

# The formal complexity of Natural Language(s)

Pascal Amsili (with the help of Timothée Bernard)

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

Cogmaster, october 2023

# Overview

## Introduction

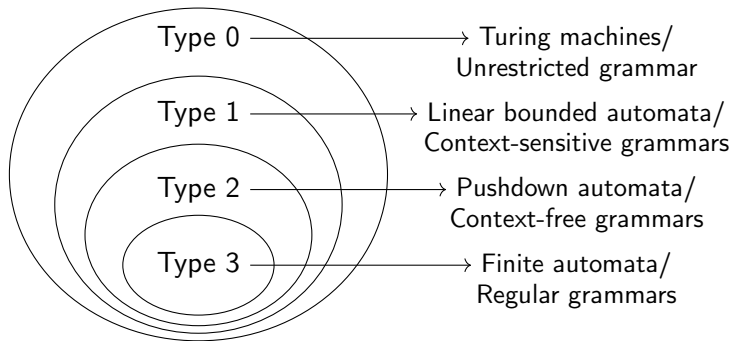
Is Natural Language regular?

Is Natural Language context-free?

Is Natural Language context-sensitive?

Linguistic formalisms

## Where are natural languages?



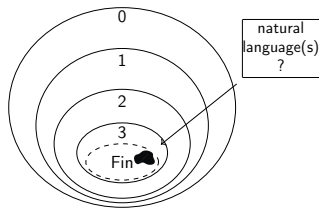
## Why should we care about the complexity of NL?

- ▶ 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

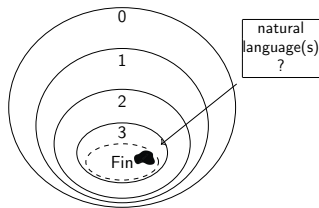
- ▶ Natural languages are all of comparable complexity or at least they can be grouped into classes of comparable complexity.
- ▶ Natural languages can be considered as formal languages:
  - ▶ Finite set of atomic symbols (morphemes?).
  - ▶ Binary grammaticality judgments for all sequences.

## Are natural languages finite?



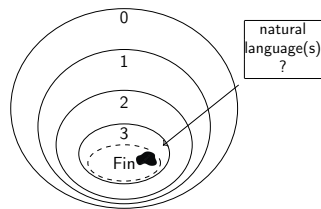
- ▶ NLs could be modelled as lists.
- ▶ It could still be interesting to use more powerful formalisms but for other reasons than complexity (conciseness, efficiency, suitability for semantics...).

## 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.

▶ lengthening:  $ab^n c$

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

▶ center-embedding:  $ab^n cd^n e$

- (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)<sup>n</sup> (hired)<sup>n</sup> 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)

# Overview

Introduction

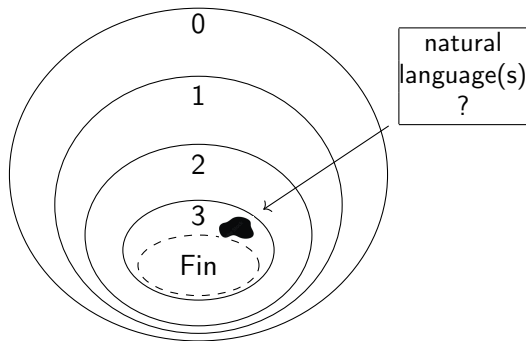
Is Natural Language regular?

Is Natural Language context-free?

Is Natural Language context-sensitive?

Linguistic formalisms

## Is natural language regular?



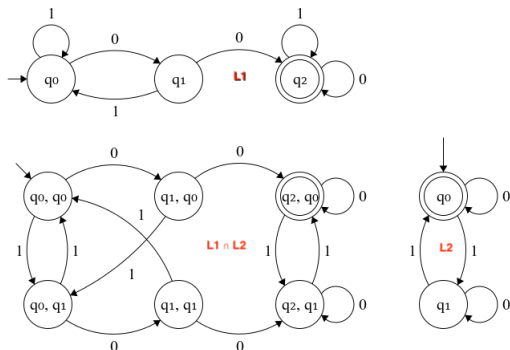
## Is NL regular? A. Closure property

### Closure property

The intersection of two regular languages is regular.

Proof: Construction of the product of two DFAs.

Example:



## 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  $q \xrightarrow{w_{i:j}} q$ .

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  $q \xrightarrow{w_{i:j}} q$ .

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.  $\forall n \in \mathbb{N}, xu^n y \in L$ .

→ “ $L$  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:

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

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

## Is NL regular? Pumping lemma (consequences)

regular	$\Rightarrow$	pumping property satisfied
pumping property <b>NOT</b> satisfied	$\Rightarrow$	<b>NOT</b> regular
pumping property satisfied	$\nRightarrow$	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 *as* than  $bs$ ).
- ▶ So no  $k \in \mathbb{N}$  works;  $L$  does not have the pumping property.

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

## Is NL regular? C. Proof (I)

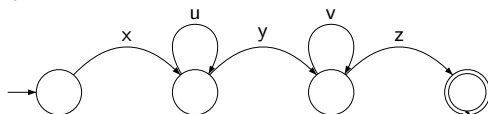
$L_1 = \{A \text{ man [that another man]}^n \text{ I saw [hired]}^m \text{ fired Sam.} \mid n, m \in \mathbb{N}\}$

This language is regular.

With

- ▶  $x = A \text{ man}$
- ▶  $u = \text{that another man}$
- ▶  $y = \text{I saw}$
- ▶  $v = \text{hired}$
- ▶  $z = \text{fired Sam}$

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



## Is NL regular? C. Proof (II)

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

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

In other words,  $\text{English} \cap L_1$  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:  $\Sigma^*$  is regular and contains all languages on  $\Sigma$ , even the most complex ones (beyond type 0).

## Is NL regular? C. Proof (III)

- ▶  $L_2 \subseteq$  English is not regular.

### Caution

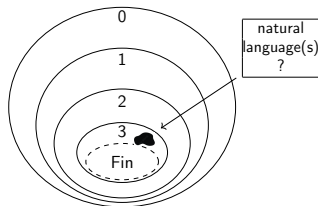
The fact that some non-regular language is a subset of English provides no indication of English being regular or not.

Ex:  $\Sigma^*$  is regular and contains all languages on  $\Sigma$ , 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 regular? **No!**



- A. Regular languages are closed under intersection.
- B.  $L_1 = \{ab^n cd^m e \mid n, m \in \mathbb{N}\}$  is regular.
  - ▶  $L_2 = \{ab^n cd^n e \mid n \in \mathbb{N}\}$  is not regular.
- C. The intersection of English with a regular language ( $L_1$ ) is not regular ( $L_2$ ).
  - ▶ Therefore, English is not regular.



# Overview

Introduction

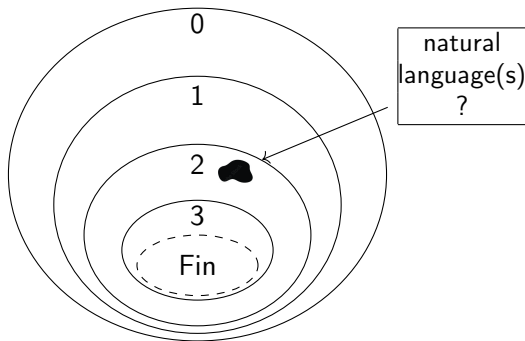
Is Natural Language regular?

**Is Natural Language context-free?**

Is Natural Language context-sensitive?

Linguistic formalisms

## Is natural language 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**.

$$S \rightarrow AB$$

$$A \rightarrow cc$$

$$| aSa$$

$$B \rightarrow b$$

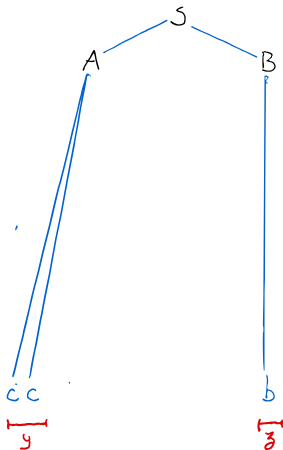
$$S \Rightarrow AB \Rightarrow ccB \Rightarrow ccb$$

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

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

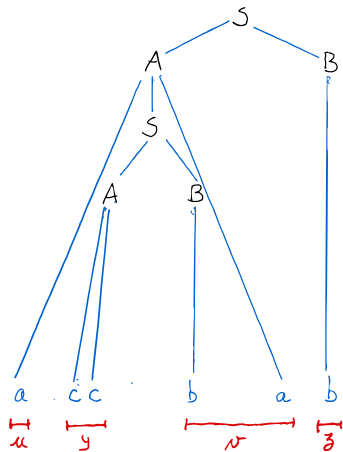
If a non-terminal  $A$  generates itself once in a derivation, since the grammar is context-free, then there is no way to prevent  $A$  from generating itself an arbitrary number of times.

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



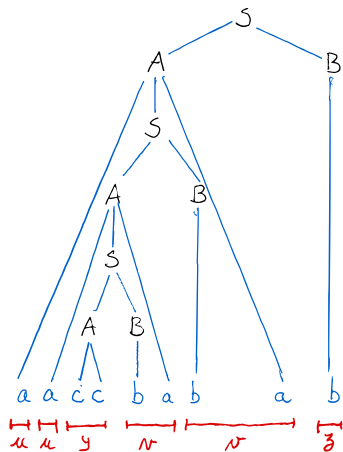
$S$	$\rightarrow$	$AB$
$A$	$\rightarrow$	$cc$
	$ $	$aSa$
$B$	$\rightarrow$	$b$

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



S	→	AB
A	→	cc
		aSa
B	→	b

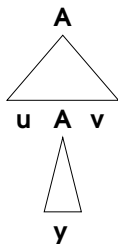
## Is NL context-free? B. Pumping lemma (intuition)



S	→	AB
A	→	cc
		aSa
B	→	b



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



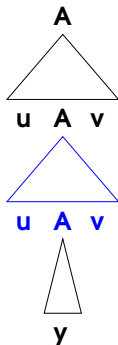
- ▶ If there is a productive derivation  $A \xRightarrow{*} y$ ,
- ▶ and a “recursive” situation  $A \xRightarrow{*} uAv$ ,
- ▶ then any identical number of embedded factors  $u$  and  $v$  can be produced.

$$A \xRightarrow{*} uAv$$

$$A \xRightarrow{*} uAv \xRightarrow{*} u^2v^2$$

$$A \xRightarrow{*} uAv \xRightarrow{*} \underbrace{u \dots u}_n A \underbrace{v \dots v}_n \xRightarrow{*} u^n v^n$$

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



$$A \xRightarrow{*} uAv$$

$$A \xRightarrow{*} uAv \xRightarrow{*} uyv$$

$$A \xRightarrow{*} uAv \xRightarrow{*} \underbrace{u \dots u}_n A \underbrace{v \dots v}_n \xRightarrow{*} u^n y v^n$$

- ▶ If there is a productive derivation  $A \xRightarrow{*} y$ ,
- ▶ and a “recursive” situation  $A \xRightarrow{*} uAv$ ,
- ▶ 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 *et al.* , 1961)

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

context-free	$\Rightarrow$	pumping property satisfied
pumping property <b>NOT</b> satisfied	$\Rightarrow$	<b>NOT</b> context-free
pumping property satisfied	$\nRightarrow$	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  $c$ s than  $a$ s, or strictly less  $a$ s than  $c$ s).
- ▶ 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 (Schieber, 1985): Cross-serial dependencies:

- (7) Jan säit das mer [em Hans]<sub>1</sub> [es huus]<sub>2</sub> [hälfed]<sub>1</sub>,  
Jan says that we Hans the house helped  
[aastriche]<sub>2</sub>.  
paint.  
'Jan says that we [helped]<sub>1</sub> [Hans]<sub>1</sub> [paint]<sub>2</sub> [the house]<sub>2</sub>.'

- ▶ 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.  
→ cross-serial dependency.
- ▶ 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]<sup>n<sub>1</sub></sup> [em Hans]<sup>n<sub>2</sub></sup> es huus haend wele  
laa<sup>n<sub>3</sub></sup> hälfe<sup>n<sub>4</sub></sup> aastrüche.

## 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:

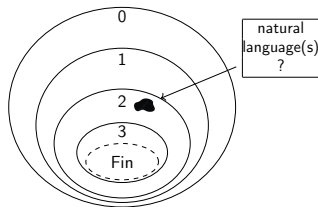
- ▶  $w = \text{Jan säit das mer}$
- ▶  $a = \text{d'chind}$
- ▶  $b = \text{em Hans}$
- ▶  $x = \text{es huus haend wele}$
- ▶  $c = \text{laa}$
- ▶  $d = \text{hälfe}$
- ▶  $y = \text{aastriiche}$

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-free? **No!**



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

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

B.  $L_2 = \{wa^n b^m x c^n d^m y \mid n, m \in \mathbb{N}\}$  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.

# Overview

Introduction

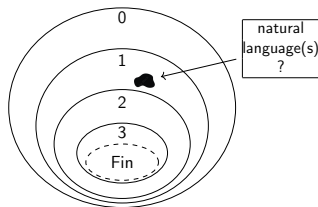
Is Natural Language regular?

Is Natural Language context-free?

**Is Natural Language context-sensitive?**

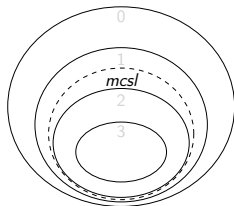
Linguistic formalisms

## 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 powerful 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.

# Overview

Introduction

Is Natural Language regular?

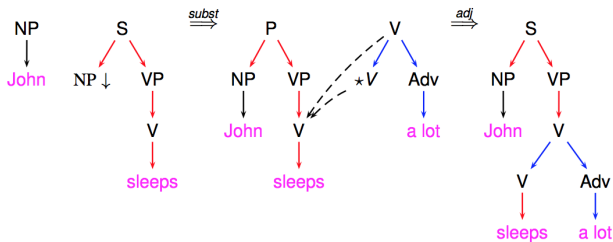
Is Natural Language context-free?

Is Natural Language context-sensitive?

Linguistic formalisms

## 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.



## TAG languages = MCSL

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

- ▶  $\{ww \mid w \in \Sigma^*\}$  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.
- ▶  $\{www \mid w \in \Sigma^*\}$  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): developed by Steedman (e.g. 2000).

Phrase structure rules are replaced with:

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

$$\begin{array}{ccc} \text{Sabine} & \text{likes} & \text{books} \\ \hline \text{NP} & (S \backslash NP) / NP & \text{NP} \end{array}$$



## CCGs define exactly the same class

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

Phrase structure rules are replaced with:

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

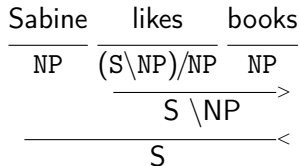
$$\frac{\frac{\text{Sabine}}{NP} \quad \frac{\text{likes}}{(S \backslash NP) / NP} \quad \frac{\text{books}}{NP}}{S \backslash NP} >$$

## CCGs define exactly the same class

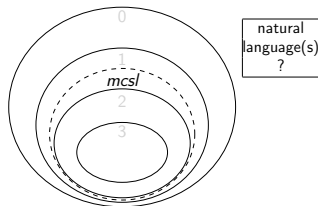
Combinatory Categorial Grammar (CCG): developed by Steedman (e.g. 2000).

Phrase structure rules are replaced with:

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

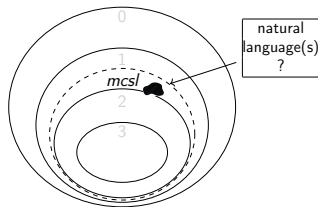


## Is NL mildly context-sensitive?



- ▶ 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?



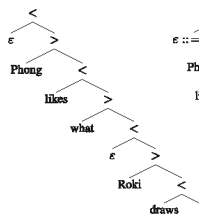
- ▶ CCG and TAG both define the same class.  
(Vijay-Shanker & Weir, 1994).
- ▶ This class is called MCSL,
- ▶ or “nearly context free”,
- ▶ or “type 1.9” in the Extended Chomsky Hierarchy.

Conjecture

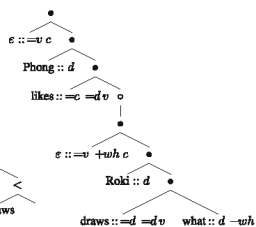
$$NL \in MCSL$$

## Another formalism defines a slightly larger class

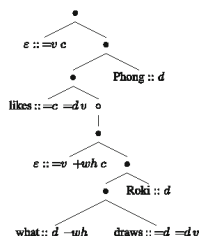
From the minimalist programme (Chomsky, 1995), a formalism called Minimalist Grammars was introduced by Stabler (2011).



(a) Derived tree



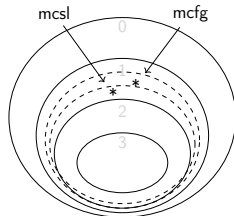
(b) Derivation tree



(c) Derivation tree

credit: Stanojević

## 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):

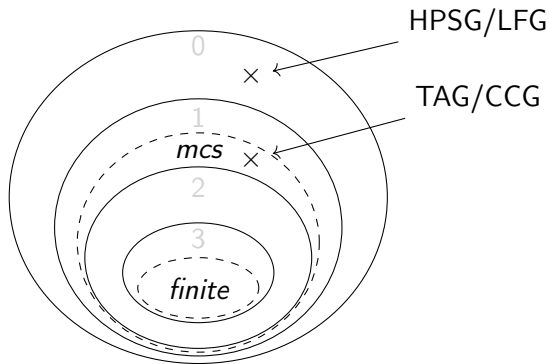
- ▶ Transformational grammars.
- ▶ HPSG.
- ▶ LFG.

## 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  $\{a^{5^i} \mid i \in \mathbb{N}\}$  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



## 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. 1995. *The Minimalist Program*. Vol. 28. Cambridge, Mass.: MIT Press.
- Gibson, Edward, & Thomas, James. 1999. Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3), 225–248.
- Joshi, Aravind Krishna. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? *Pages 206–250 of: Zwicky, Arnold M., Dowty, David R., & Karttunen, Lauri (eds), Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*. Studies in Natural Language Processing. Cambridge: Cambridge University Press.
- 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. 2000. *The syntactic process*. Vol. 24. MIT Press.
- Vijay-Shanker, K., & Weir, David J. 1994. The Equivalence of Four Extensions of Context-Free Grammars. *Mathematical Systems Theory*, 27, 511–546.