Syntax has been the core of linguistics in the US and elsewhere for many years. Noam Chomsky, professor at the MIT, has had an overwhelming influence, sometimes misleading. *Syntactic structures* (1957) has been a cult book for the past generation of linguists. Syntax can be divided into two parts:

- Formalism – How to represent syntax
- Parsing – How to get the representation of a sentence
The two most accepted formalisms use a tree representation:

- One is based on the idea of constituents
- Another is based on dependencies between words. Trees have originally been called stemmas

They are generally associated respectively to Chomsky and Tesnière. Later, constituent grammars evolved into unification grammars
Constituency can be expressed by context-free grammars. They are defined by

1. A set of designated start symbols, \( \Sigma \), covering the sentences to parse. This set can be reduced to a single symbol, such as sentence, or divided into more symbols: declarative_sentence, interrogative_sentence.

2. A set of nonterminal symbols enabling the representation of the syntactic categories. This set includes the sentence and phrase categories.

3. A set of terminal symbols representing the vocabulary: words of the lexicon, possibly morphemes.

4. A set of rules, \( F \), where the left-hand-side symbol of the rule is rewritten in the sequence of symbols of the right-hand side.
These grammars can be mapped to DCG rules as for

*The boy hit the ball*

```
sentence --> np, vp.
np --> t, n.
vp -- verb, np.
t --> [the].
n --> [man] ; [ball] ; etc.
verb --> [hit] ; [took] ; etc.
```

Generation of sentences is one of the purposes of grammar according to Chomsky
In some parsing algorithms, it is necessary to have rules in the Chomsky normal form (CNF) with two right-hand-side symbols.

Non-CNf rules:

\[ \text{lhs} \rightarrow \text{rhs1, rhs2, rhs3}. \]

can be converted into a CNF equivalent:

\[ \text{lhs} \rightarrow \text{rhs1, lhs_aux}. \]
\[ \text{lhs_aux} \rightarrow \text{rhs2, rhs3}. \]
Transformations

Rearrangement of sentences according to some syntactic relations: active/passive, declarative/interrogative, etc.
Transformations use rules – transformational rules or T rules –

*The boy will hit the ball/the ball will be (en) hit by the boy*

T1:  np1, aux, v, np2 --->
     np2, aux, [be], [en], v, [by], np1
Transformations

\[ S \rightarrow NP_1 \quad VP \quad NP_2 \]

\[ NP_1 \quad Verb \quad Aux \quad V \]

\[ NP_2 \quad Verb \quad Aux \quad be \quad en \quad V \quad by \quad NP_1 \]
## Syntactic Categories (Penn Treebank)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ADJP</td>
<td>Adjective phrase</td>
</tr>
<tr>
<td>2. ADVP</td>
<td>Adverb phrase</td>
</tr>
<tr>
<td>3. NP</td>
<td>Noun phrase</td>
</tr>
<tr>
<td>4. PP</td>
<td>Prepositional phrase</td>
</tr>
<tr>
<td>5. S</td>
<td>Simple declarative clause</td>
</tr>
<tr>
<td>6. SBAR</td>
<td>Clause introduced by subordinating conjunction or 0</td>
</tr>
<tr>
<td>7. SBARQ</td>
<td>Direct question introduced by <em>wh</em>-word or phrase</td>
</tr>
<tr>
<td>8. SINV</td>
<td>Declarative sentence with subject-aux inversion</td>
</tr>
<tr>
<td>9. SQ</td>
<td>Subconstituent of SBARQ excluding <em>wh</em>-word or phrase</td>
</tr>
<tr>
<td>10. VP</td>
<td>Verb phrase</td>
</tr>
<tr>
<td>11. WHADVP</td>
<td><em>wh</em>-adverb phrase</td>
</tr>
<tr>
<td>12. WHNP</td>
<td><em>wh</em>-noun phrase</td>
</tr>
<tr>
<td>13. WHPP</td>
<td><em>wh</em>-prepositional phrase</td>
</tr>
<tr>
<td>14. X</td>
<td>Constituent of unknown or uncertain category</td>
</tr>
</tbody>
</table>
Battle-tested industrial managers here always buck up nervous newcomers with the tale of the first of their countrymen to visit Mexico, a boatload of samurai warriors blown ashore 375 years ago.

(S
  (NP Battle-tested industrial managers here)
  always
  (VP buck
    up
    (NP nervous newcomers)
    (PP with
      (NP the tale
      (PP of

A Hand-Parsed Sentence using the Penn Treebank Annotation

(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))))))
(WP pseudo-attach*)

Unification-based Grammars

Grammatical features such as case modify the word morphology

<table>
<thead>
<tr>
<th>Cases</th>
<th>Noun groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominative</td>
<td>der kleine Ober</td>
</tr>
<tr>
<td>Genitive</td>
<td>des kleinen Obers</td>
</tr>
<tr>
<td>Dative</td>
<td>dem kleinen Ober</td>
</tr>
<tr>
<td>Accusative</td>
<td>den kleinen Ober</td>
</tr>
</tbody>
</table>

The rule

np --> det, adj, n.

outputs ungrammatical phrases as:

?-np(L, []).
[der, kleinen, Ober]; %wrong
[der, kleinen, Obers]; %wrong
[dem, kleine, Obers] %wrong
...

Cases Noun groups

Pierre Nugues
A possible solution is to use arguments: \( np(\text{case}: C) \) where the \( C \) value is a member of list \([\text{nom}, \text{gen}, \text{dat}, \text{acc}]\).

\[
np(\text{gend}: G, \text{num}: N, \text{case}: C, \text{pers}: P, \text{det}: D) \\
np(\text{gend}: G, \text{num}: N, \text{case}: C, \text{pers}: P, \text{det}: D) \rightarrow \\
\text{det}(\text{gend}: G, \text{num}: N, \text{case}: C, \text{pers}: P, \text{det}: D), \\
\text{adj}(\text{gend}: G, \text{num}: N, \text{case}: C, \text{pers}: P, \text{det}: D), \\
\text{n}(\text{gend}: G, \text{num}: N, \text{case}: C, \text{pers}: P).
\]
A Small Fragment of German

\[
det(\text{gend:masc}, \text{num:sg}, \text{case:nom}, \text{pers:3}, \text{det:def}) \rightarrow [\text{der}].
\]

\[
det(\text{gend:masc}, \text{num:sg}, \text{case:gen}, \text{pers:3}, \text{det:def}) \rightarrow [\text{des}].
\]

\[
det(\text{gend:masc}, \text{num:sg}, \text{case:dat}, \text{pers:3}, \text{det:def}) \rightarrow [\text{dem}].
\]

\[
det(\text{gend:masc}, \text{num:sg}, \text{case:acc}, \text{pers:3}, \text{det:def}) \rightarrow [\text{den}].
\]

\[
adj(\text{gend:masc}, \text{num:sg}, \text{case:nom}, \text{pers:3}, \text{det:def}) \rightarrow [\text{kleine}].
\]

\[
adj(\text{gend:masc}, \text{num:sg}, \text{case:gen}, \text{pers:3}, \text{det:def}) \rightarrow [\text{kleinen}].
\]

\[
adj(\text{gend:masc}, \text{num:sg}, \text{case:dat}, \text{pers:3}, \text{det:def}) \rightarrow [\text{kleinen}].
\]

\[
adj(\text{gend:masc}, \text{num:sg}, \text{case:acc}, \text{pers:3}, \text{det:def}) \rightarrow [\text{kleinen}].
\]

\[
n(\text{gend:masc}, \text{num:sg}, \text{case:nom}, \text{pers:3}) \rightarrow ['Ober'].
\]

\[
n(\text{gend:masc}, \text{num:sg}, \text{case:gen}, \text{pers:3}) \rightarrow ['Obers'].
\]

\[
n(\text{gend:masc}, \text{num:sg}, \text{case:dat}, \text{pers:3}) \rightarrow ['Ober'].
\]

\[
n(\text{gend:masc}, \text{num:sg}, \text{case:acc}, \text{pers:3}) \rightarrow ['Ober'].
\]
A Unification-based Formalism

Unification-based grammars use a notation close to that of DCGs

NP
\[
\begin{cases}
  \text{gend} : G \\
  \text{num} : N \\
  \text{case} : C \\
  \text{pers} : P \\
  \text{det} : D
\end{cases}
\] → DET
\[
\begin{cases}
  \text{gend} : G \\
  \text{num} : N \\
  \text{case} : C \\
  \text{pers} : P \\
  \text{det} : D
\end{cases}
\] → ADJ
\[
\begin{cases}
  \text{gend} : G \\
  \text{num} : N \\
  \text{case} : C \\
  \text{pers} : P \\
  \text{det} : D
\end{cases}
\] → N
\[
\begin{cases}
  \text{gend} : G \\
  \text{num} : N \\
  \text{case} : C \\
  \text{pers} : P
\end{cases}
\]
Some Rules

$$S \rightarrow NP \quad VP$$

$$NP \quad [\begin{array}{c}
\text{num} : N \\
\text{case} : \text{nom} \\
\text{pers} : P
\end{array}]$$

$$VP \quad [\begin{array}{c}
\text{num} : N \\
\text{pers} : P
\end{array}]$$

$$VP \rightarrow V$$

$$VP \quad [\begin{array}{c}
\text{num} : N \\
\text{pers} : P
\end{array}]$$

$$V \quad [\begin{array}{c}
\text{trans} : i \\
\text{num} : N \\
\text{pers} : P
\end{array}]$$

$$VP \rightarrow V$$

$$VP \quad [\begin{array}{c}
\text{num} : N \\
\text{pers} : P
\end{array}]$$

$$V \quad [\begin{array}{c}
\text{trans} : t \\
\text{num} : N \\
\text{pers} : P
\end{array}]$$

$$NP \quad [\begin{array}{c}
\text{case} : \text{acc}
\end{array}]$$
Feature Structures are Graphs

Structures can be embedded

\[
\begin{align*}
  f_1 &: v_1 \\
  f_2 &: \begin{cases}
    f_3 &: v_3 \\
    f_4 &: \begin{cases}
      f_5 &: v_5 \\
      f_6 &: v_6
    \end{cases}
  \end{cases}
\end{align*}
\]

**Pronoun** → **er**

\[
\begin{align*}
  \text{agreement} &: \begin{cases}
    \text{gender} &: \text{masc} \\
    \text{number} &: \text{sg} \\
    \text{pers} &: 3
  \end{cases} \\
  \text{case} &: \text{nom}
\end{align*}
\]

**Pronoun** → **ihn**

\[
\begin{align*}
  \text{agreement} &: \begin{cases}
    \text{gender} &: \text{masc} \\
    \text{number} &: \text{sg} \\
    \text{pers} &: 3
  \end{cases} \\
  \text{case} &: \text{acc}
\end{align*}
\]
Feature Structures are Graphs

\[ f_1 \rightarrow v_1 \]

\[ f_2 \rightarrow v_2 \]

\[ f_3 \rightarrow v_3 \]

\[ f_4 \rightarrow v_4 \]

\[ f_5 \rightarrow v_5 \]

\[ f_6 \rightarrow v_6 \]
The feature notation is based on the name, not on the position

\[
\begin{bmatrix}
\text{gen} & : & \text{fem} \\
\text{num} & : & \text{pl} \\
\text{case} & : & \text{acc}
\end{bmatrix}
\] and

\[
\begin{bmatrix}
\text{num} & : & \text{pl} \\
\text{case} & : & \text{acc} \\
\text{gen} & : & \text{fem}
\end{bmatrix}
\]

are equivalent

Unification is a generalization of Prolog unification

See the course book for the implementation
Dependency Grammars

Dependency grammars (DG) describe the structure in terms of links.

Each word has a head or “régissant” except the root of the sentence. A head has one or more modifiers or dependents:

Cat is the head of big and the; big is the head of very.

DG can be more versatile with a flexible word order language like German, Russian, or Latin.
The waiter brought the meal
Properties of Dependency Graphs

**Acyclic**

Each pair of words (Dep, Head) is only connected in one direction.

**Connected**

Every word in the graph is reachable from every other word.

**Projective**

Each pair of words (Dep, Head), directly connected, is only separated by direct or indirect dependents of Dep or Head.
Nonprojective Graphs (McDonald and Pereira)

\[
\begin{align*}
<\text{root}> & \quad \text{John} \quad \text{saw} \quad \text{a} \quad \text{dog} \quad \text{yesterday} \quad \text{which} \quad \text{was} \quad \text{a} \quad \text{Yorkshire} \quad \text{Terrier}
\end{align*}
\]
What would you like me to do?
Tesnière makes a distinction between essential and circumstantial complements. Essential – or core – complements are for instance subject and objects. Circumstantial – or noncore – complements are the adjuncts. Valence corresponds to the verb saturation of its essential complements.
## Valence Examples

<table>
<thead>
<tr>
<th>Val.</th>
<th>Examples</th>
<th>Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><em>it’s raining</em></td>
<td><em>raining</em></td>
</tr>
<tr>
<td>1</td>
<td><em>he’s sleeping</em></td>
<td><em>sleeping</em></td>
</tr>
<tr>
<td>2</td>
<td><em>she read this book</em></td>
<td><em>read</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>subject : she</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>object : book</em></td>
</tr>
<tr>
<td>3</td>
<td><em>Elke gave a book to Wolfgang</em></td>
<td><em>gave</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>subject : Elke</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>object : book</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>iobject : Wolfgang</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>subject : I</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>object : car</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>source : here</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>destination : street</em></td>
</tr>
</tbody>
</table>
Valence is a model of verb construction. It can be extended to more specific patterns as in the *Oxford Advanced Learner’s Dictionary* (OALD).

<table>
<thead>
<tr>
<th>Verb</th>
<th>Complement structure</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>slept</td>
<td>None (Intransitive)</td>
<td><em>I slept</em></td>
</tr>
<tr>
<td>bring</td>
<td>NP</td>
<td><em>The waiter brought the meal</em></td>
</tr>
<tr>
<td>bring</td>
<td>NP + to + NP</td>
<td><em>The waiter brought the meal to the patron</em></td>
</tr>
<tr>
<td>depend</td>
<td>on + NP</td>
<td><em>It depends on the waiter</em></td>
</tr>
<tr>
<td>wait</td>
<td>for + NP + to + VP</td>
<td><em>I am waiting for the waiter to bring the meal</em></td>
</tr>
<tr>
<td>keep</td>
<td>VP(ing)</td>
<td><em>He kept working</em></td>
</tr>
<tr>
<td>know</td>
<td>that + S</td>
<td><em>The waiter knows that the patron loves fish</em></td>
</tr>
</tbody>
</table>
## Subcategorization Frames in German

<table>
<thead>
<tr>
<th>Verb</th>
<th>Complement structure</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>schlafen</em></td>
<td>None (Intransitive)</td>
<td><em>Ich habe geschlafen</em></td>
</tr>
<tr>
<td><em>bringen</em></td>
<td>NP(Accusative)</td>
<td><em>Der Ober hat eine Speise gebracht</em></td>
</tr>
<tr>
<td><em>bringen</em></td>
<td>NP(Dative) + NP(Accusative)</td>
<td><em>Der Ober hat dem Kunde eine Speise gebracht</em></td>
</tr>
<tr>
<td><em>abhängen</em></td>
<td>von + NP(Dative)</td>
<td><em>Es hängt vom Ober ab</em></td>
</tr>
<tr>
<td><em>warten</em></td>
<td>auf + S</td>
<td><em>Er wartete auf dem Ober, die Speise zu bringen</em></td>
</tr>
<tr>
<td><em>fortsetzen</em></td>
<td>NP</td>
<td><em>Er hat die Arbeit fortgesetzt</em></td>
</tr>
<tr>
<td><em>wissen</em></td>
<td>NP(Final verb)</td>
<td><em>Der Ober weiß, das der Kunde Fisch liebt</em></td>
</tr>
</tbody>
</table>
The dependency structure generally reflects the traditional syntactic representation. The links can be annotated with grammatical function labels. In a simple sentence, it corresponds to the subject and the object.

Probably a more natural description to tie syntax to semantics.
Adjuncts form another class of functions that modify the verb. They include prepositional phrases whose head is set arbitrarily to the front preposition. Adjuncts include adverbs that modify a verb.
Dependency Parse Tree

Words: <root> Bring the meal to the table
Index: 0 1 2 3 4 5 6

<table>
<thead>
<tr>
<th>Word pos.</th>
<th>Word</th>
<th>Direction</th>
<th>Head</th>
<th>Head position</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bring</td>
<td>*</td>
<td>Root</td>
<td>1</td>
<td>Main verb</td>
</tr>
<tr>
<td>2</td>
<td>the</td>
<td>&gt;</td>
<td>meal</td>
<td>3</td>
<td>Determiner</td>
</tr>
<tr>
<td>3</td>
<td>meal</td>
<td>&lt;</td>
<td>Bring</td>
<td>1</td>
<td>Object</td>
</tr>
<tr>
<td>4</td>
<td>to</td>
<td>&lt;</td>
<td>Bring</td>
<td>1</td>
<td>Location</td>
</tr>
<tr>
<td>5</td>
<td>the</td>
<td>&gt;</td>
<td>table</td>
<td>6</td>
<td>Determiner</td>
</tr>
<tr>
<td>6</td>
<td>table</td>
<td>&lt;</td>
<td>to</td>
<td>4</td>
<td>Prepositional complement</td>
</tr>
</tbody>
</table>
$D = \{ \langle \text{Head}(1), \text{Rel}(1) \rangle, \langle \text{Head}(2), \text{Rel}(2) \rangle, \ldots, \langle \text{Head}(n), \text{Rel}(n) \rangle \}$,

The representation of *Bring the meal to the table*:

$D = \{ \langle \text{root} \rangle, \langle \text{3, det} \rangle, \langle \text{1, object} \rangle, \langle \text{1, loc} \rangle, \langle \text{6, det} \rangle, \langle \text{4, pcomp} \rangle \}$,
<sentence id="24">
  <word id="1" form="Dessutom" postag="ab" head="2"
      deprel="ADV"/>
  <word id="2" form="höjs" postag="vb.prs.sfo" head="0"
      deprel=""/>
  <word id="3" form="åldergränsen" postag="nn.utr.sin.def.nom"
      head="2" deprel="SUB"/>
  <word id="4" form="till" postag="pp" head="2" deprel="ADV"/>
  <word id="5" form="18" postag="rg.nom" head="6" deprel="DET"/>
  <word id="6" form="år" postag="nn.neu.plu.ind.nom" head="4"
      deprel="PR"/>
  <word id="7" form="." postag="mad" head="2" deprel="IP"/>
</sentence>

TMALT XML is an extended annotation
The CoNLL shared tasks organize evaluations of machine-learning systems for natural language processing. They define formats to share data between participants.

1 Dessutom _ AB AB _ 2 +A _ _
2 höjs _ VV VV _ 0 ROOT _ _
3 åldergränsen _ NN NN _ 2 SS _ _
4 till _ PR PR _ 2 OA _ _
5 18 _ RO RO _ 6 DT _ _
6 år _ NN NN _ 4 PA _ _
7 . _ IP IP _ 2 IP _ _
<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Token index, starting at 1 for each sentence.</td>
</tr>
<tr>
<td>2</td>
<td>FORM</td>
<td>Word form or punctuation.</td>
</tr>
<tr>
<td>3</td>
<td>LEMMA</td>
<td>Lemma or stem.</td>
</tr>
<tr>
<td>4</td>
<td>CPOSTAG</td>
<td>Part-of-speech tag.</td>
</tr>
<tr>
<td>5</td>
<td>POSTAG</td>
<td>Fine-grained part-of-speech tag.</td>
</tr>
<tr>
<td>6</td>
<td>FEATS</td>
<td>Unordered set of morphological features separated by a vertical bar (</td>
</tr>
<tr>
<td>7</td>
<td>HEAD</td>
<td>Head of the current token, which is either a value of ID or zero (0) if</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this is the root.</td>
</tr>
<tr>
<td>8</td>
<td>DEPREL</td>
<td>Dependency relation to the HEAD.</td>
</tr>
<tr>
<td>9</td>
<td>PHEAD</td>
<td>Projective head of current token, which is either a value of ID or zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0). The dependency structure resulting from the PHEAD column is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>guaranteed to be projective, when available in the corpus.</td>
</tr>
<tr>
<td>10</td>
<td>PDEPREL</td>
<td>Dependency relation to the PHEAD.</td>
</tr>
</tbody>
</table>
Visualizing Dependencies

Using *What’s Wrong With My NLP* (https://code.google.com/p/p/whatswrong/):

![Diagram of dependencies](image-url)
## Function Annotation Tagset (Järvinen and Tapanainen 1997)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main functions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>main</td>
<td>Main element</td>
<td><em>He doesn’t know whether to send a gift</em></td>
</tr>
<tr>
<td>qtag</td>
<td>Question tag</td>
<td><em>Let’s play another game, shall we?</em></td>
</tr>
<tr>
<td><strong>Intranuclear links</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v-ch</td>
<td>Verb chain</td>
<td><em>It may have been being examined</em></td>
</tr>
<tr>
<td>pcomp</td>
<td>Prepositional complement</td>
<td><em>They played the game in a different way</em></td>
</tr>
<tr>
<td>phr</td>
<td>Verb particle</td>
<td><em>He asked me who would look after the baby</em></td>
</tr>
</tbody>
</table>
Function Annotation Tagset (Järvinen and Tapanainen 1997)

### Verb complementation

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>subj</td>
<td>Subject</td>
<td>I gave him my <strong>address</strong></td>
</tr>
<tr>
<td>obj</td>
<td>Object</td>
<td>It has become <strong>marginal</strong></td>
</tr>
<tr>
<td>comp</td>
<td>Subject complement</td>
<td>Pauline gave it <strong>to Tom</strong></td>
</tr>
<tr>
<td>dat</td>
<td>Indirect object</td>
<td>His friends call him <strong>Ted</strong></td>
</tr>
<tr>
<td>oc</td>
<td>Object complement</td>
<td>We took a swim <strong>naked</strong></td>
</tr>
<tr>
<td>copred</td>
<td>Copredicative</td>
<td><strong>Play it again, Sam</strong></td>
</tr>
</tbody>
</table>

### Determinative functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>qn</td>
<td>Quantifier</td>
<td>I want <strong>more</strong> money</td>
</tr>
<tr>
<td>det</td>
<td>Determiner</td>
<td><strong>Other</strong> members will join...</td>
</tr>
<tr>
<td>neg</td>
<td>Negator</td>
<td><strong>It is not</strong> coffee that I like, but tea</td>
</tr>
</tbody>
</table>
**Function Annotation Tagset (Järvinen and Tapanainen 1997)**

<table>
<thead>
<tr>
<th>Type</th>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modifiers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>attr</td>
<td>Attributive nominal</td>
<td>Knowing no French, I couldn’t express my thanks</td>
</tr>
<tr>
<td>mod</td>
<td>Other postmodifiers</td>
<td>The baby, Frances Bean, was...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The people on the bus were singing</td>
</tr>
<tr>
<td>ad</td>
<td>Attributive adverbial</td>
<td>She is more popular</td>
</tr>
<tr>
<td><strong>Junctives</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cc</td>
<td>Coordination</td>
<td>Two or more cars...</td>
</tr>
</tbody>
</table>
Constituency (most textbooks) is a declining formalism
It cannot properly handle many languages: Swedish, Russian, Czech, Arabic, etc.
Dependency parsing can handle all these languages as well as English, German, French, etc.
Dependency parsing has improved considerably over the last 4 years: see CoNLL 2006 and 2007.
CoNLL 2008 and 2009 extend it to semantic parsing
However, constituency and dependency are (weakly) compatible provided that we restrict us to projective dependency graphs
It is possible to convert constituent trees into dependency graphs.
We need to identify a headword in all the PS rules, here with a star:

\[
\begin{align*}
    s &\rightarrow \text{np, vp*}. \\
    \text{vp} &\rightarrow \text{verb*, np}. \\
    \text{np} &\rightarrow \text{det, noun*}.
\end{align*}
\]

Parsers by Magerman and Collins used this to convert the Penn Treebank constituent annotation for their dependency parsers.
When projective, dependency structures are loosely compatible with constituent grammars.
A constituent tree with head-marked rules:

```
VP*
  |   |   |
  NP  Noun* Verb* NP
  |   |   |   |
  Det boy hit Det Noun*
  |     |     |     |
  The boy hit the ball
```

The resulting dependency graph:

```
The boy hit the ball
```