EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Syntax & Semantics WiSe 2022/2023

Lecture 6: Phrase Structure Grammar (PSG) I

15/11/2022, Christian Bentz

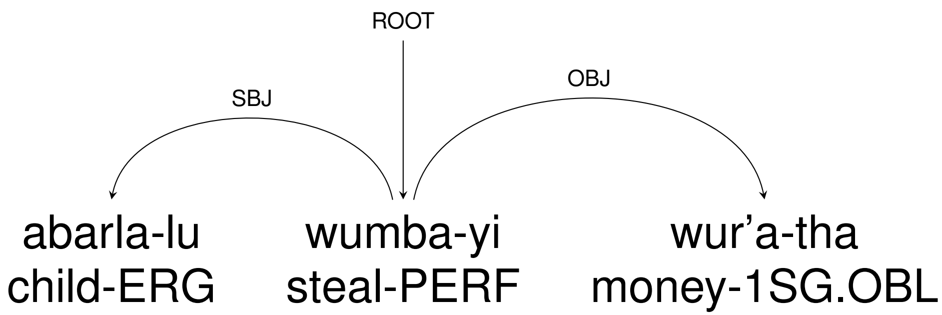
Overview

Section 1: Recap of Lecture 5

Linearization: Free Word Order

If a language has **completely free word order**, then linearization might not be required by the syntactic framework. All orders are grammatical and hence “licensed”. See the permutation examples below.

Nhanda (nha, Pama-Nyungan)

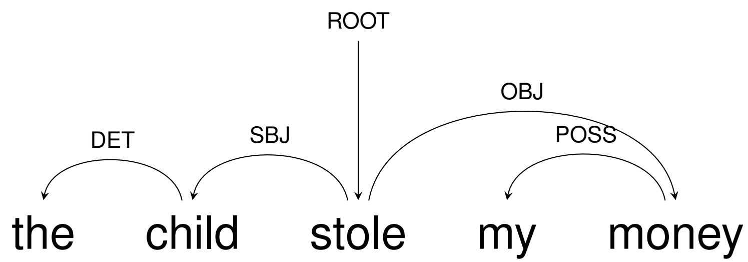


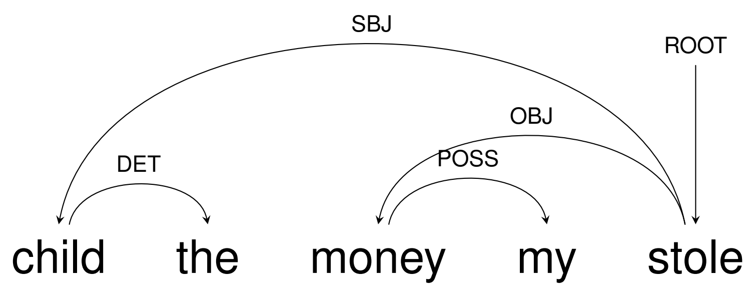
“The child stole my money.”

Adopted from Velupillai (2012), p. 282.

Linearization: Fixed Word Order

If a language has **fixed word order**, however, then the lack of linearization constraints licenses ungrammatical sentences.

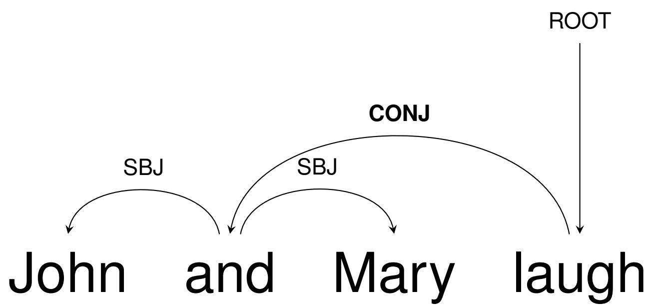




Note that both of these sentences (and all other permutations) are licensed by a dependency grammar that does not specify linearization constraints.

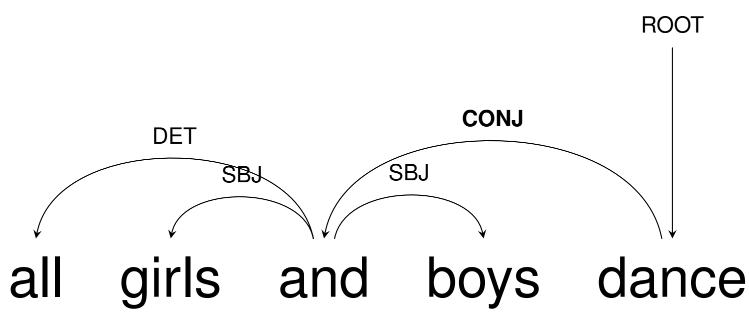
Coordination: Arrow Notation

Proper nouns:



Notes: We here need two SUBJ arrows, since both proper nouns are subjects of the sentence. In the case of noun phrases with determiners (Müller considers *all* a determiner here), the determiner also depends on the conjunction.

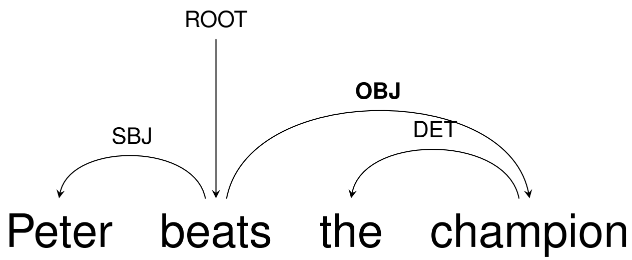
Noun phrases:



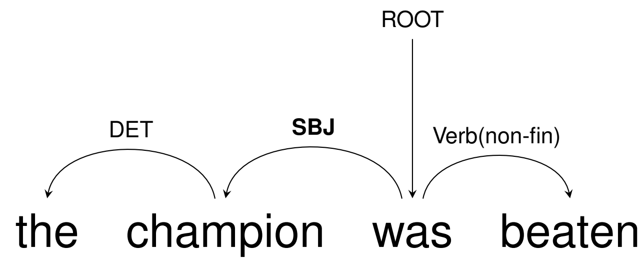
The Passive

In a **passive construction**, the object of the corresponding *active sentence* becomes the subject. If we want to further license case assignments (e.g. nominative to the subject of the active sentence and the subject of the passive sentence, while accusative to the object of the active sentence) then we have to invoke further lexical rules (see Müller (2019), pp. 373).

Active:

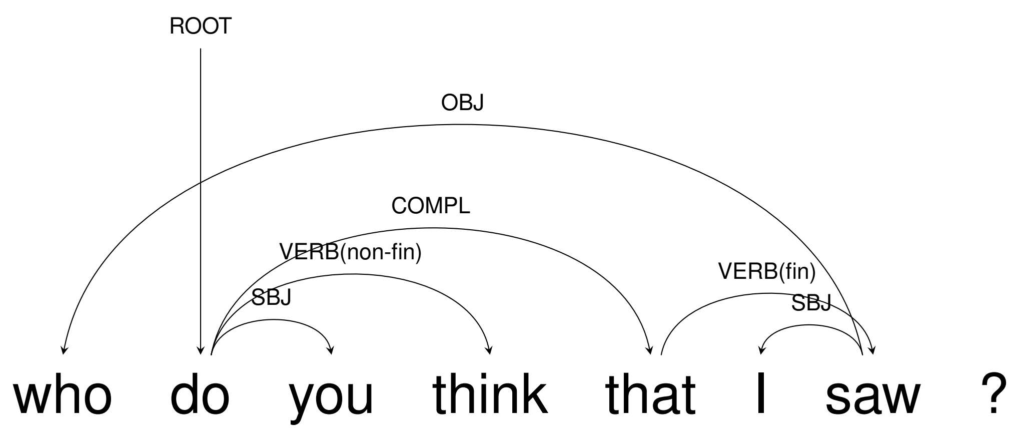


Passive:



Crossing Dependencies

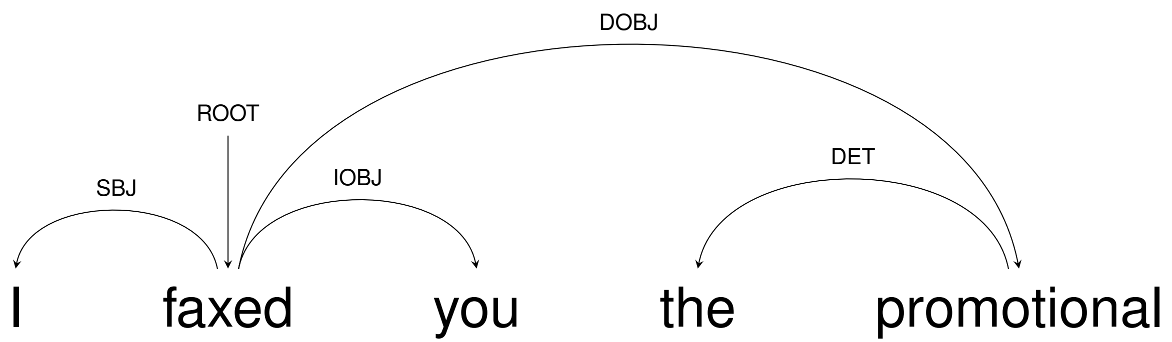
In certain syntactic constructions (and languages), dependencies might cross. Such constructions are referred to as *non-projective*. This is often seen as dispreferred from a processing and learning perspective, though there is no reason a priori why dependencies should not cross.



See the German equivalent in Müller (2019), p. 379.

Example Sentence

Lecture Notation:



Universal Dependencies Notation:

```
# sent_id = email-enronsent00_02-0047
# text = I faxed you the promotional [...]
```

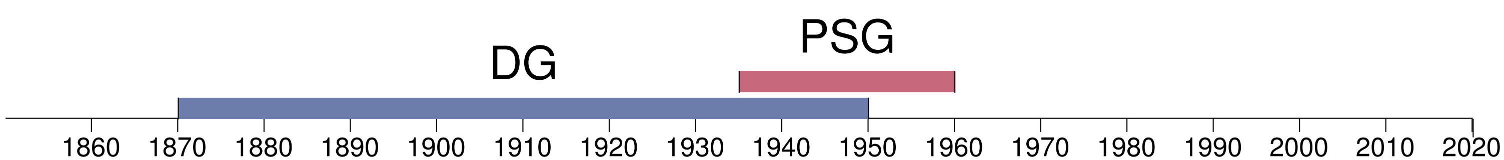
ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS
1	I	I	PRON	PRP	Case=Nom|Number=Sing|Person=1|PronType=Prs	2	nsubj	2:nsubj
2	faxed	fax	VERB	VBD	Mood=Ind|Tense=Past|VerbForm=Fin	0	root	0:root
3	you	you	PRON	PRP	Case=Acc|Person=2|PronType=Prs	2	iobj	2:iobj
4	the	the	DET	DT	Definite=Def|PronType=Art	5	det	5:det
5	prom.	prom.	NOUN	NN	Number=Sing	2	obj	2:obj

Section 2: Historical Notes

Historical Perspective

"Phrase structure grammars and associated notions of phrase structure analysis have their proximate origins in models of **Immediate Constituent (IC)** analysis. Although inspired by the programmatic syntactic remarks in Bloomfield (1933), these models were principally developed by Bloomfield's successors, most actively in the decade between the publication of Wells (1947) and the advent of transformational analyses in Harris (1957) and Chomsky (1957)."

Blevins et al. (2013). Phrase structure grammar, p. 1.



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.

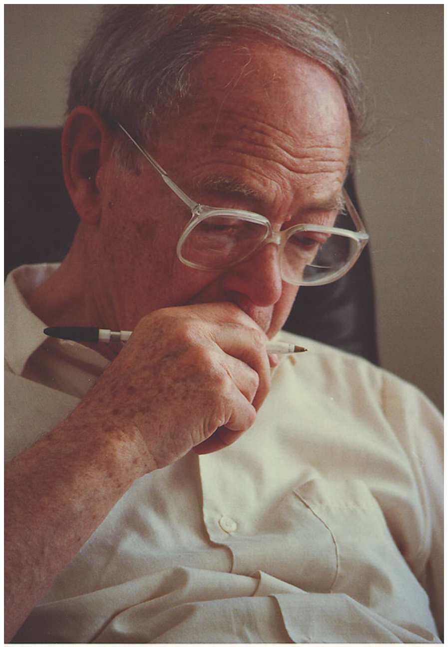
Immediate Constituent Analysis

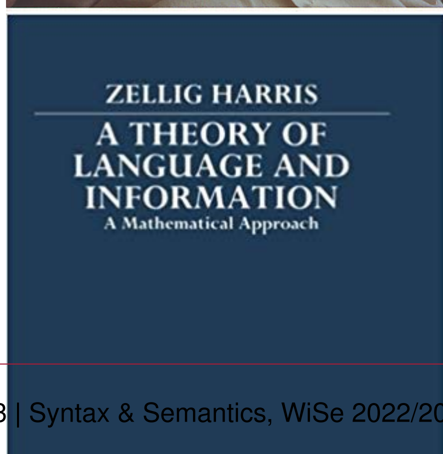
"Let us call the ICs of a sentence, and the ICs of those ICs, and so on down to the morphemes, the CONSTITUENTS of the sentence; and conversely whatever sequence is constituted by two or more ICs let us call a CONSTITUTE."

Wells (1947), Immediate Constituents, p. 84.

the || *king* ||| *of* |||| *England* | *open* ||| *ed* || *Parliament-*

Historical Aside





*Linguistic **transformations** can be viewed as an equivalence relation among sentences or certain constituents of sentences.*

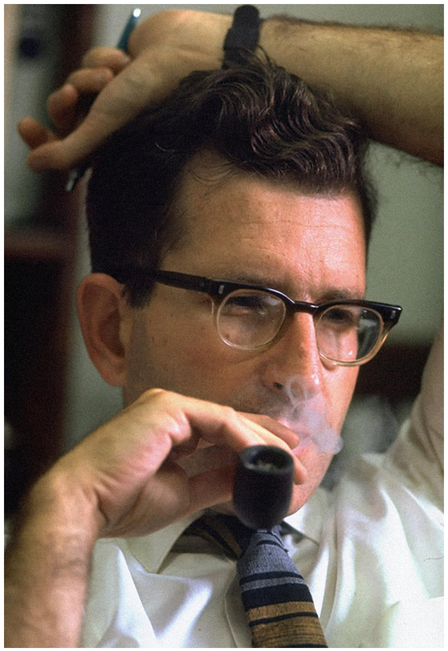
Harris, Zellig (1970). Papers in structural and transformational linguistics, p. 383.

***The theory of syntax is stated in terms related to mathematical Information Theory: as constraints on word combination**, each later constraint being defined on the resultants of a prior one. This structure not only permits a finitary description of the unbounded set of sentences, but also admits comparison of language with other notational systems, [...]*

Harris, Zellig (1991). A theory of language and information.

Section 1: Recap of Lecture 5
Section 2: Historical Notes
Section 3: Basic Definitions
Section 4: Binary Branching Trees
Section 5: Outlook
Summary
References

Historical Aside



[...] To complete this elementary communication theoretic model for language, we assign a probability to each transition from state to state. We can then calculate the "uncertainty" associated with each state and we can define the "information content" of the language as the average uncertainty, weighted by the probability of being in the associated states. ***Since we are studying grammatical, not statistical structure of language here, this generalization does not concern us.***

Chomsky, Noam (1957). Syntactic Structures, p. 20.

Section 3: Basic Definitions

Example

Assume we want to analyze/generate the following English sentence using a phrase structure grammar (PSG):

The child reads a book.

Symbols: **Terminals**

We firstly define a finite set of so-called **terminal symbols** (T). We here assume that these are words[1] in the respective language we are analyzing:

$$T = \{a, book, child, reads, the, \dots\}^2 \tag{1}$$

[1] Words are typically assumed as terminals for the analysis of natural language, but note that we could also choose morphemes, syllables, characters, etc.

[2] I here order them alphabetically, but note that the order in a set does not matter.

Symbols: **Non-Terminals**

Based on the definitions of constituency and parts of speech – as laid out in previous lectures – we can also define a finite set of so-called **non-terminal symbols** (NT).

We here assume that these consist of symbols for phrases (e.g. NP, VP, AP, etc.), parts of speech (N, V, A, etc.), as well as the starting symbol S.[3] We such arrive at:

$$NT = \{NP, VP, AP, \ldots N, V, A, \ldots S\} \tag{2}$$

[3] A glossary of all symbols used here is given at the end of this section.

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

$$\alpha \rightarrow \beta, \tag{3}$$

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 $\rightarrow$ DET N
NP $\rightarrow$ the N
NP $\rightarrow$ the tree

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:

$$\langle T, NT, S, R \rangle^4 \tag{4}$$

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

[4] S is a "distinguished member" of NT.

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 (multi)set of words and strings of words licensed by the respective rewrite rules:

$$L(\mathcal{G}) = \{(w_1), (w_2), \ldots (w_n), (w_1, w_2), \ldots (w_1, \ldots w_m)\}, \quad (5)$$

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.

Example

Assume we want to create a PSG that generates our example sentence:

The child reads a book.

Terminals

$$T = \{a, book, child, reads, the\}$$

Non-Terminals

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

Note: Here, both the concept of *constituency/headedness* and *parts-of-speech* are vital, since we need these to determine NT.

Example

Assume we want to create a PSG that generates our example sentence:

The child reads a book.

R (involving terminal symbols)

1. DET $\rightarrow$ the
2. DET $\rightarrow$ a
3. N $\rightarrow$ child
4. N $\rightarrow$ book
5. V $\rightarrow$ reads

R (only non-terminal symbols)

6. S $\rightarrow$ NP V NP
7. NP $\rightarrow$ DET N

Rewrite	R#
S	–
NP V NP	6
DET N V NP	7
DET N V DET N	7
DET N reads DET N	5
the N reads DET N	1
the child reads DET N	3
the child reads a N	2
the child reads a book	4

Terminals

$T = \{a, book, child, reads, the\}$

Non-Terminals

$NT = \{DET, N, NP, V\}$

***R* (Terminals)**

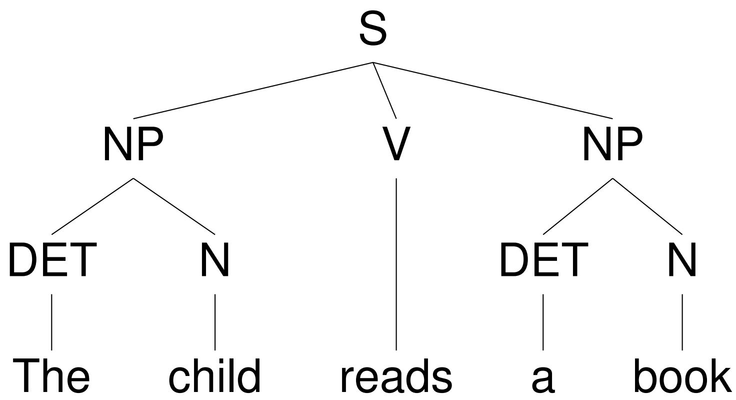
1. DET $\rightarrow$ the
2. DET $\rightarrow$ a
3. N $\rightarrow$ child
4. N $\rightarrow$ book
5. V $\rightarrow$ reads

***R* (Non-Terminals)**

6. S $\rightarrow$ NP V NP
7. NP $\rightarrow$ DET N

Note: The horizontal line indicates the point where rules exclusively defined with non-terminals ($R(NT)$) end, and rules involving terminals ($R(T)$) start. While the order of application of non-terminal rules is often important, the order of the application of terminal rules is irrelevant.

Tree Notation



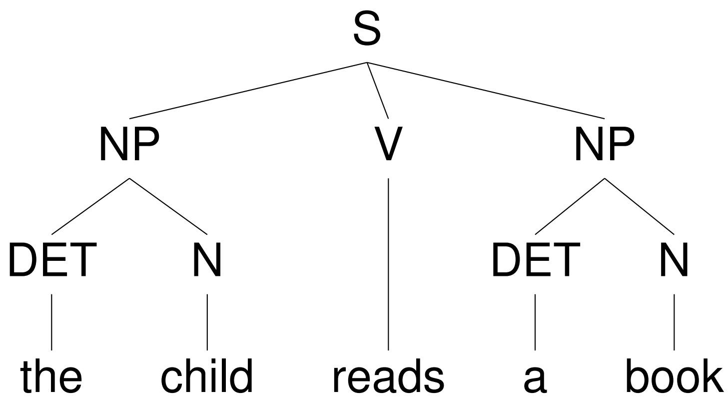
Rewrite Notation

S
NP V NP
DET N V NP
DET N V DET N

DET N reads DET N
the N reads DET N
the child reads DET N
the child reads a N
the child reads a book

Note: The *Tree Notation* and *Rewrite Notation* are structurally equivalent. Everything above the horizontal line in the *Rewrite Notation* corresponds to tree internal nodes, whereas everything below that line corresponds to the last (straight) leaves on the tree leading to the orthographic words.

Bracket Notation



Rewrite Notation

S
NP V NP
DET N V NP
DET N V DET N

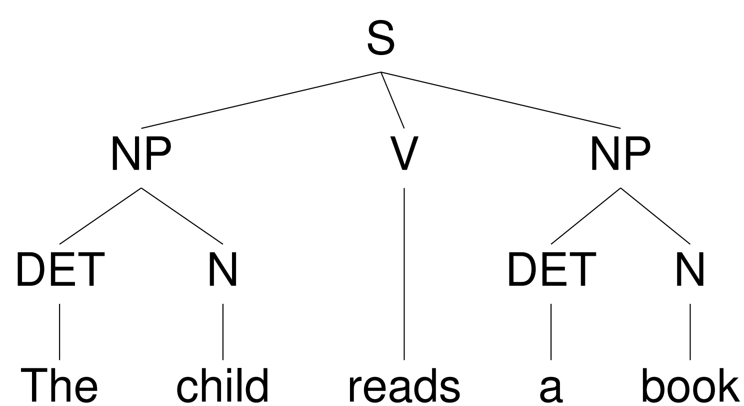
DET N reads DET N
the N reads DET N
the child reads DET N
the child reads a N
the child reads a book

[S [NP [DET [the]][N [child]]][V [reads]][NP [DET [a]][N [book]]]][5]

[5] Note: The *Bracket Notation* is yet another equivalent way to visualize the same structure. In fact, the latex code generating this slide takes the bracket notation as input to generate the above tree. There is also an online tool at ironcreek.net/syntaxtree to generate trees based on bracket notation input.

Aside: Latex Code

```
\begin{forest}
for tree={s sep=10mm, inner sep=0, l=0}
[S
    [NP
        [DET [The, tier=word]]
        [N [child, tier=word]]
    ]
    [V [reads, tier=word]]
    [NP [DET [a, tier=word]]
        [N [book, tier=word]]
    ]
]
\end{forest}
```



The Language

What are all the sentences and hence the **language (in formal terms)** that the PSG above can generate?[6]

$$L(\mathcal{PSG}) = \{(\textit{the}, \textit{child}, \textit{reads}, \textit{a}, \textit{book}), \\ (\textit{a}, \textit{child}, \textit{reads}, \textit{the}, \textit{book}), \\ (\textit{the}, \textit{book}, \textit{reads}, \textit{a}, \textit{child}), \\ (\textit{a}, \textit{book}, \textit{reads}, \textit{the}, \textit{child})\} \tag{6}$$

Note: This example nicely illustrates Chomsky's point about a strict distinction between syntax and semantics. All four strings are grammatical according to the PSG, but the latter two are rather odd from a semantic perspective.

[6] We here make the *additional assumption* that each rule has to be *applied at least once*. Otherwise, sentences such as *a child reads a book* and *a book reads a book*, as well as isolated noun phrases, and even single words would also be licensed.

Important **Take-Home-Message**

One of the most important features of PSGs is that they strongly **restrict the number of possible sentences** via **linearization constraints** in the *non-terminal rules* (inner parts of the tree). The sentences generated by the PSG above are in fact a small subset of the overall possible sentences without any linearization constraints, namely, 4 out of $5! = 120$, or around 3%.

Sentences licensed by PSG:

the child reads a book
a child reads the book
the book reads a child
a book reads the child

Possible permutations:

the child reads a book
**book the child reads a*
**a book the child reads*
**reads a book the child*
**child reads a book the*
etc.

Note: Without the constraint that each word has to occur at least once, we would have $5^5 = 3125$ different sentences.

Notation Glossary

A: adjective
AP: adjective phrase
COMPL: complementizer (i.e. *that*)
DET: determiner
N: noun
NP: noun phrase

P: preposition
PRON: pronoun
V: verb
VP: verb phrase

[6] Required in complementizer-constructions.

Faculty of Philosophy
General Linguistics

Section 4: Binary Branching Trees

Binary Branching

"[...] the question of the kind of branching structures assumed has received differing treatments in various theories. **Classical X-bar theory** assumes that a verb is combined with all its complements. In later variants of **GB**, all structures are **strictly binary branching**. Other frameworks do not treat the question of branching in a uniform way: there are proposals that assume binary branching structures and others that opt for flat structures."

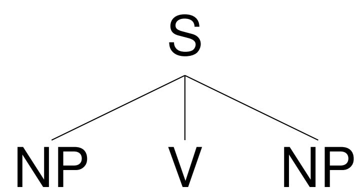
Müller (2019). Grammatical theory, p. 553.

Multifurcation

In the PSG we delevoped in the previous section, *more than two symbols* were allowed to occur on the right hand side of the rule, i.e.

$$S \rightarrow NP\ V\ NP, \tag{7}$$

yielding a so-called *multifurcation* in the tree:



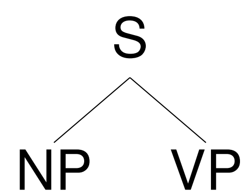
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Bifurcation

In order to restrict PSGs to a set of simpler (i.e. shorter rules), many frameworks introduce a **binarization constraint**, such that all rewrite rules have only *one symbol* on the left, and maximally *two symbols* on the right. For example,

$$S \rightarrow NP\ VP. \tag{8}$$

This yields exclusively *bifurcating* branches in the tree (except for the terminal nodes):



Example

In order to implement the *binarization constraint* for our example above we only have to introduce *VP* as a non-terminal symbol and split the rule $S \rightarrow NP\ V\ NP$ into two rules:

R (involving terminal symbols)

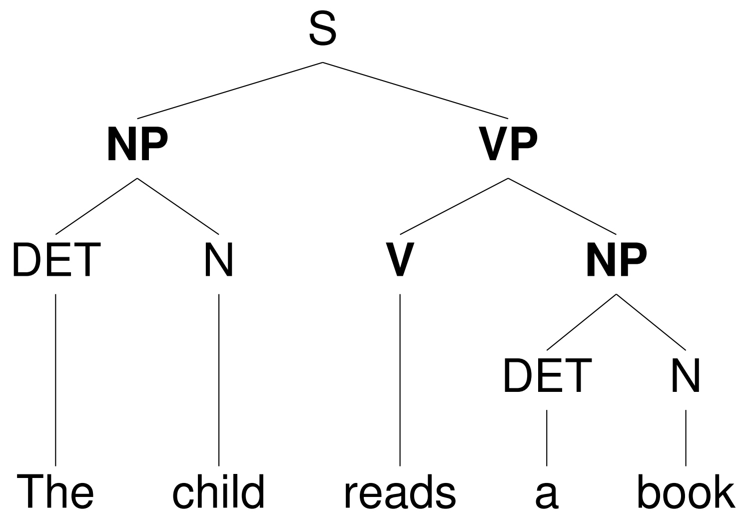
1. DET $\rightarrow$ the
2. DET $\rightarrow$ a
3. N $\rightarrow$ child
4. N $\rightarrow$ book
5. V $\rightarrow$ reads

R (only non-terminal symbols)

6. **S $\rightarrow$ NP VP**
7. **VP $\rightarrow$ V NP**
8. NP $\rightarrow$ DET N

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Tree Notation



Rewrite Notation

S
NP VP
NP V NP
DET N V NP
DET N V DET N

DET N reads DET N
the N reads DET N
the child reads DET N
the child reads a N
the child reads a book

Note: If we wanted the tree to reflect the assumption that the finite verb heads the overall sentence, then we could further introduce $S \rightarrow VP$ and then $VP \rightarrow NP\ VP$.

Section 5: Outlook

Outlook: Phrase Structure Grammar II

- Morphological Features
  - Expanding the PSG: Morphology
  - Problem: Complicated Agreement Systems
  - Problem: Implementing Morphological Features
- Syntactic Phenomena
  - Verb Position
  - Ditransitive Sentences
  - The Passive
- Pros and Cons of PSG
  - Pros (Advantages)
  - Cons (Disadvantages)

Summary

- A PSG consists of a quadruple of **terminal symbols** (T), **non-terminal symbols** (NT), **rewrite rules** (R), and a **starting symbol** (S).
- The **rewrite** notation, **bracket** notation, and **tree** notation for PSGs are structurally equivalent.
- Rewrite rules reflecting higher order constituents (VP, NP, etc.) **strongly constraint the set of possible sentences** in a language generated from the PSG.

References

References

Blevins et al. (2013). Phrase structure grammar. In: Marcel den Dikken (Ed.), *The Cambridge Handbook of Generative Syntax*. Cambridge: Cambridge University Press.

Bloomfield, Leonard (1933). *Language*. University of Chicago Press, Chicago.

Harris, Zellig S (1957). Co-occurrence and transformation in linguistic structure. *Language*, 33, 283-340.

Harris, Zellig S. (1991). *A theory of language and information. A mathematical approach*. Oxford: Oxford University Press.

Harris, Zellig S. (1970). *Papers in structural and transformational linguistics*. Springer, Dordrecht.

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

Müller, Stefan. 2019. *Grammatical theory: From transformational grammar to constraint-based approaches. Third revised and extended edition. **Volume I**.* Berlin: Language Science Press.

Partee, Barbara H., Meulen, Alice ter, Wall, Robert E. (1990). *Mathematical methods in linguistics.* Studies in Linguistics and Philosophy, Vol. 30. Dordrecht/London/Boston: Kluwe Academic Publishers.

Wells, Rulon S. (1947). Immediate constituents. *Language*, 23, 81-117.

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