EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Syntax & Semantics WS2019/2020

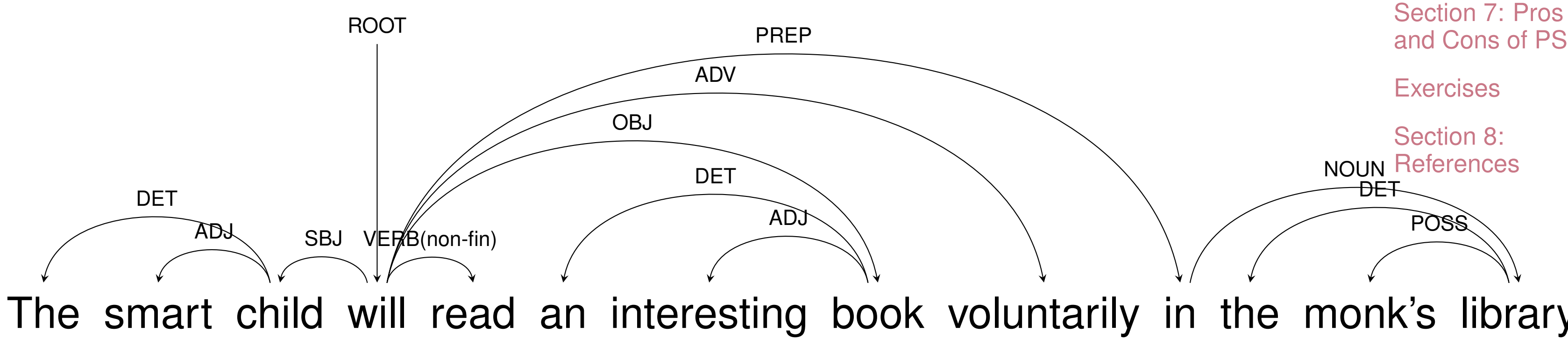
Lecture 5: Phrase Structure Grammar (PSG)

08/11/2019, Christian Bentz

Overview

Section 1: Recap of Lecture 4

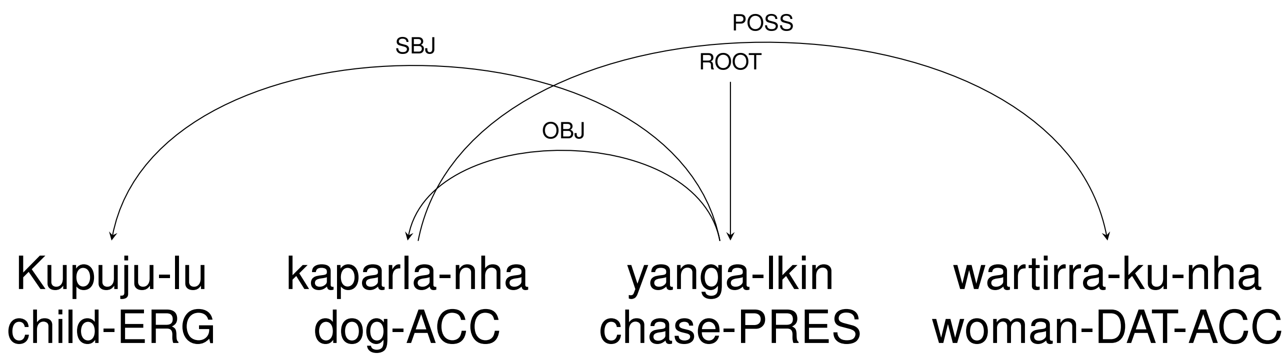
Summary: The Full Example



Linearization

The fact that dependency grammars do often not require particular rules for the *linearization* of words,[1] is the reason for why they are seen as particularly appropriate for languages with discontinuous constituents (or even no constituency at all?). Remember the example by Evans & Levinson (2009) in Lecture 2.

Thalanyji (?, Pama-Nyungan(?))



[1]Though see the discussion in Müller (2019), pp. 371, for dependency grammar accounts that additionally formulate such rules.

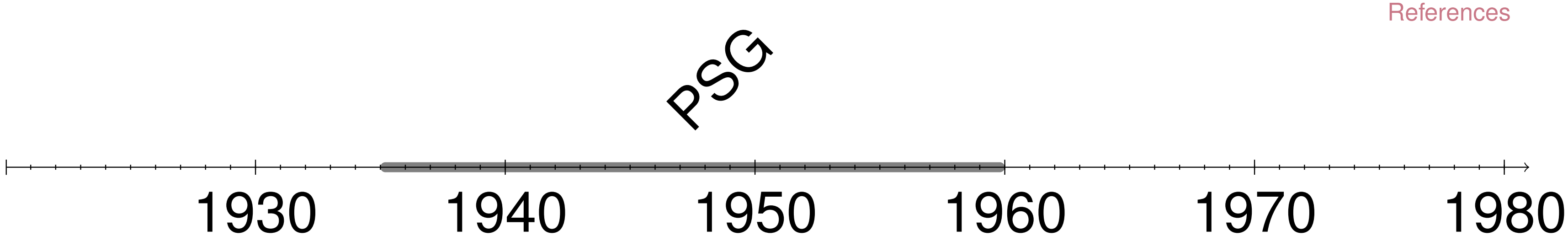
Section 2: Historical Notes

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

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.



EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Philosophy
General Linguistics

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[2] in the respective language we are analyzing:

$$T = \{a, book, child, reads, the, \dots\}^3 \quad (1)$$

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

[3] 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.[4] We such arrive at:

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

[4] 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

$$\alpha \rightarrow \beta, \quad (3)$$

where α is a string of symbols $(x_1, x_2, x_3, \ldots, x_n)$ 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:

$$\text{NP} \rightarrow \text{DET N.}$$

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:

$$\langle T, NT, S, R\rangle^{5} \tag{4}$$

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

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

Interlude: The Chomsky Hierarchy

The type of rules allowed to be part of the set R determines the generative power of the grammar $\mathcal{G}$. For example, a so-called *context-free grammar* contains a set of rewrite rules which only allow a single non-terminal symbol on the left side of the arrow. For example,

$$NP \rightarrow DET\ A\ N \tag{5}$$

A more powerful *context-sensitive grammar* would be less restrictive, i.e. allowing several symbols on the left-hand side of the rules, however, with the restriction that the left-hand side never has more symbols than the right-hand side. For example,

$$NP\ VP \rightarrow VP\ NP \tag{6}$$

For further details see Jäger and Rogers (2012). Formal language theory: refining the Chomsky hierarchy.

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 have a language defined as

$$L(\mathcal{G}) = \{(w_1), (w_2), \ldots (w_n), (w_1, w_2), \ldots (w_{n-1}, w_n), \ldots\}, \quad (7)$$

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

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

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

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	Rule
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)

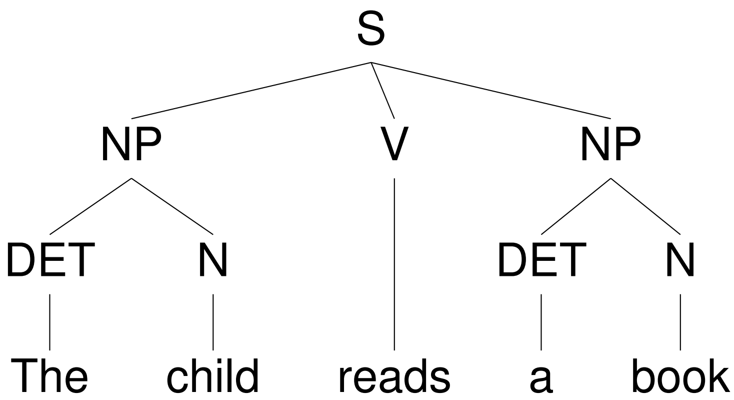
1. DET $\rightarrow$ the
2. DET $\rightarrow$ a
3. N $\rightarrow$ child
4. N $\rightarrow$ book
5. V $\rightarrow$ reads

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



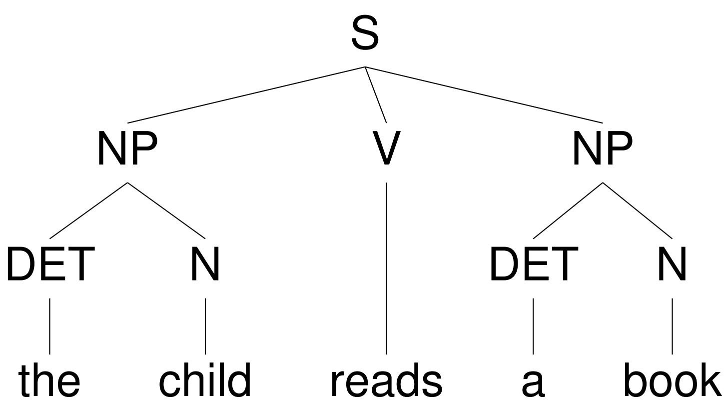
Rewrite Notation

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]]]][6]

[6] 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.

The Language

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

$$L(\mathcal{PSG}) = \{(\textit{the}, \textit{child}, \textit{reads}, \textit{a}, \textit{book}), (\textit{a}, \textit{child}, \textit{reads}, \textit{the}, \textit{book}), (\textit{the}, \textit{book}, \textit{reads}, \textit{a}, \textit{child}), (\textit{a}, \textit{book}, \textit{reads}, \textit{the}, \textit{child})\} \quad (8)$$

Important: 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* 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.

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.

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."

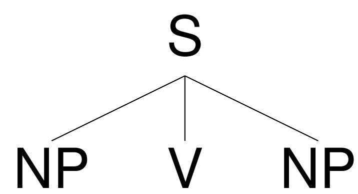
Müller (2019). Grammatical theory, p. 553.

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.

$$S \rightarrow NP\ V\ NP, \tag{9}$$

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

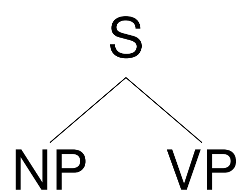


Bifurcation

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,

$$S \rightarrow NP\ VP. \tag{10}$$

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:

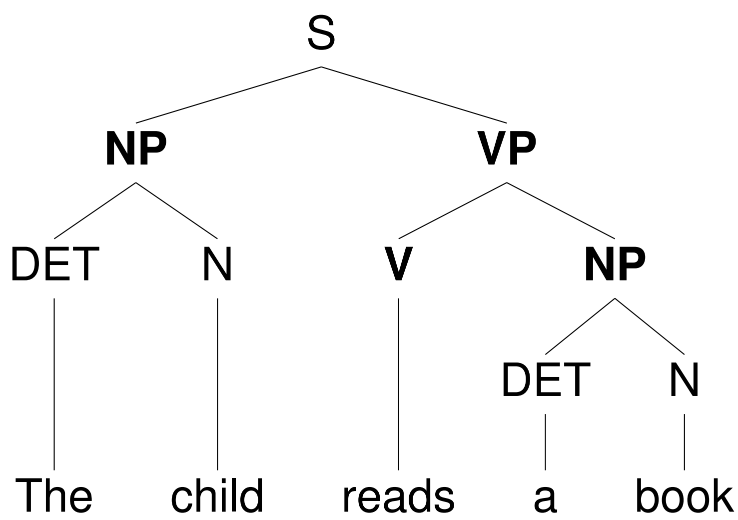
R **(involving terminal symbols)**

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

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: Morphological Features

Expanding the PSG: The Vocabulary

We can expand our PSG towards covering more of the grammatical sentences in actual English by simply adding terminal symbols, e.g. other two-place predicates (*sees*) and nouns (*tree, frog*).

Sentences licensed by PSG:

the child reads a book
the child sees a book
the child sees a tree
the frog sees a tree
etc.

Note: We will quickly run into the problem of semantics: *?The child reads a frog*. This is the point where *Chomsky's colourless green ideas* come into the picture. PSGs are geared towards grammatical licensing, regardless of semantics.

Expanding the PSG: Morphology

In order to also implement agreement between verbs, nouns and determiners, we have to expand the PSG by using morphological features.

License:

the child reads a book
the children read a book
a child reads the books
etc.

Do not license:

**the child read a book*
**the children reads a book*
**the child reads a books*
etc.

First Step: Expand the Terminals

Terminals

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

Second Step: Expand the Non-Terminals

Non-Terminals

Morphological features are here given in parentheses '()', and in upper case notation according to the Leipzig Glossing Rules.

$$NT = \{DET(SG), DET(PL), N(SG), N(PL), NP(SG), NP(PL), V(SG), V(PL), VP(SG), VP(PL)\} \quad (11)$$

Third Step: Change Rewrite Rules

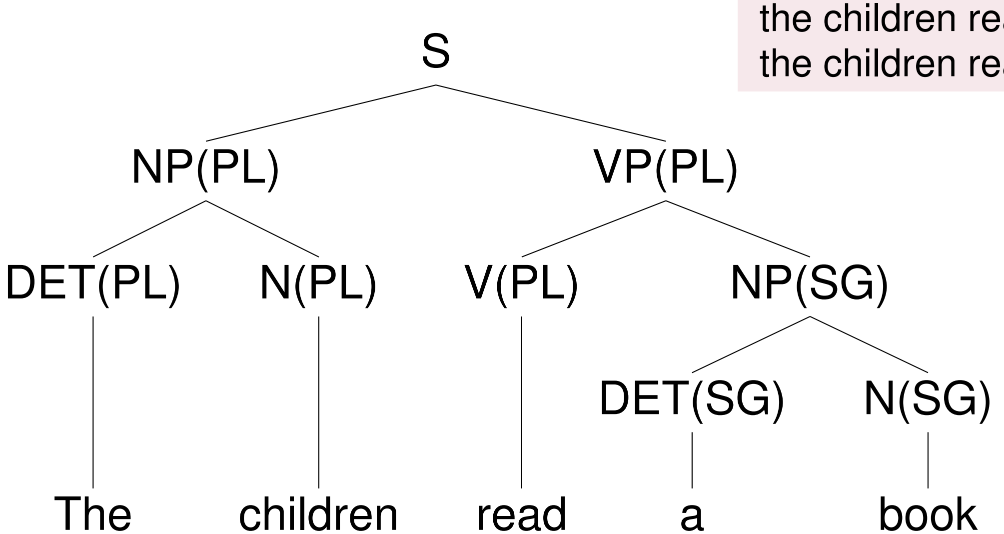
R (involving **terminal** symbols)

1. DET(SG) → the
2. DET(SG) → a
3. DET(PL) → the
4. N(SG) → child
5. N(SG) → book
6. N(PL) → children
7. N(PL) → books
8. V(SG) → reads
9. V(PL) → read

R (only **non-terminal** symbols)

6. S → NP(SG) VP(SG)
7. S → NP(PL) VP(PL)
8. NP(SG) → DET(SG) N(SG)
9. NP(PL) → DET(PL) N(PL)
10. VP(SG) → V(SG) N(SG)
11. VP(SG) → V(SG) N(PL)
12. VP(PL) → V(PL) N(SG)
13. VP(PL) → V(PL) N(PL)

Tree Notation



Rewrite Notation

S
NP(PL) VP(PL)
NP(PL) V(PL) NP(SG)
DET(PL) N(PL) V(PL) NP(SG)
DET(PL) N(PL) V(PL) DET(SG) N(SG)

DET(PL) N(PL) read DET(SG) N(SG)
the N(PL) read DET(SG) N(SG)
the children read DET(SG) N(SG)
the children read a N(SG)
the children read a book

Problem: Complicated Agreement Systems

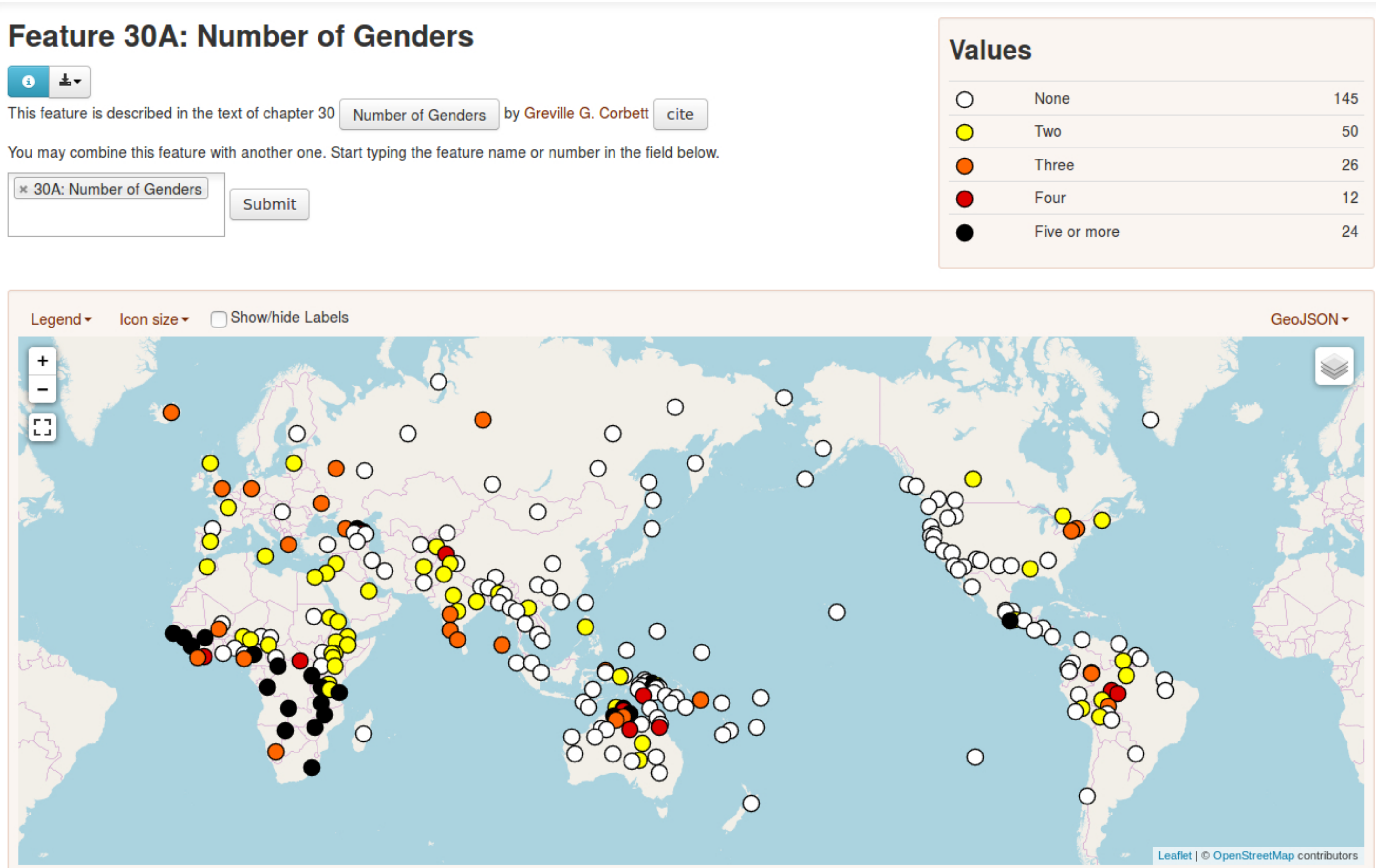
"The defining characteristic of gender is **agreement**: a language has a gender system only if we find different agreements ultimately dependent on nouns of different types. In other words, there must be evidence for gender outside the nouns themselves."

Corbett (2013). Number of Genders.

Russian (rus, Indo-European)

(1) Žurnal ležal na stole.
magazine lay.**M** on table
"The magazine lay on the table."

(2) Kniga ležal-**a** na stole.
book lay-**F** on table
"The book lay on the table."



https://wals.info/feature/30A

Example: Gender in Swahili

"In Swahili, each noun prompts the use of certain types of agreement prefixes with adjectives (e.g. -zuri "good", -kubwa "big", -moja "one", -wili "two"), pronouns (e.g. demonstrative -le "that/those"), and verbs that depend on that noun in a given phrase or sentence."

Mpiranya (2015). Swahili Grammar and Workbook.

Swahili (swh, Atlantic-Congo)

(3)	**Mw**anafunzi	**m**zuri	**yu**le	**a**li-soma	kitabu.
	student	good	that	he/she-PAST-read	book

"That good student read a book."

Problem: Implementing Morphological Features

Given productive agreement systems for gender, number, and case, it quickly becomes a formidable task to implement morphological features into a PSG. See below the examples for the word zuri “good” in Swahili.[7]

$A(SG, CL1) \rightarrow$ **m**zuri
$A(SG, CL2) \rightarrow$ **m**zuri
$A(SG, CL3) \rightarrow$ **ki**zuri
$A(SG, CL4) \rightarrow$ zuri
$A(SG, CL5) \rightarrow$ **n**zuri
$A(PL, CL1) \rightarrow$ **wa**zuri
$A(PL, CL2) \rightarrow$ **mi**zuri
$A(PL, CL3) \rightarrow$ **vi**zuri
$A(PL, CL4) \rightarrow$ **ma**zuri
$A(PL, CL5) \rightarrow$ **n**zuri

[7] This is based on my reading of the noun class system (CL) as defined by Mpiranya (2015), p. 22.

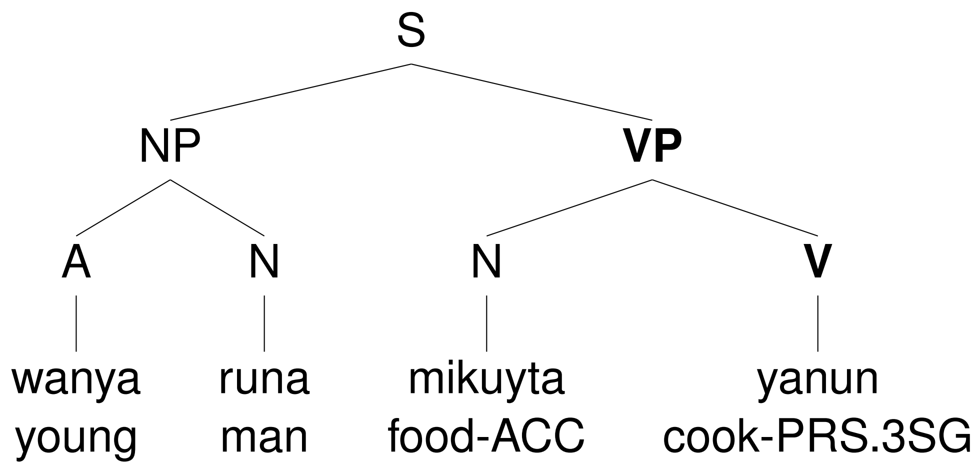
Section 6: Syntactic Phenomena

Verb Position

The position of the verb can be handled straightforwardly by changing its position on the *left and right hand side of rules*, i.e. adapting the rules of how to combine the verb with its complements (e.g. noun phrases).

Verb-final Position

Ayacucho Quechua (quy, Quechuan)



R (terminals)

1. A → wayna
2. N → runa
3. N → mikuyta
4. V → yanun

R (non-terminals)

5. S → **NP VP**
6. VP → **N V**
7. NP → A N

Rewrite Notation

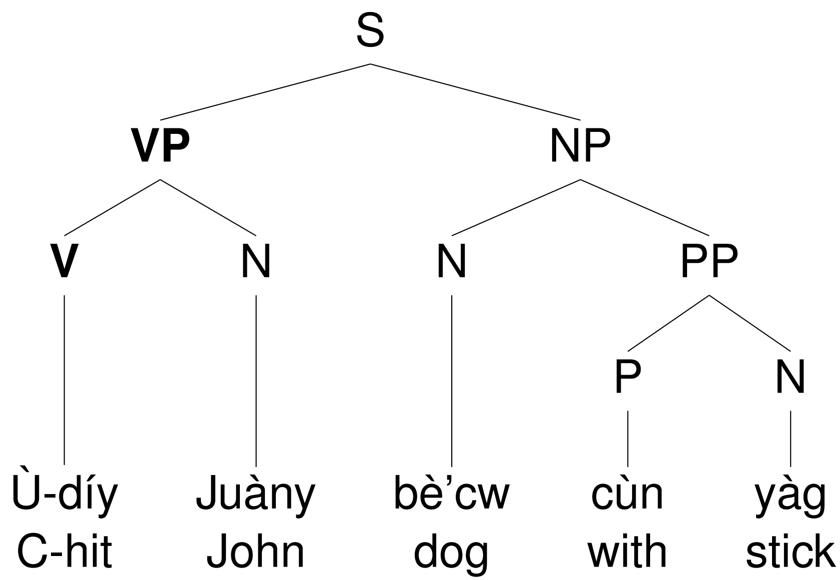
S
NP VP
NP N V
A N N V

---

wayna N N V
wayna runa N V
wayna runa mikuyta V
wayna runa mikuyta yanun

Verb-initial Position

Zapotec (???, Otomanguean)



R (terminals)

1. N → yàg
2. N → bè'cw
3. N → Juàny
4. P → cùn
5. V → Ù-díy

R (non-terminals)

5. S → **VP NP**
6. VP → **V N**
7. NP → N PP
7. PP → P N

Rewrite Notation

S
VP NP
V N NP
V N N PP
V N N P N

Ù-díy N N P N
Ù-díy Juàny N P N
Ù-díy Juàny bè'cw P N
Ù-díy Juàny bè'cw cùn N
Ù-díy Juàny bè'cw cùn yàg

Section 7: Pros and Cons of PSG

Pros (Advantages)

- Implements *linearization constraints* explicitely
- Is grounded on a solid mathematical footing (automata theory)
- Can be exdended to model morphological features
- Relatively easily implementable in computational frameworks

Cons (Disadvantages)

- The assumption that all languages need phrase structure rules for their grammatical description might not be valid (e.g. free word order)
- Implementation of morphological features can be cumbersome, especially for languages with productive morphological marking (though this is also an issue for other frameworks)
- It excludes semantic aspects from questions of grammaticality
- Without further constraints, there is an infinite number of PSGs that can generate any given sentence or set of sentences. Hence, it is unclear how to choose a particular PSG.

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Exercises: Tutorial Week 2

Exercise 1: Dependency Grammar

Take the Swiss German sentence that was mentioned in Lecture 4: *[...](dass) mer d'chind em Hans es huus lönd hälfe aanstriiche*. Dependencies for objects are already in the example in Lecture 4. Also, note that the relationships between the three verbs is such that *lönd* "let" is the main (finite) verb. The infinitive verb *hälfe* "to help" then depends on the main verb, and *aanstriiche* "to paint" in turn depends on *hälfe*. Do the following tasks:

- a) Add all other dependencies with labels. Disregard the complementizer *dass*.
- b) Calculate the average dependency length in this sentence.
- c) Calculate the number of crossing depencencies.

Exercise 2: Dependency Grammar

Translate the Swiss German sentence into English and do the same tasks as in Excercise 1. Compare the results and discuss why there are differences.

Exercise 3: Phrase Structure Grammar

Take the English sentence: *The child stole my money*. Assume that the word class (POS) of the possessive pronoun *my* is PRON. Do the following tasks:

- a) Write a PSG which can generate this sentence. Disregard morphological features. Apply the *binarization constraint*.
- b) What are all the possible sentences that your PSG can generate *without any further constraints*?
- c) What are all the possible sentences your PSG can generate *if all the non-terminal rules have to be applied at least once*?
- d) What are all the possible sentences your PSG can generate *if all the rules have to be applied at least once*?

Exercise 4: Phrase Structure Grammar

Take the corresponding sentence in Nhanda:

(4) abarla-lu wumba-yi wur'a-tha
child-ERG steal-PERF money-1SG.OBL
"The child stole my money."

Do the following tasks:

- a) Write a PSG generating the Nhanda sentence. Remember from Lecture 4 that the word order in Nhanda is completely free! Take this into account in your PSG. Disregard morphological features. Apply the binarization constraint.
- b) What are the sentences your PSG can generate *without further constraints*? How does this compare to the PSG for the English sentence(s)?
- c) How many sentences does the Nhanda PSG license if we introduce the additional constraint that all rules have to be applied at least once?

Section 8: 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.

Greville G. Corbett (2013). Number of Genders. In: Dryer, Matthew S. & Haspelmath, Martin (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/30, Accessed on 2019-10-31.)

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

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

Mpiranya, Fidèle (2015). Swahili Grammar and Workbook. London/New York: Routledge.

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.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Thank You.

Contact:

Faculty of Philosophy
General Linguistics
Dr. Christian Bentz
SFS Wihlemstraße 19-23, Room 1.24
chris@christianbentz.de
Office hours:
During term: Wednesdays 10-11am
Out of term: arrange via e-mail