EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Syntax & Semantics WiSe 2022/2023

Lecture 8: The Chomsky Hierarchy

22/11/2022, Christian Bentz

Overview

Q&As

Why is English coded as having three genders in the Chapter on Number of Genders in the WALS chapter?

Corbett writes: “Our examples have involved agreement of the verb, but there are various other targets which may agree in gender, such as adjectives, determiners, numerals and even focus particles. Most scholars working on agreement include the control of anaphoric pronouns by their antecedent (*the girl ... she*) as part of agreement. If this is accepted, as we do here, then languages in which free pronouns present the only evidence for gender will be counted as having a gender system. Of course, such languages with **pronominal gender systems** have a much less pervasive system than those like Russian. Including them, however, makes little difference to the overall picture, since they are rare (the best known example is English, which is typologically unusual in this respect); [...]”

Section 1: Recap of Lecture 7

Grammar in Formal Language Theory

A **grammar** $\mathcal{G}$ in formal language theory is then a quadruple consisting of the set of terminal symbols, non-terminal symbols, a starting symbol S, and a set of rewrite rules R:

$$\langle T, NT, S, R \rangle^{1} \tag{1}$$

Jäger and Rogers (2012). Formal language theory: refining the Chomsky hierarchy.
Partee et al. (1990). Mathematical methods in linguistics.

---

[1] S is a “distinguished member” of NT.

Rewrite Rules

In the most general definition, **rewrite rules** define how we can rewrite a string of symbols into another string of symbols. We formally have

$$\alpha \rightarrow \beta, \tag{2}$$

where α is a string of n symbols $(x_1, x_2, x_3, \ldots, x_n)$, with $n \geq 1$, for which $x_i \in (T \cup NT)$, and, likewise, β is a string of symbols $(y_1, y_2, y_3, \ldots, y_n)$ for which $y_i \in (T \cup NT)$.

In words: α and β are strings which are made up of terminal symbols, non-terminal symbols, or both. For example, a noun phrase involving a determiner and a noun can be rewritten as follows:

NP → DET N
NP → the N
NP → the tree

Language in Formal Language Theory

"The set of all strings that $\mathcal{G}$ can generate is called the language of $\mathcal{G}$, and is notated L($\mathcal{G}$)."

Jäger and Rogers (2012). Formal language theory: refining the Chomsky hierarchy, p. 1957

We thus might imagine a language as a multiset of words and strings of words licensed by the respective rewrite rules:

$$L(\mathcal{G}) = \{(w_1), (w_2), \dots (w_n), (w_1, w_2), \dots (w_1, \dots w_m)\}, \qquad (3)$$

where w_i is a terminal symbol, i.e. word in our case, n is the overall number of terminal symbols, i.e. the cardinality $|T|$; and m is the maximum length of strings (could be ∞). Note that each string here has to be licensed by the rewrite rules.

Note: $L(\mathcal{G})$ has to be a multiset, since the same strings can occur multiple times.

Passive Transformations

Passive constructions are handled in some syntactic frameworks (e.g. Government and Binding) with the same underlying deep structure as **active constructions**. Note that this is an important deviation from traditional PSGs. In a traditional PSG you would have to formulate different phrase structure rules for active and passive sentences.

Early example of a transformational rule going back to Chomsky (1957):

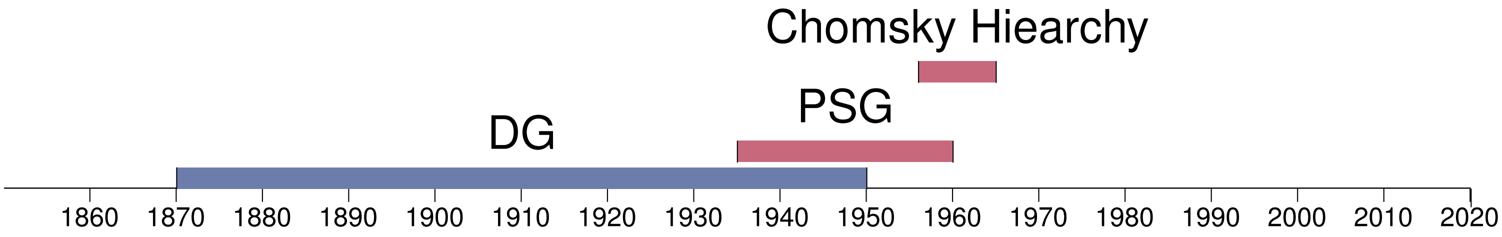
$\mathrm{NP}_1\ \mathrm{V}_2\ \mathrm{NP}_3 \rightarrow \mathrm{NP}_3\ [_{AUX}\ \mathrm{be}]\ \mathrm{V}_2\mathrm{en}\ [_{PP}\ [_{P}\ \mathrm{by}]\ \mathrm{NP}_1]$
John sees Mary $\rightarrow$ Mary $[_{AUX}$ is] seen $[_{PP}$ $[_{P}$ by] John]

Müller (2019). Grammatical theory, p. 85.

Section 2: Historical Notes

Historical Perspective

The articles which contributed to the hiearchy of grammar formalisms which later became known as the **Chomsky Hierarchy**, or **Chomsky-Schützenberger Hierarchy**, were written in the late 1950s, early 1960s. This work is strongly interlinked with the development of Phrase Structure Grammars (PSGs).



Note: The chronology bars indicate the rough time period where the first and foundational works relating to a framework were published. All of the theories discussed here still have repercussions also in current syntactic research.

Q&As Lecture 7

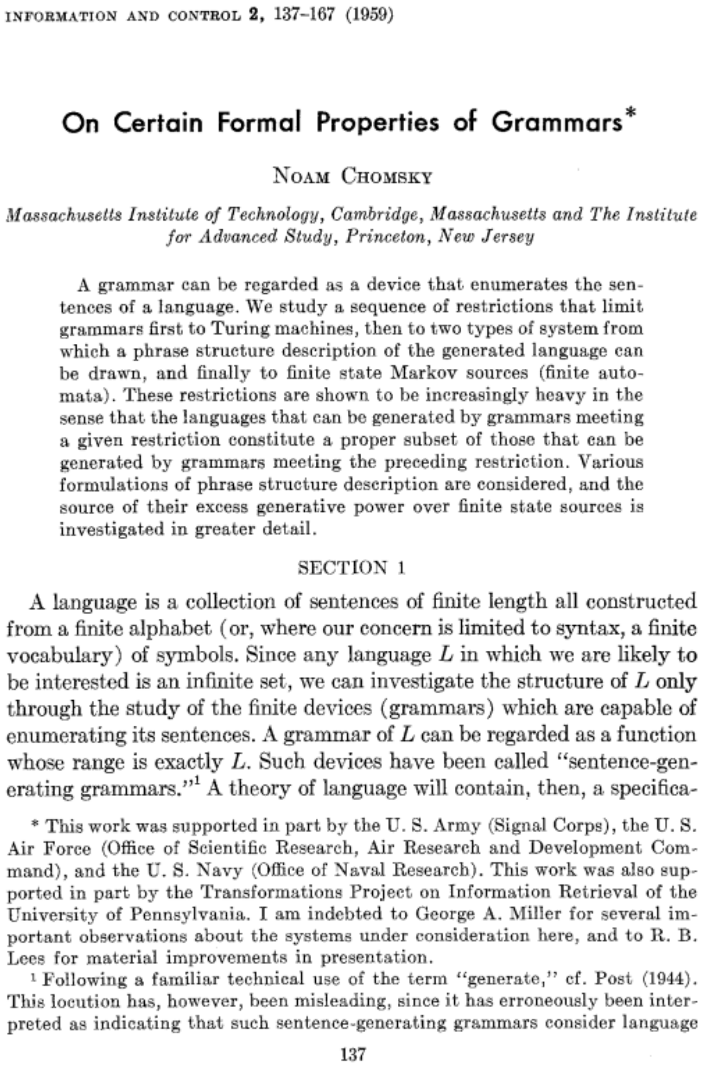
Section 1: Recap of Lecture 7

Section 2: Historical Notes

Section 3: The Chomsky Hierarchy

Section 4: The Sub-Hierarchies

References



INFORMATION AND CONTROL 2, 137–167 (1959)

On Certain Formal Properties of Grammars*

NOAM CHOMSKY

Massachusetts Institute of Technology, Cambridge, Massachusetts and The Institute for Advanced Study, Princeton, New Jersey

A grammar can be regarded as a device that enumerates the sentences of a language. We study a sequence of restrictions that limit grammars first to Turing machines, then to two types of system from which a phrase structure description of the generated language can be drawn, and finally to finite state Markov sources (finite automata). These restrictions are shown to be increasingly heavy in the sense that the languages that can be generated by grammars meeting a given restriction constitute a proper subset of those that can be generated by grammars meeting the preceding restriction. Various formulations of phrase structure description are considered, and the source of their excess generative power over finite state sources is investigated in greater detail.

SECTION 1

A language is a collection of sentences of finite length all constructed from a finite alphabet (or, where our concern is limited to syntax, a finite vocabulary) of symbols. Since any language L in which we are likely to be interested is an infinite set, we can investigate the structure of L only through the study of the finite devices (grammars) which are capable of enumerating its sentences. A grammar of L can be regarded as a function whose range is exactly L. Such devices have been called "sentence-generating grammars."[1] A theory of language will contain, then, a specifica-

---

* This work was supported in part by the U. S. Army (Signal Corps), the U. S. Air Force (Office of Scientific Research, Air Research and Development Command), and the U. S. Navy (Office of Naval Research). This work was also supported in part by the Transformations Project on Information Retrieval of the University of Pennsylvania. I am indebted to George A. Miller for several important observations about the systems under consideration here, and to R. B. Lees for material improvements in presentation.

[1] Following a familiar technical use of the term "generate," cf. Post (1944). This locution has, however, been misleading, since it has erroneously been interpreted as indicating that such sentence-generating grammars consider language

137

Section 3: The Chomsky Hierarchy

Natural Language Examples

Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References

$\alpha \rightarrow \beta$
the → a
the tree → this
the huge tree → the tree
the huge tree bends in the wind → the
the → the huge tree bends in the wind
VP → NP bends NP NP
NP VP NP NP → NP VP
DET → the
the → DET
NP → the N
NP → DET N
VP → NP VP

In a series of publications, Chomsky (1956, 1957, 1959, 1963) and Chomsky & Schützenberger (1965) discussed which *further restrictions* to re-write rules could be applied to more realistically match the generative capacity of natural languages.

Notational Conventions

- Lower case Latin letters of the beginning of the alphabet are **terminal symbols**, i.e. $a, b, c \in T$. The ones from the end of the alphabet, i.e. x, y, z are used as *variables* representing any possible terminal symbol.
- Upper case Latin letters represent **non-terminal symbols**, i.e. $A, B, C \in NT$, with X, Y, Z being *variables* again. S is the starting symbol.
- Lower case Greek letters, i.e. α, β, ω, represent **strings of terminal and non-terminal symbols**.
- We use the symbol ϵ for the **empty string**.

Regular Languages (Type 3)

Definition: Finite State Grammar

In so-called **regular**, or **finite state grammars**, the rewrite rules are restricted to two forms:

$$\mathrm{X} \rightarrow \mathrm{x}$$
$$\mathrm{X} \rightarrow \mathrm{xY}$$

In words, any non-terminal on the left-hand side of a rule can only be rewritten either into a terminal, or into a terminal followed by a non-terminal.

Jäger & Rogers (2012), p. 1958.

Notes: Jäger & Rogers (2012) just use Latin letters from the beginning of the alphabet here, i.e. A → a, A → aB. Moreover, remember from typical mathematical functions like $f(x, y) = x + y^2$, that x and y might represent different numbers, *but they do not have to*, i.e. it is possible that $x = y$. Also, we could have the rule X → Yx instead of the one above (but we could not mix them according to Jäger & Rogers 2012, footnote 6).

Beware: Different Usages of “Regular”

Interestingly, in his early writing, Chomsky uses the term **finite state** for grammars with rewrite rules of the form

$$X \rightarrow x$$
$$X \rightarrow xY$$

But he reserves the term **regular** for grammars with rewrite rules of the form

$$X \rightarrow x$$
$$X \rightarrow \mathbf{YZ},$$

where (in this particular case) $Y \neq Z$.

Chomsky (1959), p. 149.

Note: According to the definition by Chomsky (1959), binarized phrase structure grammars as discussed in the last lectures could be called “regular” grammars. However, in Jäger & Rogers (2012) the term “regular” refers to finite state rewrite rules above.

Abstract Example

Rewrite	Rule
S	–
aS	1
aaS	1
aaaS	1
aaaa	2

Terminals

$T = \{a\}$

Non-Terminals

$NT = \{S\}$

R

1. S → aS
2. S → a

Note: The language generated with this grammar is $L(\mathcal{G}_R) = \{a, aa, aaa, \ldots, a^n\}$, with $n \in \mathbb{N}$, and hence in theory we can have $n = \infty$ due to the recursive rule. If we add the constraint that each rewrite rule has to be applied at least once, then the first element of this set would be *aa*.

Natural Language Example

Rewrite	Rule
S	–
she VP	1
she saw NP	2
she saw the N	3
she saw the axolotl	4

Terminals

$T = \{axolotl, bunny, saw, she, the\}$

Non-Terminals

$NT = \{N, NP, VP, S\}$

R

1. S $\rightarrow$ she VP
2. VP $\rightarrow$ saw NP
3. NP $\rightarrow$ the N
4. N $\rightarrow$ axolotl
5. N $\rightarrow$ bunny

The language generated with this grammar is
$L(\mathcal{G}_R) = \{\text{she saw the axolotl}, \text{she saw the bunny}\}$.

Further Examples of Regular Languages $L(\mathcal{G}_R)$

- The set of strings which follows the pattern $x^n y^m$, e.g. $L(\mathcal{G}_R) = \{ab, aab, abb, aabb, \dots\}$[2]
- The set of strings such that the number of ‘a’s in them is a multiple of 4, i.e. $L(\mathcal{G}_R) = \{aaaa, aaaaaaaa, aaaaaaaaaaaa, \dots\}$
- The set of natural numbers that leave a remainder of 3 when divided by 5, i.e. $L(\mathcal{G}_R) = \{8, 13, 18, \dots\}$
- etc.

Jäger & Rogers (2012), p. 1958.

---

[2] If we include a rule which yields an empty element, e.g. $x \rightarrow \epsilon$, then *a*, *b*, and ϵ would also be part of this set.

Are Natural Languages Fully **Regular**?[3]

While there are certainly many sentences in natural languages which can be generated by regular grammars (maybe even the vast majority?), Chomsky argued in some of his earliest work that structures such as

1. **If** S_1, **then** S_2,
2. **Either** S_3, **or** S_4,
3. The **man** who said that S_5, **is** arriving today,

bear dependencies which hinder a strictly regular generation.

Chomsky (1956), p. 115.

Note: The Ss here stand in for declarative sentences. The dependencies connect the words in bold face.

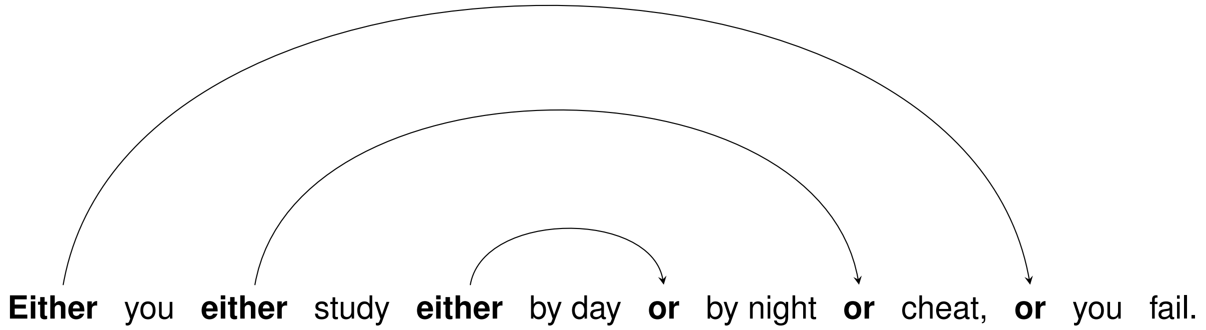
Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References

---

[3] In the sense “equivalent to finite state languages”.

Non-Regular Example

Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References



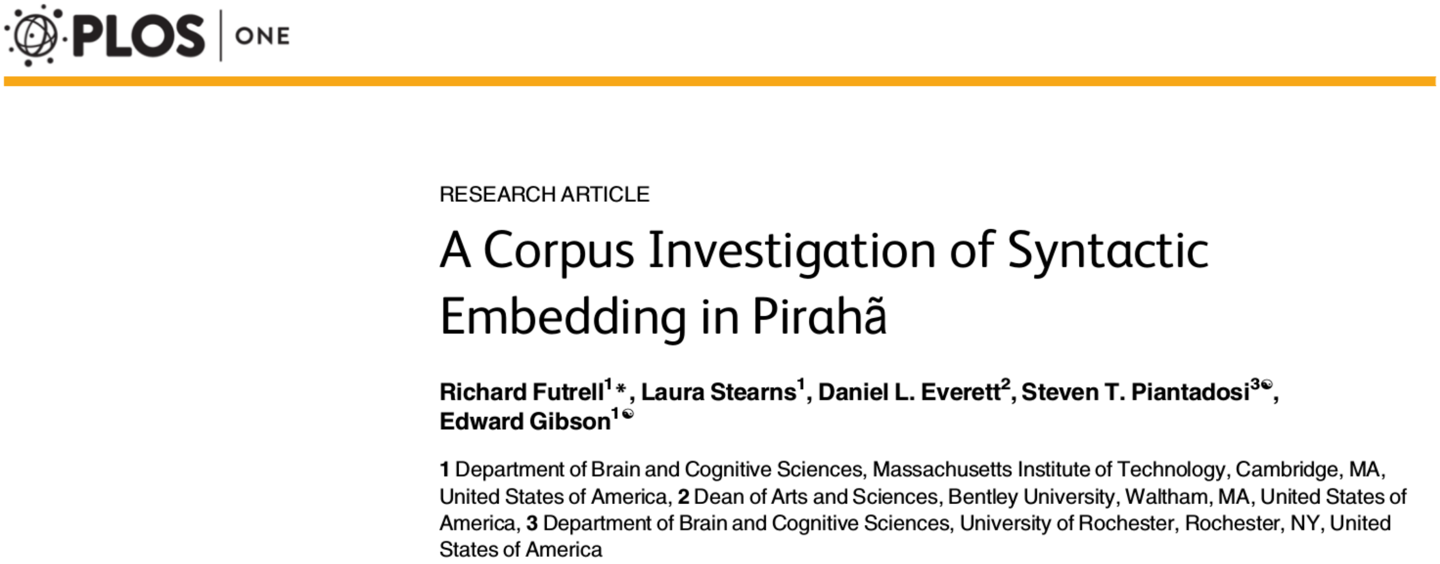
(1) **Either** you **either** study **either** by day **or** by night **or** cheat, **or** you fail.

either ... either ... either ... or ... or ... or
a ... a ... a ... b ... b ... b

$a^n b^n$

Under the assumption that such constructions can be applied productively *ad infinitum*, a regular grammar could not generate *these and only these* sentences. While the *non-regular* language $L(\mathcal{G}_{NR}) = \{ab, aabb, aaabbb, \ldots, a^n b^n\}$ is a proper subset of the regular language $L(\mathcal{G}_R) = \{ab, aab, abb, aabb, \ldots, a^n b^m\}$, i.e. $L(\mathcal{G}_{NR}) \subset L(\mathcal{G}_R)$, there is no way to identify this subset with a finite state automaton.

Pirahã (myp, Isolate): A Fully Regular Language?



PLOS | ONE

RESEARCH ARTICLE

A Corpus Investigation of Syntactic Embedding in Pirahã

Richard Futrell[1]*, Laura Stearns[1], Daniel L. Everett[2], Steven T. Piantadosi[3]☯, Edward Gibson[1]☯

1 Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, United States of America, **2** Dean of Arts and Sciences, Bentley University, Waltham, MA, United States of America, **3** Department of Brain and Cognitive Sciences, University of Rochester, Rochester, NY, United States of America

Syntactic Embedding

- Embedded possessives: *[[[the woman]'s sister]'s husband]*
- Reported Speech: *He said [that she said [that . . .]]*
- Sentential complements: *I dreamed that the Brazilian woman was there last night*
- Adverbials: *because x, x*
- Relative clauses: *the food that the man devoured*
- Coordination: *John and Mary and Bill and ...*

Futrell at al. (2016). A corpus investigation of syntactic embedding in Pirahã.

Proposal of Regular Grammar

“We found no unambiguous evidence for sentential or NP embedding in Pirahã in our corpus. The corpus is consistent with the hypothesis that Pirahã is a regular language; [...] In order to flesh out our claim that the corpus is consistent with a regular grammar, we give here a regular expression (technically an egrep expression) which is consistent with the corpus. The symbol `S` matches all sentences in the corpus:

```
 S = NPtopic?  NPtopic?  NPvoc?  NPsubj NPsubj?  NPsubj?
NPtmp?  NPloc?  NPiobj?  (JJobj | NPobj NPobj?)?  NPiobj?  V
JJobj?  NPvoc?  NPtopic?
```

where `X?` means optional `X` , `(X|Y)` means `X` or `Y`, and each of the symbols above expand into other regular expressions (ignoring morphology and null nouns/verbs) [...]”

Futrell at al. (2016). A corpus investigation of syntactic embedding in Pirahã, p. 17.

Summary: Regular Grammars (Type 3)

- **Regular grammars** are already very powerful. They can generate, for instance, a set of strings which reflects the natural numbers $\mathbb{N}$, i.e. $L(\mathcal{G}_R) = \{a, aa, aaa, \ldots, a^n\}$.
- They can (in principle) generate all sentences in natural languages too, however...
- Firstly, for certain constructions, e.g. of the a^nb^n type, they will also generate ungrammatical sentences...
- Secondly, since at least one terminal symbol has to be produced in every rewrite, the generation of sentences is not very “elegant”, in the sense that no higher level patterns (phrase structures) can be captured.

Context-Free Languages (Type 2)

Definition: Context-Free Grammar

In the original work of Chomsky, the most general rewrite rule from above, i.e.

$$\alpha \rightarrow \beta, \tag{4}$$

is further restricted by, firstly, adding a **context** to each side, i.e. $\varphi_1\_\varphi_2$, and secondly, by requiring that α is a single non-terminal X such that we have

$$\varphi_1 X \varphi_2 \rightarrow \varphi_1 \beta \varphi_2 \tag{5}$$

Now, if this context is **defined to be *null***, we call this a **context-free** grammar (hence the name). We thus have rewrite rules of the general form

$$X \rightarrow \beta \tag{6}$$

In words, we only allow one *single non-terminal symbol* on the left-hand side of the arrow, but an arbitrary string of terminals and non-terminals on the right-hand side.

Chomsky (1959), p. 142.
Jäger and Rogers (2012), p. 1957.

Abstract Example

Rewrite	Rule
S	—
aSb	1
aaSbb	1
aaaSbbb	1
aaabbb	2

Terminals

$T = \{a, b, \epsilon\}$

Non-Terminals

$NT = \{S\}$

R

1. S $\rightarrow$ aSb
2. S $\rightarrow$ ϵ

The language generated with this grammar is
$L(\mathcal{G}_{CF}) = \{ab, aabb, aaabbb, \ldots, a^n b^n\}$, with $n \in \mathbb{N}$.

Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References

Natural Language Example

Rewrite	Rule
S	—
NP V NP	1
PRON V NP	3
PRON V DET N	2
she saw the axolotl	4, 5, 6, 8

Terminals

$T = \{axolotl, bunny, saw, she, the\}$

Non-Terminals

$NT = \{N, NP, V, PRON, S\}$

R

1. S $\rightarrow$ NP V NP
2. NP $\rightarrow$ DET N
3. NP $\rightarrow$ PRON
4. DET $\rightarrow$ the
5. V $\rightarrow$ saw
6. N $\rightarrow$ axolotl
7. N $\rightarrow$ bunny
8. PRON $\rightarrow$ she

Natural Language Example

The language generated with this grammar is $L(\mathcal{G}_{CF}) =$ {she saw the axolotl, she saw the bunny, the axolotl saw she, the bunny saw she, the axolotl saw the axolotl, the axolotl saw the bunny, the bunny saw the bunny, the bunny saw the axolotl}.

Terminals

$$T = \{axolotl, bunny, saw, she, the\}$$

Non-Terminals

$$NT = \{N, NP, V, PRON, S\}$$

R

1. S → NP V NP
2. NP → DET N
3. NP → PRON
4. DET → the
5. V → saw
6. N → axolotl
7. N → bunny
8. PRON → she

Further Examples of Context-Free Languages

- *Mirror language*: the set of strings $\gamma\omega$ over a set of terminals T such that ω is the mirror image of γ, e.g. $L(\mathcal{G}_{CF}) = \{abba, abccba, abcddcba, \ldots\}$
- *Palindrome language*: the set of strings γ that are identical to their mirror image, e.g. $L(\mathcal{G}_{CF}) = \{aba, bab, abba, baab, aabaa, \ldots\}$
- Languages with strings of the form $x^n y^m z^m w^n$.
- etc.

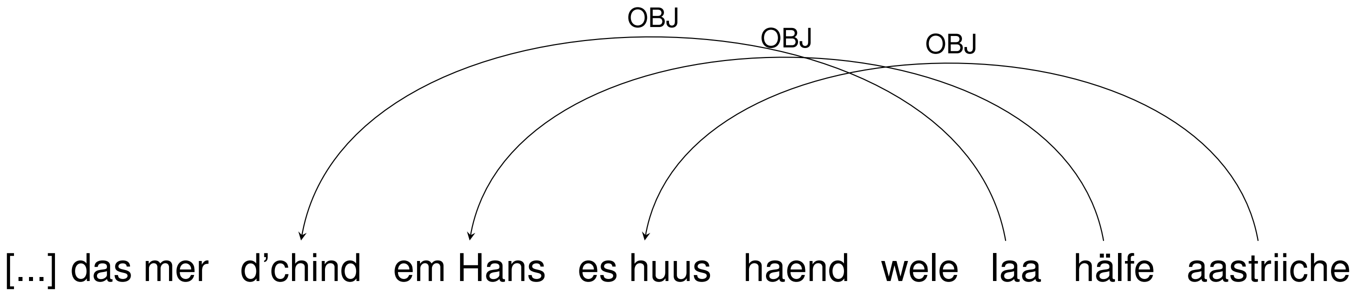
Jäger & Rogers (2012), p. 1958.

Are Natural Languages **Context-Free**?

This was a debated topic in *Formal Language Theory* (FLT) since the first formulations of the types of languages in the 1950s and 1960s. It took until the mid 1980s (!) for an alleged **non-context-free sentence structure** to be proposed and (apparently) accepted by most syntacticians.

Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References

Swiss German (Indo-European)



[...] das mer d'chind em Hans es huus haend wele laa hälfe aastriiche

“[...] that we have wanted to let the children help Hans paint the house.”

Shieber (1984). Evidence against the context-freeness of natural language.

Non-Context-Free Example (?)

(2)	[...]	das mer	[**d'chind**]	[**em**	**Hans**]	es	huus	haend	wele	[**laa**]	[**haelfe**]	aastriiche
	[...]	that we	children.**ACC**	the.**DAT**	Hans	the.ACC	house	have	wanted	**let**	**help**	paint

"[...] that we have wanted to let the children help Hans paint the house."

Since *laa* assigns accusative case to *d'chind*, and *haelfe* assigns dative case to *em Hans*, Shieber argues that we are looking at an $a^m b^n c^m d^n$ pattern, with

a = *d'chind*,
b = *em Hans*,
c = *laa*,
d = *hälfe*.

This pattern has been shown before to constitute an example of a non-context-free language.

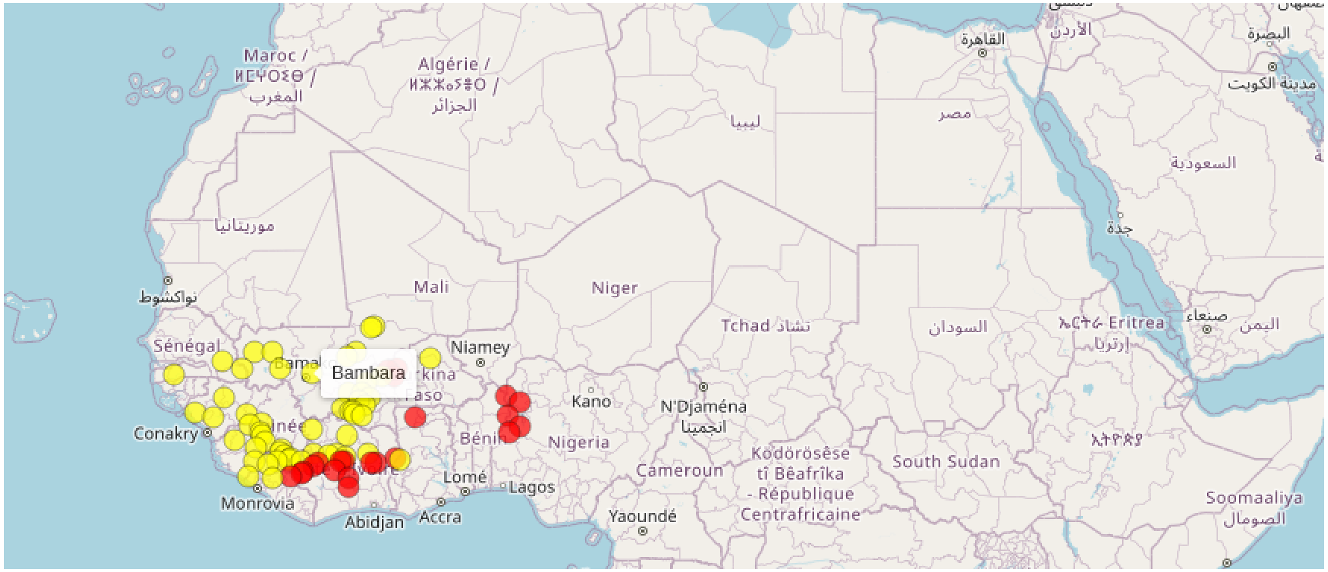
Some Problems with Shieber's Argument

- At the face of it, there is no repetition of *any* of the terminal elements in Shieber's example. That is – even without further consulting Swiss German speakers – it seems clear that Swiss German would not allow patterns of the kind: *d'chind d'chind* [...] *laa laa*. So we are rather looking at $a^1 b^1 c^1 d^1$ patterns.
- Shieber seems to use *m* and *n* here as *indeces* rather than variables standing in for power numbers. However, in FLT such a usage is highly unusual. Note that, for instance, the $a^n b^n$ pattern proposed by Chomsky does actually refer to repetitions of 'a's and 'b's, e.g. the terminals *either* and *or*.

Another proof based on Bambara (Mande, bam)

> “In this paper I look at the possibility of considering the vocabulary of a natural language as a sort of language itself. In particular, I study the weak generative capacity of the vocabulary of Bambara, and show that vocabulary is not context free. This result has important ramification the theory of syntax of natural language.”

Culy (1985). The complexity of the vocabulary of Bambara, p. 345.



Bambara Vocabulary Structure

Noun *o* Noun construction:

(3)

wulu	o	wulu
dog	PRT	wulu

“whichever dog”

Agentive construction: Noun + Transitive Verb + *la*

(4) wulunyinina

wulu	nyini	la
dog	search.for	PRT

“one who searches for dogs”, i.e. “dog searcher”

Agentive construction with recursive application:

(5) wulunyininanyinina

wulu	nyini	la	nyini	la
dog	search.for	PRT	search.for	PRT

“one who searches for dog searchers”

Bambara Vocabulary Structure

Noun o Noun construction and Agentive construction together:

(6) wulunyinina o wulunyinina

wulu	nyini	la	o	wulu	nyini	la
dog	search.for	PRT	PRT	dog	search.for	PRT

“whichever dog searcher”

(7) wulunyininanyinina o wulunyininanyinina

wulu	nyini	la	nyini	la	o	wulu	nyini	la	nyini	la
dog	search.for	PRT	search.for	PRT	PRT	dog	search.for	PRT	search.for	PRT

“whichever searcher of dog searchers”

Q&As Lecture 7
Section 1: Recap of Lecture 7
Section 2: Historical Notes
Section 3: The Chomsky Hierarchy
Section 4: The Sub-Hierarchies
References

Culy's proof in a nutshell

A subset of the Vocabulary ($\mathcal{B}$) in Bambara is of the form:

$$\mathcal{B}' = \{\text{wulu}^m\text{nyinina}^n\,\text{o}\,\text{wulu}^m\text{nyinina}^n | m, n \geq 1\}. \tag{7}$$

if we assume that this is the case for all nouns *a* and the suffix *-nyinina* (*b*), this yields the more general form:[4]

$$\mathcal{B}' = \{\text{a}^m\text{b}^n\text{a}^m\text{b}^n | m, n \geq 1\}. \tag{8}$$

Since $\mathcal{B}'$ is not context free, and a subset of the overall vocabulary ($\mathcal{B}' \subset \mathcal{B}$), it follows that $\mathcal{B}$ is also not context free.

Culy (1985), p. 349.

Problem: The formulations in the above equations are not fully correct, since, according to Culy's analysis of Bambara vocabulary, $m = 1$ and $n \geq 1$. In other words, the noun cannot be multiplied, only the suffix. So, strictly speaking, we look at a pattern ab^{n}ab^n.

---

[4] The "o" is here dropped in the general form without loss of generality.

Summary: Context-Free Grammars (Type 2)

- **Context-free grammars** are more powerful than regular grammars by allowing more diverse sentences to be generated.
- If we take the **binarized version** of CFG, this essentially boils down to having one additional rule pattern compared to regular grammars: $X \rightarrow YZ$[5]
- It has been argued (and largely accepted) that some (very rare) patterns in natural languages might require **non-context-free** rules to be generated.

---

[5] In this case, it is allowed that $Y = Z$.

Context-Sensitive Languages (Type 1)

Definition: Context-Sensitive Grammar

In the original work of Chomsky, the most general rewrite rule from above, i.e.

$$\alpha \rightarrow \beta, \tag{9}$$

is further restricted by, firstly, adding a **context** to each side, i.e. $\varphi_1\_\varphi_2$, and secondly, by requiring that α is a single non-terminal X such that we have

$$\varphi_1 X \varphi_2 \rightarrow \varphi_1 \beta \varphi_2 \tag{10}$$

Now, if this context **may be *null*** (but does not have to be), we call this a **context sensitive** grammar (hence the name).

Chomsky (1959), p. 142.

Definition: Alternative Version

Chomsky subsequently proves that it is possible to give an alternative definition of Type 1 languages by stating the original most general rule:

$$\alpha \rightarrow \beta, \tag{11}$$

with the additional condition that β **is at least as long as** α, i.e.

$$l(\alpha) \leq l(\beta), \tag{12}$$

where $l()$ is a function which gives the length in number of symbols. More precisely, Chomsky proves that this weakening of the original restriction ($\varphi_1 X \varphi_2 \rightarrow \varphi_1 \beta \varphi_2$) “will not increase the class of generated languages.”[6]

We will work with this **alternative definition** in the lecture series.

Chomsky (1959), p. 145.
Jäger & Rogers (2012), p. 1957.

---

[6] Sometimes this latter definition of context-sensitive grammar in equation (11) and with the length restriction is referred to as *non-contracting grammar*.

Abstract Example

Rewrite	Rule
S	—
XY	1
acY	3
acbYd	4
acbbdd	5
abcbdd	6
abbcdd	6

Terminals

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

Non-Terminals

$NT = \{S, X, Y\}$

R

1. S $\rightarrow$ XY
2. X $\rightarrow$ aXc
3. X $\rightarrow$ ac
4. Y $\rightarrow$ bYd
5. Y $\rightarrow$ bd
6. **cb** $\rightarrow$ **bc** (context-sensitive !)

The language generated with this grammar is $L(\mathcal{G}_{CS}) = \{acbd, abbcdd, aabbccdd, \ldots, a^m b^n c^m d^n\}$, with $n, m \in \mathbb{N}$. Note that $n = m$ is possible.

Natural Language Examples

$\alpha \rightarrow \beta$ with $l(\alpha) \leq l(\beta)$

the → a ✓

the tree → this x

the huge tree → the tree x

the huge tree bends in the wind → the x

the → the huge tree bends in the wind ✓

VP → NP bends NP NP ✓

NP VP NP NP → NP VP x

DET → the ✓

the → DET ✓

NP → the N ✓

NP → DET N ✓

VP → NP VP ✓

Note: It seems like the passive transformation could still be in the domain of context-sensitive grammars.

$NP_1\ V_2\ NP_3 \rightarrow NP_3\ [_{AUX}\ \text{be}]\ V_2\text{en}\ [_{PP}\ [_{P}\ \text{by}]\ NP_1]$ ✓(?)

Further Examples of Context-Sensitive Languages

- *Copy language*: the set of strings $\omega = \gamma\gamma$ over a set of terminals T such that ω consists of two identical halves, e.g. $L(\mathcal{G}_{CS}) = \{aa, abab, abcabc, abcdabcd, \ldots\}$
- Languages with strings of the form $x^n y^n z^n$, e.g. $L(\mathcal{G}_{CS}) = \{abc, aabbcc, aaabbbcc, \ldots\ a^n b^n c^n\}$
- Languages with strings of the form $x^n y^n z^n w^n v^n$, e.g. $L(\mathcal{G}_{CS}) = \{abcde, aabbccddee, \ldots\ a^n b^n c^n d^n e^n\}$
- The set of all *prime numbers* (where each number x is represented by a string of length $l(x)$).
- etc.

Jäger & Rogers (2012), p. 1958.

Are Natural Languages **Context-Sensitive**?

Arguments such as the one by Shieber (1984) and Culy (1985) have led most FLT syntactians to assume that natural languages are at least **mildly context-sensitive**. There seem to be no generally accepted proposals of natural language structures which would require to go beyond the context-sensitive level.

Recursively Enumerable Languages (Type 0)

Definition: Type-0 Grammar

If the most general rewrite rule

$$\alpha \rightarrow \beta, \tag{13}$$

is not further restricted, we would call this a **Type-0 grammar**.

Chomsky (1959), p. 143.

Natural Language Examples

$$\alpha \rightarrow \beta$$

the tree → this

the huge tree → the tree

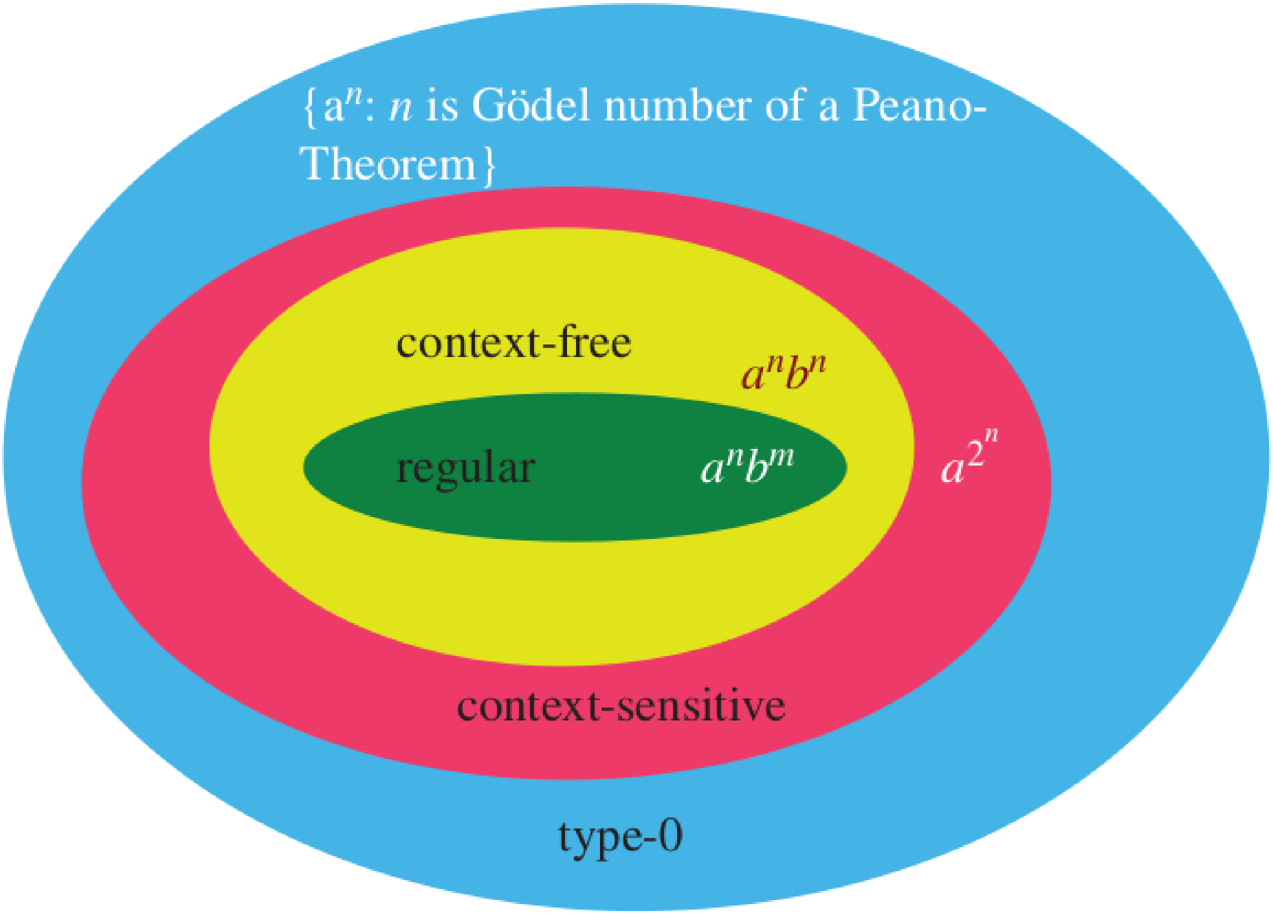
the huge tree bends in the wind → the

NP VP NP NP → NP VP

If we really wanted to perform arbitrary transformations from any string α to any string β, then this would be a **Type-0 grammar**.

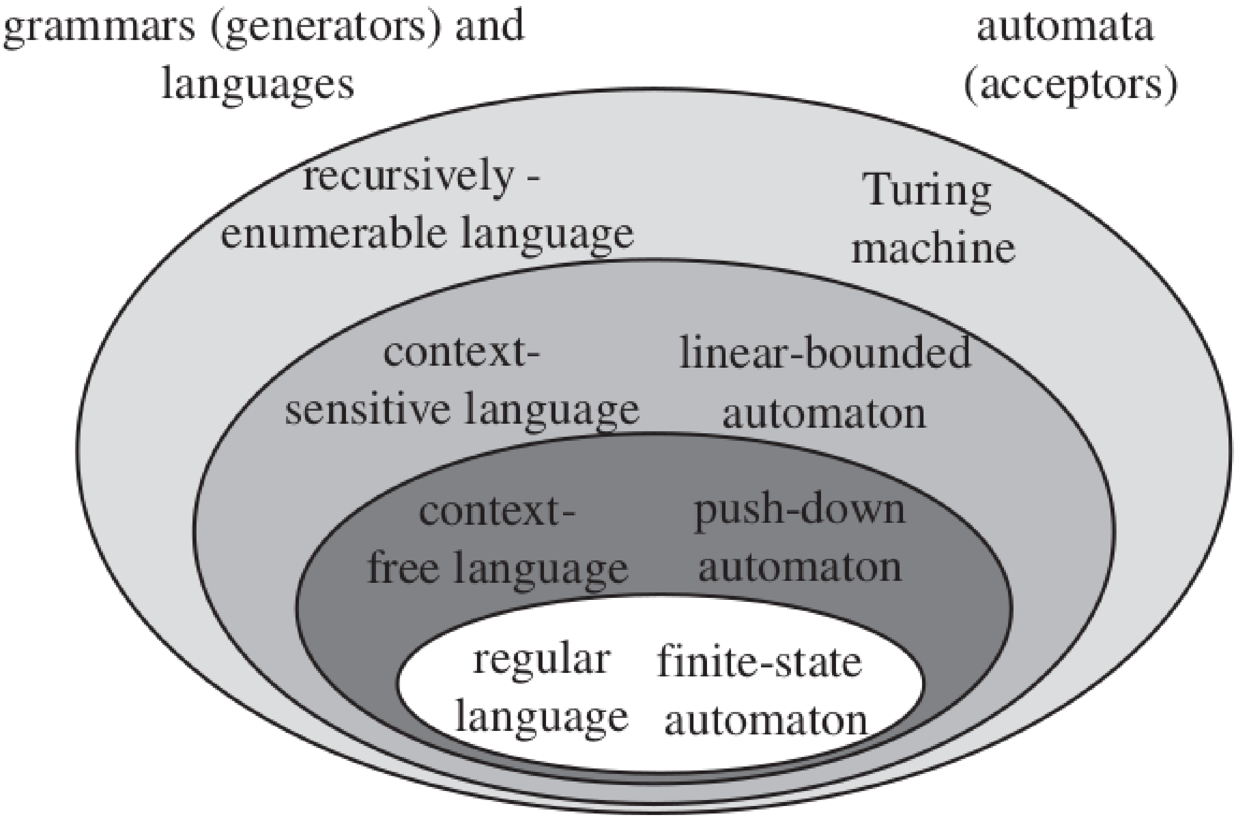
The Classical Hierarchy

The Classical Hierarchy



Jäger & Rogers (2012), p. 1959.

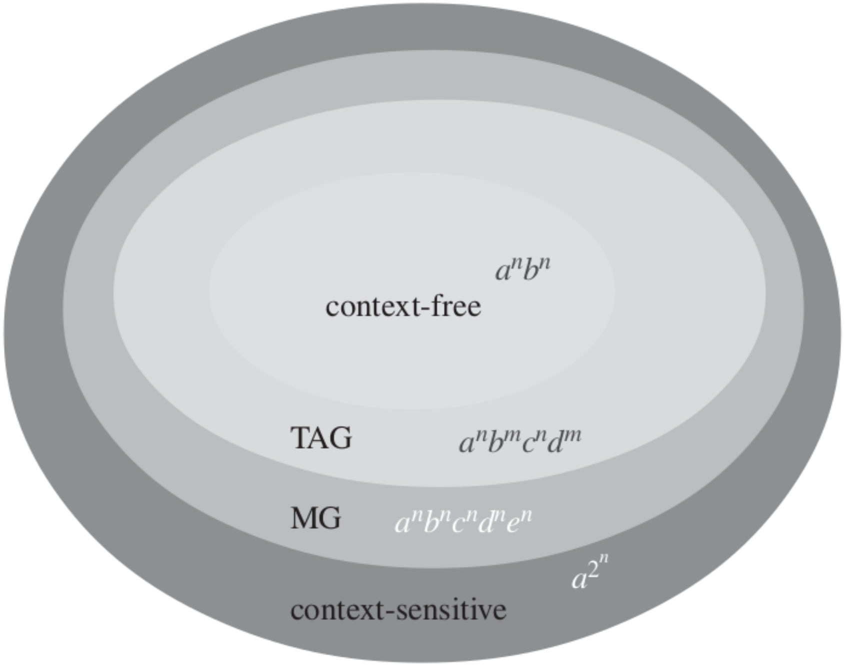
The Classical Hierarchy



Fitch & Friederici (2012).

Section 4: The Sub-Hierarchies

Languages between CF and CS



TAG: Tree Adjoining Grammar
MG: Minimalist Grammar (formalized)

Jäger & Rogers (2012), p. 1961.

Further Literature:

Discussion of Tree Adjoining Grammars and their relation to the Chomsky hiearchy.

Joshi, A. (1985). How much context-sensitivity is necessary for characterizing structural descriptions – tree adjoining grammars.

Formalization of a Minimalist Grammar.

Stabler, E. (1997). Derivational Minimalism.

Old Georgian as a non-MG, i.e. context sensitive, language.

Michaelis, J. & Kracht, M (1996). Semilinearity as a syntactic invariant.

References

References

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions in Information Theory* 2, 113–124.

Chomsky, N. (1957). *Syntactic structures*. The Hague, The Netherlands: Mouton.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control* 2, 137–167.

Chomsky, N. & Schützenberger, M. P. (1965). The algebraic theory of context-free languages. *Studies in logic and the foundations of mathematics*. North Holland Publishing Company, Amsterdam.

Culy, C. (1985). The complexity of the vocabulary of Bambara. *Linguistics and Philosophy*, Vol. 8, No. 3, pp. 345-351.

Fitch, T. & Friederici, A. D. (2012). Artificial grammar learning meets formal language theory: an overview. *Philosophical Transactions of the Royal Society*, 367, pp. 1933-1955.

Jäger, Gerhard, and Rogers, James (2012). Formal language theory: refining the Chomsky hierarchy. *Philosophical Transactions of the Royal Society*, 367, pp. 1956-1970.

Shieber, Stuart (1984). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*.

Thank You.

Contact:

Faculty of Philosophy
General Linguistics
Dr. Christian Bentz
SFS Wilhelmstraße 19-23, Room 1.24
chris@christianbentz.de

Office hours:
During term: Wednesdays 10-11am
Out of term: arrange via e-mail