Natural language syntax: parsing and complexity

Timothée Bernard and Pascal Amsili

Université Paris Cité, Université Sorbonne Nouvelle
timothee.bernard@u-paris.fr, pascal.amsili@ens.fr

Ljubljana, Slovenia – August 7-11, 2023
ESSLLI foundational course in Language and Computation

Overview of the course

- Day 1: Formal languages and syntactic complexity.
- Day 2: The complexity of natural language.
- Day 3: Historic algorithms for parsing.
- Day 4: Modern approaches to parsing.
- Day 5: Neural networks and error propagation.

Day 2

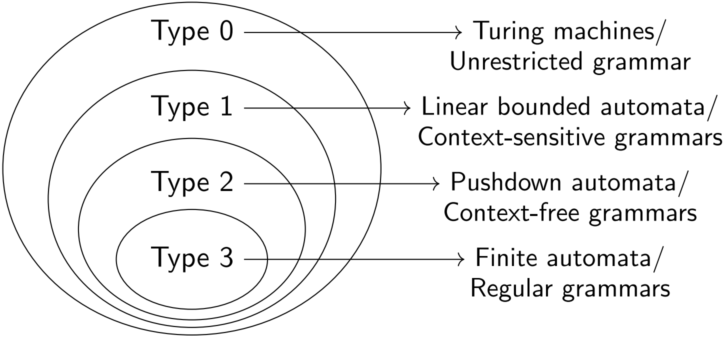
Recap from Day 1

- Languages are sets of words (finite sequences of symbols).
- Automata are finite state machines with or without additional memory.
- Grammars are finite sets of rewriting rules.
- The parsing problem for a grammar consists in finding derivations.
- All solvable problems can be expressed as parsing problems.
- The Chomsky-Schützenberger hierarchy is a hierarchy of classes of languages, of models of automata, and of grammatical formalisms.
- For interpreted languages, syntactic complexity is not semantic expressivity.

Today's content

- The complexity of natural language(s).
- Closure properties of formal languages.
- Pumping lemmas (regular & context-free).
- Syntactic formalisms used in formal linguistics.
- The complexity of these formalisms.

Where are natural languages?



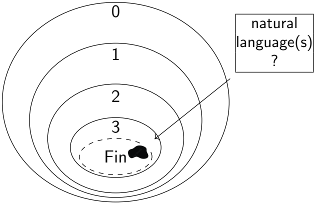
Why should we care about the complexity of NL?

- Theoretical understanding of (natural) language.
- Appropriateness of linguistic (syntactic) formalisms.
- Lower bound for the complexity of NLP tasks.
- Predictions about human language processing and acquisition.

Hypotheses

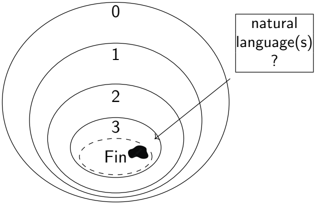
- Natural languages are all of comparable complexity
  or at least they can be grouped into classes of comparable complexity.
- Natural languages can be considered as formal languages:
  - Finite set of atomic symbols (morphemes?).
  - Binary grammaticality judgments for all sequences.

Are natural languages finite?



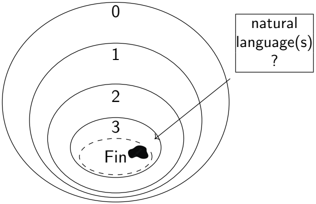
- NLs could be modelled as lists.
- It could still be interesting to use more powerful formalisms but for other reasons than complexity (conciseness, efficiency, suitability for semantics...).

Are natural languages finite?



- NLs could be modelled as lists.
- It could still be interesting to use more powerful formalisms but for other reasons than complexity (conciseness, efficiency, suitability for semantics...).
- Requires a bound on the length of well-formed sentences...

Are natural languages finite?



- NLs could be modelled as lists.
- It could still be interesting to use more powerful formalisms but for other reasons than complexity (conciseness, efficiency, suitability for semantics...).
- Requires a bound on the length of well-formed sentences...

... which is not realistic, if language is, as proposed by Humboldt (frequently quoted by Chomsky) "an infinite use of finite means"

An infinite number of well-formed sentences (data)

- It is possible to build up arbitrarily long sentences.
- lenghtening: ab^nc

(1) a. Sam took her knife.
b. Sam took her lovely knife.
c. Sam took her lovely little knife.

- center-embedding: ab^ncd^ne

(2) a. A foreman was fired.
b. A foreman who an employee talked with was fired.
c. A foreman who an employee that Mary recently hired talked with was fired.

An infinite number of well-formed sentences (discussion)

(3) A man (that another man)^n (hired)^n fired Sam.

Some rather simple cases may seem hard to parse because of cognitive limitations (working memory...):

(4) #The patient who the nurse who the clinic had hired admitted met Jack.

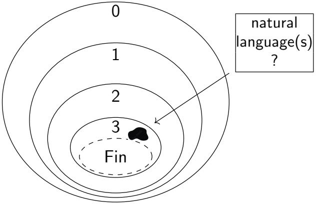
... but with appropriate help (punctuation, selection restrictions...) most speakers accept arbitrarily complex sentences and recognise them as well formed:

(5) Isn't it true that example sentences [that people [that you know] produce] are more likely to be accepted? (De Roeck et al, 1982)

(6) A book [that some Italian [I've never heard of] wrote] will be published soon by MIT Press. (Frank, 1992)

(Gibson & Thomas 1999)

Is natural language regular?



A. Regular languages are closed under intersection.

- $L_1 = \{ab^n cd^m e \mid n, m \in \mathbb{N}\}$ is regular.

B. $L_2 = \{ab^n cd^n e \mid n \in \mathbb{N}\}$ is not regular.

C. The intersection of English with a regular language (L_1) is not regular (L_2).

- Therefore, English is not regular.

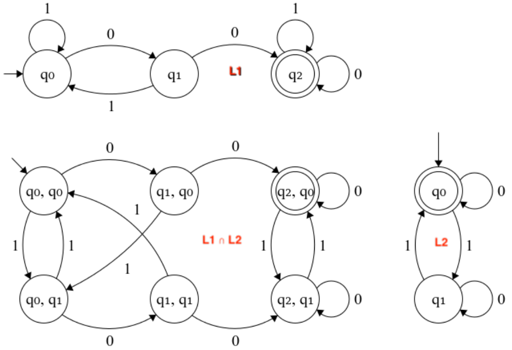
Is NL regular? A. Closure property

Closure property

The intersection of two regular languages is regular.

Proof: Construction of the product of two DFAs.

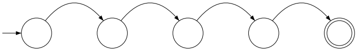
Example:



credit: Martin Alessandro

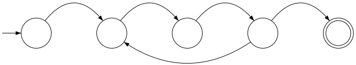
Is NL regular? B. Pumping lemma (intuition)

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 $\mathcal{L}(A)$.



Is NL regular? B. 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 $q \overset{w_{i:j}}{\to} q$.
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 $\mathcal{L}(A)$.



Is NL regular? B. 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. $\{a^nbc \mid n \in \mathbb{N}\}$) 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^ny (i.e. a^nbc) belongs to the language.

Is NL regular? 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.

Is NL regular? 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 *a*s than *b*s).
- So no $k \in \mathbb{N}$ works; L does not have the pumping property.

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

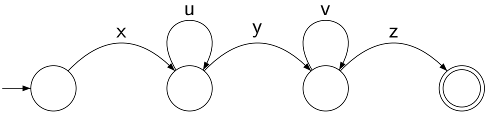
Is NL regular? C. Proof (I)

$L_1 = \{$A man [that another man]^n I saw [hired]^m fired Sam. $\mid n, m \in \mathbb{N}\}$
This language is regular.

With

- $x =$ A man
- $u =$ that another man
- $y =$ I saw
- $v =$ hired
- $z =$ fired Sam

$L_1 = \{xu^n yv^m z \mid n, m \in \mathbb{N}\}$.



Is NL regular? C. Proof (II)

$L_1 = \{$A man [that another man]^n I saw [hired]^m fired Sam. $\mid n, m \in \mathbb{N}\}$.

Sentences of L_1 are well-formed in English iff $n = m$.

In other words, English $\cap$ L_1 is $L_2 = \{xu^n yv^n z \mid n \in \mathbb{N}\}$.

We have seen that this language is not regular.

Is NL regular? C. Proof (III)

- $L_2 \subseteq$ English is not regular.

Caution

The fact that some non-regular language is a subset of English provides no indication of English being regular or not.
Ex: Σ^* is regular and contains all languages on Σ, even the most complex ones (beyond type 0).

Is NL regular? C. Proof (III)

- $L_2 \subseteq$ English is not regular.

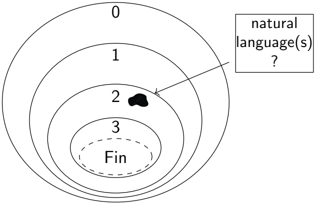
Caution

The fact that some non-regular language is a subset of English provides no indication of English being regular or not.
Ex: Σ^* is regular and contains all languages on Σ, even the most complex ones (beyond type 0).

But:

- The intersection of English with a regular language (L_1) is not regular, therefore English is not regular.

Is natural language context-free?



A. Context-free languages are closed under intersection with a regular language.

- $L_1 = \{wa^n b^m xc^k d^l y \mid n, m, k, l \in \mathbb{N}\}$ is regular.

B. $L_2 = \{wa^n b^m xc^n d^m y \mid n, m \in \mathbb{N}\}$ is not context-free.

C. The intersection of Swiss German with a regular language (L_1) is not context-free (L_2).

- Therefore, Swiss German is not context-free.

Is natural language context-free? A. Closure property

Closure property

The intersection of a context-free language with a regular language is context-free.

Proof: by construction of a cross-product push-down automaton which can recognise the intersection.
(other proofs, based on CF grammars, possible)

Is NL context-free? B. Pumping lemma (intuition)

If L is an infinite context-free language,
if a word is long enough, then, in its derivation, there is (at least) one non-terminal symbol that generates itself with additional material.

$$\begin{array}{lcl} S & \rightarrow & A\ B \\ A & \rightarrow & cc \\ & | & aSa \\ B & \rightarrow & b \end{array}$$

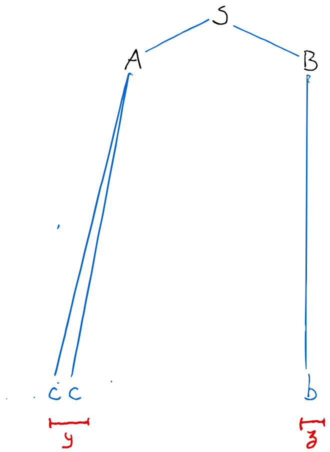
$S \Rightarrow AB \Rightarrow ccB \Rightarrow ccb$

$S \Rightarrow AB \Rightarrow sSaB \Rightarrow aABaB \Rightarrow \ldots \Rightarrow accbab$

Is NL context-free? B. Pumping lemma (intuition)

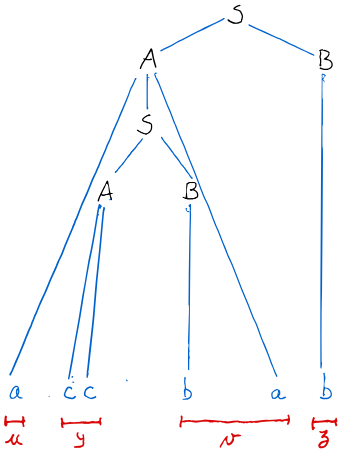
If a non-terminal A generates itself once in a derivation, since the grammar is context-free, then there is no way to prevent A from generating itself an arbitrary number of times.

Is NL context-free? B. Pumping lemma (intuition)



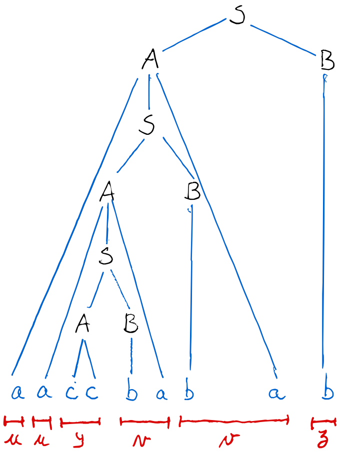
$$
\begin{array}{lcl}
S & \to & A\ B \\
A & \to & cc \\
 & | & aSa \\
B & \to & b
\end{array}
$$

Is NL context-free? B. Pumping lemma (intuition)



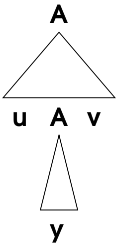
$$
\begin{array}{lcl}
S & \rightarrow & A\ B \\
A & \rightarrow & cc \\
 & | & aSa \\
B & \rightarrow & b
\end{array}
$$

Is NL context-free? B. Pumping lemma (intuition)



$$
\begin{array}{lcl}
S & \rightarrow & A\ B \\
A & \rightarrow & cc \\
 & | & aSa \\
B & \rightarrow & b
\end{array}
$$

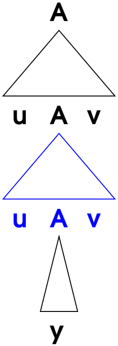
Is NL context-free? B. Pumping lemma (intuition)



- If there is a productive derivation $A \overset{*}{\Rightarrow} y$,
- and a "recursive" situation $A \overset{*}{\Rightarrow} uAv$,
- then any identical number of embedded factors u and v can be produced.

$$A \overset{*}{\Rightarrow} uAv$$
$$A \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} uyv$$
$$A \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} \underbrace{u\dots u}_{n} A \underbrace{v\dots v}_{n} \overset{*}{\Rightarrow} u^n y v^n$$

Is NL context-free? B. Pumping lemma (intuition)



$$A \overset{*}{\Rightarrow} uAv$$
$$A \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} uyv$$
$$A \overset{*}{\Rightarrow} uAv \overset{*}{\Rightarrow} \underbrace{u \ldots u}_{n} A \underbrace{v \ldots v}_{n} \overset{*}{\Rightarrow} u^n y v^n$$

- If there is a productive derivation $A \overset{*}{\Rightarrow} y$,
- and a "recursive" situation $A \overset{*}{\Rightarrow} uAv$,
- then any identical number of embedded factors u and v can be produced.

Is NL context-free ? Pumping Lemma (definition)

Pumping lemma

Let L be a context-free language.
$\exists k \in \mathbb{N}$ such that
$\forall w \in L$ such that $|w| \geq k$,
$\exists x, u, y, v, z$ such that $w = xuyvz$ and that

1. $|uv| \geq 1$;
2. $|uyv| \leq k$;
3. $\forall n \in \mathbb{N}$, $xu^n yv^n z \in L$.

(Bar-Hillel, Perles & Shamir 1961)

Is NL context-free? B. Pumping lemma (consequences)

context-free	$\Rightarrow$	pumping property satisfied
pumping property **NOT** satisfied	$\Rightarrow$	**NOT** context-free
pumping property satisfied	$\not\Rightarrow$	context-free

To prove that L is

context-free provide a context-free grammar;

not context-free show that the pumping property is not satisfied.

Is NL context-free? Pumping lemma (example)

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

- Consider any $k \in \mathbb{N}$.
- Consider $w = a^k b^k c^k \in L$ ($|w| \geq k$).
- If $w = xuyvz$ with $|uv| \geq 1$ and $|uyv| \leq k$, then uyv either contains no c, or contains no a.
- But then, $xu^0yv^0z = xyz \notin L$ (either strictly less cs than as, or strictly less as than cs).
- So no $k \in \mathbb{N}$ works; L does not have the pumping property.

A similar reasoning applies to $\{xu^n yv^n zw^n t \mid n \in \mathbb{N}\}$.

Is NL context-free? C. Proof

Swiss German data (Shieber 1985).
Cross-serial dependencies:

(7) Jan säit das mer [em Hans]$_1$ [es huus]$_2$ [hälfed]$_{1'}$ [aastriiche]$_{2'}$.
Jan says that we Hans the house helped paint.
'Jan says that we [helped]$_{1'}$ [Hans]$_1$ [paint]$_{2'}$ [the house]$_2$.'

- In Swiss German, subordinate clauses can have a structure where all NPs precede all Vs.
- It is possible to have all dative NPs before all accusative NPs and all dative-subcategorizing Vs before all accusative-subcategorizing Vs. $\rightarrow$ cross-serial dependancy.
- The number of verbs requiring a dative has to be equal to the number of dative NPs, similarly for accusative.
- The number of verbs in a subordinate clause is limited only by performance.

Is NL context-free? C. Proof

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

Hypothesis

Considering the well-formedness of (8), the following sentence is correct iff $n_1 = n_3$ and $n_2 = n_4$:

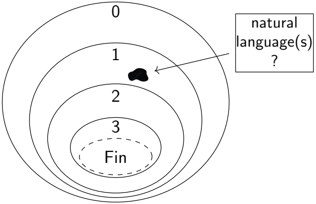
(9) Jan säit das mer [d'chind]^{n_1} [em Hans]^{n_2} es huus haend wele laa^{n_3} hälfe^{n_4} aastriiche.

Is NL context-free? C. Proof

- $L_1 = \{wa^n b^m xc^l d^k y \mid n, m, k, l \in \mathbb{N}\}$ is regular.
- With:
  - $w =$ Jan säit das mer
  - $a =$ d'chind
  - $b =$ em Hans
  - $x =$ es huus haend wele
  - $c =$ laa
  - $d =$ hälfe
  - $y =$ aastriiche

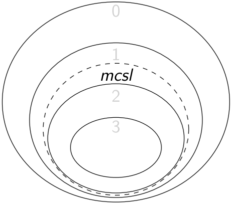
  Swiss German $\cap$ L_1 is $L_2 = \{wa^{n_1} b^{n_2} xc^{n_1} d^{n_2} y \mid n_1, n_2 \in \mathbb{N}\}$.
- L_2 is not CF ($\rightarrow$ pumping lemma, CF version), so Swiss German is not CF either.

Is natural language context-sensitive?



- Almost certainly.
- But this class seems much too large (it includes languages very far from (any) natural language).
- Joshi 1985: what's needed is a class of grammars/languages that are only slightly more powerfull than CFGs.

Looking for a smaller class



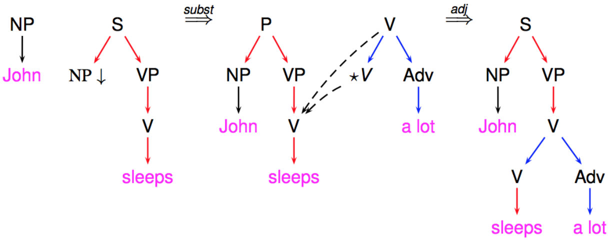
Mildly context-sensitive grammars:
(Joshi 1985)

- limited cross-serial dependencies (cf. Swiss German);
- constant growth (a^{2^i} should not belong to the class);
- polynomial parsing;

Formal definition still needed; note that parsing depends on the grammar rather than on the language.

Tree Adjoining Grammars

- Tree Adjoining Grammars (TAG): introduced by Joshi (1985).
- Elementary units are (anchored) trees rather than sequences of letters.
- A grammar contains rules for rewriting trees, based on two operations: adjunction and substitution.



Inria, FMRG

TAG languages = MCSL

Tree Adjoining Grammars define the class of Mildly Context Sensitive Languages (MCSL).

- $\{ww \mid w \in \Sigma^\star\}$ is MCS.
- $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ is MCS.
- $\{a^n b^n c^n d^n \mid n \in \mathbb{N}\}$ is MCS.
- $\{a^i b^j c^i d^j \mid i, j \in \mathbb{N}\}$ is MCS.

- $\{a^n b^n c^n d^n e^n \mid n \in \mathbb{N}\}$ is not MCS.
- $\{www \mid w \in \Sigma^\star\}$ is not MCS.
- $\{ab^h ab^i ab^j ab^k ab^l \mid h > i > j > k > l \geq 1\}$ is not MCS.
- $\{a^{2^i} \mid i \in \mathbb{N}\}$ is not MCS.

CCGs define exactly the same class

Combinatory Categorial Grammar (CCG): developped by Steedman (e.g. 2000).
Phrase structure rules are replaced with:

- categories: *likes*: (S\NP)/NP;
- general combinatory rules.

$$\frac{\text{Sabine}}{\mathtt{NP}} \quad \frac{\text{likes}}{\mathtt{(S\backslash NP)/NP}} \quad \frac{\text{books}}{\mathtt{NP}}$$

CCGs define exactly the same class

Combinatory Categorial Grammar (CCG): developped by Steedman (e.g. 2000).
Phrase structure rules are replaced with:

- categories: *likes*: (S\NP)/NP;
- general combinatory rules.

$$
\begin{array}{ccc}
\text{Sabine} & \text{likes} & \text{books} \\
\hline
\texttt{NP} & \texttt{(S\backslash NP)/NP} & \texttt{NP} \\
 & \multicolumn{2}{c}{\overline{\qquad\qquad\qquad}{>}} \\
 & \multicolumn{2}{c}{\mathsf{S}\ \backslash \mathsf{NP}}
\end{array}
$$

CCGs define exactly the same class

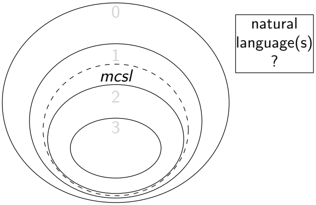
Combinatory Categorial Grammar (CCG): developped by Steedman (e.g. 2000).

Phrase structure rules are replaced with:

- categories: *likes*: (S\NP)/NP;
- general combinatory rules.

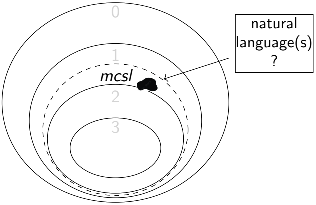
$$
\dfrac{\dfrac{\text{Sabine}}{\mathtt{NP}} \quad \dfrac{\dfrac{\text{likes}}{\mathtt{(S\backslash NP)/NP}} \quad \dfrac{\text{books}}{\mathtt{NP}}}{\text{S}\,\backslash\text{NP}}>}{\text{S}}<
$$

Is NL mildly context-sensitive?



- CCG and TAG both define the same class. (Vijay–Shanker & Weir 1994).
- This class is called MCSL,
- or "nearly context free",
- or "type 1.9" in the Extended Chomsky Hierarchy.

Is NL mildly context-sensitive?



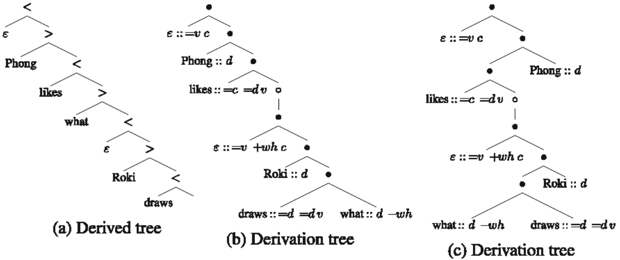
- CCG and TAG both define the same class.
  (Vijay–Shanker & Weir 1994).
- This class is called MCSL,
- or “nearly context free”,
- or “type 1.9” in the Extended Chomsky Hierarchy.

Conjecture

NL $\in$ MCSL

Another formalism defines a slightly larger class

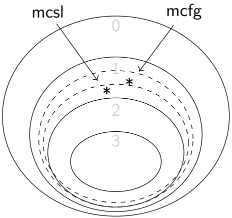
From the minimalist programme (Chomsky 1995), a formalism called Minimalist Grammars was introduced by Stabler (2011).



(a) Derived tree
(b) Derivation tree
(c) Derivation tree

credit: Stanojevič

MG are equivalent to MCFG



Other classes of languages:

- minimalist grammars (MG).
- multiple CFG (MCFGs).
- linear context-free rewrite systems (LCFRSs).
- etc.

Theorem (Stabler 2011)

$$CF \subsetneq \boxed{TAG \equiv CCG} \subsetneq \boxed{MCFG \equiv LCFRS \equiv MG} \subsetneq CS$$

Even more powerful formalisms?

Even if we assume that natural languages all belong to, say, the class of MCS languages, it might be a good idea to use even more powerful formalisms that may offer benefits regarding:

- conciseness,
- elegance,
- appropriateness for parsing,
- ...

At least three well-known syntactic formalisms have the property of being Turing-equivalent (i.e. type 0):

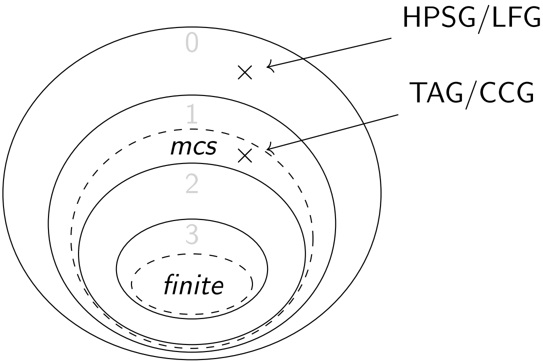
- Transformational grammars.
- HPSG.
- LFG.

Complexity can be elegant

- The language $\{a^n b^n \mid 1 \leq n \leq 1000\}$ is finite and therefore can be described by a regular grammar (with around 1000 non-terminal symbols).
- The CFG $S \rightarrow aSb \mid ab$ is a very small grammar that generates a possibly useful approximation.

- The language $\{a^{5i} \mid i \in \mathbb{N}\}$ can be described by a regular grammar with at least 5 non-terminal symbols.
- The CFG $S \rightarrow aaaaaS \mid \epsilon$ is a smaller grammar that generates exactly the same language.

A refined hierarchy



Day 2: Summary

- There are theoretical and practical reasons for determining where NL is in the Chomsky-Schützenberger hierarchy.
- center-embedding (very common) $\rightarrow$ NL is not regular.
- cross-serial dependencies (less common) $\rightarrow$ NL is not context-free.
- Good candidates: TAG/CCG and MCFG/LCFRS/MG.
- It can make sense to use much more powerful formalisms (e.g. HPSG).

- Bar-Hillel, Yehoshua, Micha Perles & Eliahu Shamir. 1961. On formal properties of simple phrase structure grammars. *STUF-Language Typology and Universals* 14(1-4). 143–172.
- Chomsky, Noam. 1995. *The minimalist program*. Cambridge, Mass.: MIT Press.
- Gibson, Edward & James Thomas. 1999. Memory limitations and structural forgetting: the perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes* 14(3). 225–248.
- Joshi, Aravind K. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Arnold M. Zwicky, David R. Dowty & Lauri Karttunen (eds.), *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives* (Studies in Natural Language Processing), 206–250. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511597855.007. (10 September, 2020).

- Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8(3). 333–343. https://doi.org/10.1007/BF00630917. (30 July, 2018).
- Stabler, Edward P. 2011. Computational Perspectives on Minimalism. In Cedric Boeckx (ed.), *The Oxford Handbook of Linguistic Minimalism*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199549368.013.0027.
- Steedman, Mark. 2000. *The syntactic process*. Vol. 24. MIT Press.
- Vijay–Shanker, K. & David J. Weir. 1994. The equivalence of four extensions of context–free grammars. *Mathematical Systems Theory* 27. 511–546.