrow BC &
\end{array}
$$

generated language : words with an equal number of a, b, and c.

Example 2: type 0

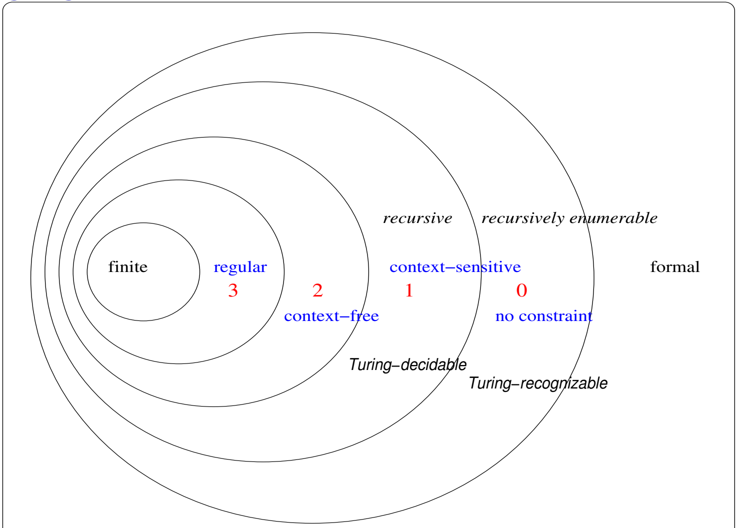
$$
\text{Type 0:} \quad
\begin{array}{lll}
S \rightarrow \$S'\$ & Aa \rightarrow aA & \$a \rightarrow a\$ \\
S' \rightarrow aAS' & Ab \rightarrow bA & \$b \rightarrow b\$ \\
S' \rightarrow bBS' & Ba \rightarrow aB & A\$ \rightarrow \$a \\
S' \rightarrow \varepsilon & Bb \rightarrow bB & B\$ \rightarrow \$b \\
 & & \$\$ \rightarrow \#
\end{array}
$$

Example 2: type 0 (cont'd)



$	a	A	b	B	$
a	$	A	b	B	$
a	$	A	b	$	b
a	$	b	A	$	b
a	b	$	A	$	b
a	b	$	$	a	b
a	b	#		a	b

Language families



Remarks

- There are others ways to classify languages,
  - either on other properties of the grammars;
  - or on other properties of the languages
- Nested structures are preferred, but it's not necessary
- When classes are nested, it is expected to have a growth of complexity/expressive power

Overview

Sorbonne Nouvelle

Motivation

Why an inquiry into the formal complexity of Natural Language(s) ?

- It gives us knowledge about the **structure** of natural languages,
- It helps us assess the **adequation** of linguistic formalisms,
- It gives bound for the **complexity** of NLP tasks,
- It provides us with **predictions** about human language processing.

Hypotheses

We assume that:

- We can talk about "natural language" in general: all languages have a similar structure, a similar power
- Natural languages are recursively enumerable, i.e. they are formal languages
- Natural languages are infinite

$\Rightarrow$ Under these hypotheses, it is possible to ask the question: what is the complexity of natural languages ?

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

(4) A stranger arrived.

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

(4) A tall stranger arrived.

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

(4) A tall handsome stranger arrived.

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

(4) A dark tall handsome stranger arrived.

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

(4) A dark tall handsome stranger arrived suddenly.

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

   (4) A dark tall handsome stranger arrived suddenly.

2. More interestingly, arbitrary long sentences can be built through center-embedding. In this case, there is a dependancy between arbitrary far apart elements:

   (5) The cats hunt.

   center-embedding: embedding a phrase in the middle of another phrase of the same type

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

   (4) A dark tall handsome stranger arrived suddenly.

2. More interestingly, arbitrary long sentences can be built through center-embedding. In this case, there is a dependancy between arbitrary far apart elements:

   (5) The cats the neighbor owns hunt.

   center-embedding: embedding a phrase in the middle of another phrase of the same type

An infinite number of sentences

1. Arbitrary long sentences can be built by adding new material:

   (4) A dark tall handsome stranger arrived suddenly.

2. More interestingly, arbitrary long sentences can be built through center-embedding. In this case, there is a dependancy between arbitrary far apart elements:

   (5) The cats the neighbor who arrived owns hunt.

   center-embedding: embedding a phrase in the middle of another phrase of the same type

An infinite number of sentences (cont'd)

Consider the 3 structures:

- If S_1, then S_2.
- Either S_1 or S_2.
- The man who said S_1 is coming today.

1. The colored items are *dependent* one from the other
2. It is possible to create nested sentences of arbitrary length:

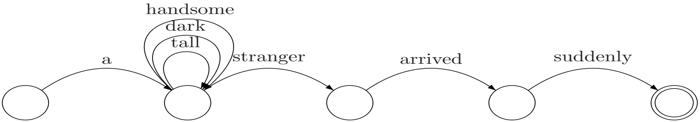
(6) If either the man who said S_a is coming today, or S_b, then S_c.

⇒ A look at various ways to form infinite sentences gives access to complexity.

Overview

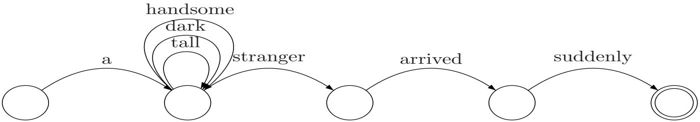
Preliminaries: a word on lexicon

(7) A dark tall handsome stranger arrived suddently.



Preliminaries: a word on lexicon

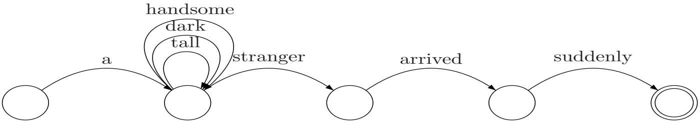
(7) A dark tall handsome stranger arrived suddently.



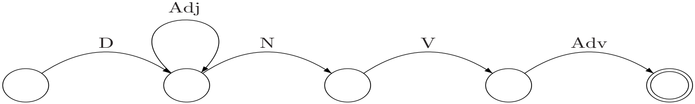
Let's leave aside lexicon issues

Preliminaries: a word on lexicon

(7) A dark tall handsome stranger arrived suddently.



Let's leave aside lexicon issues



Chomsky's first attempt

Consider the 3 structures:

- If S_1, then S_2.
- Either S_1 or S_2.
- The man who said S_1 is coming today.

1. The colored items are *dependent* one from the other
2. It is possible to create nested sentences of arbitrary length:

(8) If either the man who said S_a is coming today, or S_b, then S_c.

Since such sentences are instances of mirroring and since the mirror language is not regular, then English is not regular (Chomsky, 1957, p. 22).

Fallacious claim: **a regular language may contain a non regular sub-language**

Classical argument I

Let's consider the sentence(s):

(9) A man fired another man.

Classical argument I

Let's consider the sentence(s):

(9) A man that a man hired fired another man.

Classical argument I

Let's consider the sentence(s):

(9) A man that a man that a man hired hired fired another man.

Classical argument I

Let's consider the sentence(s):

(9) A man that a man that a man hired hired fired another man.
A man (that a man)^2 (hired)^2 fired another man.

Classical argument I

Let's consider the sentence(s):

(9) A man that a man that a man hired hired fired another man.
A man (that a man)^2 (hired)^2 fired another man.

The sentences (10) are all well-formed sentences (for any n).

(10) A man (that a man)^n (hired)^n fired another man.

Classical Argument II

Let x = that a man
y = hired
w = a man
v = fired another man

- wx^*y^*v is regular
- English $\cap\ wx^*y^*v = wx^ny^nv$ (10)
- If English is regular, then wx^ny^nv must be regular (for the intersection of two regular languages is regular)
- But wx^ny^nv is not regular (pumping lemma).
  Contradiction $\Rightarrow$ English is not regular.

(Schieber, 1985)

Discussion

Counter arguments :

- Natural languages are finite
  - productivity doesn't seem to be bound
  - a list of all possible sentences, supposedly finite, is still too long for a human to learn
- People are bad at interpreting embedding: there might be a limit
  - there are indeed constraints on performance,
  - but in writing, or with an appropriate intonation, there doesn't seem to be a hard-wired limit

Discussion: processing problems with nested structures

Psycholinguistic evidence that (11b) is more accepted than (11a) (Fodor, Frazier)

(11) a. The patient who the nurse who the clinic had hired admitted met Jack.
b. The patient who the nurse who the clinic had hired met Jack.

Other factors:

(12) a. The pictures which the photographer who I met yesterday took were damaged by the child.
b. ?The pictures which the photographer who John met yesterday took were damaged by the child.

(13) a. Isn't it true that example sentences [that people [that you know] produce] are more likely to be accepted? (De Roeck et al, 1982)
b. A book [that some Italian [I've never heard of] wrote] will be published soon by MIT Press (Frank, 1992)

(Gibson & Thomas, 1997)

Examples

Bad examples :

(14) A girl that the man that the doctor knows like was fired.

Good examples:

(15) A foreman that an employee who were recently hired talked with was fired.

Overview

Formal Languages

Regular Languages

Formal Grammars

Formal complexity of Natural Languages

- Introduction
- Are NL regular?
- Are NL context-free?
- Are NL context-sensitive?

Pumping lemma: intuition

1. If a word is long enough, then there is (at least) one non terminal symbol appearing several times in its derivation

"long enough" ?

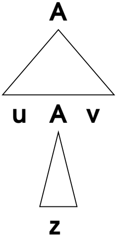
$$\begin{array}{lll} S & \rightarrow & A\ B \\ A & \rightarrow & abaccabca \\ & | & abSba \\ B & \rightarrow & ccccc \end{array}$$

Minimal length : 14:

$$S \rightarrow AB \rightarrow abaccabcaB \rightarrow abaccabcaccccc$$

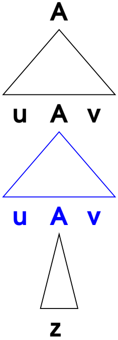
Pumping lemma: intuition

2 Let's call this non terminal symbol A.



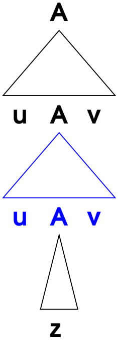
Pumping lemma: intuition

2 Let's call this non terminal symbol *A*.



Pumping lemma: intuition

2 Let's call this non terminal symbol A.



$$A \xrightarrow{*} uAv$$
$$A \xrightarrow{*} uAv \xrightarrow{*} uzv$$
$$A \xrightarrow{*} uAv \xrightarrow{*} uuAvv \xrightarrow{*} \underbrace{u \dots u}_{n} z \underbrace{v \dots v}_{n}$$

Pumping Lemma for CF languages

Def. 20 (Star lemma – CF languages)

If L is context-free, there exists $p \in \mathbb{N}$ such that:
$\forall w$ s.t. $|w| \geqslant p$,
w can be factorized $w = rstuv$,
with:

$$|su| \geqslant 1$$
$$|stu| \leqslant p$$
$$\forall i \geqslant 0, \quad rs^i tu^i v \in L$$

(Bar-Hillel *et al.* , 1961)

Pumping lemma: Consequences

The pumping lemma gives us a tool to prove that a language is **not** context-free.

$\mathcal{L}$ context-free	$\Rightarrow$	pumping lemma ($\forall i, rs^i tu^i v \in \mathcal{L}$)
pumping lemma	$\not\Rightarrow$	$\mathcal{L}$ context-free
NO pumping lemma	$\Rightarrow$	$\mathcal{L}$ **NOT** context-free

to prove that $\mathcal{L}$ is

context-free provide a type 2 grammar

not context-free show that the pumping lemma does not apply

Results: expressivity

- well-parenthetized words (dyck's language) is context-free
  $S \rightarrow (S)S \mid \varepsilon$
- $a^n b^n (n \geqslant 0)$ is a context-free language
  $S \rightarrow aSb \mid \varepsilon$
- $ww^R, w \in \Sigma^*$ (mirror language) is a context-free language
  $S \rightarrow aSa \mid bSb \mid \varepsilon$
- $ww, w \in \Sigma^*$ (copy language) is not context-free
  proof: pumping lemma
- $a^n b^n c^n$ is not context-free
  proof: pumping lemma
- $a^m b^n c^m d^n$ is not context-free
  proof: pumping lemma
- $xa^m b^n yc^m d^n z$ is not context-free
  proof: pumping lemma

Closure properties I

- CF languages are closed under rational operations
  - union (gather all the rules, avoiding name conflicts, and adding a new start rule $S \rightarrow S_1|S_2$),
  - product ($S \rightarrow S_1 S_2$),
  - and Kleene star ($S \rightarrow S_1 S \mid \varepsilon$).

Closure properties II : intersection

- CF languages are not closed under intersection

Example

$L_1 = \{a^i b^i c^j \mid i, j \geq 0\}$ is context-free:
$$S \rightarrow XY$$
$$X \rightarrow aXb \mid \varepsilon$$
$$Y \rightarrow cY \mid \varepsilon$$

$L_2 = \{a^i b^j c^j \mid i, j \geq 0\}$ is also context-free:
$$S \rightarrow XY$$
$$X \rightarrow aX \mid \varepsilon$$
$$Y \rightarrow bYc \mid \varepsilon$$

But $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ is not contex-free.

Closure properties III: other results

- CF languages are not closed under complement (since they are not closed under intersection)
- CF languages are closed under intersection with a regular language
- a sub-class of CF languages, *deterministic CF languages* are closed for set complement, but not for union (one can easily define an intrinsequely non deterministic language as the union of two "independant" languages)

Final argument I

After many attempts by various scholars, attempts which are severely critized and ruined in (Gazdar & Pullum, 1985), Schieber (1985) came up with a widely accepted answer:

1. In swiss-german, subordinate clauses can have a structure where all NPs precede all Vs. (16)

(16) Jan säit das mer NP* es huus haend wele V* aastrüche
Jan said that we NP* the house have wanted V* paint
'Jan said that we have wanted (that) V* NP* paint the house'

2. Among those subordinate clauses, those where all the dative NPs precede all the accusative NPs are well-formed. (17)

(17) … das mer d'chind em Hans es huus haend wele laa hälfe aastrüche
… that we the_children.ACC Hans.DAT the house.ACC have wanted let help paint
'… that we have wanted to let the children help Hans to paint the house'

Final argument II

3. The number of verbs requiring a dative has to be equal to the number of dative NPs, the same for accusative.
4. The number of verbs in a subordinate clause is limited only by performance

Let R be the language:

R = {Jan säit das mer (d'chind)^h (em Hans)^i es huus haend wele (laa)^j (hälfe)^k aastrüche, $i, j, k, h \geqslant 1$}

Then let L = Swiss-German $\cap$ R =

{Jan säit das mer (d'chind)^m (em Hans)^n es huus haend wele (laa)^m (hälfe)^n aastrüche, $m, n \geqslant 1$}

L is not context-free, whereas R is regular.

$\Rightarrow$ Swiss-German is not context-free.

Overview

Current proposal

1. The context-sensitive class seems too big: for instance $\{a^{2^i} \;/\; i \geqslant 0\}$ is context-sensitive.
2. Joshi (1985) proposed a subclass of type 1 languages, namely the class of *mildly context-sensitive languages* (MCSL), this class has the following properties:
   - ww is MCS
   - $a^n b^n c^n$ is MCS
   - $a^n b^n c^n d^n$ is MCS
   - $a^i b^j c^i d^j$ is MCS
   - $a^n b^n c^n d^n e^n$ is not MCS
   - www is not MCS
   - $ab^h ab^i ab^j ab^k ab^l, h > i > j > k > l \geqslant 1$ is not MCS
   - a^{2^i} is not MCS

Current proposal

1. The context-sensitive class seems too big: for instance $\{a^{2^i} \ / \ i \geqslant 0\}$ is context-sensitive.
2. Joshi (1985) proposed a subclass of type 1 languages, namely the class of *mildly context-sensitive languages* (MCSL), this class has the following properties:
   - ww is MCS
   - $a^n b^n c^n$ is MCS
   - $a^n b^n c^n d^n$ is MCS
   - $a^i b^j c^i d^j$ is MCS
   - $a^n b^n c^n d^n e^n$ is not MCS
   - www is not MCS
   - $ab^h ab^i ab^j ab^k ab^l, h > i > j > k > l \geqslant 1$ is not MCS
   - a^{2^i} is not MCS

Conjecture : NL $\in$ MCSL

More about MCSL

Interesting properties of MCSL:

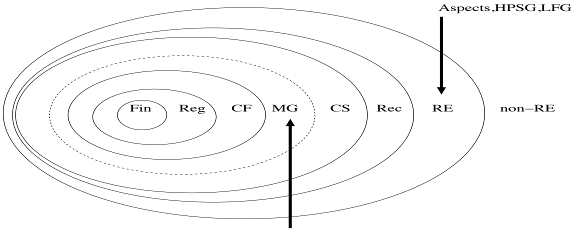
- restricted growth: if L is MCS, there is k such that for all words $w \in L$, there is a word w' s.t. $|w'| \leqslant |w| + k$
- word problem for MCSL are of a polynomial complexity

These properties are arguably common with natural languages

The formalism introduced by Joshi, *Tree Adjoining Grammars*, defines the class of MCSL.

Minimalist grammars (Stabler, 2011)

Minimalist grammars (MGs), as defined here by (5), (6) and (8), have been studied rather carefully. It has been demonstrated that the class of languages definable by minimalist grammars is exactly the class definable by multiple context free grammars (MCFGs), linear context free rewrite systems (LCFRSs), and other formalisms [62,64,66,41]. MGs contrast in this respect with some other much more powerful grammatical formalisms (notably, the 'Aspects' grammar studied by Peters and Ritchie [76], and HPSG and LFG [5,46,101]):



The MG definable languages include all the finite (Fin), regular (Reg), and context free languages (CF), and are properly included in the context sensitive (CS), recursive (Rec), and recursively enumerable languages (RE). Languages definable by tree adjoining grammar (TAG) and by a certain categorial combinatory grammar (CCG) were shown by Vijay Shanker and Weir to be sandwiched inside the MG class [103].[4] With all these results,

Theorem 1. $CF \subset \boxed{TAG \equiv CCG} \subset \boxed{MCFG \equiv LCFRS \equiv MG} \subset CS.$

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.

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.