# Preparatory Course: Syntax 2

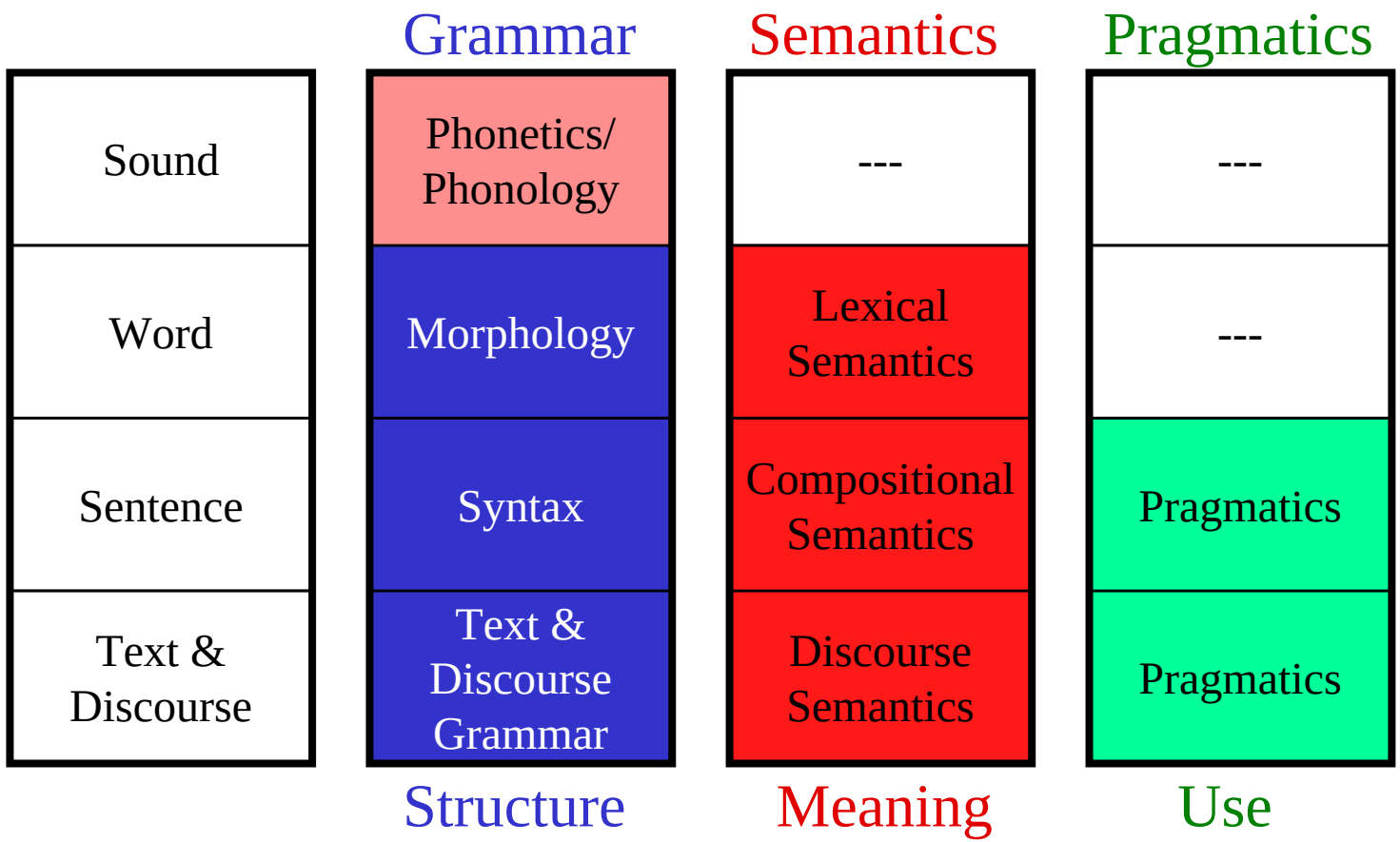
Lecture 2 (13.10.2008)

**PD Dr.Valia Kordoni**

Email: kordoni@coli.uni-sb.de

http://www.coli.uni-saarland.de/courses/msc-prep-08/

# Units of Language – Subfields of Linguistics



# Morphology and Syntax

- Morphology investigates the structure of words
- Syntax investigates the structure of sentences.

- In a way, syntax is the morphology of sentence, or, taken the other way round, morphology is the syntax of words.
- But: Sentence structure differs from word structure, in various respects.

# Observation 1: Constituents

- A simple morphological rule of German:
  - The comparative morpheme occupies the first position of the ending (= the second position of the word)
  - *schnell*+*er*+*es* [ fast+er, n, sg]
- A simple syntactic rule of English:
  - The finite verb occupies the second position of a declarative sentence
  - *John* + *gave* + *Mary* + *a* + *book*

# Constituents [1]

- Counter-examples (1)
  - Yesterday John gave Mary a book.
  - But John gave Mary a book.
- Counter-examples (2)
  - The student gave Mary a book.
  - The friendly student gave Mary a book.
  - The friendly student which I told you about yesterday gave Mary a book.

# Constituents [2]

- Counter-examples (1)
  - Yesterday John gave Mary a book.
  - But John gave Mary a book.
- Counter-examples (2)?
  - The student gave Mary a book.
  - The friendly student gave Mary a book.
  - The friendly student which I told you about yesterday gave Mary a book.
- The verb is still in second place, if we count constituents rather than words.

# Arbitrarily long and complex sentences [1]

- The mouse escaped into the garden.
- The mouse that the cat chased escaped into the garden.
- The mouse that the cat which Mary owns chased escaped into the garden.

# Arbitrarily long and complex sentences [2]

- *Er hat die Übungen gemacht.*
- *Der Student hat die Übungen gemacht.*
- *Der interessierte Student hat die Übungen gemacht.*
- *Der an computerlinguistischen Fragestellungen interessierte Student hat die Übungen gemacht.*
- *Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester hat die Übungen gemacht.*
- *Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, der im Hauptfach Informatik studiert, hat die Übungen gemacht.*
- *Der an computerlinguistischen Fragestellungen interessierte Student im ersten Semester, der im Hauptfach, für das er sich nach langer Überlegung entschieden hat, Informatik studiert, hat die Übungen gemacht.*

# Structural ambiguity

- Morphology talks about sequences of morphemes.
- To talk about syntactic regularities requires reference to constituent structure.
- Semantic interpretation of sentences also requires information about constituent structure:
  - *Pick up* | *a big red block.*
- in particular, if sentences are structurally ambiguous:
  - *John saw the man with the telescope.*

# Syntactic ambiguity

- *John saw the man with the telescope*
- *John saw the man with the telescope*
- *Young students and professors attended the party.*
- *Young students and professors attended the party.*

# Tests for constituency

- Substitution test: Word sequences that can be systematically substituted for a single word (e.g., proper name or personal pronoun) form a constituent:
- *The student gave Mary a book.*
- *The friendly student gave Mary a book.*
- *The friendly student which I told you about yesterday gave Mary a book.*
- *Mary gave John a book.*
- *Mary gave the student a book.*
- *Mary gave the friendly student which I told you about yesterday a book.*
- Compare with:
- Yesterday John gave Mary a book.
- Mary gave yesterday John a book.

# Syntactic Categories

- Constituents that are substitutable for each other can be subdivided into larger classes that share distribution and structural properties, the Syntactic Categories, e.g.:
  - Noun phrases, consisting of a pronoun, a proper name, or a complex structure with a common noun as syntactic head element – NP
  - Prepositional phrases (*with the telescope*, *into the garden*) – PP
  - Adjective phrases (*friendly*, *very friendly*, *interested in linguistics*) - AP

# Categories and Functions

- Syntactic categories denote classes of constituents with similar internal structure, in particular, the category /part-of-speech of their
- lexical head.
- Grammatical functions characterise the external role of a constituent in its syntactic context, e.g.
  - Complements: Subject, (Direct, indirect, prepositional) Object
  - Modifier / Adjunct

# Syntactic Description with CFGs

- CFG is a formalism that allows to model the concept for grammaticality for natural languages, by specifying the set of grammatically correct sentences, and assigning them their appropriate grammatical structures (in terms of their parse trees).
- Is it a realistic and reasonable aim to describe the set of grammatically correct sentences of a language?
  - What to do with ungrammatical input?
  - What does 'grammatical' mean after all? – Graded grammaticality!
- Is a CFG the appropriate formalism to describe the grammar of a language?

# Syntactic Processing with CFGs [1]

- Morphological analysers are finite-state automata (or transducers) working in linear time.
- The syntax of programming languages is recursive, and therefore described by CFGs. Because the languages typically are unambiguous, and described by deterministic CFGs, parsers for programming languages are also linear time.
- Unfortunately, grammars of natural languages are ambiguous and non-deterministic. The best algorithms (Earley Algorithm, Chart Parsing) take quadratic time to find one parse, and cubic time to find all parses.

# Syntactic Processing with CFGs [2]

- Good news: There are techniques to compile CFGs down to FSAs for many applications, without loosing much coverage (e.g., by constraining recursion depth; "finite-state technology")
- Bad news: Constituent structure is only the tip of the iceberg: More descriptive power is needed to describe syntactic structure of natural languages appropriately. Modern grammar formalisms like LFG or HPSG come in the format of typed feature structures with a context-free backbone.

# Variable Word-Order in German

*Peter hat der Dozentin das Übungsblatt heute ins Büro gebracht.*

Peter has the lecturer the exercise-sheet today into-the office brought

*Das Übungsblatt hat Peter der Dozentin heute ins Büro gebracht.*

*Der Dozentin hat Peter heute das Übungsblatt ins Büro gebracht.*

*Ins Büro hat heute Peter der Dozentin das Übungsblatt gebracht.*

*Heute hat Peter das Übungsblatt der Dozentin ins Büro gebracht.*

*Ins Büro hat das Übungsblatt der Dozentin Peter heute gebracht.*

** Ins Büro heute Peter das Übungsblatt hat gebracht der Dozentin.*

** Ins heute Büro der Peter Dozentin das hat Übungsblatt gebracht.*

# More syntactic phenomena

- Agreement
- Subcategorisation
- Long-distance Dependencies

# Computational Grammar Formalisms

Computational Grammar formalisms share several properties:

- Descriptive adequacy
- Precise encodings (implementable)
- Constrained mathematical formalism
- Monostratalism
- (Usually) high lexicalism

# Descriptive Adequacy

Some researchers try to explain the underlying mechanisms, but we are most concerned with being able to *describe* linguistic phenomena

- Provide a structural description for every well-formed sentence
- Gives us an accurate encoding of a language
- Gives us broad-coverage, i.e., can (try to) describe all of a language

  → No notion of core and periphery phenomena

# Precise Encodings

**Mathematical Formalism**: formal way to generate sets of strings

Precisely define:

- elementary structures
- ways of combining those structures

=> Such an emphasis on mathematical precision makes these grammar formalisms more easily implementable

# Constrained Mathematical Formalism

A formalism must be **constrained**, i.e., it cannot be allowed to specify all strings

- Linguistic motivation: limits the scope of the theory of grammar
- Computational motivation: allows us to define efficient processing models

# Monostratal Frameworks

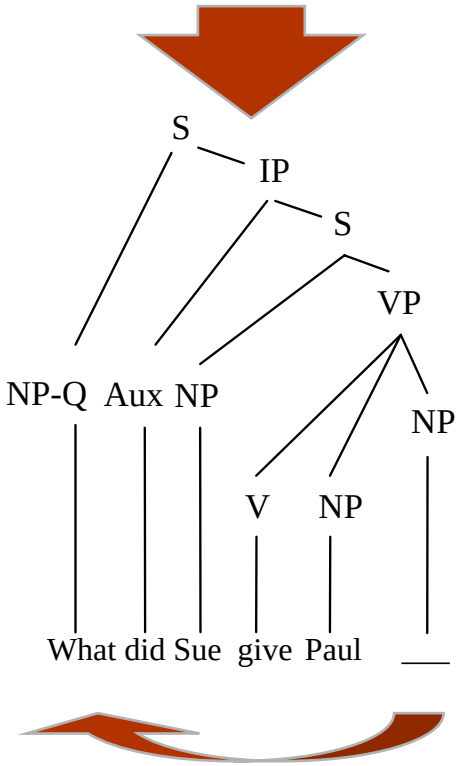
Only have one (surface) syntactic level

- Make no recourse to movement
- Augment your basic (phrase structure) tree with information that can describe „movement“ phenomena

=> Without having to refer to movement, easier to process sentences on a computer

# This should be avoided!

## Sue gave Paul an old penny



# Lexical

In the past, rules applied to broad classes and only some information was put in the lexicon, e.g., subcategorisation information

- Linguistic motivation: lexicon is the best way to specify some generalisations: *He told/*divulged me the truth*
- Computational motivation: can derive lexical information from corpora (large computer-readable texts)

=> Shift more of the information to the lexicon; each lexical item may be a complex object

# Context-Free Grammars (CFGs)

Context-Free Grammars (CFGs) are one kind of constrained mathematical formalism, a precise way of encoding syntactic rules:

- elementary structures: rules composed of non-terminal and terminal elements
- combine rules by rewriting them

# Context-Free Rules

Example of a set of rules:

- S → NP VP
- NP → Det N
- VP → V NP
- ...

But these rules are rather impoverished.

# Are CFGs good enough?

- Data from various languages show that CFGs are not powerful enough to handle all natural language constructions
- CFGs are not easily lexicalised
- CFGs become complicated once we start taking into account agreement features, verb subcategorisations, unbounded dependency constructions, raising constructions, etc.

We need more refined formalisms...

# Beyond CFGs

Move beyond CFGs, but stay „mathematical“:

- Extend the basic model of CFGs with, for instance, complex categories, functional structure, feature structures, ...
- Eliminate CFG model (or derive it some other way)

# Computational Grammar Frameworks

- Dependency Grammar (DG)
- Tree-Adjoining Grammar (TAG)
- Combinatory Categorial Grammar (CCG)
- Lexical Functional Grammar (LFG)
- Head-Driven Phrase Structure Grammar (HPSG)

# Dependency Grammar (DG)

- The way to analyse a sentence is by looking at the relations between words
- A verb and its valents/arguments drive an analysis, which is closely related to the semantics of a sentence
- No grouping, or constituency, is used

# Tree-Adjoining Grammar (TAG)

- Elementary structures are trees of arbitrary height
- Trees are rooted in lexical items, i.e., lexicalised
- Put trees together by substituting and adjoining them, resulting in a final tree which looks like a CFG-derived tree

# Combinatory Categorial Grammar (CCG)

- 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, and other constructions

# Lexical Functional Grammar (LFG)

- Functional structure (subject, object, etc.) divided from constituent structure (tree structure)
  - kind of like combining dependency structure with phrase structure
- Can express some generalisations in f-structure; some in c-structure; i.e., not restricted to saying everything in terms of trees

# Head-driven Phrase Structure Grammar (HPSG)

- Sentences, phrases, and words all uniformly treated as linguistic signs, i.e., complex objects of features
- Similar to LFG in its use of feature architecture
- Uses an inheritance hierarchy to relate different – types of objects (e.g., nouns and determiners are both types of nominal)