

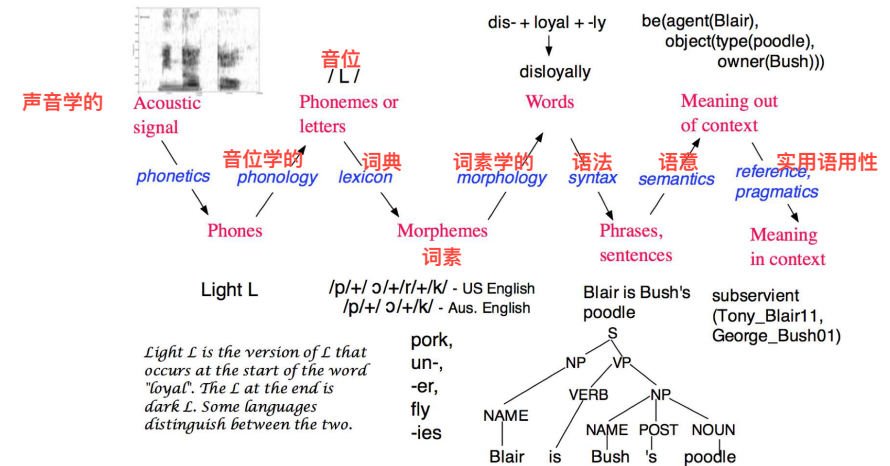
# COMP9414: Artificial Intelligence

## Lecture 7a: Grammars and Parsing

Wayne Wobcke

e-mail: w.wobcke@unsw.edu.au

## Linguistics Landscape



## This Lecture

- Overview of Natural Languages
- Syntax and Grammar for Natural Languages
- (Simple) Semantics for Natural Languages
- Pragmatics for Natural Languages

实用功能

## Natural Language Processing

- Syntax
  - Linguistic Knowledge
  - Grammars and Parsing
  - Probabilistic Parsing
- Semantics
  - Semantic Interpretation and Logical Form
- Pragmatics
  - Discourse Processing
  - Speech Act Theory
- 演说 (Spoken) Dialogue Systems

## Related Disciplines

- Linguistics
  - ▶ Study of language in the abstract and particular languages
- Psycholinguistics
  - ▶ Psychological models of human language processing
- Neurolinguistics
  - ▶ Neural models of human language processing
- Logic
  - ▶ Study of formal reasoning

## Central Problem – Ambiguity

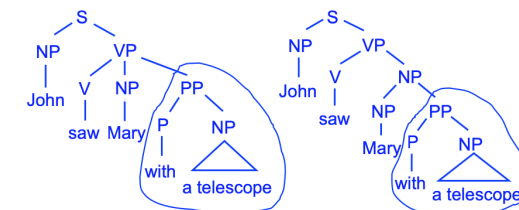
- Natural languages exhibit **ambiguity**
  - “The fisherman went to the bank” (lexical)
  - “The boy saw a girl with a telescope” (structural)
  - “Every student took an exam” (semantic)
  - “The table won’t fit through the doorway because it is too [wide/narrow]” (pragmatic)
- Ambiguity makes it difficult to interpret meaning of phrases/sentences
  - ▶ But also makes inference harder to define and compute
- **Resolve** ambiguity by mapping to unambiguous representation

## NLP Applications

- Chatbots
  - ▶ Customer service, e.g. CBA, Amtrak, Lyft, Spotify, Whole Foods
- Personal Assistants
  - ▶ Siri, Alexa, Google Assistant
- Information Extraction
  - ▶ Financial reports, news articles
- Machine (Assisted) Translation
  - ▶ Weather reports, EU contracts, Canada Hansard
- Social Robotics
  - ▶ Home care robots

## Structural Ambiguity

“John saw Mary with a telescope”



- Different interpretation → different representation

“John sold a car to Mary” and “Mary was sold a car by John”

- Same interpretation → same representation

## Syntax

- Linguistic Knowledge and Grammars
- Context Free Grammars
- Parsing
  - ▶ Top Down Parsing
  - ▶ Bottom Up Parsing
  - ▶ Chart Parsing
  - ▶ Deterministic Parsing
  - ▶ Probabilistic Parsing

## Methodology

- 自主 (性) Autonomy of Syntax
  - John promised to work
  - \*John persuaded to work
- vs
  - \*John was promised to work (by someone else)
  - John was persuaded to work (by someone else)
- Shows ‘promise’ and ‘persuade’ have different properties

\* means ungrammatical

## Framework (Chomsky)

- descriptive vs prescriptive 规范性
  - ▶ Goal not to dictate use of language, but describe how language, especially spoken language, is actually used
- sentence vs utterance 说话方式
  - ▶ Consider sentences (as an abstraction over utterances)
- competence vs performance
  - ▶ Focus on underlying linguistic knowledge
- descriptive vs explanatory adequacy
  - ▶ Aim to explain how linguistic knowledge is acquired

## Lexical Items (Basic Words)

- Open class
  - ▶ Nouns: denote objects (e.g. cat, John, justice)
  - ▶ Verbs: denote actions, events (e.g. buy, break, believe)
  - ▶ Adjectives: denote properties of objects (e.g. red, large)
  - ▶ Adverbs: denote properties of events (e.g. quickly)
- Closed class (function words)
  - 介词 ▶ Prepositions: at, in, of, on, . . .
  - ▶ Articles: the, a, an
  - ▶ Conjunctions: and, or, if, then, than, . . .

## Sentence Forms

- Declarative (indicative)
  - ▶ Bart is listening.
- Yes/No question (interrogative)
  - ▶ Is Bart listening?
- Wh-question (interrogative)
  - ▶ When is Bart listening?
- Imperative
  - ▶ Listen, Bart!
- Subjunctive
  - ▶ If Bart were listening, he might hear something useful.

## Verb Phrases

- Distribution over sentences
  - ▶ Verb phrases: occur as “predicate” with a range of “subjects”
    - John ⟨verb phrase⟩
    - The dog ⟨verb phrase⟩
    - Any noun phrase ⟨verb phrase⟩
  - ▶ Examples
    - . . .
- Notice ⟨verb phrase⟩ depends on ⟨noun phrase⟩

## Noun Phrases

- Distribution over sentences
  - ▶ Noun phrases: occur as “subject” with a range of “predicates”
    - ⟨noun phrase⟩ ate the bone
    - ⟨noun phrase⟩ saw the bird in the sky
    - ⟨noun phrase⟩ believes that  $2 + 2 = 4$
  - ▶ Examples
    - John, The dog, The big ugly dog,
    - The man in the red car,
    - The oldest man in the world with a beard,
    - The oldest man who lives in China, . . .
- Sentences need not “make sense”

## Inside Noun Phrases

- Within noun phrase
  - ▶ Main item (the **head** of the phrase): noun
  - ▶ Optional **specifiers**
    - Determiners (articles, demonstratives, quantifiers)
    - Adjectives and other nouns
  - ▶ Mandatory **arguments**
    - Depend on head (e.g. capital ⟨of France⟩)
  - ▶ Optional **modifiers**
    - Adjectival phrases (e.g. larger than Spain)
    - Prepositional phrases (e.g. in the park)
    - Relative clauses (e.g. who likes beer)
  - ▶ Order specifiers, head, modifiers in English

## Inside Verb Phrases

- Within verb phrase
  - ▶ Main item (the **head** of the phrase): verb
  - ▶ Optional **specifiers**
    - Auxiliary verbs (e.g. do, does, will, might, . . .)
    - Adverbs (e.g. quickly)
  - ▶ Mandatory **arguments**
    - depend on head (e.g. bought ⟨a book⟩ ⟨for Henry⟩)
  - ▶ Optional **modifiers**
    - Adverbial phrases (e.g. more quickly than Henry)
  - ▶ Notice similar structure to noun phrases

## Context Free Grammars

- Nonterminal symbols (grammatical categories)
- Terminal symbols (lexical items)
- Start symbol (a nonterminal) e.g. ⟨sentence⟩
- Rewrite rules
  - ▶ nonterminal → sequence of nonterminals, terminals  
e.g. ⟨sentence⟩ → ⟨noun phrase⟩ ⟨verb phrase⟩
- Open question: is English context free?

## Prepositional Phrases

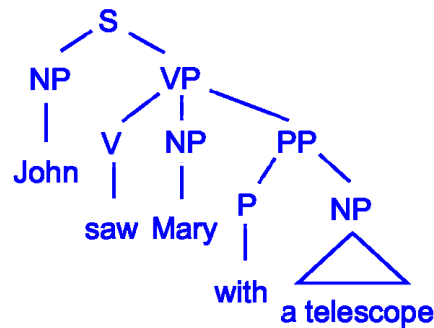
- Within prepositional phrase
  - ▶ Main item (the **head** of the phrase): preposition
  - ▶ Mandatory **arguments**
    - ⟨noun phrase⟩ (e.g. in the park)
- Nouns, verbs, etc. are just the heads of phrases

## Typical (Small) Grammar

$S \rightarrow NP VP$   
 $NP \rightarrow [Det] Adj^* N [AP \mid PP \mid Rel\ Clause]^*$   
 $VP \rightarrow V [NP] [NP] PP^*$   
 $AP \rightarrow Adj PP$   
 $PP \rightarrow P NP$   
 $Det \rightarrow a \mid an \mid the \mid \dots$   
 $N \rightarrow John \mid park \mid telescope \mid \dots$   
 $V \rightarrow saw \mid likes \mid believes \mid \dots$   
 $Adj \rightarrow hot \mid hotter \mid \dots$   
 $P \rightarrow in \mid \dots$

Special notation: \* is “0 or more”; [ . . . ] is “optional”

## Syntactic Structure



Syntactically ambiguous = more than one parse tree

## Rightmost Derivation

```

S
⇒ NP VP
⇒ NP V NP PP
⇒ NP V NP P NP
⇒ NP V NP P Det N
⇒ NP V NP P Det telescope
⇒ NP V NP P a telescope
⇒ ...
⇒ ...
⇒ ...
  
```

## (Leftmost) Derivation of Example

```

S
⇒ NP VP
⇒ N VP
⇒ John VP
⇒ John V NP PP
⇒ John saw NP PP
⇒ John saw N PP
⇒ John saw Mary PP
⇒ John saw Mary P NP
⇒ John saw Mary with NP
⇒ John saw Mary with Det N
⇒ John saw Mary with a N
⇒ John saw Mary with a telescope
  
```

⇒ means “rewrites as”

## Parsing

- Aim is to compute a derivation of a sentence (hence tree)
- Methods
  - ▶ Top down
    - Start with S, apply rewrite rules until sentence reached
  - ▶ Bottom up
    - Start with sentence, apply rewrite rules “in reverse” until S is reached
  - ▶ Chart parsing
    - Chart records parsed fragments and hypotheses
    - Can mix top down and bottom up strategies

## Top-Down Parsing

- Use a stack to record working hypothesis
- Start with S as only symbol on stack
- At each step
  - ▶ Rewrite top of stack T using grammar rule  $T \rightarrow \text{RHS}$  i.e. replace T by RHS (in reverse order), **OR**
  - ▶ Match word on top of stack to next word in sentence
- Apply backtracking on failure
- Accept sentence when stack is empty and all words in sentence matched; reject sentence when no rules to try
- Produces leftmost derivation

## Bottom Up Parsing

- Use a stack to record parsed (left-right) fragment
- Start with stack empty
- At each step
  - ▶ Rewrite sequence at top of stack using rule  $T \rightarrow \text{RHS}$  i.e. replace RHS (in reverse) by T, **OR**
  - ▶ Move word from input to stack
- Apply backtracking on failure
- Accept sentence when input empty and stack contains S; reject sentence when no more rules to try
- Produces rightmost derivation (in reverse)

## Example

STACK	INPUT
S	John saw Mary with a telescope
VP NP	John saw Mary with a telescope
VP N	John saw Mary with a telescope
VP John	John saw Mary with a telescope
VP	saw Mary with a telescope
PP NP V	saw Mary with a telescope
PP NP saw	saw Mary with a telescope
PP NP	Mary with a telescope
...	...
...	...

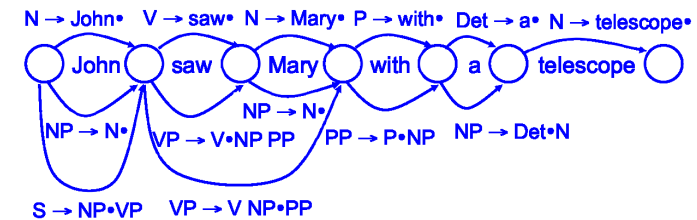
## Example

STACK	INPUT
	John saw Mary with a telescope
John	saw Mary with a telescope
N	saw Mary with a telescope
NP	saw Mary with a telescope
NP saw	Mary with a telescope
NP V	Mary with a telescope
NP V Mary	with a telescope
NP V N	with a telescope
...	...
...	...

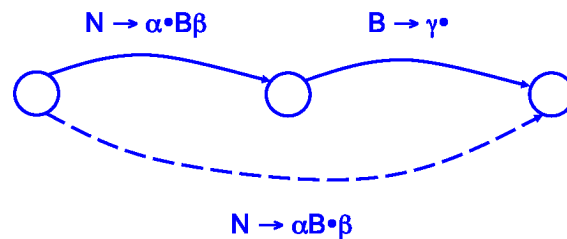
## Chart Parsing

- Use a **chart** to record parsed fragments and hypotheses
- Hypotheses  $N \rightarrow \alpha \bullet \beta$  where  $N \rightarrow \alpha \beta$  is a grammar rule **means** “trying to parse  $N$  as  $\alpha \beta$  and have so far parsed  $\alpha$ ”
- One node in chart for each word gap, start and end
- One arc in chart for each hypothesis
- At each step, apply **fundamental rule**
  - ▶ If chart has  $N \rightarrow \alpha \bullet B \beta$  from  $n_1$  to  $n_2$  and  $B \rightarrow \gamma \bullet$  from  $n_2$  to  $n_3$  add  $N \rightarrow \alpha B \bullet \beta$  from  $n_1$  to  $n_3$
- Accept sentence when  $S \rightarrow \alpha \bullet$  is added from start to end
- Can produce any sort of derivation

## Example Chart



## Fundamental Rule



## Top Down Chart Parsing

- Start with  $S \rightarrow \bullet \alpha$  from start node to start node for all rules  $S \rightarrow \alpha$  where  $S$  is start symbol
- When adding  $N \rightarrow \alpha \bullet B \gamma$  from  $n_1$  to  $n_2$ 
  - ▶ Also add  $B \rightarrow \bullet \beta$  from  $n_2$  to  $n_2$  for each rule  $B \rightarrow \beta$
- Exercise: Trace top down chart parser on example



## Bottom Up Chart Parsing

- Start with arcs for each lexical item
- When adding  $C \rightarrow \alpha \bullet$  from  $n_1$  to  $n_2$ 
  - ▶ Also add  $B \rightarrow \bullet C \gamma$  from  $n_1$  to  $n_2$  for each rule  $B \rightarrow C \gamma$
- Exercise: Trace bottom up down chart parser on example

## Deterministic Parsing

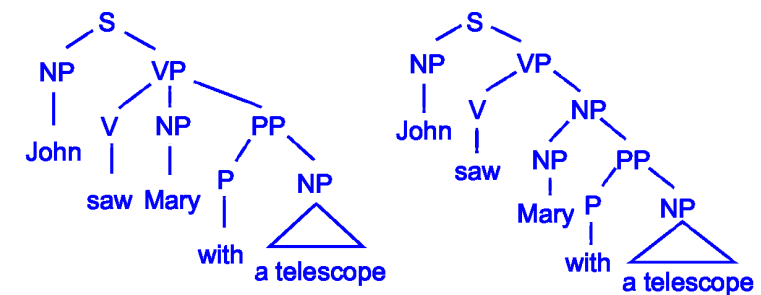
- Motivation
  - ▶ People don't notice ambiguity . . .
  - ▶ But sometimes have trouble
    - “The horse raced past the barn fell”
    - “We painted all the walls with cracks”
    - “The man kept the dog in the house”
- Can we do what the “human parser” does?

## Compare and Contrast

- Top Down Parsing
  - ▶ Simple, Memory efficient
  - ▶ Much repeated work, may loop infinitely
- Bottom Up Parsing
  - ▶ Less repeated work, harder to control
- Chart Parsing
  - ▶ Memory inefficient (especially with features)
  - ▶ No repeated work, difficult to control

## Heuristics

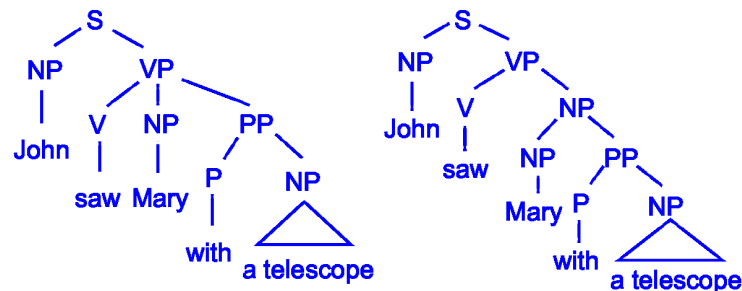
### Minimal Attachment



- Minimize size of parse tree

## Heuristics

### Right Association



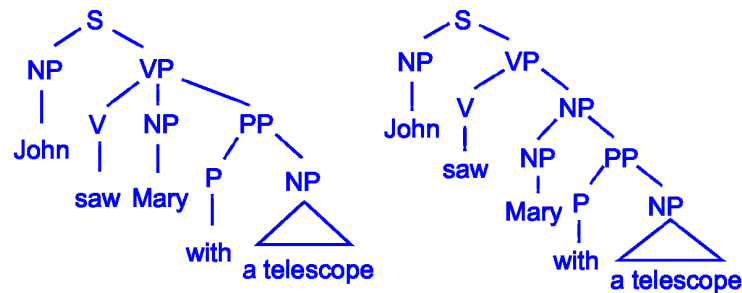
- Always attach to rightmost (lower) nodes

## Probabilistic Context Free Grammars

- Associate probabilities with grammar rules
  - Requires parsed corpus (e.g. Penn Treebank)
  - Count number of times rule used in parsing corpus sentences
- Probability of parse tree
  - $\Pi_r$  probability rule  $r$  \*  $\Pi_w$  probability of word  $w$  given category
  - Assuming independence (again)

## Heuristics

### Lexical Preference



- Try to fill most common subcategorization frame

## Probabilistic Chart Parsing

- Start with probabilities calculated by part of speech tagger
- Multiply probabilities when applying fundamental rule
- Best-First Chart Parsing
  - Examine most likely constituents first (priority queue)
    - Various ways to estimate these probabilities!
  - When adding  $A \rightarrow \alpha B \beta$ , try to extend to  $A \rightarrow \alpha B \beta^1 \beta^2$
  - Never constructs constituents with lower probability than parse

## Summary

---

- Syntactic Knowledge
  - ▶ Grammatical categories defined by distribution
  - ▶ Much determined by properties of lexical items
- Context Free Grammars
  - ▶ Useful and powerful formalism
  - ▶ Relatively efficient parsers
  - ▶ Limited when dealing with complex phenomena
- Parsing
  - ▶ Top down method is easy to understand, but not efficient
  - ▶ Bottom up method is more efficient