# Dependency Grammars

## Syntactic Theory
## Winter Semester 2009/2010

Antske Fokkens

Department of Computational Linguistics
Saarland University

27 October 2009

# Outline

1. Introduction
2. Phrase Structures
3. Dependency Grammars
   - Introduction
   - Dependency relations
   - Properties of Dependencies

# Outline

1 Introduction

2 Phrase Structures

3 Dependency Grammars
- Introduction
- Dependency relations
- Properties of Dependencies

# Overview of this lecture

In the next three lectures, we will discuss Dependency Grammars:

- Dependencies and Phrase Structures:
  - basic objectives of syntactic analysis
  - properties of phrase structure grammars
- Basic definitions of Dependencies
  - What are dependencies?
  - Example analyses
- Differences and Relations between Dependencies and Phrase Structures
- Syntactic Theory/CL and Dependencies
  - Meaning to Text Theory
  - Prague Dependency Treebank

# Syntactic Analysis

- Syntax investigates the structure of expressions
- Some reasons for performing syntactic analysis:
  - To understand something about how language works
  - To analyze language: how can we relate speech/written text to meaning?

# Focus of syntactic analysis

There are two properties of syntactic structure that a syntactic analysis may be directed to:

1. Composition of expressions:
   - Hierarchical analysis of the subparts of an expression (top down)
   - Main parts of interest: **phrases** and **constituents**, i.e. groupings of words
2. The relations between elements of the expressions
   - Hierarchical analysis of the head (main item) in the expression and items that depend on it
   - Main part of interest: **syntactic functions** or **relations**, i.e. how do words in the expression relate to each other

# Phrase Structures vs Dependencies

- **Phrase Structure** Analysis refers to the analysis that divides expressions into its subparts. It looks at how an expression is built up
- **Dependency** Analysis refers to the analysis that looks at the relations that are found in an expression. It looks at what role parts of the expression play to convey the meaning
- Every Syntactic Theory contains Phrase Structure Analyses or Dependency Analyses, and possibly both

# Outline

1 Introduction

2 Phrase Structures

3 Dependency Grammars
- Introduction
- Dependency relations
- Properties of Dependencies

# Phrase Structure Analysis

- Words may be combined into **phrases**
- A phrase (or word) may occur as a **constituent** of an expression
- Constituents occur as parts of phrases
- E.g. in *The child chases the cat*:

  *the* and *cat* form a noun phrase,
  the noun phrase *the cat* occurs as a constituent of *chases the cat*
  *chases the cat* is a verb phrase

# Compositional Syntactic Analysis

- How to determine the composition of an expression?
  → identify its constituents
- Constituents can be identified by the following tests:
  - Substitution
  - Movement
  - Questions
  - Coordination (not always reliable)

# Syntactic Categories (1/2)

Phrases and words often share syntactic properties:

- They may have the same **syntactic distribution**, i.e. they may occur in the same positions to form well-formed expressions
- They may have the same **morphological** properties, e.g. English verbs bear tense, nouns do not; Latin nouns bear case, verbs do not.

When phrases (or words) share such properties, they generally are of the same **syntactic category**

# Syntactic Categories (2/2)

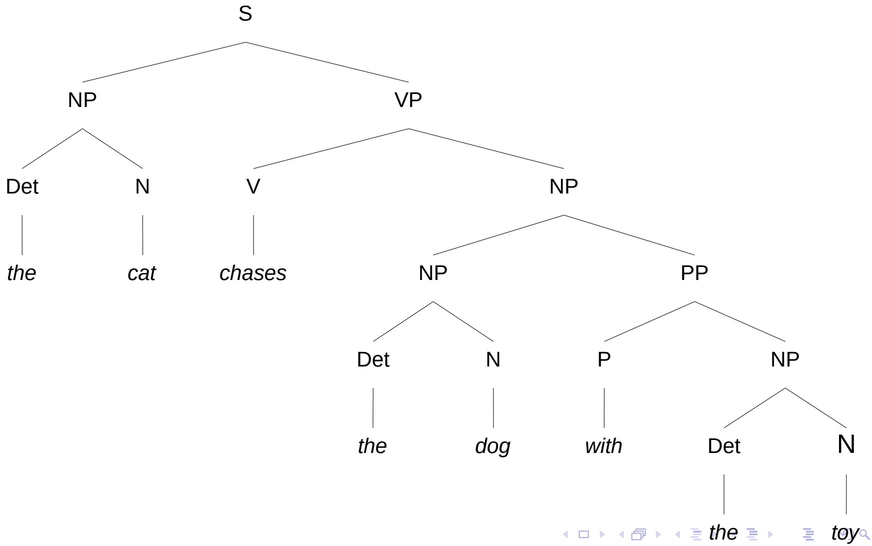
- We distinguish two kinds of syntactic categories:
  - Phrases belong to a **phrasal category**
  - Words belong to a **lexical category**

| | **The** | **cat** | **chased** | **the** | **dog** | **with** | **the** | **toy** |
|---|---|---|---|---|---|---|---|---|
| *lex-cat*: | det | N | V | det | N | P | det | N |
| *phr-cat*: | NP | | | NP | | | NP | |
| *phr-cat*: | | | | | | PP | | |
| *phr-cat*: | | | | NP | | | | |
| *phr-cat*: | | | VP | | | | | |
| *phr-cat*: | S | | | | | | | |

# Heads in Phrase Structure Grammar

- Phrases are **headed**
- The **head** of a phrase is a designated item (within the phrase) that determines the syntactic properties of the entire phrase
- Typically we refer to a lexical item when we talk about the head of a phrase, but it may be also be one of the phrases immediate constituents.
  e.g. in *John likes Mary* the head of S may be VP (as immediate “head” constituent of S) or *likes* (the lexical head of the sentence)
- Generally, the category of the head percolates up to its mother (N is head of NP, V is head of VP)

# Typical Representation: A PS-Tree



# Some Basic Syntactic Categories for English

- Lexical Categories:
  - A: Adjective
  - Adv: Adverb
  - Det: Determiner
  - N: Noun
  - P: Preposition
  - V: Verb
- Phrasal Categories:
  - AP: Adjectival Phrase
  - AdvP: Adverbial Phrase
  - NP: Noun Phrase
  - PP: Prepositional Phrase
  - VP: Verb Phrase

# Phrase Structure Rules (PSR)

Phrase Structure Rules describe how phrases are composed:

e.g. NP → Det N
S → NP VP

- These rules say:
  - A noun phrase may consist of a determiner and a noun
    A sentence may consist of a noun phrase and a verb phrase
  - A determiner and a noun can form a noun phrase
    A verb phrase and a noun phrase can form a sentence
- Phrase Structure Rules determine linear order:
  - The determiner must precede the noun in a noun phrase
  - The noun phrase must precede the verb phrase in a sentence

# Properties of Phrase Structure Rules

- **generativity:** PSR describe which strings may be formed
- **recursivity:** Rules may be reapplied:

  NP → NP PP
  PP → P NP

- PRS may represent the hierarchical structure of expressions: phrases have an internal structure
- They can reveal structural ambiguity when applied to analyze a string, e.g. PP attachment:
  [I [[saw [the man]] [with the telescope]]] (VP → VP PP)
  [I [saw [the man [with the telescope]]]] (NP → NP PP)

# Context Free Grammar

- The Phrase Structure Rules on the previous slides are Context Free Rules.
- Context Free Grammars (consisting of a collection of context free rules) are used as basic methods to describe phrase structures
- In the Chomskyan tradition, CFGs are also called Phrase Structure Grammars
- But (as we will see later in the course) a grammar that is called 'Phrase Structure Grammar' is not necessarily a CFG...

# Formal Definition of CFG

A context-free grammar *G* is a quadruple:

G = ($V_{NT}$, $V_T$, P, S) where

- $V_{NT}$: a set of non-terminal symbols (phrases)
- $V_T$: a set of terminal symbols (lexical items)
- P: a set of production rules of the form A $\rightarrow \alpha$, where A is a terminal symbol and $\alpha$ is a sequence of terminal and non-terminal symbols.
- S: a designated start symbol, S $\in V_{NT}$

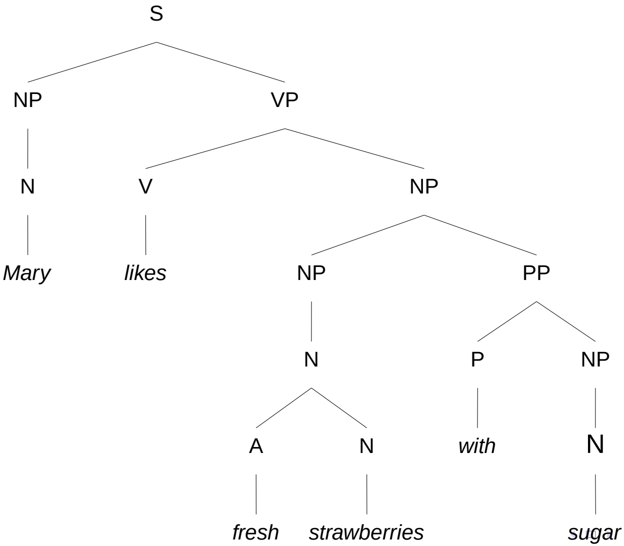
Note that A on the left-hand side of a production rule must be exactly one symbol: this determines the **context-free** character of the grammar: A $\rightarrow \alpha$ regardless of the context of A (i.e. not possible to say that the rule applies given A precedes/follows B).

# Simple Example Grammar for English

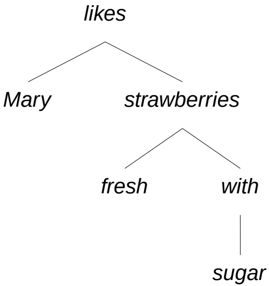
S → NP VP
NP → Det A* N PP*
VP → V (NP) (NP|PP)
VP → VP PP
PP → P NP
X → $X^+$ CONJ X, where X = NP,VP,PP,N, or V

- Exercise:
  1. Make up a small lexical (containing at least one item of each lexical category)
  2. Look at what sentences your grammar and lexicon produce:
     - What does it capture?
     - What goes wrong (subcategorization, agreement...)?

# Example PS-Tree



# Another Kind of Tree



# Outline

1 Introduction

2 Phrase Structures

3 Dependency Grammars

- Introduction
- Dependency relations
- Properties of Dependencies

# Dependency Grammars

- Dependency Grammars: collection name of grammars that address syntactic relations in a sentence
- There are many dependency grammars, using different frameworks, based on different theories
- We will focus on representations that correspond (more or less) to Mel’čuk’s ’surface syntactic dependencies’ (Mel’čuk 1988).
  This level of dependency is typically meant when people talk about dependency grammars

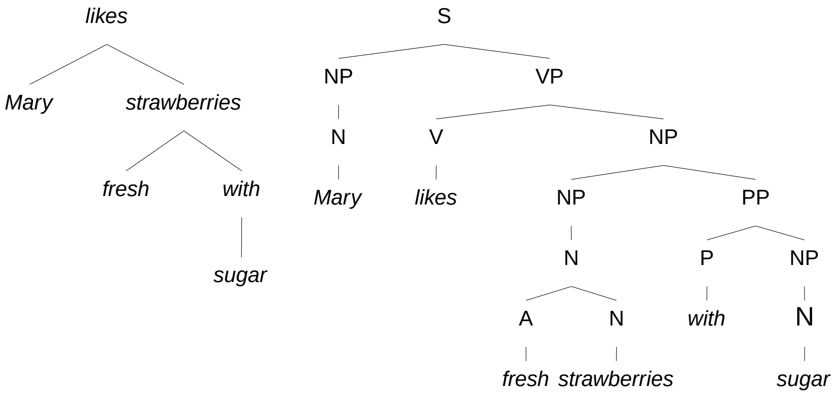
# Why dependencies?

- Syntax studies how words are combined to form complex expressions
- The idea is to understand how these expressions come to convey their meaning
- Consider the following sentence:

| | | | | |
|---|---|---|---|---|
| *ratio* | *hominem* | *hominum* | *appetentem* | *fecit* |
| rationality-NOM | man-ACC | man-GEN | longing-for-ACC | make-3S |

- How can we interpret this sentence?
- Does Phrase-Structure tell us?

# A Dependency Tree versus a PS Tree



*Mary likes fresh strawberries with sugar*

# Dependency relations

- Dependency relations aim at revealing the syntactic structure of a sentence
- The syntactic structure of a sentence can be expressed by:
  - Word Order
  - Prosody
  - Inflection
  - Lexical items
- Mel'čuk: the means that are used to express syntactic structure cannot be part of the structure itself (i.e. no phrase structure in syntactic representation)

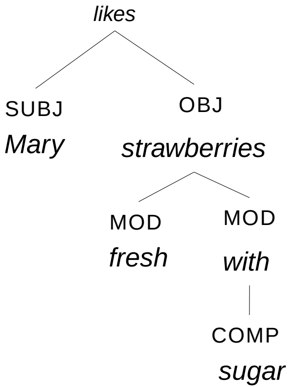
# Dependency relations

A or A → B or B ← A

B

- There is a **dependency relation** between A and B
- In the representations above, we say that:
  - A is a **head**, B its **dependent**
  - A **governs** B
  - B **is dependent on** A
- We are not only interested in the direction of the relation that exists between A and B

# Labeled Dependencies



We'll look into these relations in more detail later

# Properties of dependency relations (1/2)

Dependency relations must be:

- **antisymmetric**:
  - if A $\rightarrow$ B, then B $\nrightarrow$ A
  - If A governs B, then B does not govern A
    E.g. noun-compounds:
    - *student politics* $\neq$ *politics student*
    - *politics* $\rightarrow$ *student* $\neq$ *student* $\rightarrow$ *politics*
- **antireflexive**:
  - If A $\rightarrow$ B, then A $\neq$ B
  - No item governs itself
    - No wordform is linearly ordered or inflected with respect to itself
    - This property also follows from the antisymmetric property of dependency relations

Based on Mel'čuk (1988: p. 21-22)

# Properties of dependency relations (2/2)

- **antitransitive**:
  - if A $\rightarrow$ B, and B $\rightarrow$ C, then A $\nrightarrow$ C
  - If A governs B, and B governs C, then A does not govern C
  - Dependency relations are always *direct*
  - e.g. a *very smart student*:
    *student* $\rightarrow$ *smart* $\rightarrow$ *very*
    *student* $\nrightarrow$ *very*
- **labeled**:
  - In *John loves Mary*:*love* $\rightarrow$ *John*, *love* $\rightarrow$ *Mary*:
    *love* $\rightarrow_{SUBJ}$ *John*, *love* $\rightarrow_{OBJ}$ *Mary*
  - Russian:
    - *žena-vrač*: ŽENA $\rightarrow_{r1}$ VRAČ (wife who is a doctor)
    - *žena-vrača*: ŽENA $\rightarrow_{r2}$ VRAČ (doctor's wife)
  - In CL, unlabeled dependencies are found as well

Based on Mel'čuk (1988: p. 21-22)

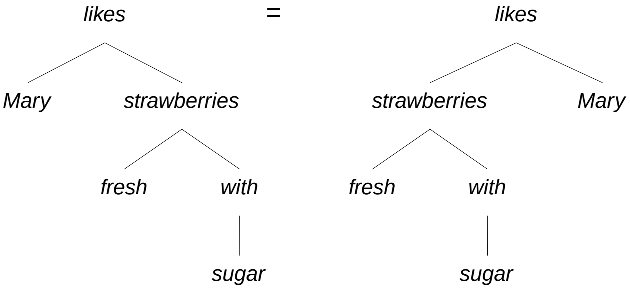
# Dependency trees

- A dependency tree is a **connected directed labeled graph**
  - Its **nodes** or **vertices** are labeled with reduced word forms (no inflection)
  - Its **branches** or **arcs** are labeled with the names of syntactic relations
- It must have exactly one node that does not depend on any other node: the **root** or **top node** of the tree
- In Mel'čuk's approach (and many others), no syntactic node may simultaneously depend on two or more nodes

Based on Mel'čuk

# Dependency trees

Linear order is not represented in dependency trees:



# Dependency Analysis

To determine the syntactic structure of a sentence, we need to determine 3 things:

1. Which items stand in a direct relation
2. When there is a dependency between A and B which is the dependent (direction of the relation)?
3. What is the relation between A and B (the label)?

In most cases, the easiest way to find the syntactic dependencies of an expression is by identifying the **root**

# Identifying Dependency relations

How to identify dependency relations?

- Syntax has means to express syntactic dependency relations:
  - Phrase structure and word order (the subject in English (NP left of the VP))
  - Morphology (the subject in Latin or German (nominative case))
- Syntactic properties of lexical items: a verb's arguments are its dependents in a syntactic structure

# Identifying the direction of the dependency

- Identifying the direction of a dependency is not always a clear-cut issue
- The following guidelines have been proposed to identify heads (Zwicky 1985; Schneider 1998; Hudson 1990)
  - A subcategorizes for B (*John runs*: *runs* subcategorizes for a subject)
  - A determines concord/agreement with some other element (*John runs*, *John has run*)

    ASF: Careful! This may conflict with the subcategorization guideline, people's opinions differ here
  - A carries the inflection (red books, not ∗reds book)
  - A belongs to a category which has the same distribution as A + B (I like red books/John/books)
  - A is obligatory
  - A + B is a hyponym of A (red book is a hyponym of book)

Based on Kordoni (2008b)

# Dependency relations

- What labels do we need for our dependencies?
  → what distinctions need to be made?
- A fundamental distinction: **arguments** that are selected for by their head versus **adjuncts** that some something additional about the head
- A head **sub-categorizes** for its **arguments**: their presence is often (but not always!) obligatory
- An **adjunct** is per definition optional

# Argument or Adjunct?

- Lexical items come with a subcategorization frame: a list of the arguments it takes
- Because most arguments are required, they can generally not be ommitted. Recall:
  1. I put the book on the table
  2. *I put the book
- However, some items have optional arguments, or an ambiguous sub-categorization frame. Recall:
  1. Kim told Sam (a lie)
  2. Mary ate (strawberries)
- There are tests that show whether an element is an argument or an adjunct. One of them is the ’anaphora test’

# Anaphora test (argument or adjunct)

Recall:

- The anaphor *do so* can be used to contrast between adverbials (adjuncts), but not between arguments:
  1. Kim went to the movies on Thursday, and Sam did so on Friday.
  2. *Kim put the book on the table, and Sam did so on the chair.
  3. *Sam told Kim a lie, and Bill did so the truth.
- Similarly, the anaphor *one* in NPs cannot be used to contrast between functional arguments:
  1. John met the King from England, and I the one from France.
  2. *John met the King of England, and I the one of France.

# Argument relations

- Arguments are dependents of the lexical item that subcategorizes for them
- How do we distinguish between arguments when there is more than 1?
  → there are different argument relations
- The exact labeling may differ from one framework (or even linguist) to another. Some relation names that you may find:
  - ARG$_1$, ARG$_2$, ARG$_3$,...,ARG$_N$
  - *predicative*, *1st completive*, *2nd completive*,..
  - *subject*, *object*, *indirect object*
- We will use *subject, object* and *indirect object* in this lecture

# Bibliography I

- Haegeman, Liliane (1991). *Introduction to Government and Binding Theory*. Oxford, UK: Blackwell Publishers.
- Hudson, Richard A. (1990). *English Word Grammar*. Oxford, UK: Blackwell. http://www.phon.ucl.ac.uk/home/dick/ewg.htm.
- Kahane, Sylvain (2003). The Meaning-Text Theory. *Dependency and Valency. Handbooks of Linguistics and Communication Sciences 25 1-2*. Berlin, Germany: De Gruyter.
- Kordoni, Valia (2008a). Syntactic Theory Lectures 1 and 2. Course slides.
- Kordoni, Valia (2008b). Syntactic Theory Lectures 3 and 4. Course slides.
- Mel'čuk, Igor (1988) *Dependency Syntax: Theory and Practice*. Albany, USA: State University of New York Press.
- Ouhalla, Jamal (1994). *Introducing Transformational Grammar*. New York, USA: Oxford University Press.

# Bibliography II

- Sag, Ivan A., Thomas Wasow and Emily M. Bender (2003). *Syntactic Theory*. A Formal Introduction. Palo Alto, USA: CSLI Publications.
- Schneider, Gerold (1998). *A Linguistic Comparison of Constituency, Dependency and Link Grammar. Lizentiatsarbeit, Institut für Informatik der Universität Zürich*. http://www.ifi.unizh.ch/cl/study/lizarbeiten/lizgerold.pdf.
- Zwicky, Arnold (1985). Heads. *Journal of Linguistics* 12, 1-30.