UNIVERSITÄT DES SAARLANDES

Preparatory Course for Master's Students: Syntax II

October 2011

Alexis Palmer,
based on slides from Judith Köhne

Recap

- Syntax aims to account for grammaticality
- Syntactic categories (lexical, functional, phrasal) help us generalize
- Constituency tests: substitution, movement, question, coordination
- PS-rules one way to encode this knowledge

Building up the sentence - Syntactic constituents

- Heads
  - Contains the most important semantic information
  - Usually obligatory
  - Selects for dependents of a certain category (P selects NP)
  - Often triggers agreement (V in VP)
  - Requires certain case of dependents

Building up the sentence - Syntactic constituents

- Constituency tests: Movement test

Tim embarrasses Anna in front of the bakery.

Anna, Tim embarrasses in front of the bakery.

In front of the bakery, Tim embarrasses Anna.

Embarrass Anna in front of the bakery, Tim did.

*Front of the bakery, Tim embarrasses Anna.

* Embarrasses, Tim Anna in front of the bakery.

Building up the sentence - Syntactic constituents

- Constituency tests: Question test

Tim embarrasses Anna in front of the bakery.

Who embarrasses Anna in front of the library? *Tim*

Whom does Tim embarrass in front of the library? *Anna*

Where does Tim embarrass Anna? *In front of the library*

In front of what does Tim embarrass Anna? *the library*

What does Tim do? *Embarrass Anna in front of the library*

Building up the sentence - Syntactic constituents

- Constituency tests: Coordination test

Tim embarrasses Anna in front of the bakery.

Tim and Tom…

Tim embarrasses and kisses Anna…

Tim embarrasses Anna and Tina…

Tim embarrasses Anna in front of the library and the church

Tim embarrasses Anna in front of the library and close to the church

*Tim embarrasses Anna in front of the and a library.

Syntactic categories

A syntactic category is a set of expressions

- Parts of speech (noun, verb, determiner etc.)
- Constituents (NP, VP, etc.) / heads of constituents

which share (morpho-)syntactically relevant features

- Distribution
- Morpho-syntactic features

Syntactic categories

[[[The]$_{\text{DET}}$ [boy]$_{\text{N}}$]$_{\text{NP}}$ [[loves]$_{\text{V}}$ [[the]$_{\text{DET}}$ [tasty]$_{\text{ADJ}}$ [cake]$_{\text{N}}$]$_{\text{NP}}$]$_{\text{S}}$

[[[This]$_{\text{DET}}$ [flower]$_{\text{N}}$]$_{\text{NP}}$ [[needs]$_{\text{V}}$ [[some] $_{\text{DET}}$ [water]$_{\text{N}}$]$_{\text{NP}}$]$_{\text{VP}}$] $_{\text{S}}$

- Shared distribution of determiners

Syntactic categories

[[[The]$_{DET}$ [boy]$_{N}$]$_{NP}$ [[loves]$_{V}$ [[the]$_{DET}$ [tasty]$_{ADJ}$ [cake]$_{N}$]$_{NP}$]$_{S}$

[[[This]$_{DET}$ [flower]$_{N}$]$_{NP}$ [[needs]$_{V}$ [[some] $_{DET}$ [water]$_{N}$]$_{NP}$]$_{VP}$]$_{S}$

- Shared distribution of noun phrases

Syntactic categories

- Terminological/definitional issues
  - Often (and in particular by formal syntacticians) only distribution is considered as criterium and only constituents or heads of constituents are meant
  - Used in such a way, the notion of ‘syntactic categories’ is a purely syntactic one (in particular for describing generative grammar)
  - Often the terms syntactic categories, parts of speech and word classes are used in a substitutable way

What does a sentence need? - Syntactic functions

[The boy]$_{SUB}$ [loves]$_{PRED}$ [the tasty cake]$_{DIR\_OBJ}$

[He]$_{SUB}$ [put]$_{PRED}$ [it]$_{DIR\_OBJ}$ [on the table]$_{PREP\_OBJ}$

- Syntactic/sentence functions describe the *roles* of constituents in the sentence
- Complements (subject, objects) & adjuncts/modifiers
- Syntactic *requirements, argument structure*
- *Again, the universality of (some) syntactic functions is controversial*

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

The new teacher reads a short book in the library.

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

The new teacher reads a short book in the library.

Det A N V Det A N P Det N

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

[The new teacher] reads [a short book] [in the library].

SUBJECT OBJECT ADJUNCT

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

The new teacher reads a short book in the library.

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

The new teacher reads a short book in the library.

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

[The new teacher] reads [a short book] [in the library].

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

The new teacher reads a short book in the library.

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

[The new teacher] [reads [a short book] [in the library]].

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

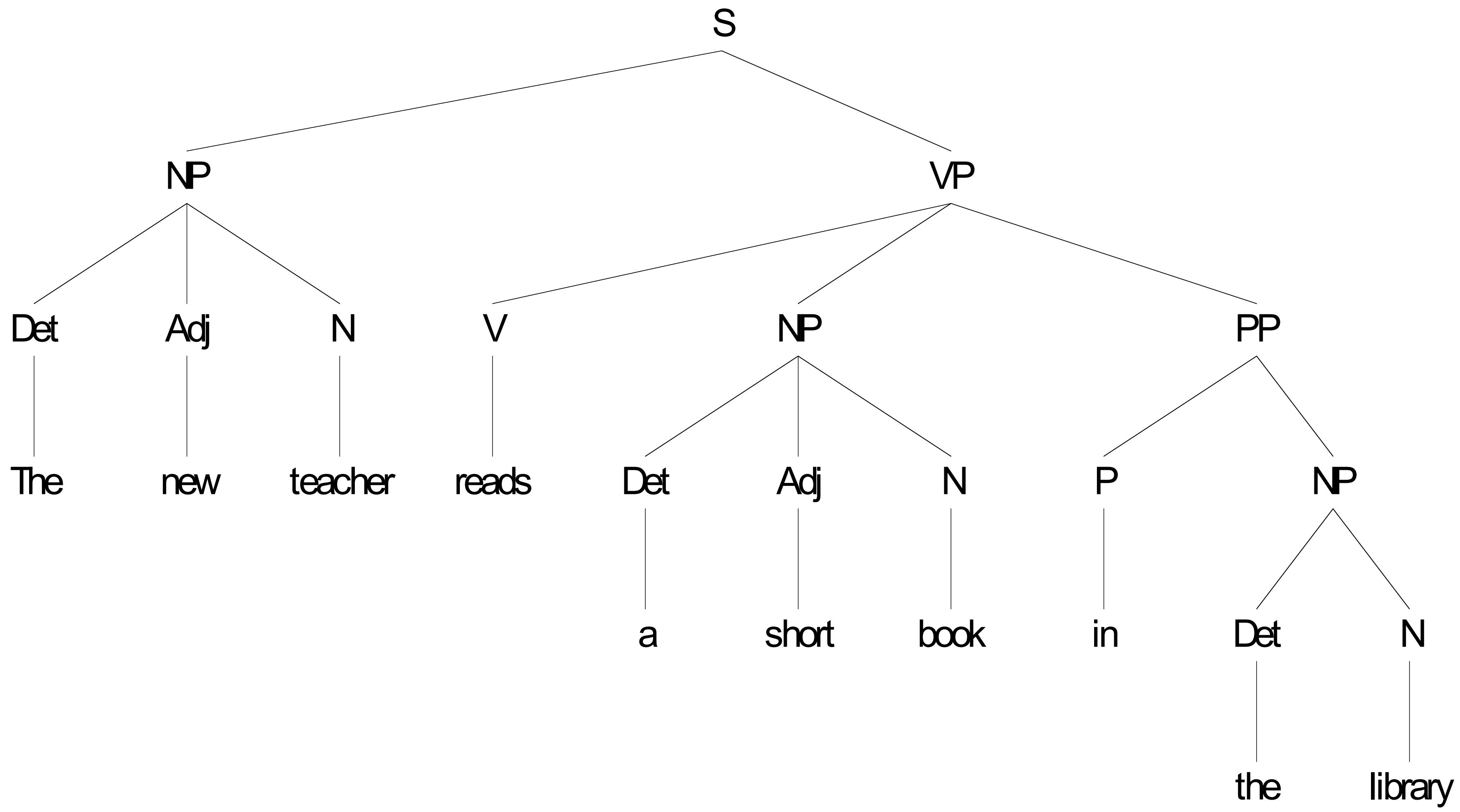
The new teacher reads a short book in the library.

Exercise I

- For the following sentence, define
  - the parts of speech
  - syntactic functions
  - Constituents (movement test, question test, coordination test)

[The new teacher] [[[reads] [a short book]] [in [the library]]].

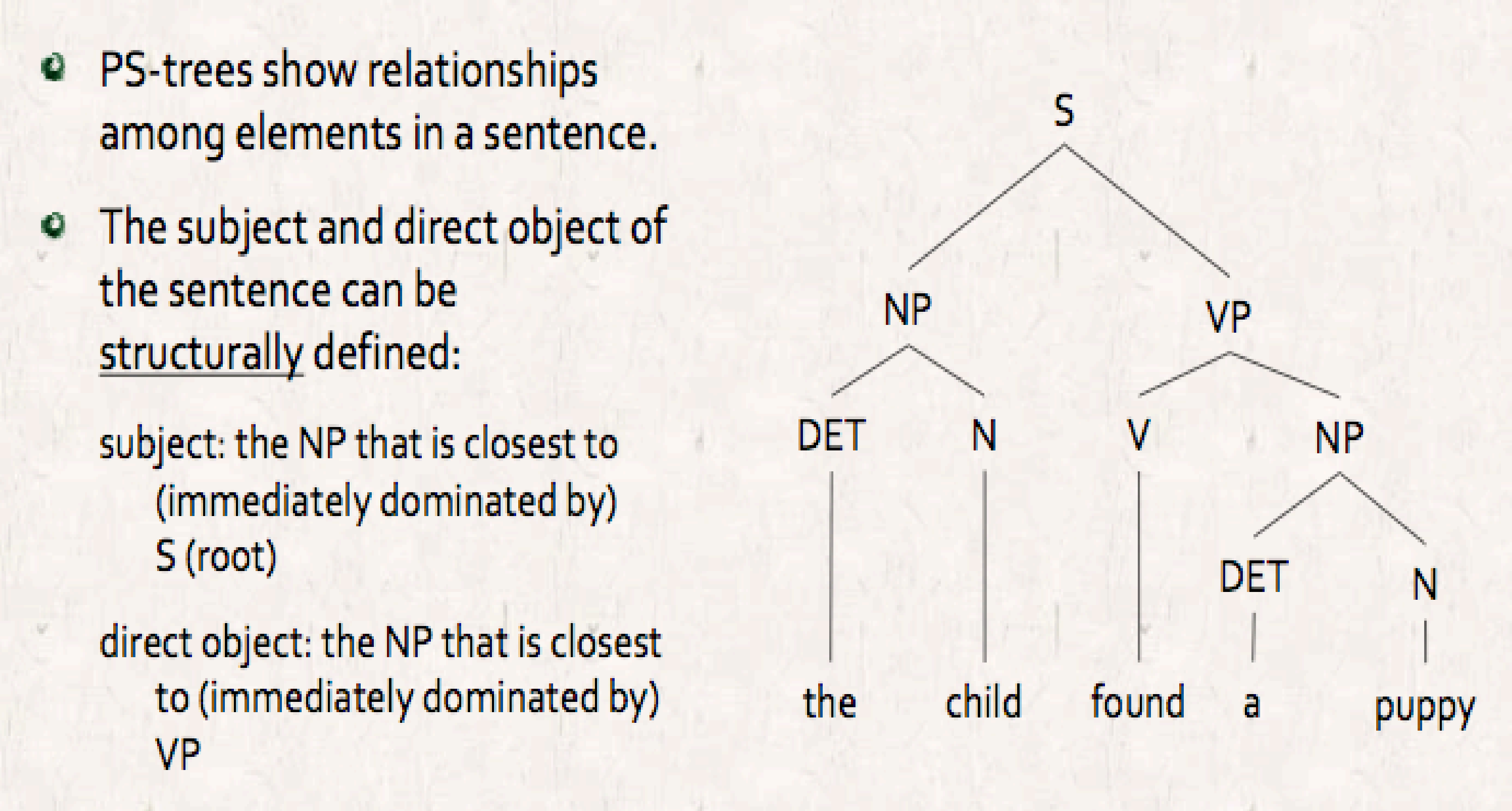
Exercise I


S
NP
VP
Det
Adj
N
V
NP
PP
The
new
teacher
reads
Det
Adj
N
P
NP
a
short
book
in
Det
N
the
library

Next

- X’ (X-bar) theory
- Structural ambiguity
- Grammars & grammatical formalisms

Heads and Complements

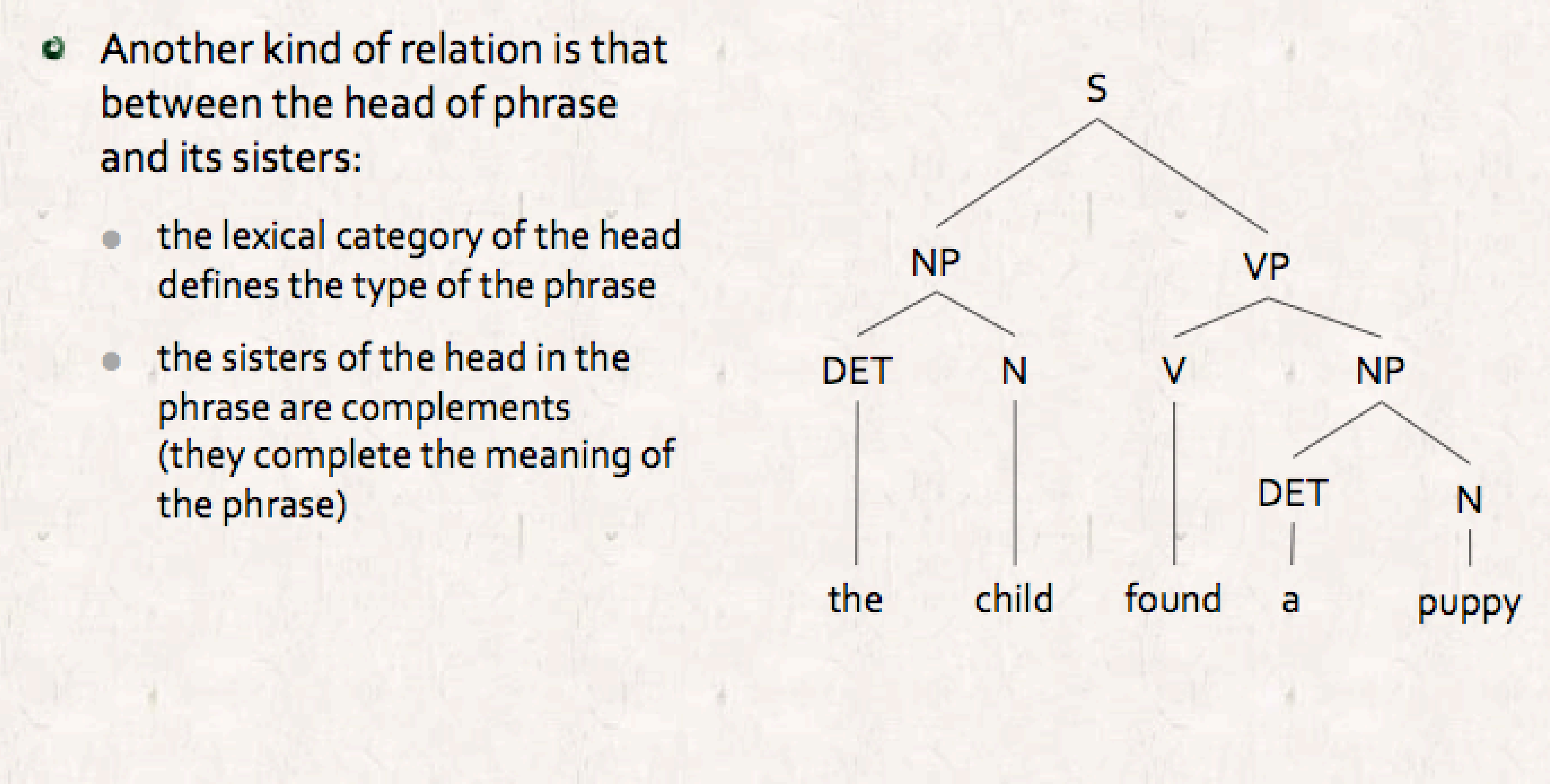


- PS-trees show relationships among elements in a sentence.
- The subject and direct object of the sentence can be structurally defined:

subject: the NP that is closest to (immediately dominated by) S (root)

direct object: the NP that is closest to (immediately dominated by) VP

Heads and Complements

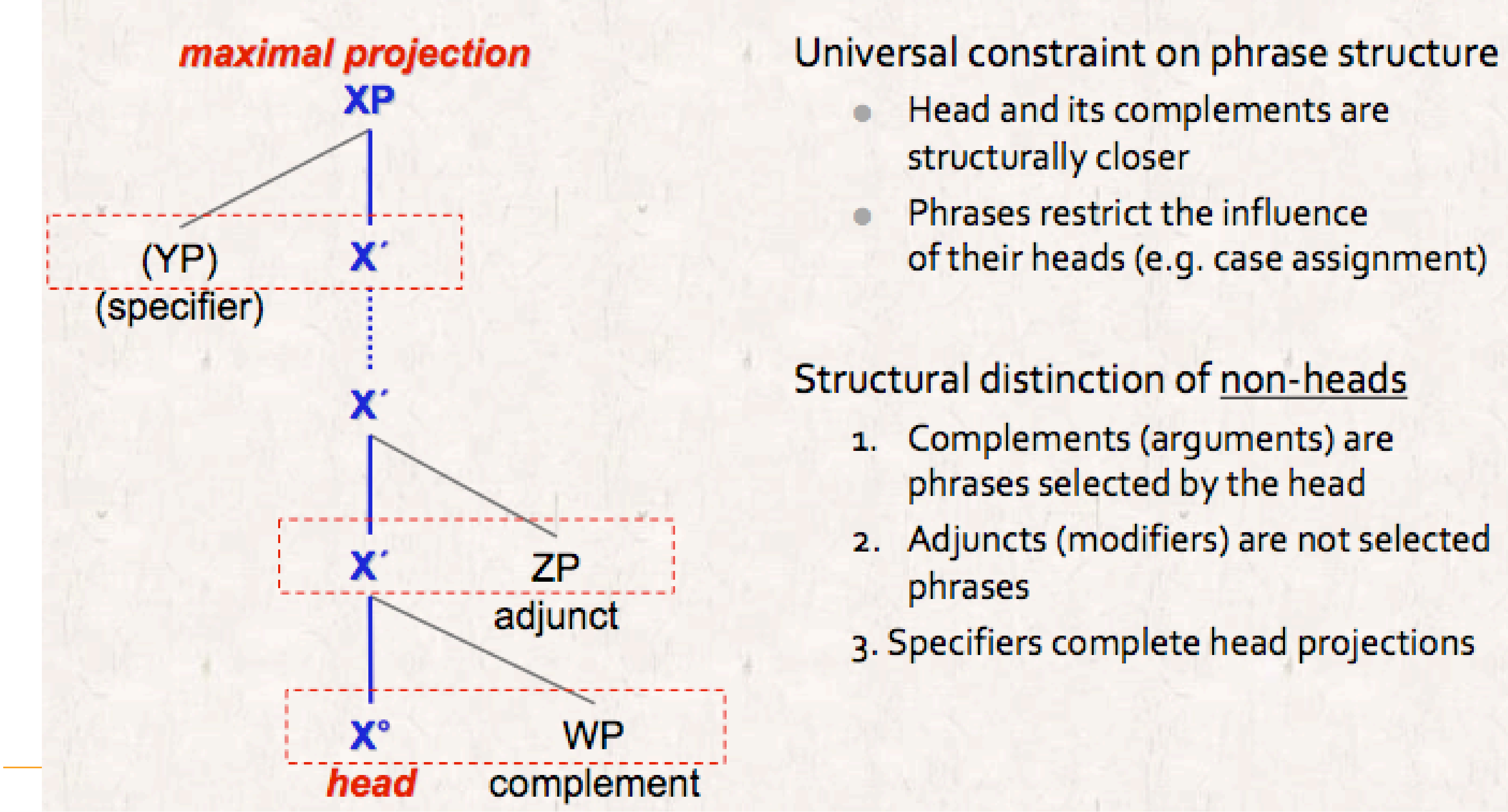


- Another kind of relation is that between the head of phrase and its sisters:
  - the lexical category of the head defines the type of the phrase
  - the sisters of the head in the phrase are complements (they complete the meaning of the phrase)

Heads and Complements

- The information about the complement(s) selected by a particular lexical item is referred to as subcategorization, or c-selection
- Complements
  - Are closer to the head than modifiers
  - Combine w/ lexical head at an intermediate phrasal level
- Modifiers
  - May iteratively combine with such an intermediate phrase
  - *But* not all modifiers have the same status (e.g. determiners vs. adjectives, with respect to nouns)

Phrases as Head Projections: X' (X-bar) Scheme



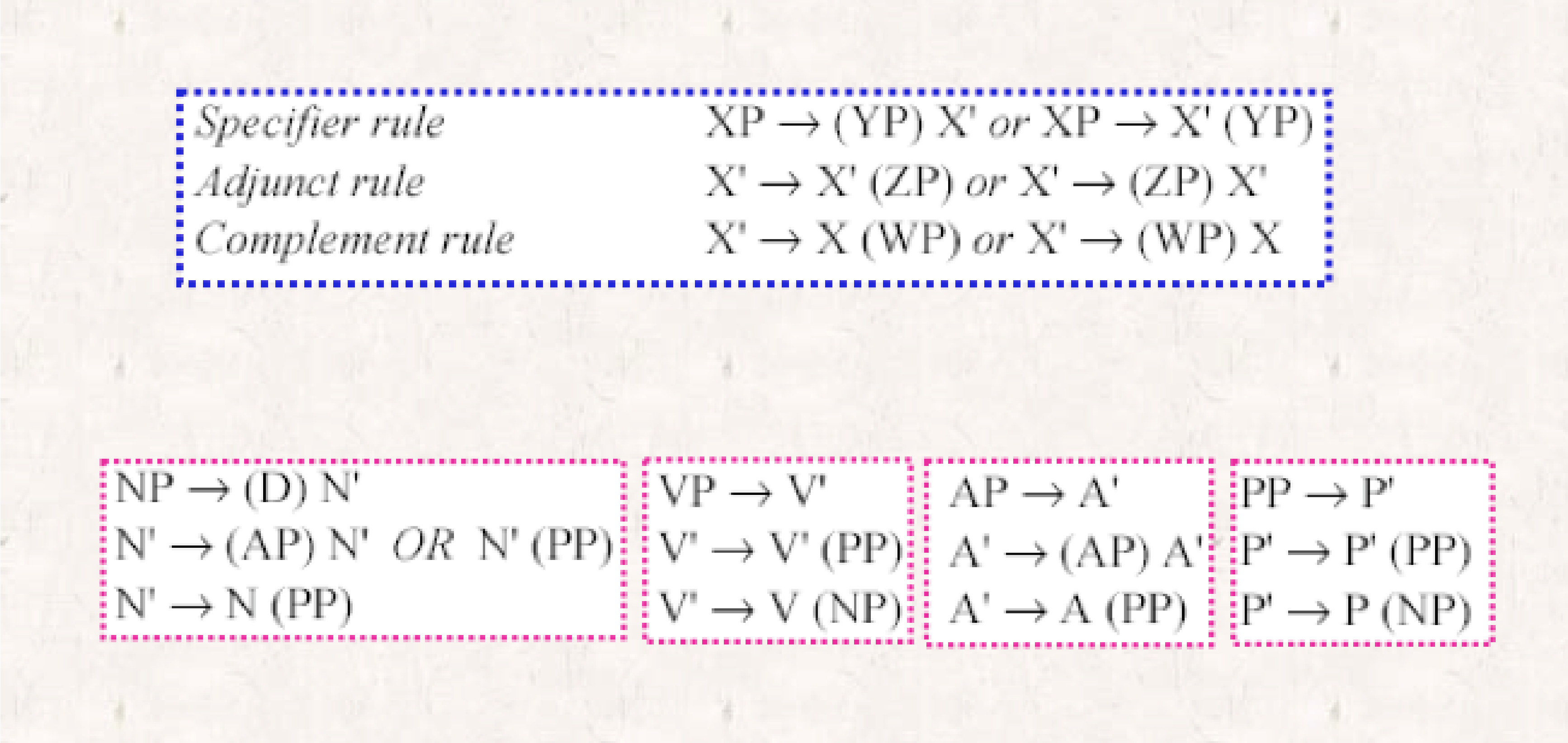
Universal constraint on phrase structure

- Head and its complements are structurally closer
- Phrases restrict the influence of their heads (e.g. case assignment)

Structural distinction of <u>non-heads</u>

1. Complements (arguments) are phrases selected by the head
2. Adjuncts (modifiers) are not selected phrases
3. Specifiers complete head projections

Generalization of X’ rules



Specifier rule	XP → (YP) X' *or* XP → X' (YP)
Adjunct rule	X' → X' (ZP) *or* X' → (ZP) X'
Complement rule	X' → X (WP) *or* X' → (WP) X

NP → (D) N'
N' → (AP) N' *OR* N' (PP)
N' → N (PP)

VP → V'
V' → V' (PP)
V' → V (NP)

AP → A'
A' → (AP) A'
A' → A (PP)

PP → P'
P' → P' (PP)
P' → P (NP)

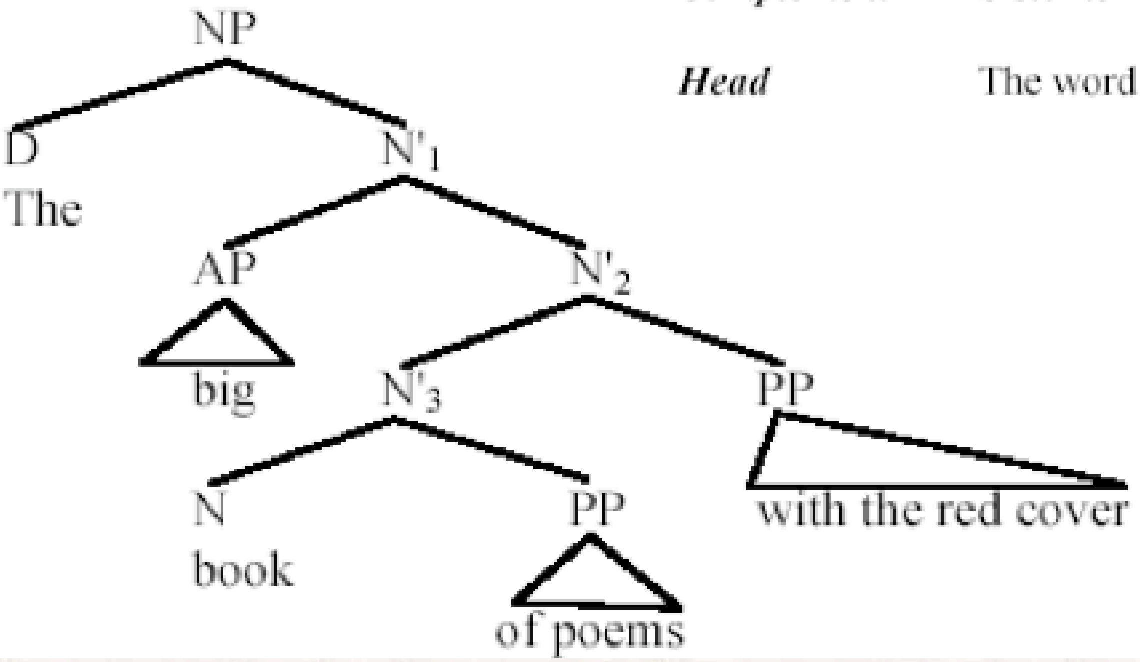
Generalization of X' rules

Specifier Sister to X', daughter of XP.

Adjunct Sister to X', daughter of X'.

Complement Sister to X, daughter of X'.

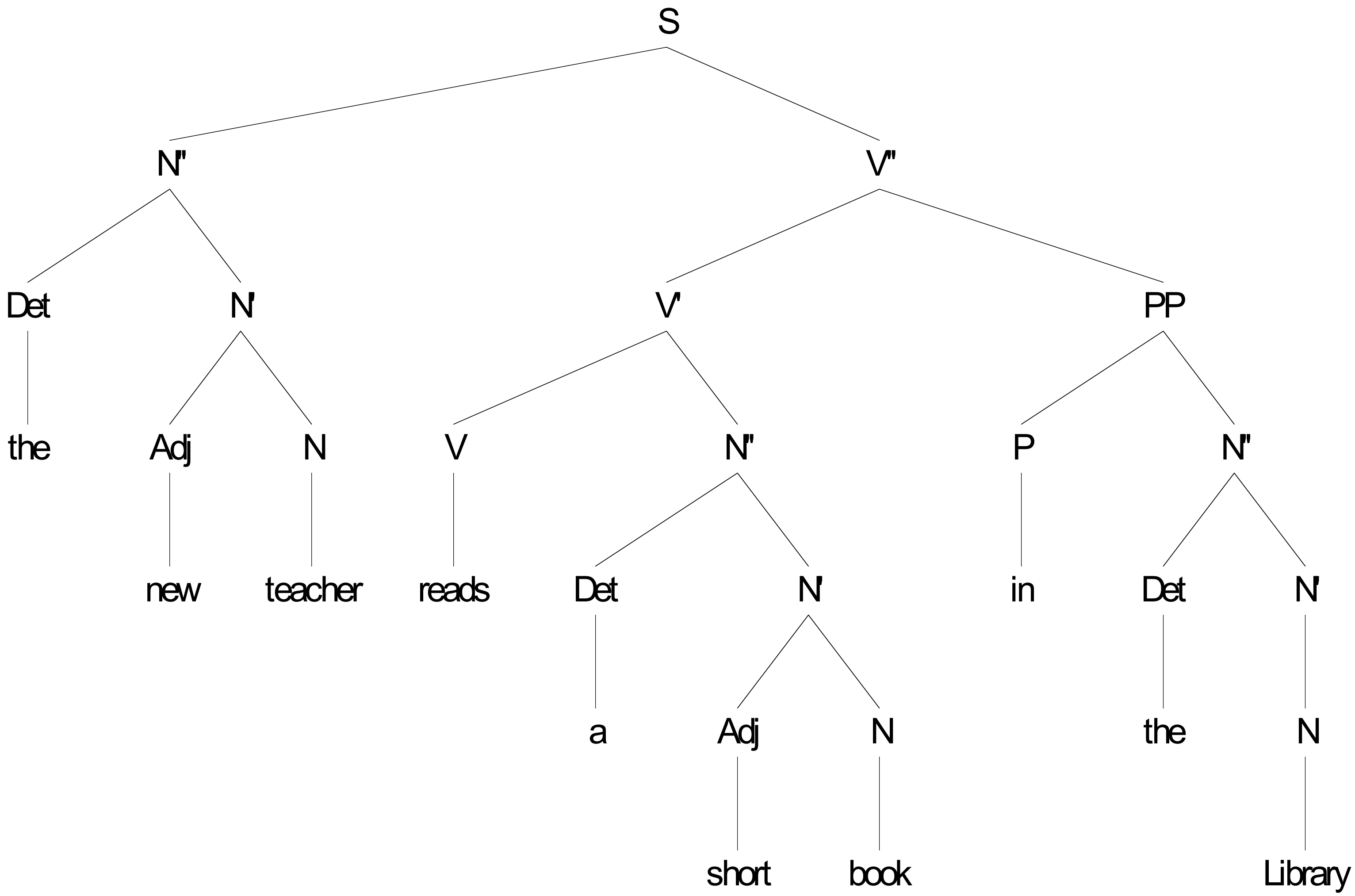
Head The word that gives its category to the phrase.



Exercise I

- Draw X’-style tree for sentence ‘The new teacher reads a short book in the library’

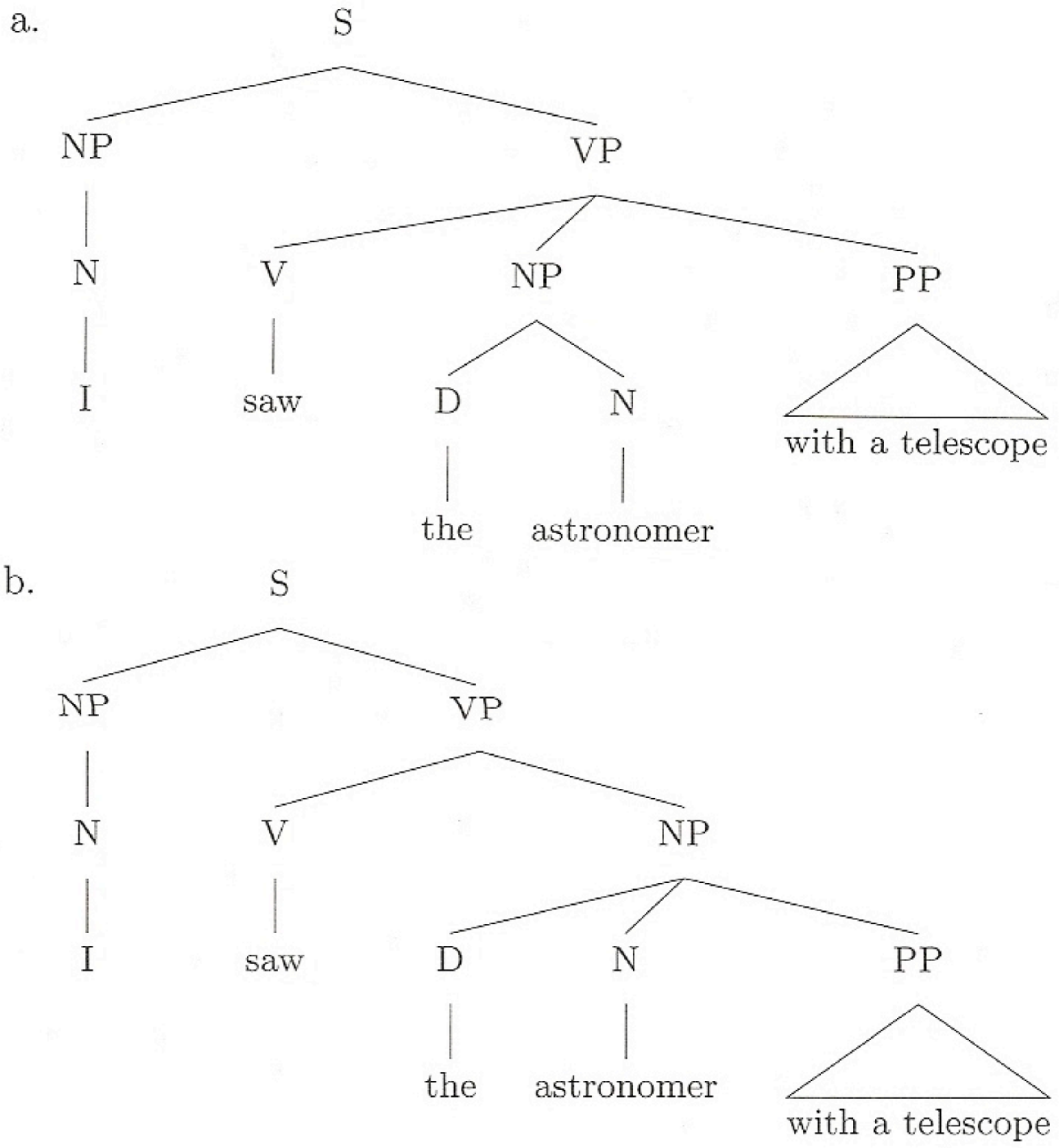
Exercise I


S
N"
V"
Det
N'
V'
PP
the
Adj
N
V
N"
P
N"
new
teacher
reads
Det
N'
in
Det
N'
a
Adj
N
the
N
short
book
Library

Structural Ambiguity

- One sentence can have different phrase structures > different meanings
- The girl watched the man with the telescope.

- [[I]$_{NP}$ [saw [the astronomer [with the telescope]$_{PP}$]$_{NP}$]$_{VP}$]$_{S}$.
- [[I]$_{NP}$ [saw [the austronomer]$_{NP}$ [with the telescope]$_{PP}$]$_{VP}$]$_{S}$.



Sag et al. (2003), p.28

Exercise I

Assume the following grammar:

S → NP VP

VP → V (NP) (PP)

NP → Det A* N PP*

PP → P NP

V: watched

P: beside, with

Det: the

N: birds, fleas, dog, flowers

A: big, brown

Draw both tree structures for the folowing sentence:

(1) The big brown dog with fleas watched the birds beside the flowers.

Grammar/Syntactic Theories

Grammar and Grammar Theories

- Grammar (prescriptive): body of rules how to build sentences (as grammar learned in school)
- Grammar (descriptive): the rules a language follows, including syntactic, semantic, and phonological rules
- Mental grammar is unconscious
- Universal grammar
- Grammar Theories: Formal frameworks for describing grammar

Grammar and Grammar Theories

- Chomsky Hierarchy: containment h. of formal grammars
- Formal grammars: terminal & non-terminal symbols, production rules, start symbol
- Hierarchy
  - Type-0 Unrestricted (Turing machine)
  - Type-1 Context-sensitive (all language that can be recognized by a linear bounded automaton)
  - Type-2 Context-free (non-deterministic pushdown automaton)
  - Type-3 Regular (finite-state automaton)

Context-Free Grammars (CFG)

- A context-free grammar consists of
- Two finite, non-empty sets of symbols, terminal symbols and non-terminal symbols
- A finite, non-empty set of context-free rules (A ➔ B_1 … B_N)
- A start symbol *S*

Context-Free Grammars (CFG)

- S → NP VP
- VP → V NP (NP) (PP)
- NP → (DET) (A) N
- PP → P NP

Context-Free Grammars (CFG)

- S → NP VP
- VP → V NP (NP) (PP)
- NP → (DET) (A) N (PP)
- PP → P NP

- Recursion
- Sentences can be infinitely long

Context-Free Grammars (CFG)

- S → NP VP
- VP → V NP (NP) (PP)
- NP → (DET) (A) N
- PP → P NP

- Rules are local and simply provide constraints on well-formed structure

Phrase Structure Grammar (PSG)

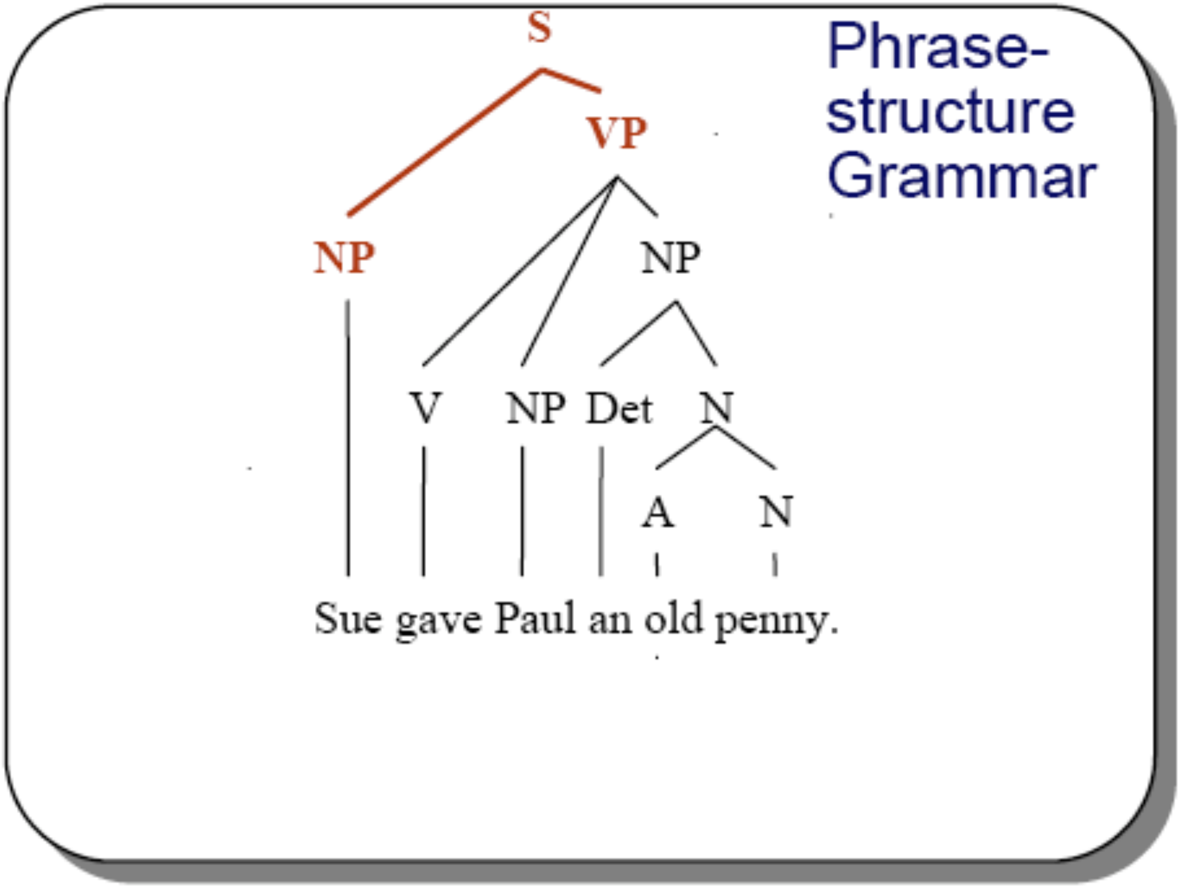
- Beginnings PSG: American Structuralism (Wundt; Bloomfield 1935)
  - Division of complex expressions into other complex expressions, relations among them
  - Description of corpora sentences

Phrase Structure Grammar (PSG)

- Birth of *Generative PSG*: Chomsky 1957
  - Grammaticality *ratings*, *acceptability* > which rules and principles follow
  - Set of sentences considered infinite
  - Syntax as *generator*: generate *all* acceptable sentences (*completeness*) but *only* those (*correctness*)

Phrase Structure Grammar (PSG)

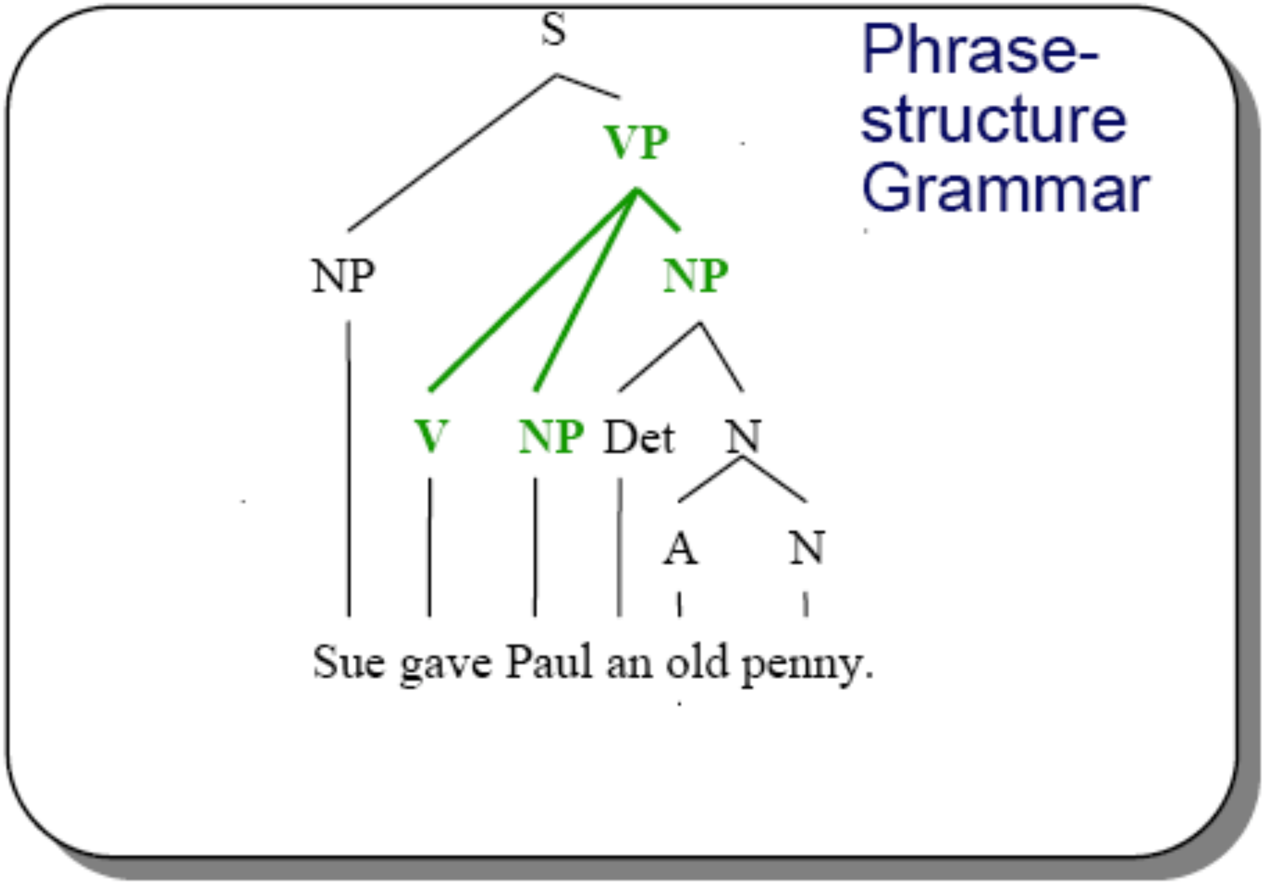
S → NP VP



Phrase Structure Grammar (PSG)

S → NP VP

VP → V NP NP

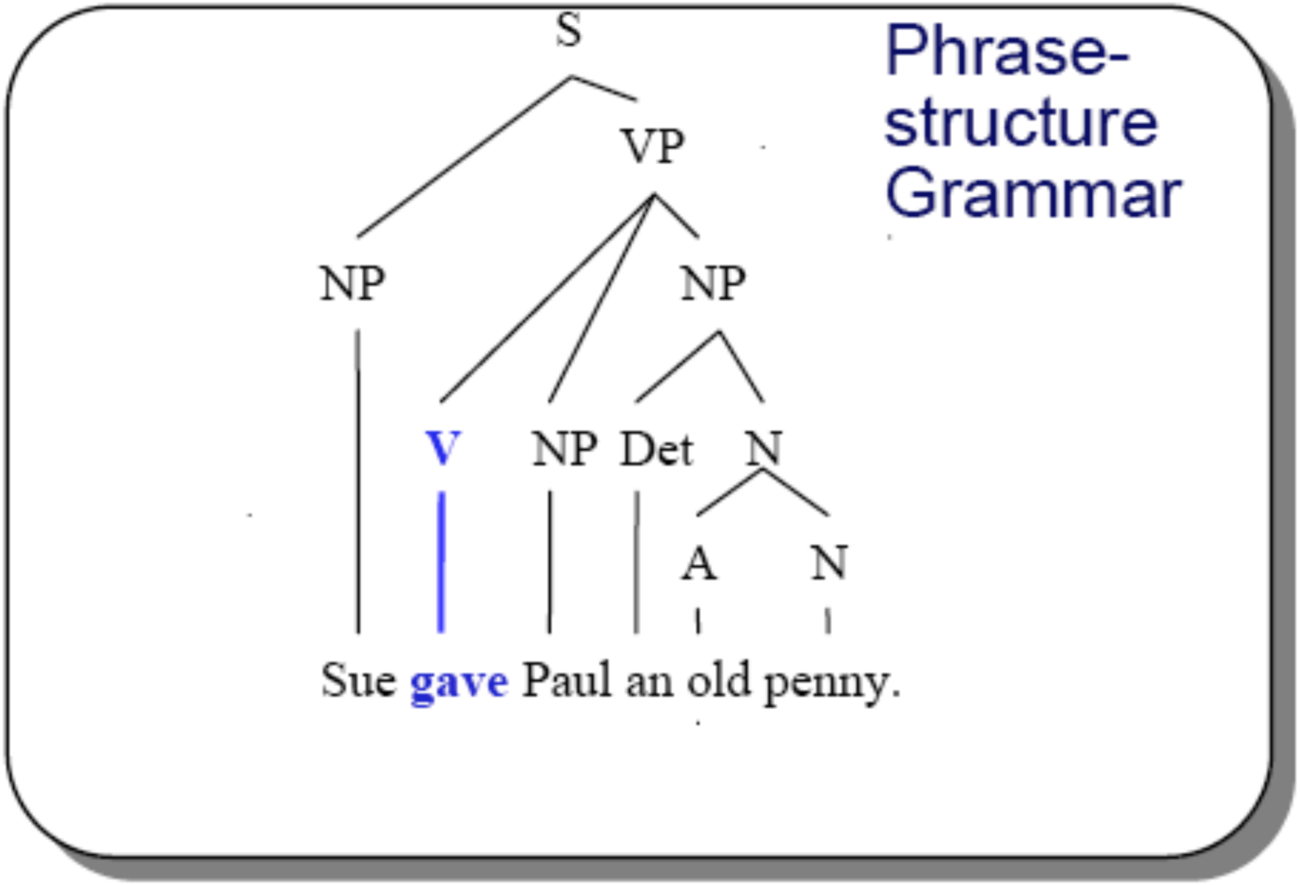


Phrase Structure Grammar (PSG)

S → NP VP

VP → V NP NP

V → gave



Phrase-structure Grammar

A note on Transformational Grammar (GB, P&P)

- Syntax should not just describe the structure of sentences/constructions but also the systematic relations *between sentences, e.g. active-passive*

  (1) Angela welcomes Guido

  (2) Guido is welcomed (by Angela)

- (1) and (2) are said to have the same *Deep Structure* but (via *transformations*) different *Surface Structures*

A note on Transformational Grammar (GB, P&P)

- Transformational Grammar Formalisms are not very applicable for Computational Linguistics
  - Complex
  - Not precise
  - Error-prone

Context-Free Grammars (CFG)

- CFGs allow to decide whether a sentence is grammatical and to assign each sentence an appropriate grammatical structure

- However, what does grammaticality mean? What to do with ‘ungrammatical’ input?

- Is CFG the right way to describe the grammar of natural languages?

Applications for CFG

- Morphological Parsers
- Syntax of programming languages
- However, problem with natural languages

Limitations of CFG

- Agreement (*The dogs sees a house.)
- Subcategorization/ argument structure /verb requirements (*The boy sleeps a house.)
- It is possible to introduce new categories (VP1, VP2…) but we would need a lot and the formal relations would not be accounted for

Generalized/Head-driven PSG (Gazdar, Pollard, Sag)

- Generative PSG with feature-structures (features & values)
- Syntax, semantics, and phonology
- Possibility to integrate argument structure requirements and agreement limitations
- HPSG: Systematise the notion of head (direct way to express what a phrase and its head have in common, i.e. agreement)
- Very local syntactic rules (as PS rules)

HPSG

- Signs
  - Words
  - Phrases
- Represented by ATTRIBUTE-*value* Matrixes (AVMs) (e.g. CASE *nom*)
- Each ATTRIBUTE is of a certain *type* (e.g. SYN has the type *synsem*)

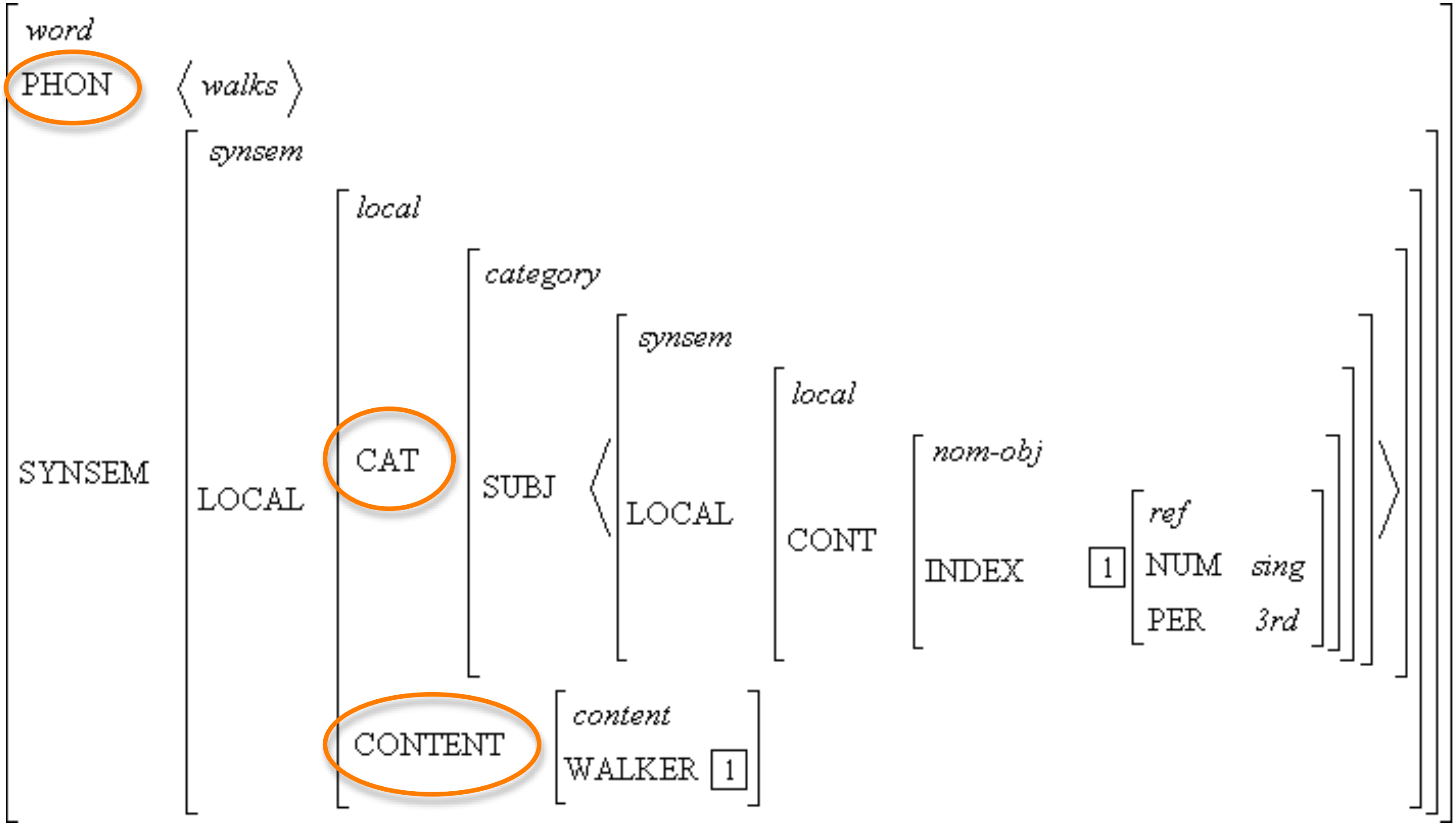
HPSG

- PHON (phonology), CATEGORY (syntax), CONTENT (semantics), CONTEXT (pragmatics)

- CATEGORY: HEAD, COMP (complements), SPEC (specifier)
- HEAD: AGR (agreement)

HPSG

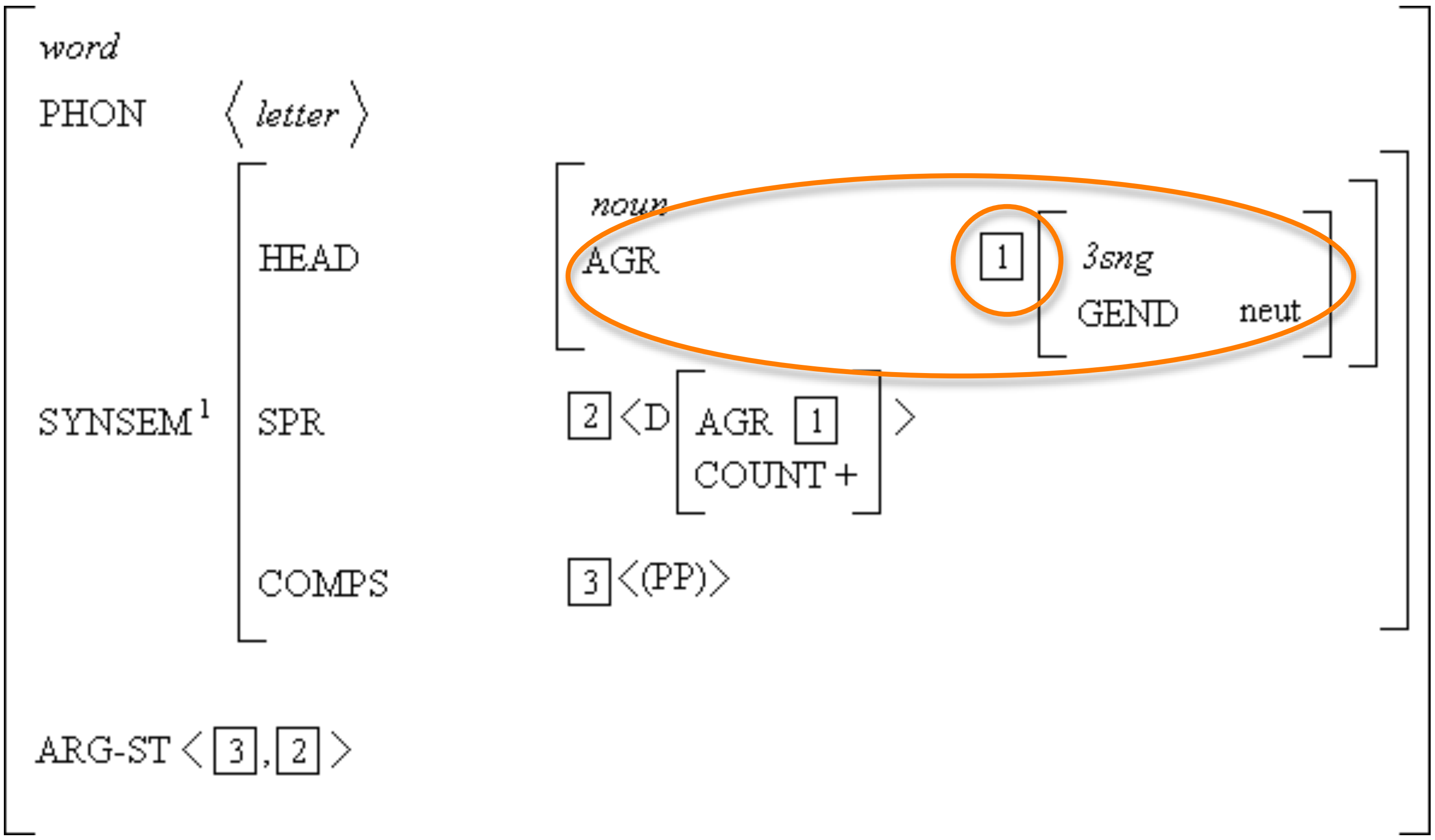
(1)



$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{SUBJ} & \left\langle \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CONT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \textit{ref} & \\ \text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \textit{content} & \\ \text{WALKER} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

HPSG

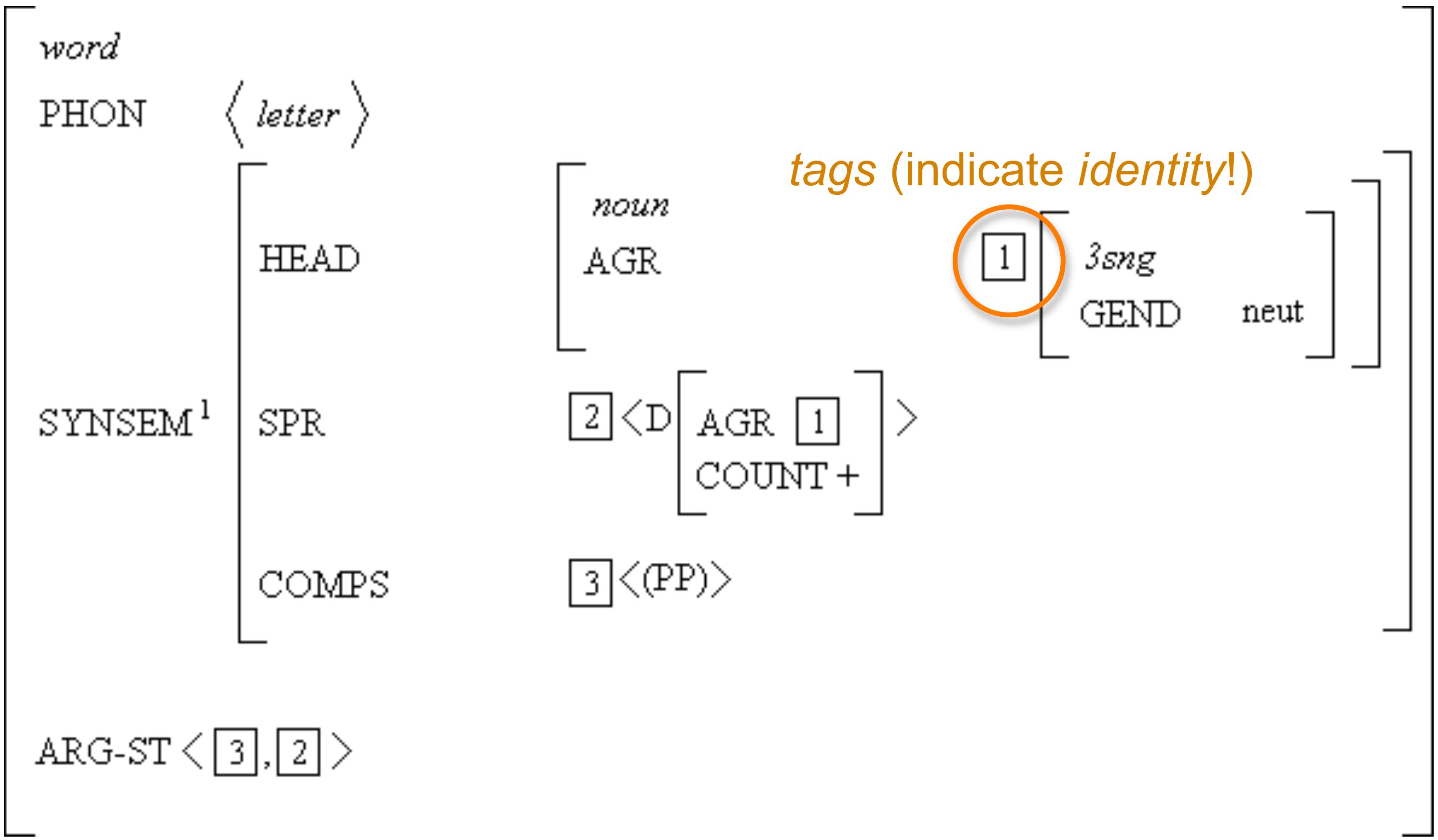
(2)



$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{letter} \rangle \\ \text{SYNSEM}^1 & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1} \begin{bmatrix} \textit{3sng} & \\ \text{GEND} & \text{neut} \end{bmatrix} \end{bmatrix} \\ \text{SPR} & \boxed{2} \langle \text{D} \begin{bmatrix} \text{AGR} & \boxed{1} \\ \text{COUNT} & + \end{bmatrix} \rangle \\ \text{COMPS} & \boxed{3} \langle (\text{PP}) \rangle \end{bmatrix} \\ \text{ARG-ST} \langle \boxed{3}, \boxed{2} \rangle & \end{bmatrix}$$

HPSG

(2)



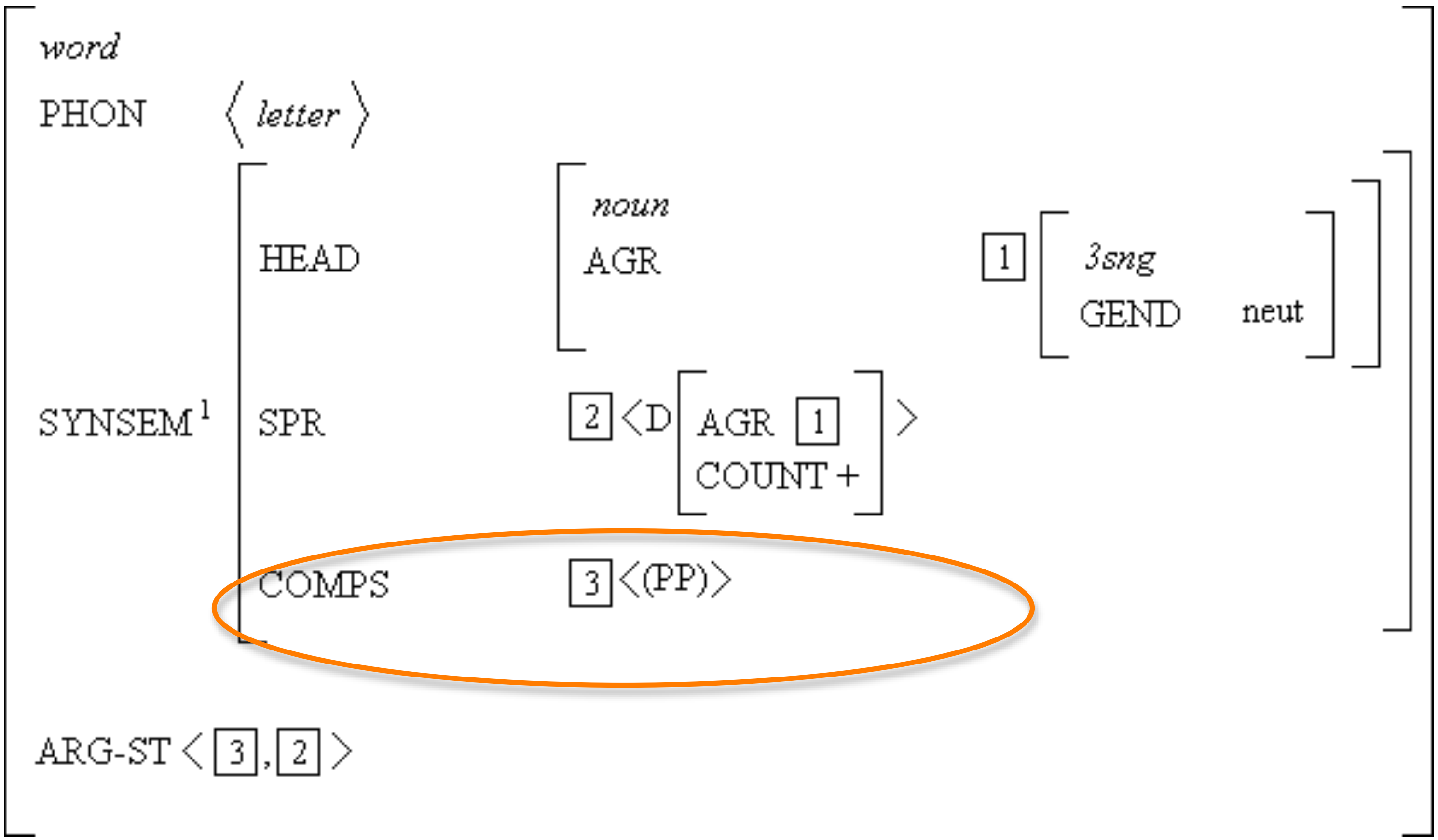
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{letter} \rangle \\ \text{SYNSEM}^{1} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \boxed{1}\begin{bmatrix} \textit{3sng} & \\ \text{GEND} & \text{neut} \end{bmatrix} \end{bmatrix} \\ \text{SPR} & \boxed{2}\langle \text{D}\begin{bmatrix} \text{AGR} & \boxed{1} \\ \text{COUNT} & + \end{bmatrix}\rangle \\ \text{COMPS} & \boxed{3}\langle(\text{PP})\rangle \end{bmatrix} \\ \text{ARG-ST}\langle \boxed{3}, \boxed{2} \rangle & \end{bmatrix}$$

tags (indicate *identity*!)

Riehemann (1995), p.1

HPSG

(2)



word

PHON ⟨ *letter* ⟩

SYNSEM[1]

HEAD

noun

AGR

1

3sng

GEND neut

SPR

2 ⟨D

AGR 1

COUNT + ⟩

COMPS

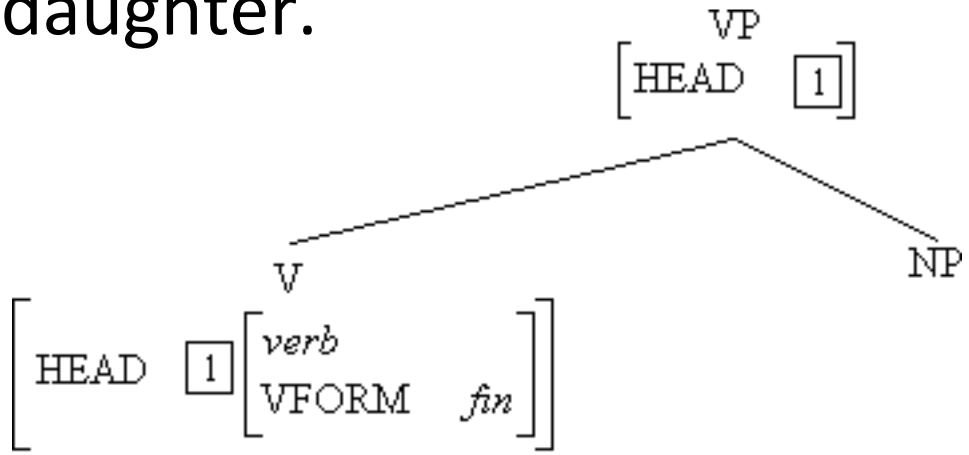
3 ⟨(PP)⟩

ARG-ST ⟨ 3 , 2 ⟩

Carnie (2002), p.363

HPSG – Rules & principles

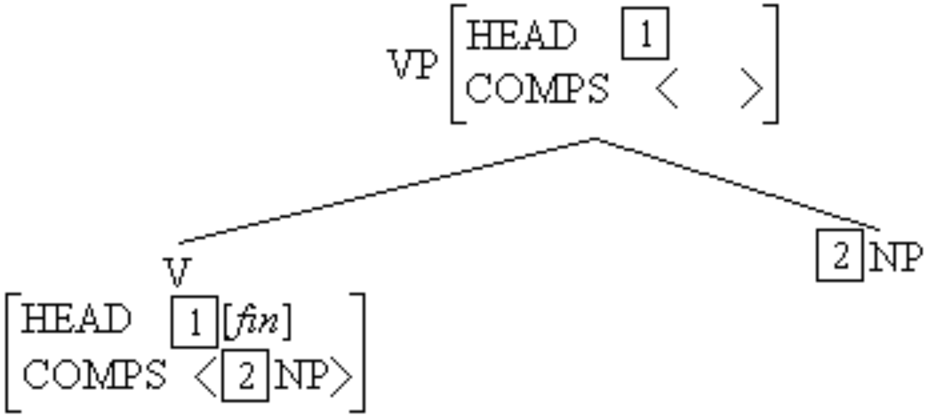
- Feature structures interact with rules and principles > well-formed expressions
- e.g., Head Feature Principle (HFP): The HEAD value of a headed phrase is identified with that of its head-daughter.



Kim (2000), p.9

HPSG – Rules & principles

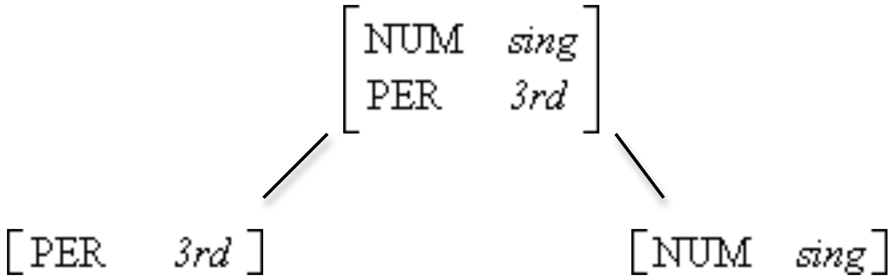
- e.g., Valence Principle (VALP): For each valence feature F, the F value of a headed phrase is the daughter's F value minus the realized non-head-daughters.



Kim (2000), p.10

HPSG – Rules & principles

- HPSG is *unification-based*: phrases to be combined are unified



Generalized/Head-driven PSG

- Advantages for CL
  - Relatively simple: Only one level of syntactic structure
  - Well-understood and precise
  - Restrictive (limited range of potential solutions) ot any descriptive problem

Syntactic Theory in Computational Linguistics

- *Parsing*: the input is a sentence and the output is a syntactic analysis and/or the acceptability of the sentence (sentence or not)

- *Generation*: the input is a meaning representation, the output a valid sentence

- Both tasks are often subparts of practical applications, e.g., *Machine Translation* (MT) and *Dialogue systems*

Syntactic Theory in Computational Linguistics

- The grammars that are useful for parsing or generation should meet some criteria:
  - Accuracy: gives correct analysis
  - Precision: tells a computer exactly what to do
  - Efficiency: is able to parse a sentence and to return one or only a small number of parses
  - Usefulness: it is relatively easy to map a syntactic structure to its meaning

More Grammars

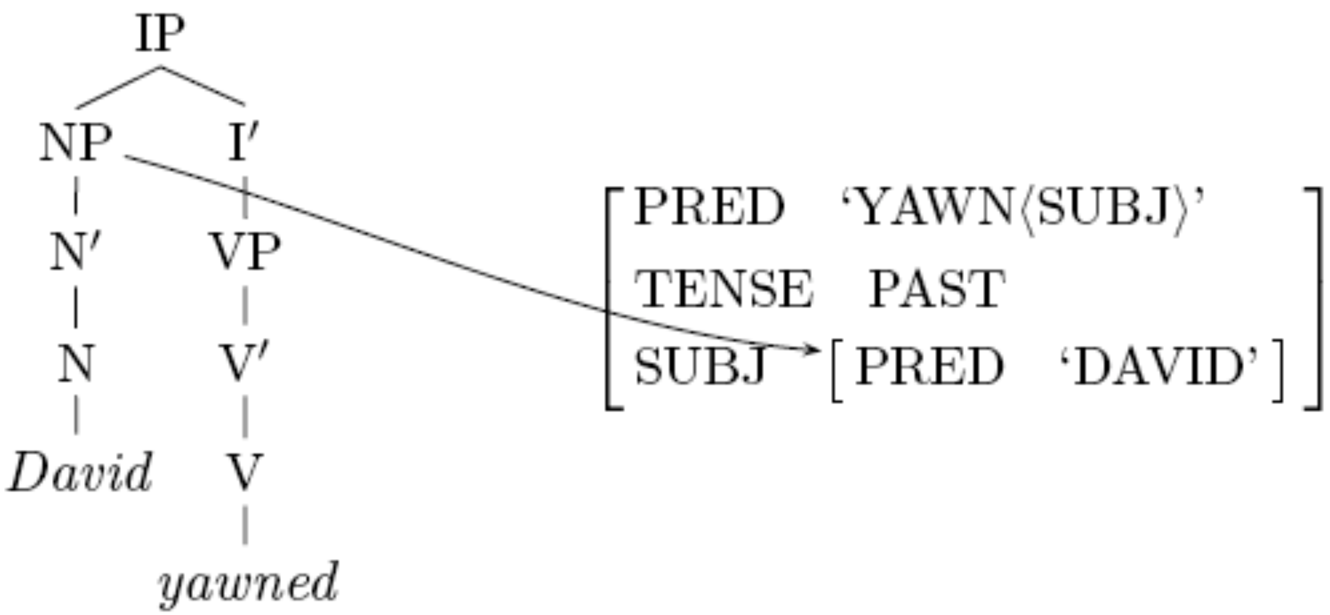
- LFG (Lexical-Functional Grammar)
- CCG (Combinatory Categorial Grammar)
- TAG (Tree-Adjoining Grammar)
- DG (Dependency Grammar)
- CG (Construction Grammar)
- R&R (Role and Reference Grammar)

Lexical-Functional Grammar (Bresnan & Kaplan)

- Unification-based generative grammar
- Focus on syntax (but including on its relationship to morphology and semantics)
- f-structure (representation of grammatical fuctions) & c-structure (structure of syntactic constituents
- Further structures for argument structure, morphology, phonology, semantics, & information st.
- Grammatical-function changing operations (passivization) are described as lexical

LFG (Bresnan & Kaplan)

(33) *David yawned.*

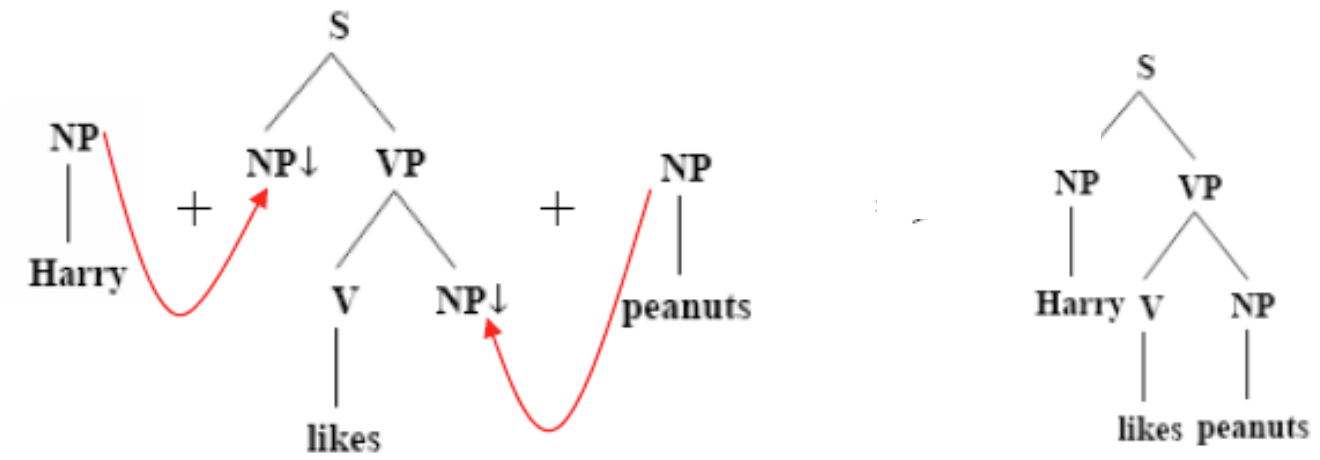


Dalrymple, 2006, p.15

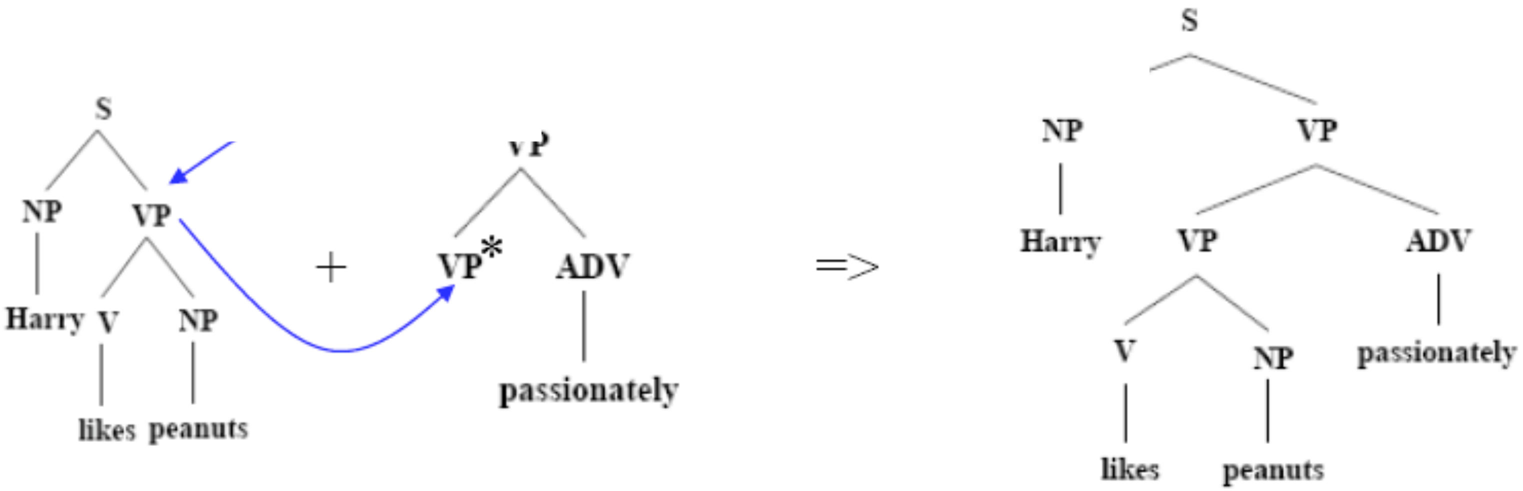
Tree-Adjoining Grammar (Joshi)

- Elementary structures are (lexicalised) trees of arbitrary height
- Rules for rewriting the nodes of trees as other trees
- Substitution (NP > NP) & Adjoining (auxiliary trees, e.g., adjuncts)

TAG (Joshi)



Substitution



Adjoining

Combinatory Categorial Grammar (Steedman)

- Categorial Grammar derives sentences in a proof-solving manner, maintaining a close link with a semantic representation
- Lexical categories specify how to combine words into sentences
- CCG has sophisticated mechanisms that deal nicely with coordination, extraction, long-distance dependencies and other constructions

CCG (Steedman)

(1) likes := $(S\backslash NP)/NP$

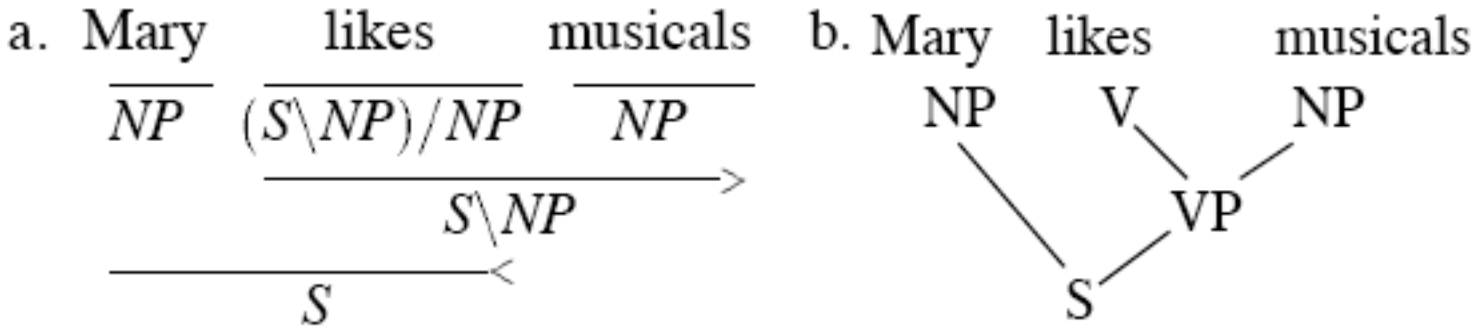
(2) *Forward Application:* $(>)$

$$X/Y \quad Y \quad \Rightarrow \quad X$$

(3) *Backward Application:* $(<)$

$$Y \quad X\backslash Y \quad \Rightarrow \quad X$$

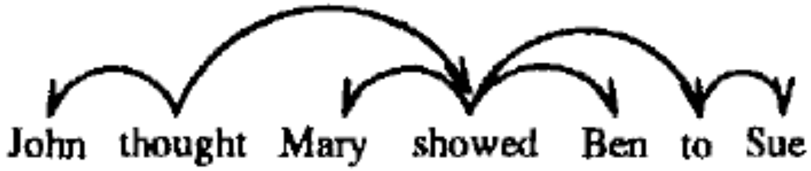
(4)



Steedman (1996), p.2

Dependency Grammar (Tesnière)

- A sentence is analysed based on the relations between words
- The verb and ist arguments(valents) drive the analysis (close relationship to the semantics of a sentence)
- No groupings (phrases, constituents)



Further Reading

- Joshi, A.K., Levy, L.S., & Takahashi, M (1975). Tree adjunct grammars. Journal of Computer and System Sciences, 10(1). [TAG]
- Borsley, R.D. (1996). Modern Phrase Structure Grammar. Cambridge, MA: Blackwell Publishers.
- Bresnan, J. (2001). Lexical-Functional Syntax. Malden (MA): Blackwell Publishers. [LFG]
- Carnie, A.. 2002. Syntax: A Generative Introduction, Oxford: Blackwell Publishing. [Minimalism (~Chomsky)]
- Chomsky, N. (1957). Syntactic Structures. The Hague: Mouton.
- Dalrymple, M. (2006). Lexical Functional Grammar. In Brown, Keith (Eds.), Encyclopedia of Language and Linguistics. Elsevier.
- Donohue, Mark (1999a). A Grammar of Tukang Besi. Berlin/New York: Mouton de Gruyter.
- Kim, J. (2000). The Grammar of Negation: A Constraint-Based Approach. Chapter 1: Introduction and Theoretical Foundations. Stanford: CSLI Publications.
- Klenk, U. (2003). Generative Syntax. Tübingen: Gunter Narr Verlag.
- Ouhalla, J. (1994). Transformational Grammar. London: Edward Arnold.
- Pollard, C. & Sag, I (1994). Head-Driven Phrase Structure Grammar. Chicago: The University of Chicago Press.
- Riehemann, S. (1995). The HPSG Formalism. Unpublished manuscript: Stanford University. <http://www-csli.stanford.edu/~sag/L221a/hand2-formal.pdf.
- Sag, I. Wasow, T. & Bender, E. (2003). Syntactic Theory. Stanford: CSLI Publications. [HPSG]
- Steedman, M. (2000). The Syntactic Process. MIT Press. [CCG]
- Steedman, M. (1996). A very short Introduction to CCG. Ms. www.inf.ed.ac.uk/teaching/courses/ics/papers/ccgintro.pdf.