Human Communication 1
Lecture 7

What type?

S → aS

S → aABb

aAb → Aa

Aa → a

B → Bb

Bb → b

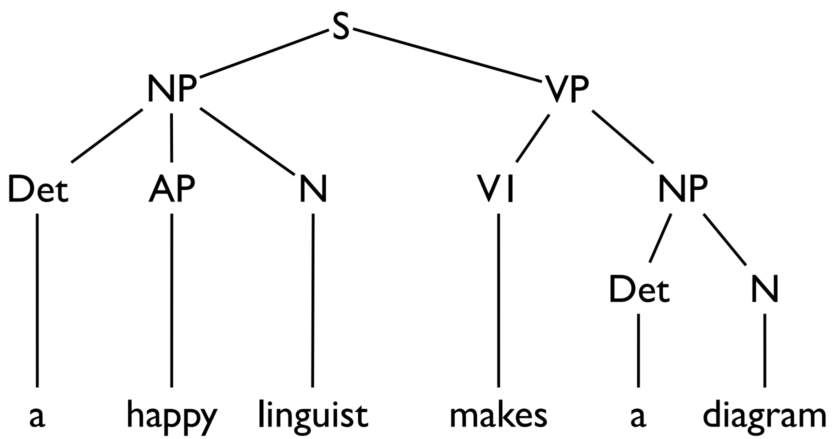
A theoretical ideal: Modularity

- The approaches we have seen have one appealing property: modularity
- Syntactic and semantic processing are separated, i.e. semantic information is not used in syntactic processing
- In other words: syntactic processing is isolated from semantic processing

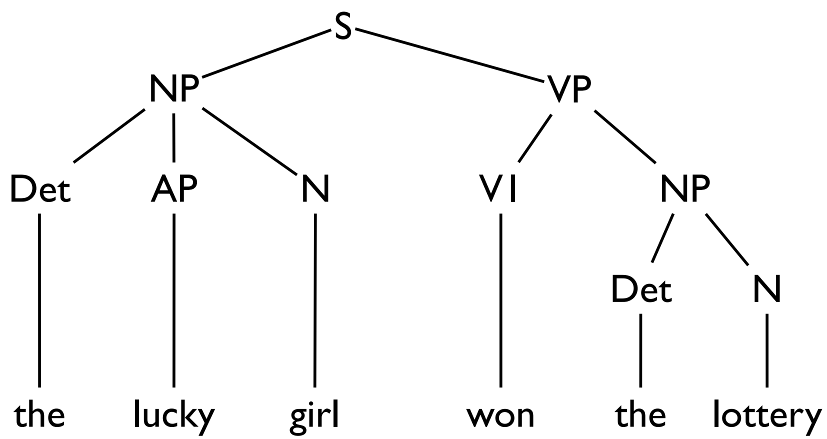
Example

- *A happy linguist makes a diagram.*
- *The lucky girl won the lottery.*

A happy linguist makes a diagram



The lucky girl won the lottery



Reminder: levels of analysis

Pragmatics	meaning in context
Semantics	sentence meaning
Syntax	sentence structure
Morphology	word structure
Phonology	sound structure
Phonetics	speech sounds

Generalisation

- Two sentences
  - *A happy linguist makes a diagram.*
  - *The lucky girl won the lottery.*
- Different meaning (different on semantic level)
- But: syntactic structure is identical

A theoretical ideal: Modularity

- Modularity makes it easier
  - to create a parsing algorithm
  - to predict the behaviour of the algorithm
- But: this is not quite realistic as a model of human language

Some standard questions

- How can we apply models of the kind we've seen so far for the processing of human language?
- What is the current engineering practice? (computational linguistics)
- What do we learn about humans? (cognitive science, psycholinguistics)

Example grammar

NP → Det N	N → tree
Det → a	N → trees
Det → an	N → tree's
Det → the	N → ox
	N → ox's
	N → oxen

Example grammar

NP → Det N	N → tree
Det → a	N → trees
Det → an	N → tree's
Det → the	N → ox
	N → ox's
	N → oxen

"a trees" is in the language

Example grammar

NP → $Det_{singular}$ $N_{singular}$	$N_{singular}$ → tree
NP → Det_{plural} N_{plural}	N_{plural} → trees
$Det_{singular}$ → a	$N_{singular}$ → tree's
$Det_{singular}$ → an	$N_{singular}$ → ox
$Det_{singular}$ → the	$N_{singular}$ → ox's
Det_{plural} → the	N_{plural} → oxen

Generalisation

- N → tree ⇒ $N_{singular}$ → tree
- N → trees ⇒ N_{plural} → trees
- Problem: $N_{singular}$ and N_{plural} do not capture the generalisation except as a mnemonic

Attribute grammars

- Instead of a subscript that creates two distinct symbols, attribute grammars use a complex symbol with an attribute–value list
  - N[num=singular] → ox
  - N[num=plural] → oxen
- These attribute–value lists are also called *feature structures*

Definition

"An attribute grammar is a formal way to define attributes for the productions of a formal grammar, associating these attributes to values. The evaluation occurs in the nodes of the abstract syntax tree, when the language is processed by some parser or compiler."

http://en.wikipedia.org/wiki/Attribute_grammar

Attribute Grammar

- Context free grammar
- Every nonterminal has attributes
- Attributes are either synthesised or inherited
- Attribute–value list defines values of attributes

Attributes

- Synthesised attributes pass information up the parse tree (if attribute from left hand side is computed from attribute in the right hand side)
- Inherited attributes pass information down the parse tree or from left siblings to the right siblings
- Attribute values are assumed to be available from the context

Example

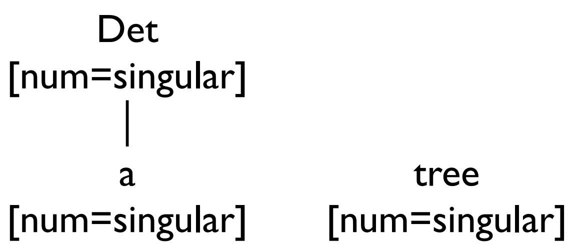
a tree

Example

a	tree
[num=singular]	[num=singular]

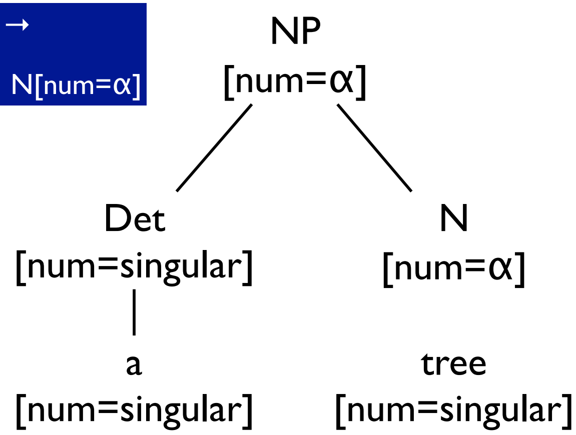
Example

Det[num=singular] →
a[num=singular]



Example

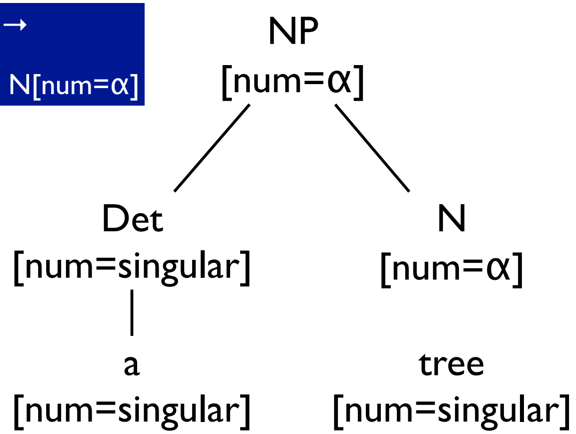
NP[num=α] →
Det[num=α] N[num=α]



Example

NP[num=α] →
Det[num=α] N[num=α]

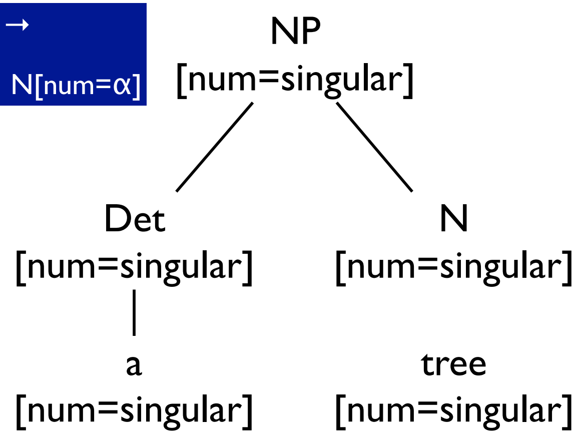
α=singular



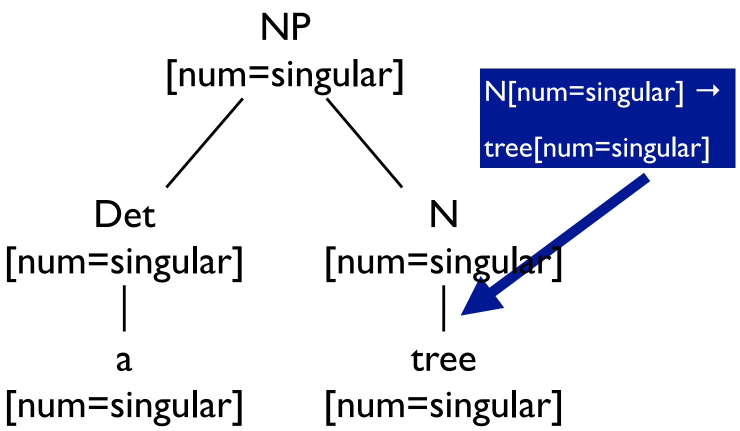
Example

NP[num=α] →
Det[num=α] N[num=α]

α=singular



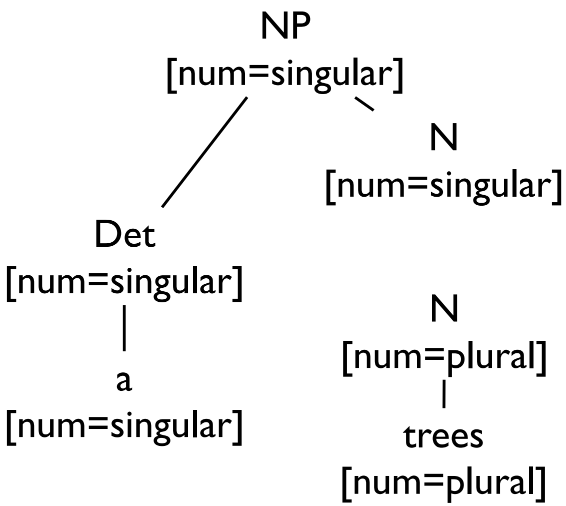
Example



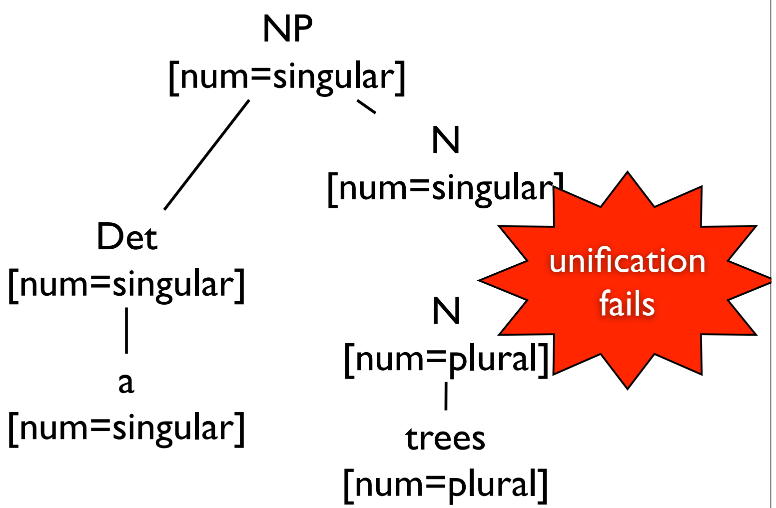
Unification

- The last step of the example used *unification* of two feature structures: [num=singular]
- tree[num=singular] and N[num=singular] both have the same feature structure, so they can be unified

Example



Example



Unification

- If two feature structure are identical, they can be unified
- If two feature structure are incompatible, unification is not possible (and the parse fails)

Example

N
[num=singular]

N
[num=singular, case=genitive]
|
tree's
[num=singular, case=genitive]

Example

N
[num=singular, case=genitive]

N
[num=singular, case=genitive]
|
tree's
[num=singular, case=genitive]

Unification

- If two feature structure are incompatible, unification is not possible (and the parse fails)
- If two feature structure are compatible (including identical), they can be unified
- The further down in the tree a node, the more specific (longer) its feature structure