Dependency Grammars
Lecture 3

Syntactic Theory
Winter Semester 2009/2010

Antske Fokkens

Department of Computational Linguistics
Saarland University

3 November 2009

Outline

Outline

Overview of lecture on Dependency Grammars

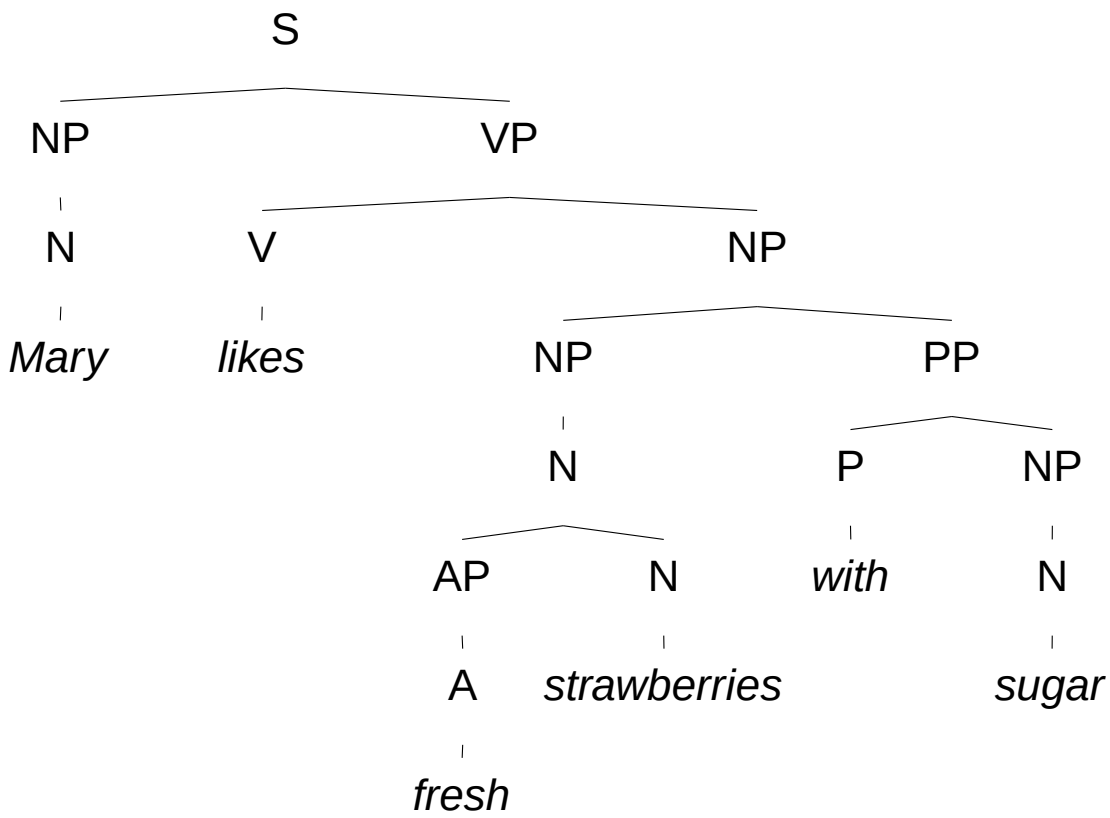
- Dependencies and Phrase Structures:
  - basic objectives of syntactic analysis
  - properties of phrase structure grammars
- Basic definitions of Dependencies
  - What are dependencies?
  - Example analyses
- Differences and Relations between Dependencies and Phrase Structures
- Syntactic Theory/CL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

Syntactic relations in Phrase Structures

- Phrase Structures focus on the composition of phrases into chunks, on how words group together to form phrases
- Phrase structure is what syntactic analysis is mainly about in these approaches, but syntactic relations are implicitly present in PS-trees
- When the head of the phrase is well-defined, and the tree distinguishes between arguments and adjuncts, dependency structures can be derived from the PS-tree

From Phrase Structures to Dependencies

E.g. :



Converting a PS-tree to dependencies

Steps to take:

1. Add grammatical relations (based on definitions on the structure) to mother-daughter connections in tree
2. Start at the root of the tree
3. Identify lexical head of the phrase
4. Percolate the lexical head up to its maximal projection
5. Remove redundant nodes from the tree
6. Repeat steps 3-5 for all maximal projections in the tree

Going from Dependencies to Phrase Structure

- Dependencies can be derived from phrase structures, because phrases consist of a head and its dependents (if it has any)
- Similarly, you can derive phrase structures from dependencies by grouping heads and their dependents together
- Just like we needed definitions on structures to derive the labels for our dependencies, some additional information is necessary to derive a well-formed PS-tree

From Dependencies to Phrases

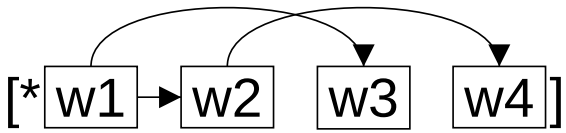
- To derive a PS-tree from a dependency representation it is necessary to define
  1. how constituents of a phrase are ordered relative to each other (if linear order is not registered somehow in the dependency representation)
  2. how to map relations to the correct $\overline{\text{X}}$-level formation

Projectivity or Adjacency

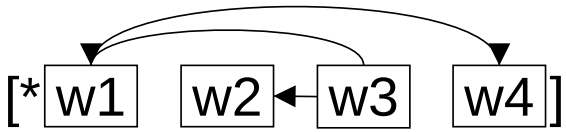
- Both Mel'čuk (1988) and Hudson (2007) mention the tendency of words to form continuous phrases as an important property of language
- It seems to hold cross-linguistically; there are exceptions in most languages, but they generally concern 'marked' structures (except maybe Dutch and Swiss German)
- According to Mel'čuk (1988) this observation was first made by Hays and Lecref (around 1960), but note that it was already (implicitly) used in transformational syntax
- In Dependency Grammars this property of word order is captured by the **Projectivity** or the **Adjacency** principle.

Projectivity/Adjacency (1)

- A sentence is projective if and only if among the arcs of dependency linking its wordforms:
  (i) No arc crosses another arc:



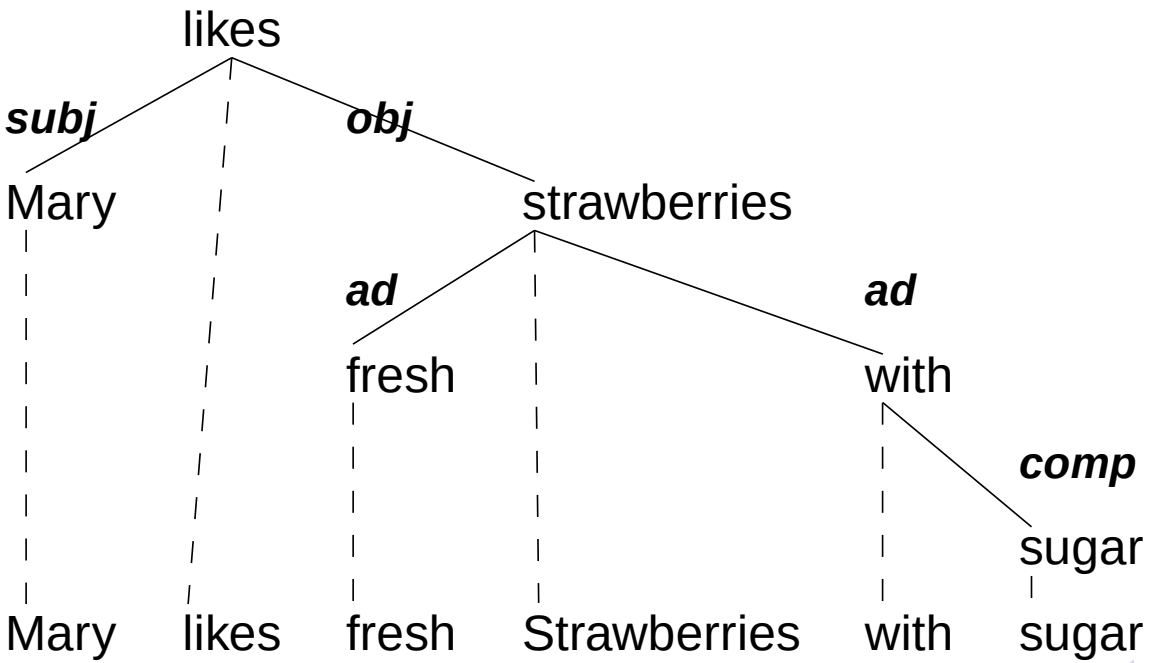
(ii) No arc crosses the top node:



Mel'čuk (1988; p.35-36)

Projectivity/Adjacency (2)

- A sentence is projective if and only if we can draw a dependency tree from which each node can be connected by a vertical line to its corresponding form in the surface string without crossing another line



Projectivity as principle

- Word Grammar assumes **strict projectivity** (Hudson 2003)
- In other words: all well-formed expressions must be projected
- Word Grammar must thus find a way to deal with discontinuous phrases

Outline

Meaning-Text Theory (MTT)

- Put forward in Moscow by Žolkovskij and Mel'čuk (1965, 1967) as a model for machine translation
- Its objective is to reveal explicit rules that express the correspondence between meaning and text
- Meaning-Text Theory is meant to be a model of linguistic knowledge, and not a cognitive model of language usage
- Though much ignored in main-stream linguistics in Western Europe and the US, MTT has been highly influential in linguistics in Eastern European school and computational linguistics, where the popularity of dependency approaches is increasing

We will follow Kahane (2003) in our presentation of MTT

Some characteristics of MTT

Kahane (2003) mentions the following characteristics of MTT

- Focus on dependencies rather than constituents
- Highly lexicalized ('massive relocation of syntactic information into the lexicon')

Language, Meaning and Text (Postulate 1)

A **natural language** *L* is a 'logical device that establishes the correspondence between the set of possible **meanings** of *L* and the set of possible **texts** of *L*.' (Kahane 2003)

- **Meanings**: distinguishable entities that form an infinite countable set, formalized by **semantic representations**.
  Meaning is invariant of synonymic transformations and only refers to information that is conveyed by language (Mel'čuk 1988)
- **Texts**: distinguishable entities that form an infinite countable set, formalized by **phonetic representations**.
  Text is the physical form of any utterance
- A description of correspondence between semantic and phonetic representation is equivalent to describing all acceptable sentences of the language

Meaning to text Model (Postulate 2)

a language *L* is 'described by a **Meaning-Text-Model** (MTM)

- an MTM is a symbolic model
- it includes a finite set of rules defining correspondence between the set of meanings of *L* and the set of texts of *L*

Some remarks on MTMs (1/2)

- a Meaning-Text Model is in principle bidirectional, but it is developed in a *synthesis direction* only (i.e. from meaning to text)
  - language production is a more linguistic task than interpretation, where extra-linguistic factors such as context play a role
  - Grammar restrictions (e.g. (**do*)/*make a decision* vs.(**make*)/*do someone a favor*) need to be accounted for in production, but are uninteresting for interpretation

Some Remarks on MTMs (2/2)

- Correspondence between Meaning and Text is many to many
  - **synonymy** leads to many possible ways to express a sentence
  - **ambiguity** leads to more than one interpretation for a given expression

Layers of linguistic representation (Postulate 3)

There are two intermediate representation levels between the semantic representation and the phonetic representation: a syntactic representation and a morphological representation

- Each level of representation (except for the semantic representation) is divided in a deep - and a surface level
- The fact that syntactic and morphological levels are seen as **intermediate** levels between semantics and phonetics is particular to MTT
- This makes the correspondences between each level completely modular, i.e. you can change mapping from semantics to syntax, without effecting mapping from syntax to morphology

Overview of Meaning-Text Theory

Semantic representation (or the meaning)

⇕ semantics

Deep-syntactic representation

⇕ deep syntax

Surface-syntactic representation

⇕ surface syntax

Deep-morphological representation

⇕ deep morphology

Surface-morphological representation

⇕ surface morphology

Deep-phonological representation

⇕ phonology

Surface-phonological representation

Based on Kahane (2003, p3)

Semantic Representation (1/2)

- A semantic representation present the meaning of a set of synonymous expressions
  $\rightarrow$The concept of 'meaning' is based on the concept of 'same meaning'
- The precision of 'synonymy' may be domain dependent (e.g. law text versus journal text)
- It represents the meaning of a sentence, but also the 'dictionary meaning' definitions of 'semantemes' (units of semantic analysis)

Kahane (2003, p.4-5)

Semantic Representation (2/2)

- There is no analysis of meaning in the direction of truth-conditions, absurdity of expressions (since it does not seem necessary for translation)
- The semantic representation contains what the speaker intends to say
  - Choices during the synthesis process may change the original meaning (lexical items have their own semantic nuances)
  - Inflectional meaning (e.g. tense) may be represented, but in a descriptional way (e.g. 'at any time', 'in the future')

Kahane (2003, p.4-8)

Structure of Semantic representation

- A semantic representation is a connected directed graph. The nodes represent **semantemes**: meaning units similar to dictionary entries
- **Semantemes** are functors: they introduce arguments which are **semantic actants**. A **semantic name** is a semanteme without arguments
- The semantic representation's branches represent **semantic dependencies** between a **semanteme** and its **semantic actant**
- Each dependency is labeled with a number *i*
  $\longrightarrow$ each dependency is distinct, but not semantic

A Semantic Representation

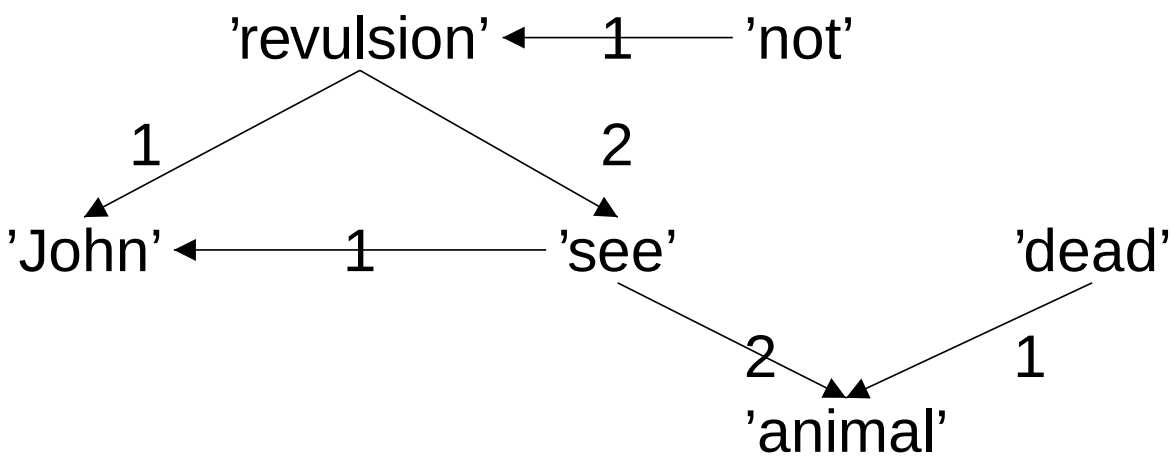
Consider the following sentences:

- John feels no revulsion at the sight of a dead animal
- John does not feel revulsion in seeing a dead animal
- John experiences no revulsion at the sight of a dead animal
- John experiences no revulsion when he sees some dead animal
- John is not revolted by the sight of a dead animal

Kahane (2003, p.5)

A semantic representation

The following tree represents the meaning of the sentences on the previous slides:



Adapted from Kahane (2003, p.6)

The direction of semantic dependencies

- Semantically, when a semanteme ’A’ expresses a property of semanteme ’B’, semanteme ’B’ is a semantic dependent of ’A’
- E.g. the trees of expressions such as *small river*, *smart student*, *the river swelled*, *the student passed*
- small smart swell pass

Adjuncts (modifiers) versus arguments (actants)(1/2)

- Semantically, an adjective governs the noun it modifies (it expresses a property of the noun)
- Syntactically, it depends on the noun (it is optional, the noun tends to bear the inflection)
- This is a typical property of adjuncts or 'modifiers'
- Arguments that are subcategorized for by their head, depend on the head in both semantic and syntactic representations

Adjuncts versus Arguments (2/2)

- When the semantic dependency has the same direction as the syntactic dependency, B is an *actant* (= argument, ASF) of A
- When the syntactic and semantic dependencies have opposite direction, B is a *modifier* (= adjunct, ASF) of A

Kahane (2003, p.6)

Other aspects of semantic representation

There is more to say about the semantic representation in MTT:

- It may represent *information structure*
- Semantic representation can be encoded in a more logical style:
  x: 'John' $\wedge$ y: 'animal' $\wedge$ p : 'dead'(y) $\wedge$: e: 'see'(x,y) $\wedge$ w: 'revulsion'(x,e) $\wedge$ q: 'not'(w)
- These representations differ from representations in Fregean logic because variables refer to the *meaning* of words
- These properties are outside of scope for this class, but if you are interested, read Kahane (2003), or Mel'čuk (1988)

Deep syntactic representation

- The deep syntactic representation specifies words and the syntactic relations between them
- Coming from the Semantic Representation, the deep syntactic representation more or less adds the lexical choices that are made to express a certain meaning
- Coming from the Surface Syntactic Representation, it represents **generalized lexemes** (excluding words and inflection that are required by the grammar of the language)

Deep syntactic structure

- The deep syntactic structure is a dependency tree in which nodes are **generalized lexemes**
- Typically, these are **semantically full lexemes**
- A lexeme may be accompanied by a **semantic grammeme**, i.e. an inflectional element that has a meaning (e.g. number, definiteness or natural gender for nouns, tense and aspect for verbs)
- Functional words or inflection marking that is required by the grammar are not part of the deep syntactic structure (e.g. selected prepositions, auxiliaries, agreement and case marking)

Generalized Lexemes

A generalized lexeme may be

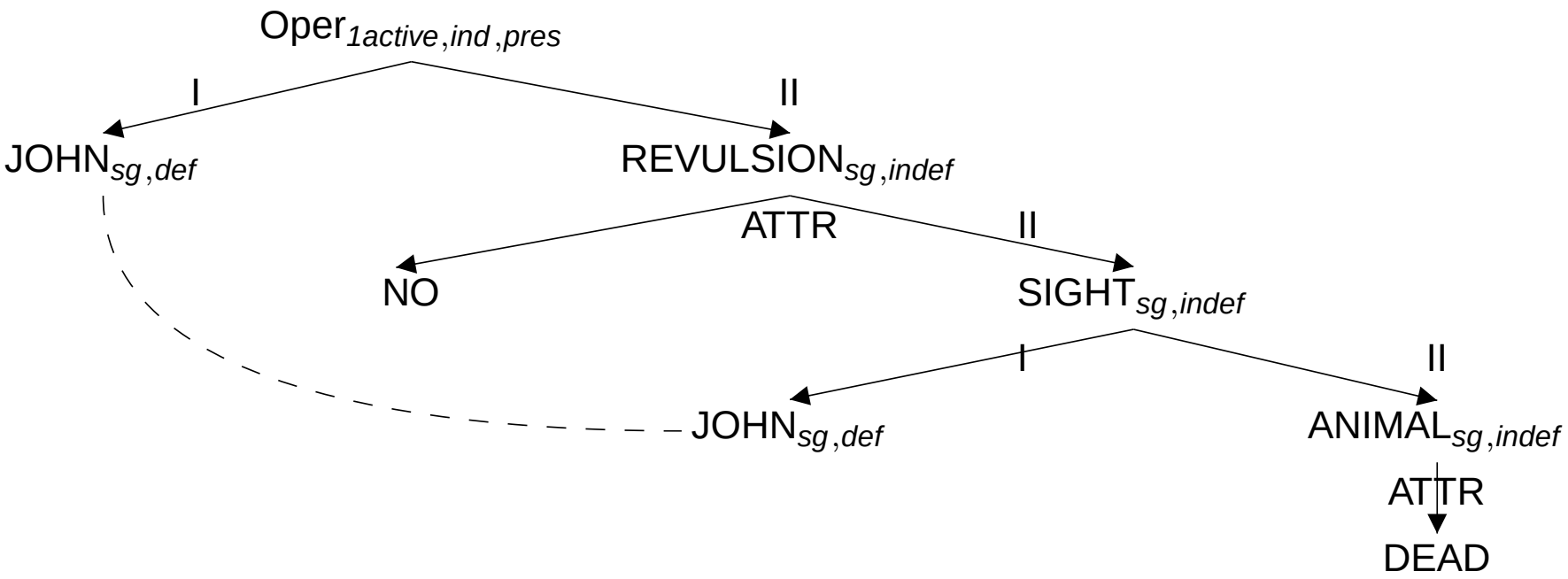
- A **full lexeme**
- A **fictitious lexeme** may represent a meaning that is expressed by a syntactic structure
  - E.g. Russian: *sto metrov* (100m) vs *metrov sto* (approx. 100m)
- A **phraseme** is a group of words that semantically forms a whole (e.g. *pull someone's leg*)
- A **lexical function** is a function that allows to describe collocations
  - E.g. **Magn** = 'very'
    Magn(*belief*) = staunch
    Magn($work_V$) = as a Trojan, one's guts out
- Each **lexeme** may be accompanied by a semantic **grammeme**

Branches in a Deep syntactic structure

- The branches of a deep syntactic structure are labeled by a deep syntactic relations coming from a small and universal set
- We distinguish:
  - **Actant** (argument) relations with syntactically meaningful labels I, II, III, IV, V, VI (where I is the subject)
  - An **attributive** (adjunct) relation (for all modifying adjuncts)
  - A **coordinative** relation (for coordination structures)
  - An **appendancy** relation (for interjections, direct addresses)

A deep syntactic structure

The deep syntactic representation of:
John feels/experiences no revulsion at the sight of a dead animal



Adapted from Kahane (2003, p.14)

The Surface Syntactic Representation

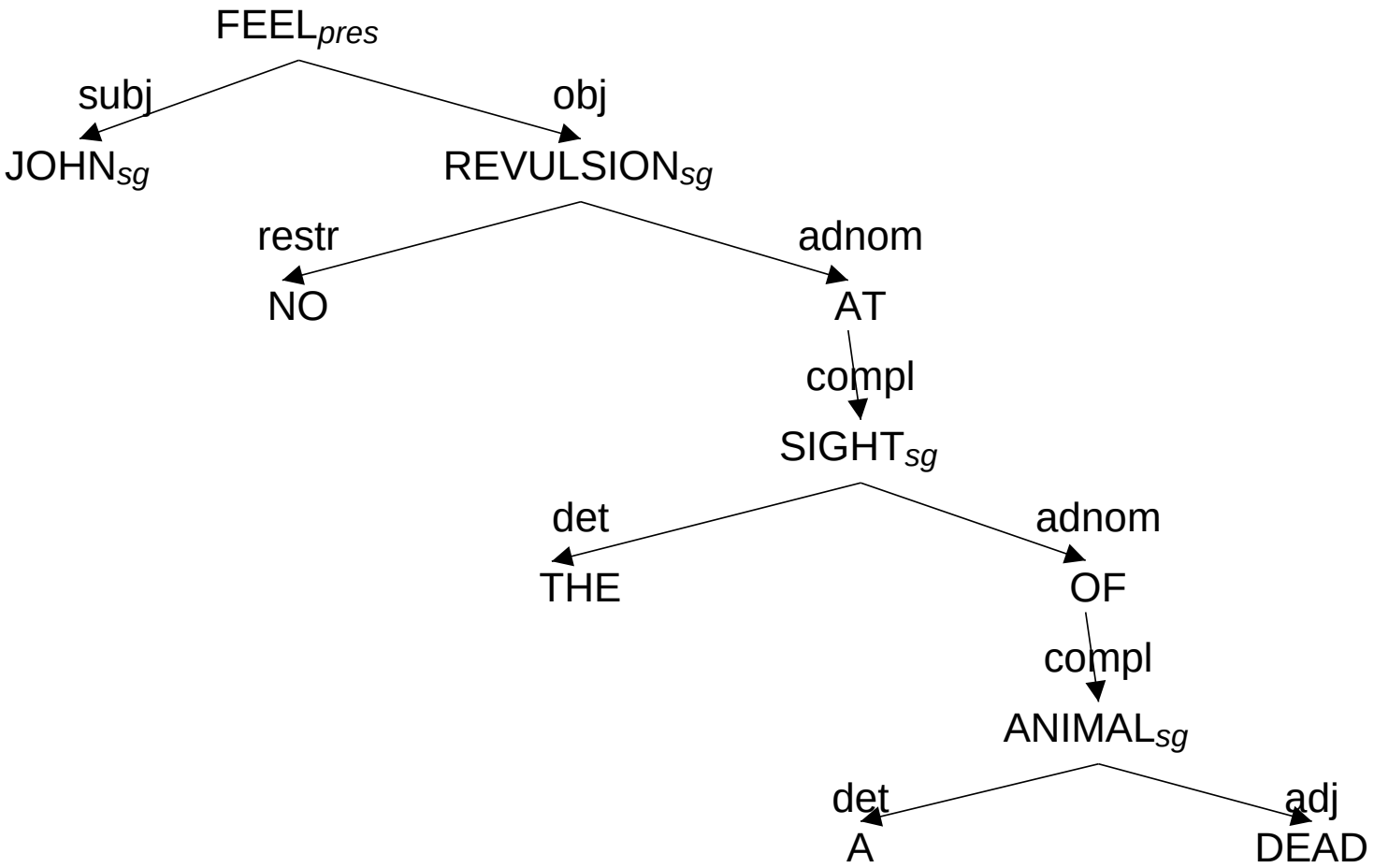
- The unordered dependency trees we have seen in previous lectures correspond (more or less) to surface syntactic representations in MTT
- Coming from the deep structure:
  - all surface lexemes and grammemes are present
  - phrasemes are expanded to surface trees
  - fictitious lexemes are replaced by syntactic relations
  - lexical functions are replaced by the lexemes that appear on the surface

Surface Syntactic Dependency trees

Some properties of syntactic dependency trees

- Even though the tree nodes are the actual lexical items of the sentence, the mapping from nodes to words on the surface is not one-to-one:
  - A language may allow to drop elements (e.g. 'to be' in present indicative form in Russian)
  - A language may have amalgamated word forms
    - E.g. French (à + la) *à la maison* versus (à + le) *au château*
- Arcs and branches of a surface syntactic tree are labeled with language specific syntactic relations

A Surface-syntactic Tree



Adapted from Kahane (2003, p.18)

Deep-morphological representations

- The deep-morphological representation specifies the word forms of a sentence in their linear order
- The morphemes of each word form are specified, but the internal structure of the words is not represented yet

  E.g. the representation of:
  John feels no revulsion at the sight of a dead animal

JOHN$_{sg}$ FEEL$_{pres,3,sg}$ NO REVULSION$_{sg}$ AT THE SIGHT$_{sg}$ OF A DEAD ANIMAL$_{sg}$

Adapted from Kahane (2003, p.17)

Different layers of representation

- MTT uses to represent different levels a Meaning-Text Model
- One of the advantages is that this allows us to capture seemingly 'double dependencies'
- E.g.
  1. Wash the dish clean (wash → clean, dish → clean)
  2. We heard Mary singing (heard → singing, Mary → singing)
  3. John was running (was → John, running → John)
- In the sentences above, there is only one syntactic dependency (wash → clean, heard → singing, and was → John); the other dependencies are of a semantic nature

The Lexicon

- MTT is a highly lexicalized system
- A lexical entry of a Meaning-Text Model has three components:
  - A **semantic** component: the lexicographic definition or 'semantic decomposition' of the item
  - A **syntactic** component: a 'government pattern' (= subcategorization frame) specifying the deep syntactic relations of each argument and its surface syntactic realization
    $\rightarrow$ here additional conditions (such as selectional restrictions) can be defined
  - A **lexical co-occurrence** component: defines lexical functions, describing which lexical items co-occur with the head word

Based on Kahane (2003; p.19)

Lexical entry REVULSION (1/3)

Semantic definition

X's revulsion for Y $\equiv$ X's (strong) negative emotion about Y similar to what people normally experience when they are in contact with something that makes them sick and such that it causes that X wants to avoid any contact with Y.

Kahane (2003, p.19)

Lexical entry REVULSION (2/3)

Government Pattern

X = I	Y = II
1. N's	1. *against* N
2. A_{poss}	2. *at* N
	3. *for* N
	4. *toward* N

(1) $C_{II.2}$: N denotes something that happens and can be seen or felt

(2) $C_{II.4}$: N denotes people

Kahane (2003, p.19)

Lexical entry REVULSION (3/3)

Lexical Function (selected functions)

Magn+Labor$_{12}$	: fill [N=X with $\sim$]
Oper$_1$	: experience, feel $\sim$
Magn	: deep $<$ extreme $<$ utmost
AntiMagn	: slight
Syn$_{cap}$	: repugnance; repulsion; disgust; loathing; distaste
Anti$_{cap}$	: attraction

X refers to X from the government pattern definition

Based on Kahane (2003)

Outline

The Prague Dependency Treebank

- The Prague Dependency Treebank (Hajicova 2000, PDT) is a joint project between Charles University in Prague and Masaryk University in Brno
- It is a manually annotation project that provides rich linguistic annotation of Czech data
- Annotations include morphology, syntax, semantics/pragmatics 'and beyond'
  (http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/html/ch01.html, accessed October 20th 2009)

PDT 1.0 and 2.0

- The Prague Dependency Treebank has two releases:
  - PDT 1.0: contains morphological and surface syntactic annotations
  - PDT 2.0: tectogrammatical representation
- The project has two main goals:
  - Empirical testing of linguistic theory developed at the Prague Linguistics School
  - Develop data that can be used for Machine Learning

Main components of the PDT

- The Morphological layer
- The Analytical layer
- The Tectogrammatical layer
- PDT-Vallex (Dictionary for lexical entries)

The Morphological layer

- The morphological layer provides three kinds of information:
  - the surface form
  - the base form (nominative, infinitive, etc.)
  - a tag specifying the morphemes that are present
- Morphological annotation is applied to individual tokens (no analysis of complex forms)
- The tagset was taken from a morphological dictionary for Czech (developed at UFAL)

The analytical representation

- A rooted tree that represents the surface syntactic structure
- The mapping from nodes in the dependency tree to words on the surface is one to one: no ellipses, traces, etc.
- The representation is a dependency tree
- Order of the nodes corresponds to the original linear order of words in the sentence

Tectogrammatical layer

- A rooted dependency tree consisting of labeled edges and nodes
- Represents the deep syntactic structure of the sentence
- Nodes are (almost exclusively) full semantic items, occasionally accompanied by grammetemes
- Mapping of nodes is not one to one with words on the surface (zero nodes to match the theory and prepositions are not present in this representation)

Valency

- **Valency** refers to the ability of a word to take arguments
- Each word may take a specific number of arguments, this is specified by its **frame**
- Some of them may be obligatory, some may be optional
  - Sometimes a distinction is made between **semantic valency** (all arguments) and **syntactic valency** (optional arguments)
  - When the term **subcategorization** is used, focus lies more on the syntactic properties of the arguments
  - 'Valency' in the PDT may refer to all three (optional and obligatory arguments and relevant syntactic information)

Valency labels

- Arguments are labeled according to their semantic role
- We will not discuss semantic roles in detail here, but here is an example of semantic roles:
  1. the boy broke the window with a stone.
     ACTOR PATIENT MEANS

     the stone broke the window.
     MEANS PATIENT
  2. the window broke.
     PATIENT

PDT-Vallex

- A dictionary that contains verbs, deverbal nouns and adjectives found in the corpus
- Each item has the following information:
  - Individual sense of the item
  - A corresponding valency frame:
    - zero or more valency slots, each labeled with a syntactic or semantic relation
    - it is marked 'optional' or 'obligatory'
    - it contains surface syntactic and morphological information
- Each item is linked to the place(s) where it was found in the corpus

For more information...

see:

http://ufal.mff.cuni.cz/pdt/Corpora/PDT$_1$.0/Doc/whatis.html

and

http://ufal.mff.cuni.cz/pdt/Corpora/PDT$_2$.0/Doc/whatis.html

Outline

Some advantages of Dependency Grammars (1/2)

- A Meaning-Text Model is modular: correspondences can be defined independent of each other
- There is a close connection to semantics:
  - Clean treatment of active-passive alternation, dative shift, etc.
  - syntactic analysis gears towards representations that are suited for Machine Translation

Some advantages of Dependency Grammars (2/2)

- Some phenomena are easier to treat (Hudson):
  - subject-verb agreement: the relation between subject and verb is direct (not a 'second cousin')
  - selection of specific prepositions, lexical case assignment is also captured by a direct relation between a head and its dependent (e.g. *look at*, *depend on*)
  - Non-constituent coordination is not much of an issue, e.g. *I had coffee at eleven and tea at four*
  - Free word order and discontinuous constituents
- PS-trees contain a lot of redundancy (passing up information to N', N", etc.)

Partially based on Kordoni (2008b)

Difficulties for Dependency Grammars

Dependency has difficulties with groupings

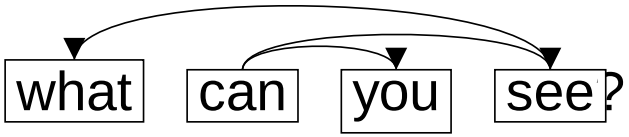
- Coordination: how to capture the symmetry in coordination structures?
- How to integrate dependents that modify both coordinands?
- Modification of a restricted expression:
  - E.g. *I lived in Bordeaux in 2001*
  - *in 2001* depends on *lived*, but I lived at other places in other years

Partially based on Kordoni (2008b)

Outline

Structure sharing in Word Grammar (1/2)

Two challenges for dependency structures in Word Grammar:

- Strict projectivity is assumed: what to do with non-projective sentences?
  - 

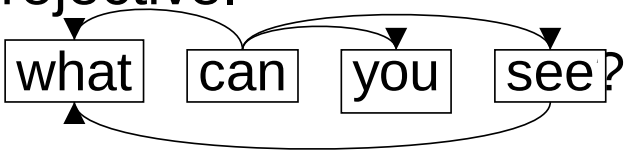
Structure sharing in Word Grammar (2/2)

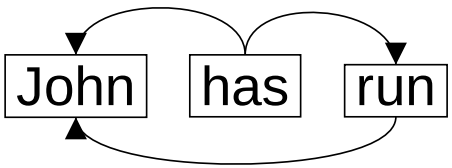
- We have seen that it is not always trivial to identify dependencies:
  What to do when a dependent seems to be governed by more than one head?
  - John has run
  - It keeps raining
  - He washed the dish clean
- Hudson accounts for such examples through *structure-sharing*

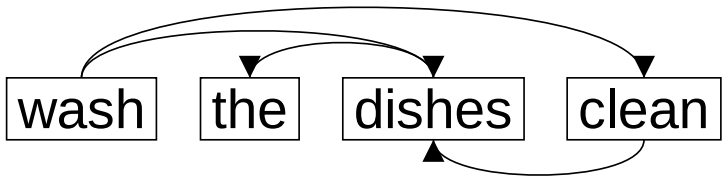
Structure sharing in Word Grammar (2/2)

- We have seen that it is not always trivial to identify dependencies:
  What to do when a dependent seems to be governed by more than one head?
  - John has run (*has* agrees with *John*, but 'John' is the 'runner')
  - It keeps raining (*raining* selects *it*, but *keep* agrees)
  - He washed the dish clean (*clean* says something about *wash* and *dish*)
- Hudson accounts for such examples through *structure-sharing*

Structure Sharing

- Word Grammar explicitly allows for **structure sharing**, i.e. it allows items to depend on two or more items
- Elements that appear in a non-projective position, are said to be **extracted** by another item, in relation to which they are projective.
  - 

  - This makes at least part of the structure projective...
- Similarly, we can account for 'double dependencies':
  - 

  - 

Condition on structure sharing

- When an item is governed by multiple heads, there are restrictions on what these heads may be: not any two items can share a dependent
- Structure-sharing:
  - If A governs B, A may licence structure-sharing between A and B. If structure sharing occurs, A and B both govern a third item C.
- The condition that structure-sharing may only occur between a head and its direct depends allows to define structure-sharing as a property of the head

Bibliography I

- Haegeman, Liliane (1991). *Introduction to Government and Binding Theory*. Oxford, UK: Blackwell Publishers.
- Hajicova, Eva (2000). Dependency-Based Underlying-Structure Tagging of a Very Large Czech Corpus In T.A.L., vol. 41, n.1, pp. 47-66
- Hudson, Richard A. (1984). *Word Grammar*. New York, USA: Blackwell.
- Hudson, Richard A. (1990). *English Word Grammar*. Oxford, UK: Blackwell. http://www.phon.ucl.ac.uk/home/dick/ewg.htm.
- Hudson, Richard A. (2007). *Language Networks - The new word grammar*. New York, USA: Oxford Press.
- 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.

Bibliography II

- 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.
- 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.