

Special Topics in Computer Science

# NLP in a Nutshell

**CS492B Spring Semester 2009**

**Jong C. Park**

Computer Science Department

Korea Advanced Institute of Science and Technology

# **CHOMSKY'S GRAMMAR IN SYNTACTIC STRUCTURES**

# Chomsky's Grammar in Syntactic Structures

- Introduction
  - Constituency
  - Dependency

# Chomsky's Grammar in Syntactic Structures

- Three levels of a grammar
  - phrase-structure (PS) rules expressing constituency
  - transformation rules complementing PS rules
  - morphophonemic component handling the final word generation
    - [en + arrive] → arrived

# Chomsky's Grammar in Syntactic Structures

## ■ Constituency: A Formal Definition

### ■ Example PS Rules

sentence  $\rightarrow$  np, vp.

np  $\rightarrow$  t, n.

vp  $\rightarrow$  verb, np.

t  $\rightarrow$  [the].

n  $\rightarrow$  [man]; [ball]; etc.

verb  $\rightarrow$  [hit]; [took]; etc.

# Chomsky's Grammar in Syntactic Structures

## ■ Constituency: A Formal Definition

Fig. 10.1. Generation of sentences.

<i>Sentence</i>	0
<i>NP + VP</i>	1
<i>T + N + VP</i>	2
<i>T + N + Verb + NP</i>	3
<i>the + N + Verb + NP</i>	4
<i>the + man + Verb + NP</i>	5
<i>the + man + hit + NP</i>	6
<i>the + man + hit + T + N</i>	7
<i>the + man + hit + the + N</i>	8
<i>the + man + hit + the + ball</i>	9

# Chomsky's Grammar in Syntactic Structures

## ■ Constituency: A Formal Definition

### ■ Chomsky normal form (CNF)

- Rules in the CNF have either two nonterminal symbols to their right-hand side or one nonempty terminal symbol.
- Example
  - lhs  $\rightarrow$  rhs1, rhs2.
  - lhs  $\rightarrow$  [a]
- cf. weak equivalence

# Chomsky's Grammar in Syntactic Structures

## ■ Transformations

■ T1 : np1, aux, v, np2 →  
np2, aux, [be], [en], v, [by], np1

### ■ Example

– the man will hit the ball →

the ball will be (en hit) by the boy

■ T2: affix, v → v, affix, #

### ■ Example

– the ball will be hit en # by the boy

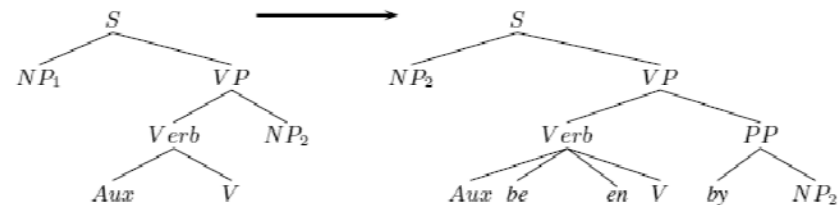
■ Morphophonemic rules: hit en # → hit



# Chomsky's Grammar in Syntactic Structures

## ■ Transformations

Fig. 10.2. A tree-to-tree mapping representing the active/passive transformational rule.



# Chomsky's Grammar in Syntactic Structures

## ■ Transformations

### ■ Other common transformations

- Negations: *John comes* → *John doesn't come.*
- Yes/no questions: *they arrive* → *do they arrive; ...*
- Interrogatives: *John ate an apple* → *did John eat an apple; ...*
- Conjunction: (*the scene of the movie was in Chicago; the scene of the play was in Chicago*) → *the scene of the movie and of the play was in Chicago*
- Topicalization: *the waiter brought the meal to the table* → *to the table, the waiter brought the meal; ...*

# Chomsky's Grammar in Syntactic Structures

- Transformations and *Movements*
  - A movement is a sentence rearrangement where a constituent is moved to another location.
    - The moved constituent leaves a trace.

# Chomsky's Grammar in Syntactic Structures

## ■ Transformations and Movements

Table 10.1. Movements to obtain the passive of sentence *The man hit the ball*. Traces are represented by—. Original positions of traces are in bold.

Movements	Traces	Passives
First Movement	<b>The man</b> hit ...	... -- is hit by the man
Second Movement	... hit <b>the ball</b>	The ball is hit --

Table 10.2. Questions beginning with a *wh*-word and their traces (—).

Question	Traces
Who ate an apple in the dining room?	—ate an apple in the dining room
What did John eat in the dining room?	John ate — in the dining room
Which apple did John eat in the dining room?	John ate — in the dining room
Where did John eat an apple?	John ate an apple —

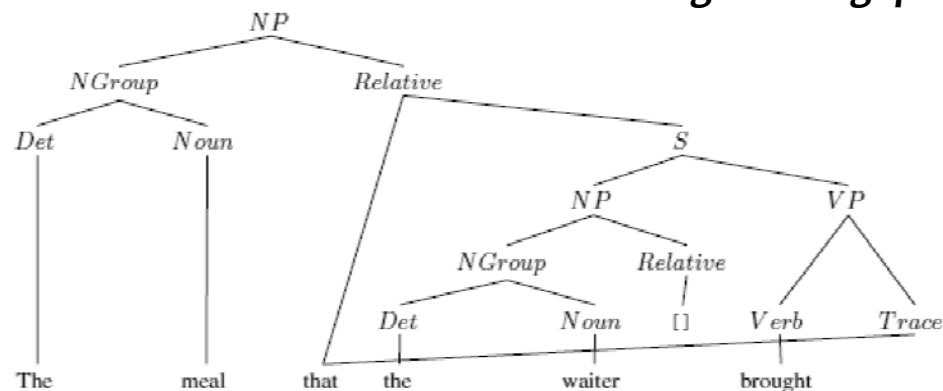
# Chomsky's Grammar in Syntactic Structures

## ■ Gap Threading

- a technique to parse *wh*-movements

## ■ Gap Threading to Parse Relative Clauses

Fig. 10.3. The parse tree of *The meal that the waiter brought with gap threading.*



# Standard Phrase Categories for English

- Sample category names
  - Noun phrases (NPs)
  - Verb phrases (VPs)
  - Adjective phrases (AdjPs)
  - Adverbial phrases (AdvPs)
  - Prepositional phrases (PPs)

**Table 10.3. The Penn Treebank phrase labels. After Marcus et al. (1993).**

	<b>Categories</b>	<b>Description</b>
1.	ADJP	Adjective phrase
2.	ADVP	Adverb phrase
3.	NP	Noun phrase
4.	PP	Prepositional phrase
5.	S	Simple declarative clause
6.	SBAR	Clause introduced by subordinating conjunction or <i>o</i> (see below)
7.	SBARQ	Direct question introduced by <i>wh</i> -word of <i>wh</i> -phrase
8.	SINV	Declarative sentence with subject-aux inversion
9.	SQ	Subconstituent of SBARQ excluding <i>wh</i> -word of <i>wh</i> -phrase
10.	VP	Verb phrase
11.	WHADVP	<i>wh</i> -adverb phrase
12.	WHNP	<i>wh</i> -noun phras
13.	WHPP	<i>wh</i> -prepositional phrase
14.	X	Constituent of unknown or uncertain category
	<b>Null elements</b>	
1.	*	“Understood” subject of infinitive or imperative
2.	<i>o</i>	Zero variant of <i>that</i> in subordinate clauses
3.	T	Trace – marks position where moved <i>wh</i> -constituent is interpreted
4.	NIL	Marks position where preposition is interpreted in pied-piping context

# Standard Phrase Categories for English

Fig. 10.4. Bracketed text in the Penn Treebank. After Marcus et al. (1993, p. 325).

```
( (S
  (NP Battle-tested industrial managers
    here)
  always
  (VP buck
    up
    (NP nervous newcomers)
    (PP with
      (NP the tale
        (PP of
          (NP (NP the
              (ADJP first
                (PP of
                  (NP their countrymen)))
            (S (NP *)
              to
              (VP visit
                (NP Mexico))))
          (NP (NP a boatload
              (PP of
                (NP (NP samurai warriors)
                  (VP-1 blown
                    ashore
                    (ADVP (NP 375 years)
                      ago))))))
            (VP-1 *pseudo-attach*)))))))))
  .)
```



# Unification-Based Grammars

## ■ Features

Table 10.4. Inflection imposed to noun group *der kleine Ober* ‘the small waiter’ by the case feature in German.

Cases	Noun groups
Nominative	<i>der kleine Ober</i>
Genitive	<i>der kleinen Obers</i>
Dative	<i>dem kleinen Ober</i>
Accusative	<i>den kleinen Ober</i>

# Unification-Based Grammars

## ■ Representing Features in Prolog

- `np(case: C)`

- `np(gend:G, num:N, case:C, pers:P, det:D)`

- `np(gend:G, num:N, case:C, pers:P, det:D) -->`  
`det(gend:G, num:N, case:C, pers:P, det:D),`  
`adj(gend:G, num:N, case:C, pers:P, det:D),`  
`n(gend:G, num:N, case:C, pers:P).`

# Unification-Based Grammars

## ■ A Formalism for Features and Rules

$$\begin{array}{c}
 NP \rightarrow DET \quad ADJ \quad N \\
 \left[ \begin{array}{l} gend : G \\ num : N \\ case : C \\ pers : P \\ det : D \end{array} \right] \quad \left[ \begin{array}{l} gend : G \\ num : N \\ case : C \\ pers : P \\ det : D \end{array} \right] \quad \left[ \begin{array}{l} gend : G \\ num : N \\ case : C \\ pers : P \\ det : D \end{array} \right] \quad \left[ \begin{array}{l} gend : G \\ num : N \\ case : C \\ pers : P \end{array} \right] \\
 \\
 S \rightarrow NP \quad VP \\
 \left[ \begin{array}{l} num : N \\ case : nom \\ pers : P \end{array} \right] \quad \left[ \begin{array}{l} num : N \\ pers : P \end{array} \right] \\
 \\
 VP \rightarrow V \\
 \left[ \begin{array}{l} num : N \\ pers : P \end{array} \right] \quad \left[ \begin{array}{l} trans : i \\ num : N \\ pers : P \end{array} \right]
 \end{array}$$

# Unification-Based Grammars

## ■ A Formalism for Features and Rules

$$VP \rightarrow V \quad NP$$
$$\begin{bmatrix} num : N \\ pers : P \end{bmatrix} \quad \begin{bmatrix} trans : t \\ num : N \\ pers : P \end{bmatrix} \quad [case : acc]$$

$$NP \rightarrow Pronoun$$
$$\begin{bmatrix} gen : G \\ num : N \\ pers : P \\ case : C \end{bmatrix} \quad \begin{bmatrix} gen : G \\ num : N \\ pers : P \\ case : C \end{bmatrix}$$

$$DET \rightarrow der$$
$$\begin{bmatrix} gend : masc \\ num : sg \\ case : nom \\ det : de f \end{bmatrix}$$

# Unification-Based Grammars

## ■ Features Organization

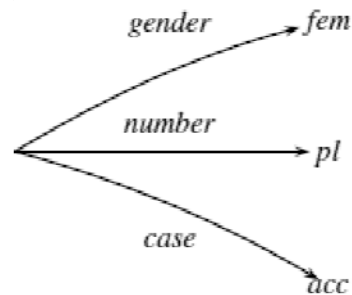
$$\begin{bmatrix} \textit{feature}_1 : \textit{value}_1 \\ \textit{feature}_2 : \textit{value}_2 \\ \vdots \\ \textit{feature}_n : \textit{value}_n \end{bmatrix}$$

$$\begin{bmatrix} \textit{gen} : \textit{fem} \\ \textit{num} : \textit{pl} \\ \textit{case} : \textit{acc} \end{bmatrix} \quad \begin{bmatrix} \textit{num} : \textit{pl} \\ \textit{case} : \textit{acc} \\ \textit{gen} : \textit{fem} \end{bmatrix}$$

# Unification-Based Grammars

## ■ Features Organization

Fig. 10.5. Graph representing a feature structure.

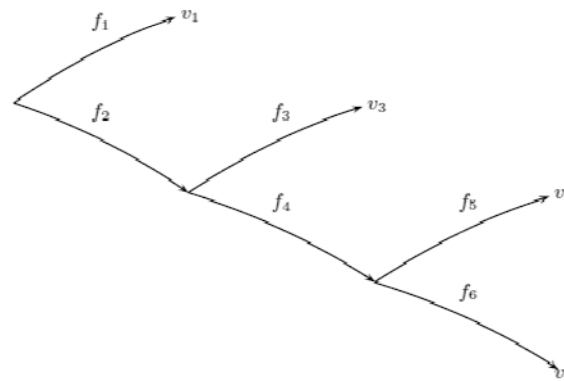


# Unification-Based Grammars

## ■ Features Organization

$$\left[ \begin{array}{l} f_1 : v_1 \\ f_2 : \left[ \begin{array}{l} f_3 : v_3 \\ f_4 : \left[ \begin{array}{l} f_5 : v_5 \\ f_6 : v_6 \end{array} \right] \end{array} \right] \end{array} \right]$$

Fig. 10.6. Graph corresponding to embedded feature structures.



# Unification-Based Grammars

## ■ Features Organization

### ■ grouping

$$\begin{array}{c} \text{Pronoun} \\ \left[ \begin{array}{l} \text{agreement : } \left[ \begin{array}{l} \text{gender : } \textit{masc} \\ \text{number : } \textit{sg} \\ \text{pers : } 3 \end{array} \right] \\ \text{case : } \textit{nom} \end{array} \right] \end{array} \rightarrow \textit{er}$$

$$\begin{array}{c} \text{Pronoun} \\ \left[ \begin{array}{l} \text{agreement : } \left[ \begin{array}{l} \text{gender : } \textit{masc} \\ \text{number : } \textit{sg} \\ \text{pers : } 3 \end{array} \right] \\ \text{case : } \textit{acc} \end{array} \right] \end{array} \rightarrow \textit{ihn}$$

$$\begin{array}{c} \text{NP} \\ \left[ \begin{array}{l} \text{agreement : } X \\ \text{case : } C \end{array} \right] \end{array} \rightarrow \begin{array}{c} \text{Pronoun} \\ \left[ \begin{array}{l} \text{agreement : } X \\ \text{case : } C \end{array} \right] \end{array}$$

$$\left[ \begin{array}{l} \text{cat : } \textit{np} \\ \text{agreement : } X \\ \text{case : } C \end{array} \right] \rightarrow \left[ \begin{array}{l} \text{cat : } \textit{pronoun} \\ \text{agreement : } X \\ \text{case : } C \end{array} \right]$$



# Unification-Based Grammars

## ■ Features and Unification

### ■ Unification of feature structures

- Structures merge the set of all their features, checking that identical features have compatible values.
- Variables unify with values and substructures.
- Feature structure unification is denoted  $\cup$ .

# Unification-Based Grammars

## ■ Features and Unification

$$\begin{bmatrix} \text{feature}_1 : v_1 \\ \text{feature}_2 : v_2 \end{bmatrix} \cup \begin{bmatrix} \text{feature}_2 : v_2 \\ \text{feature}_3 : v_3 \end{bmatrix} = \begin{bmatrix} \text{feature}_1 : v_1 \\ \text{feature}_2 : v_2 \\ \text{feature}_3 : v_3 \end{bmatrix}.$$

$$\begin{matrix} [\text{feature}_1 : v_1] & [\text{feature}_1 : v_2] \end{matrix}$$

$$\begin{bmatrix} f_1 : v_1 \\ f_2 : X \end{bmatrix} \cup \begin{bmatrix} f_5 : v_5 \\ f_2 : \begin{bmatrix} f_3 : v_3 \\ f_4 : v_4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} f_1 : v_1 \\ f_2 : \begin{bmatrix} f_3 : v_3 \\ f_4 : v_4 \end{bmatrix} \\ f_5 : v_5 \end{bmatrix}$$

$$\begin{bmatrix} f_1 : v_1 \\ f_2 : X \end{bmatrix} \cup \begin{bmatrix} f_5 : X \\ f_2 : \begin{bmatrix} f_3 : v_3 \\ f_4 : v_4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} f_1 : v_1 \\ f_2 : \begin{bmatrix} f_3 : v_3 \\ f_4 : v_4 \end{bmatrix} \\ f_5 : \begin{bmatrix} f_3 : v_3 \\ f_4 : v_4 \end{bmatrix} \end{bmatrix}$$

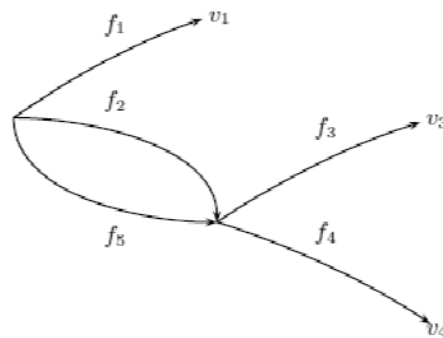
# Unification-Based Grammars

## ■ Features and Unification

$$\left[ \begin{array}{l} f_1 : v_1 \\ f_2 : \left[ \begin{array}{l} f_3 : v_3 \\ f_4 : v_4 \end{array} \right] \end{array} \right] \quad \left[ \begin{array}{l} f_5 : \left[ \begin{array}{l} f_3 : v_3 \\ f_4 : v_4 \end{array} \right] \\ f_2 : X \end{array} \right]$$

$$\left[ \begin{array}{l} f_1 : v_1 \\ f_2 : [1] \left[ \begin{array}{l} f_3 : v_3 \\ f_4 : v_4 \end{array} \right] \\ f_5 : [1] \end{array} \right]$$

Fig. 10.7. Graph with re-entrant feature structures.



# Unification-Based Grammars

## ■ A Unification Algorithm for Feature Structures

$$[case : nom] \cup [gender : masc] = \begin{bmatrix} case : nom \\ gender : masc \end{bmatrix}$$

■ struct(case:nom) and struct(gender:masc) ?

■ Using the anonymous variable

■ struct(case: nom, gender: \_)

■ struct(case: \_, gender: masc)

■ Using incomplete lists

■ [case: nom | X]

■ [gender: masc | Y]

# Unification-Based Grammars

## ■ A Unification Algorithm for Feature Structures

$$\left[ \begin{array}{l} \text{cat} : \text{np} \\ \text{agreement} : \left[ \begin{array}{l} \text{gender} : \text{masc} \\ \text{number} : \text{sg} \\ \text{pers} : 3 \end{array} \right] \\ \text{case} : \text{acc} \end{array} \right]$$

- [cat: np,  
agreement: [gender: masc, number: sg, pers: 3 | \_],  
case: acc | \_]

# Dependency Grammars

## ■ Presentation

### ■ Example

#### ■ The very big cat

Fig. 10.8. Dependency graph of the noun group *The very big cat*.

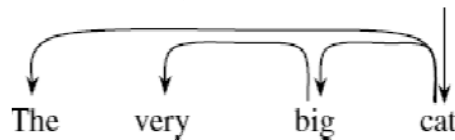
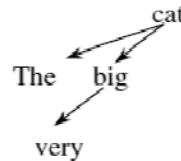


Fig. 10.9. Tree representing dependencies in the noun group *The very big cat*.



# Dependency Grammars

Fig. 10.10. Dependency graph or stemma of the sentence *The waiter brought the meal*.

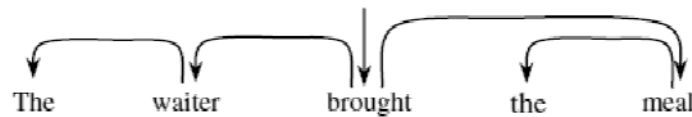
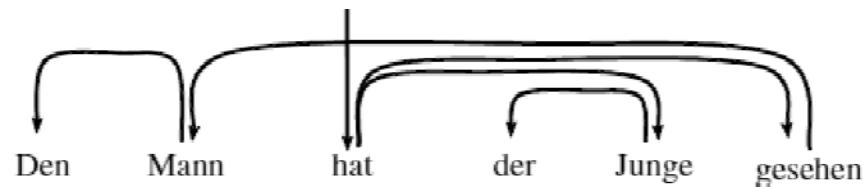


Fig. 10.11. Dependency graph of *Den Mann hat der Junge gesehen*, modified from Broker (1998).

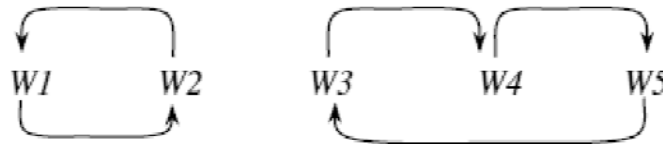


# Dependency Grammars

## ■ Properties of a Dependency Graph

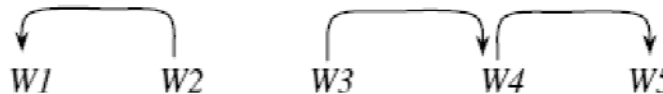
- Principle 1: Dependency graphs are acyclic.

Fig. 10.12. Cyclic dependencies in a graph.



- Principle 2: Dependency graphs should be connected.

Fig. 10.13. A nonconnected graph spanning sentence  $w_1, w_2, w_3, w_4, w_5$



- Principle 3: All the dependents of a word form a contiguous sequence.



# Dependency Grammars

## ■ Properties of a Dependency Graph

Fig. 10.14. Dependency graph of *What would you like me to do?* After Jarvinen and Tapanainen (1997).

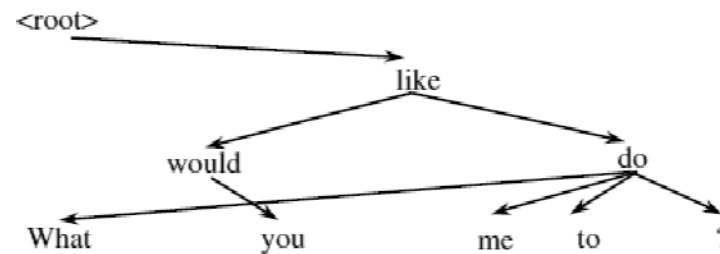
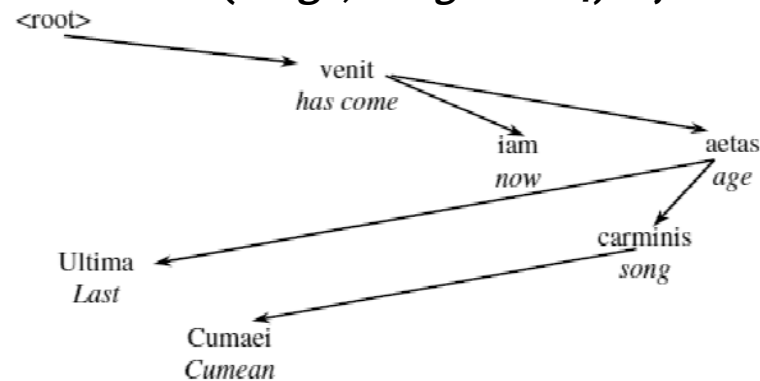


Fig. 10.15. Dependency graph of *Ultima Cumaei venit iam carminis aetas.* 'The last era of the Cumean song has now arrived' (Vergil, *Eclogues IV.4*). After Covington (1990).



# Dependency Grammars

## ■ Valence

- the number of essential complements of a verb
  - 0, for verbs describing weather, *it's raining, snowing*
  - 1, corresponding to the subject of intransitive verbs, *he's sleeping, vanishing*
  - 2, the subject and object of transitive verbs, *she read this book*
  - 3, the subject and two objects – direct and indirect objects – of ditransitive verbs, *Elke gave a book to Wolfgang, I said it to my sister*
  - 4, the subject, object, source, and destination of certain verbs like *move* or *shift*, *I moved the car from here to the street*

# Dependency Grammars

## ■ Valence

Table 10.5. Valence values and examples, where *iobject* denotes the indirect object.

Valences	Examples	Frames
0	<i>it's raining</i>	<i>raining</i> []
1	<i>he's sleeping</i>	<i>sleeping</i> [subject : <i>he</i> ]
2	<i>she read this book</i>	<i>read</i> [subject : <i>she</i> object : <i>book</i> ]
3	<i>Elke gave a book to Wolfgang</i>	<i>gave</i> [subject : <i>Elke</i> object : <i>book</i> iobject : <i>Wolfgang</i> ]
4	<i>I moved the car from here to the street</i>	<i>moved</i> [subject : <i>I</i> object : <i>car</i> source : <i>here</i> destination : <i>street</i> ]

# Dependency Grammars

## ■ Valence

Table 10.6. Verb-complement structures in English

Verb	Complement structure	Example
<i>slept</i>	None (Intransitive)	<i>I slept</i>
<i>bring</i>	NP	<i>The waiter brought the meal</i>
<i>bring</i>	NP + to + NP	<i>The waiter brought the meal to the patron</i>
<i>depend</i>	on + NP	<i>It depends on the waiter</i>
<i>wait</i>	for + NP + to + VP	<i>I am waiting for the waiter to bring the meal</i>
<i>keep</i>	VP(ing)	<i>He kept working</i>
<i>know</i>	that + S	<i>The waiter knows that the patron loves fish</i>

# Dependency Grammars

## ■ Valence

Table 10.7. Verb–complement structures in French.

Verb	Complement structure	Example
<i>dormir</i>	None (Intransitive)	<i>J'ai dormi</i>
<i>apporter</i>	NP	<i>Le serveur a apporté un plat</i>
<i>apporter</i>	NP + à + NP	<i>Le serveur a apporté un plat au client</i>
<i>dépendre</i>	de + NP	<i>Ça dépend du serveur</i>
<i>attendre</i>	que + S(Subjunctive)	<i>Il a attendu que le serveur apporte le plat</i>
<i>continuer</i>	de + VP(INF)	<i>Il a continué de travailler</i>
<i>savoir</i>	que + S	<i>Le serveur sait que le client aime le poisson</i>

# Dependency Grammars

## ■ Valence

Table 10.8. Verb-complement structure in German.

Verb	Complement structure	Example
<i>schlafen</i>	None (Intransitive)	<i>Ich habe geschlafen</i>
<i>bringen</i>	NP(Accusative)	<i>Der Ober hat eine Speise gebracht</i>
<i>bringen</i>	NP(Dative) + NP(Accusative)	<i>Der Ober hat dem Kunde eine Speise gebracht</i>
<i>abhängen</i>	von + NP	<i>Es hängt vom Ober ab</i>
<i>warten</i>	auf + S	<i>Er wartete auf dem Ober, die Speise zu bringen</i>
<i>fortsetzen</i>	NP	<i>Er hat die Arbeit fortgesetzt</i>
<i>wissen</i>	NP(Final verb)	<i>Der Ober weiß, das der Kunde Fisch liebt</i>

# Dependency Grammars

## ■ Dependencies and Functions

Fig. 10.16. Dependency graph of the sentence *The waiter brought the meal.*

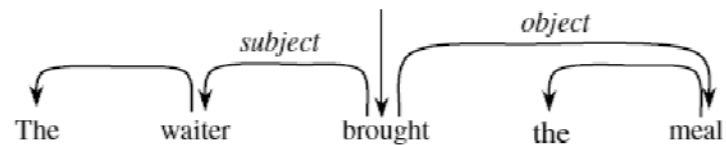
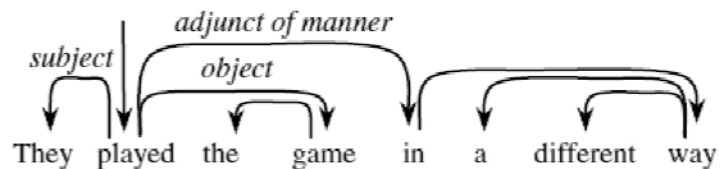


Fig. 10.17. Dependency graph of the sentence *They played the game in a different way.* After Järvinen and Tapanainen (1997).



# Dependency Grammars

## ■ Dependencies and Functions

Fig. 10.18. Dependency graph of the sentence *John Smith, the president of IBM, announced his resignation yesterday.* After Collins (1996).

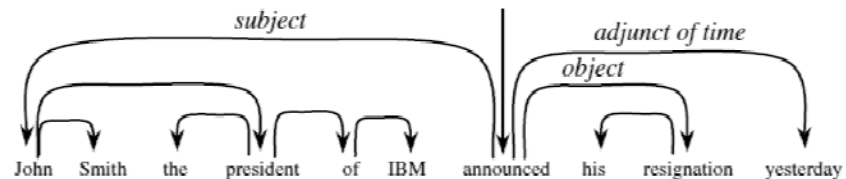




Table 10.9 (1) Main functions used by Järvinen and Tapanainen (1997) in their functional dependency parser for English. Intranuclear links combine words inside a *noeud* (a constituent). *Verb* complementation links a verb to its core complements. *Determinative* functions generally connect determiners to nouns. *Modifiers* are pre- or postmodifiers of a noun, i.e., dependents of a noun before or after it.

Name	Description	Example
<b>Main functions</b>		
main	Main element, usually the verb	<i>He doesn't <b>know</b> whether to send a gift</i>
qtag	Question tag	<i>Let's play another game, <b>shall we?</b></i>
<b>Intranuclear links</b>		
v-ch	Verb chain, connects elements in a complex verb group	<i>It <b>may have been being</b> examined</i>
pcomp	Prepositional complement, connects a preposition to the noun group after it.	<i>They played the game <b>in a different way</b></i>
phr	Prepositional complement, connects a preposition to the noun group after it.	<i>He asked me who would look <b>after</b> the baby</i>
<b>Verb complementation</b>		
subj	Subject	
obj	Object	<i>I gave him <b>my address</b></i>
comp	Subject complement, the second argument of a copula.	<i>It has become <b>marginal</b></i>
dat	Indirect object	<i>Pauline gave it <b>to Tom</b></i>
oc	Object complement	<i>His friends call him <b>Ted</b></i>
copred	Copredicative	<i>We took a swim <b>naked</b></i>
voc	Vocative	<i>Play it again, <b>Sam</b></i>

Table 10.9 (2) Main functions used by Järvinen and Tapanainen (1997) in their functional dependency parser for English. Intranuclear links combine words inside a *noeud* (a constituent). *Verb* complementation links a verb to its core complements. *Determinative* functions generally connect determiners to nouns. *Modifiers* are pre- or postmodifiers of a noun, i.e., dependents of a noun before or after it.

Name	Description	Example
<b>Determinative functions</b>		
qn	Quantifier	<i>I want <b>more</b> money</i>
det	Determiner	<i><b>Other</b> members will join ...</i>
neg	Negator	<i>It is <b>not</b> coffee that I like, but tea</i>
<b>Modifier</b>		
attr	Attributive nominal	<i><u><b>Knowing</b> no French</u>, I couldn't express my thanks.</i>
mod	Other postmodifiers	<i>The baby, <u>Frances <b>Bean</b></u>, was...</i>
ad	Attributive adverbial	<i>The people <u><b>on the bus</b></u> were singing She is <b>more</b> popular</i>
<b>Junctives</b>		
cc	Coordination	<i><u>Two <b>or</b> more</u> cars.</i>

# Dependency Grammars

**Table 10.10. Adverbial functions used by Järvinen and Tapanainen (1997). Adverbial functions connect adjuncts to their verb.**

Name	Description	Example
<b>Adverbial functions</b>		
tmp	Quantifier	<i>I want <b>more</b> money</i>
dur	Determiner	<i><b>Other</b> members will join ...</i>
frq	Negator	<i>It is <b>not</b> coffee that I like, but tea</i>
qua	Quantity	<i>It is weighed <u>almost a ton</u></i>
man	Manner	<i>They will support him, however <b>grudgingly</b>...</i>
loc	Location	<i>I don't know <b>where</b> to meet him</i>
sou	Source	<i>They travelled slowly <b>from</b> Hong Kong</i>
goa	Goal	<i>They moved <b>into</b> the kitchen every stick of furniture they possessed</i>
cnd	Condition	<i><u>If I were <b>leaving</b></u>, you should know about it</i>
meta	Clause adverbial	<i>Will somebody <b>please</b> open the door?</i>
cla	Clause initial element	<i><b>In</b> the view of the authorities, Jones was...</i>

# Dependency Grammars

Fig. 10.19. Stemma representing *The people on the bus were singing*. After Järvinen and Tapanainen (1997).

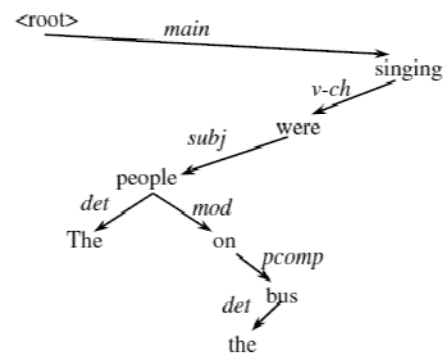


Fig. 10.20. Stemma representing *Anna, my best friend, was here last night*. After Järvinen and Tapanainen (1997).

