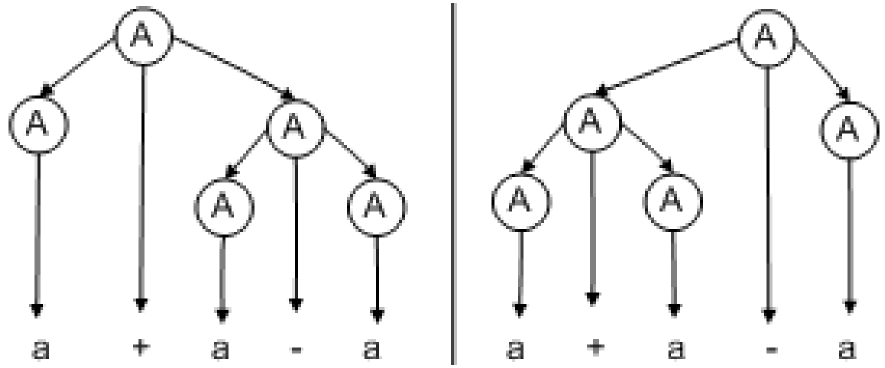
Syntax/Context Free Grammar

(Natural Language Processing)

Randil Pushpananda
rpn@ucsc.cmb.ac.lk



Grammar

Britannica Dictionary definition of GRAMMAR:

- The set of rules that explain how words are used in a language
- Speech or writing judged by how well it follows the rules of grammar

Syntax

- In linguistics, syntax is the set of rules, principles, and processes that control the structure of sentences in a given language, specifically the word order.

- To the movies we are going: any sense?

- Eats boy a the cookie: any sense?

Syntax

- In linguistics, syntax is the set of rules, principles, and processes that control the structure of sentences in a given language, specifically the word order.

- Incorrect - To the movies we are going.
- Correct - We are going to the movies.

- Incorrect - Eats boy a the cookie.
- Correct - The boy eats a cookie.

Why do we need Syntax?

- Languages are recursive
  - *recursion* is a phenomenon where a linguistic rule can be applied to the result of the application of the same rule.
    - S -> S and S
    - NP -> N NP

    - Ex:
      - Alex has a red car.
      - Alex, whom you know very well, has a red car.
      - Alex, whom you know very well, has a red car which is parked there.

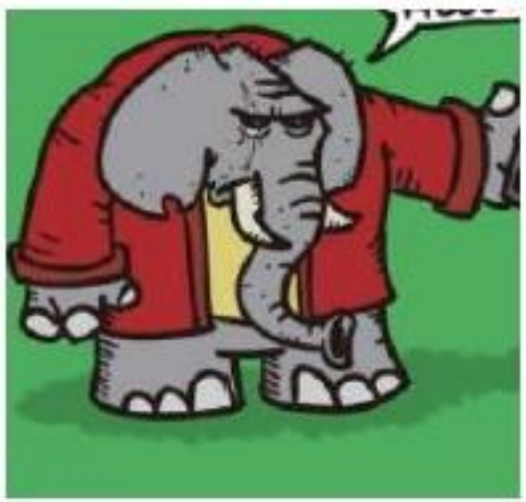
Why do we need Syntax?

- Languages are highly ambiguous

Why do we need Syntax?

- Languages are highly ambiguous

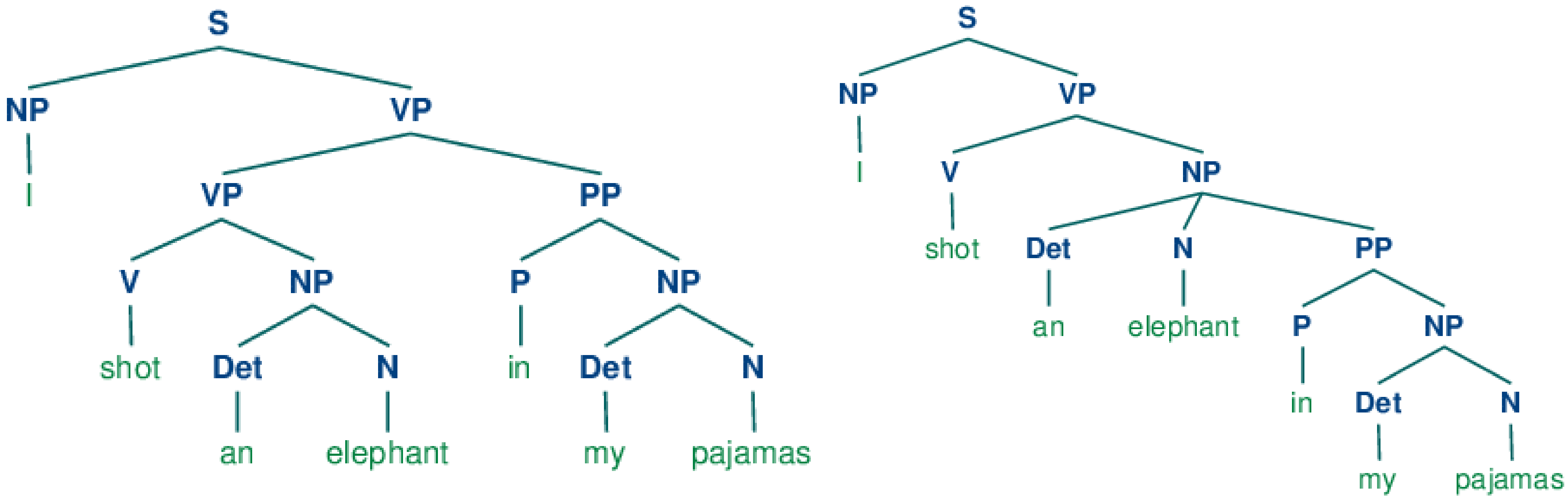
One morning in Africa,
I shot an elephant in my pajamas;
how he got into my pajamas I'll never know.



Famous joke by the American comedian Groucho Marx!

Why do we need Syntax?

- Languages are highly ambiguous

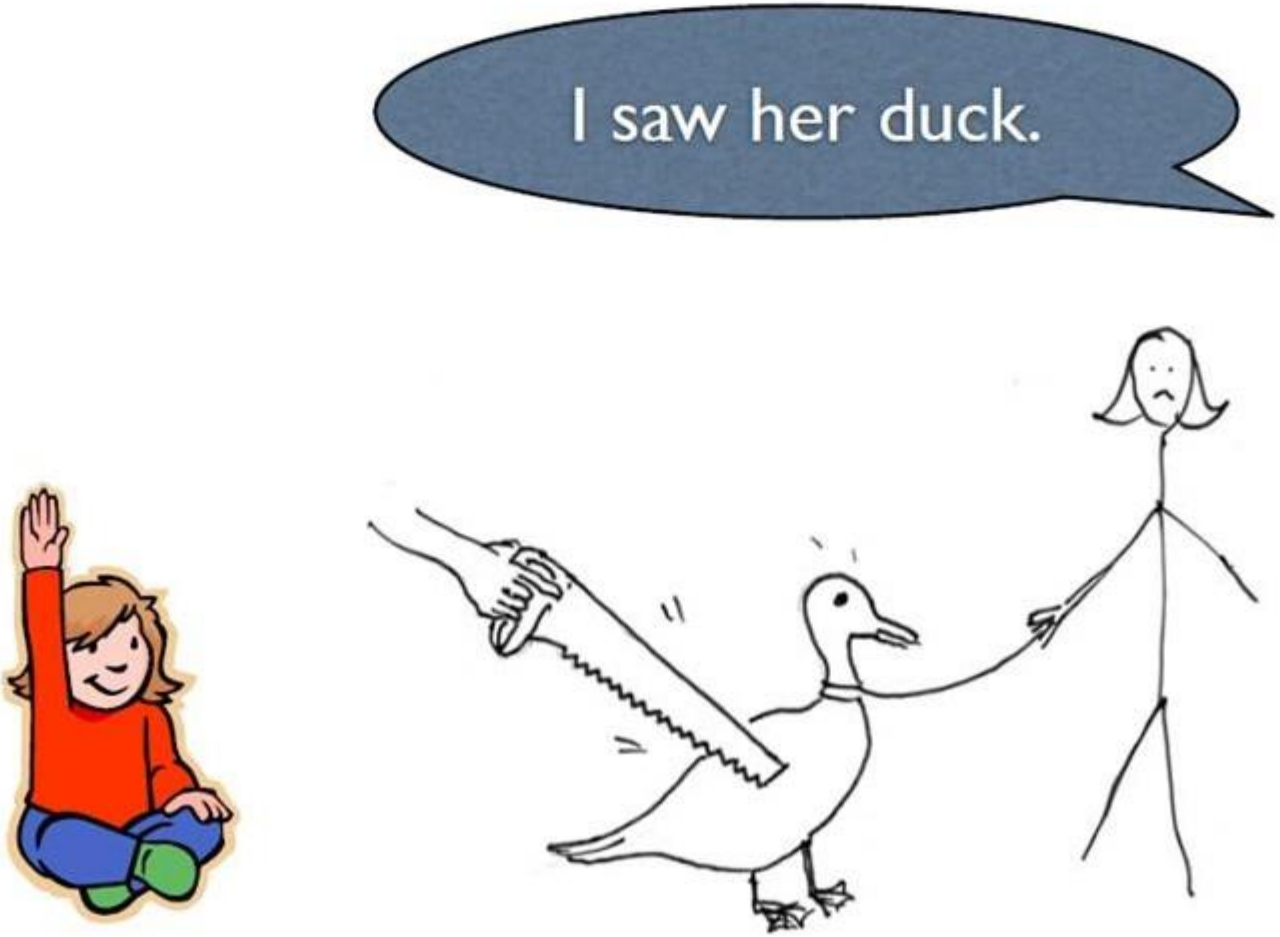


NLP is all about ambiguities

- to middle school kids: what does this sentence mean?

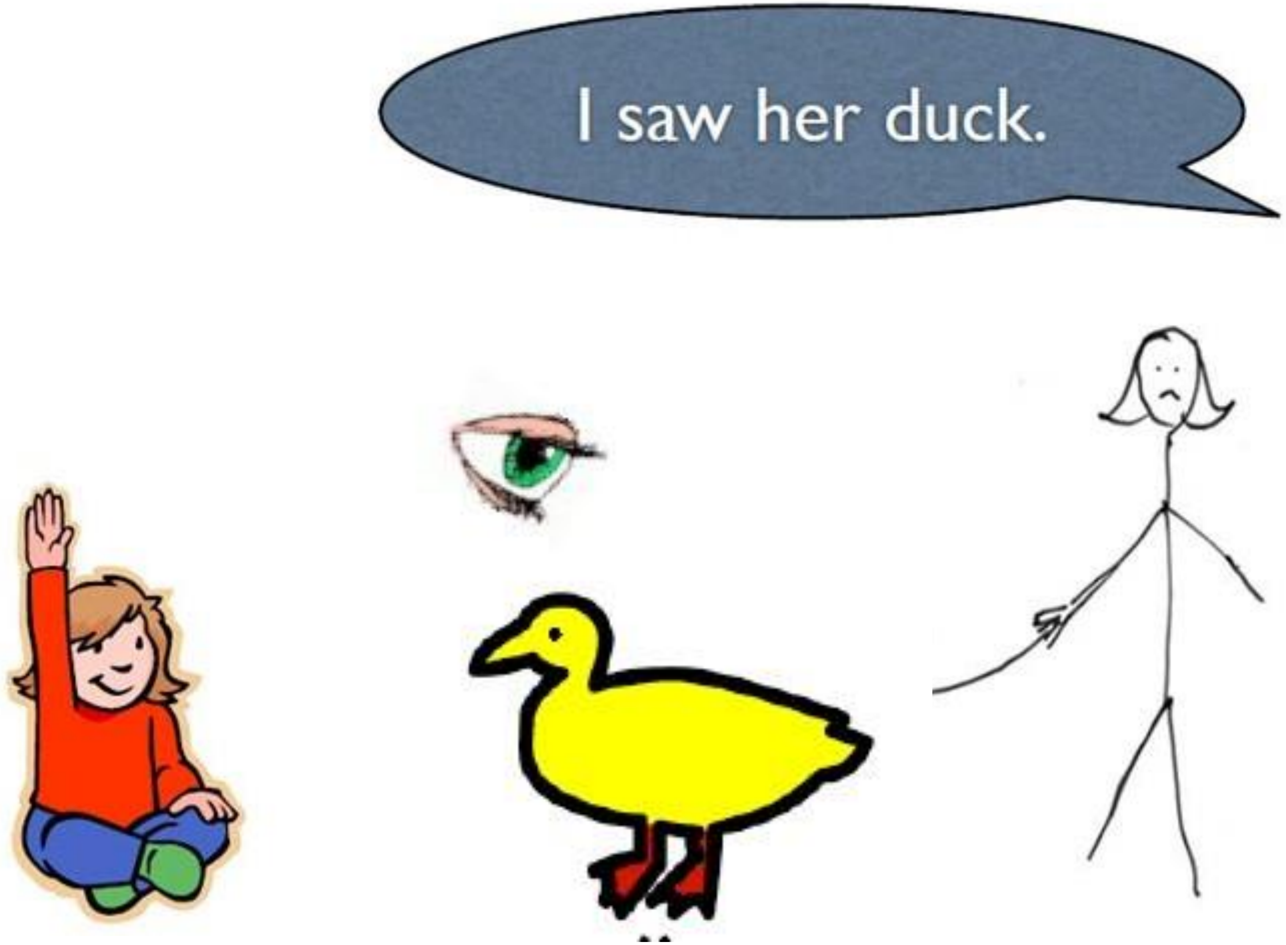
NLP is all about ambiguities

- to middle school kids: what does this sentence mean?



NLP is all about ambiguities

- to middle school kids: what does this sentence mean?



NLP is all about ambiguities

- I saw a man on a hill with a telescope.
  1. There's a man on a hill, and I'm watching him with my telescope.
  2. There's a man on a hill, who I'm seeing, and he has a telescope.
  3. There's a man, and he's on a hill that also has a telescope on it.
  4. I'm on a hill, and I saw a man using a telescope.
  5. There's a man on a hill, and I'm sawing him with a telescope.

Syntactic Analysis

- Syntax expresses the way in which words are arranged together.
- The kind of implicit knowledge of your native language that you had mastered by the time you were 3 or 4 years old without explicit instruction
  - Do these word sequences fit together?

    I saw you yesterday

    you yesterday I year

    colorless green ideas sleep furiously (Chomsky)

    furiously sleep ideas green colorless
- NLP uses syntax to produce a structural analysis of the input sentence

Context Free Grammars

- A *context-free grammar (CFG)* is a list of rules that define the set of all well-formed sentences in a language.
- Each rule has a left-hand side, which identifies a syntactic category, and a right-hand side, which defines its alternative component parts, reading from left to right.

$$S \rightarrow NP\ VP$$

Context Free Grammars

- Why Context-Free?
  - The notion of context in CFGs has nothing to do with the ordinary meaning of the word context in language.
  - All it really means is that the non-terminal on the left-hand side of a rule can be replaced regardless of context
    - Context-sensitive grammars allow context to be placed on the left-hand side of the rewrite rule
- In programming languages, and other uses of CFGs in Computer Science, notably XML, CFGS are
  - Unambiguous
    - Assign at most, 1 structural description to a string
  - Parsable in time linearly proportional to the length of the string

Context Free Grammars

- Capture constituency and ordering
  - Ordering is
    - What are the rules that govern the ordering of words and bigger units in the language
  - Constituency is
  - How do words group into units and what we say about how the various kinds of units behave
  - A constituent is a sequence of words that behave as a unit
    - John talked [to the children] [about drugs].
    - John talked [about drugs] [to the children].
    - *John talked drugs to the children about (random reorder)
  - Constituents can be expanded or substituted for:
    - I sat [on the box/right on top of the box/there]
  - Other properties: Coordination, regular internal structure, no intrusion, fragments, semantics, …

Context Free Grammar

- A Context-Free Grammar is a 4-tuple where

$$G = (N, \sum, R, S).$$

Context Free Grammar consists of:

- Non-terminal symbols

  S, NP, VP, etc. representing the constituents or categories of phrases

- Terminal symbols

  car, man, house, representing words in the lexicon

  - The rewrite rules will include lexical insertion rules (e.g. N = *car* | *man* | *house*)

- Rewrite rules / productions

  S → NP VP | VP

  (note use of | symbol to give alternate rhs of rules)

- A designated start symbol S

- A derivation is a sequence of rewrite rules applied to a string that exactly covers the items in that string

Derivation of Syntax from grammar rules

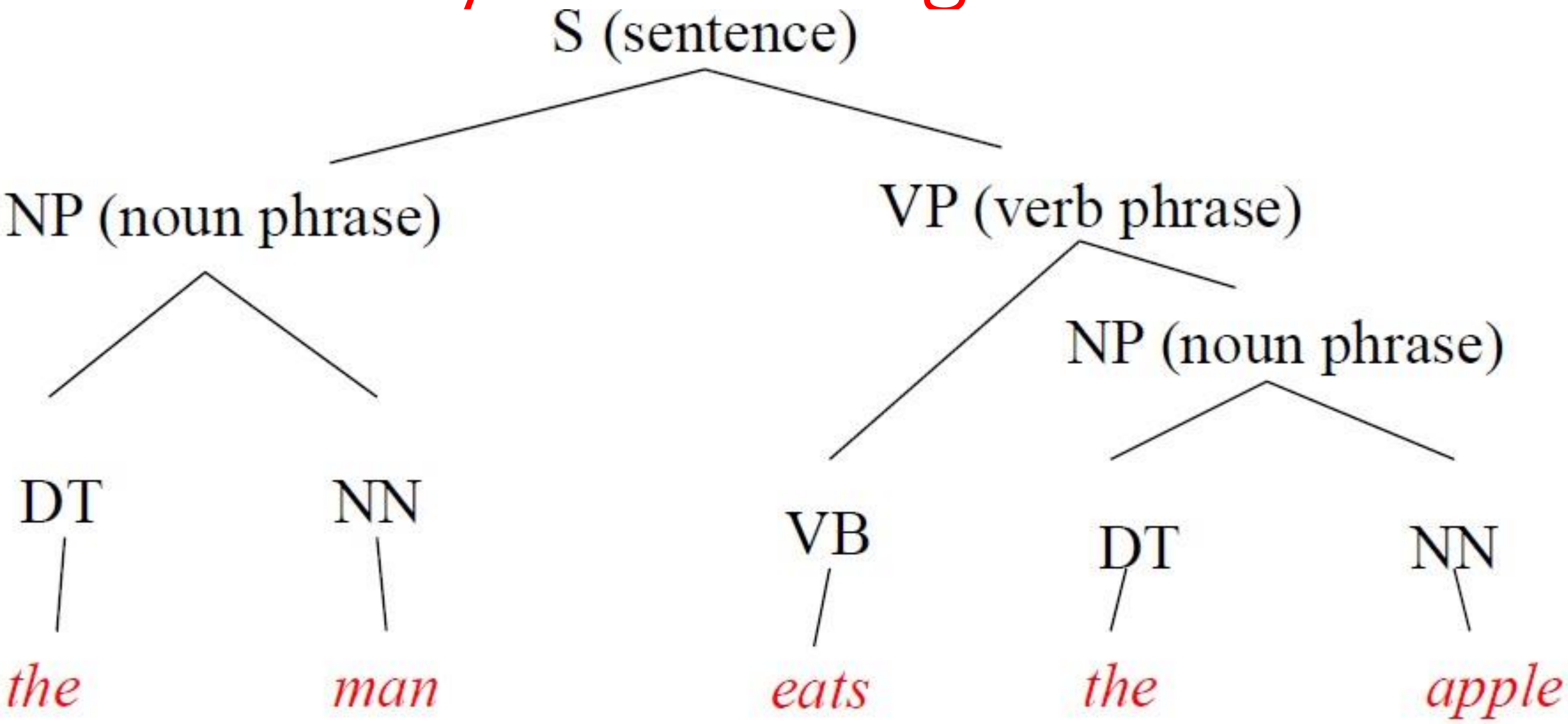
the *man* *eats* *the* *apple*

Context Free Grammar Rules:

S → NPVP
NP → DT NN
VP → VB NP
VP → VB

DT → *the* | ...
NN → *man* | apple | … (add words)
VB → *eats* | ...

Derivation of Syntax from grammar rules



Context Free Grammar Rules:

S → NPVP
NP → DT NN
VP → VB NP
VP → VB

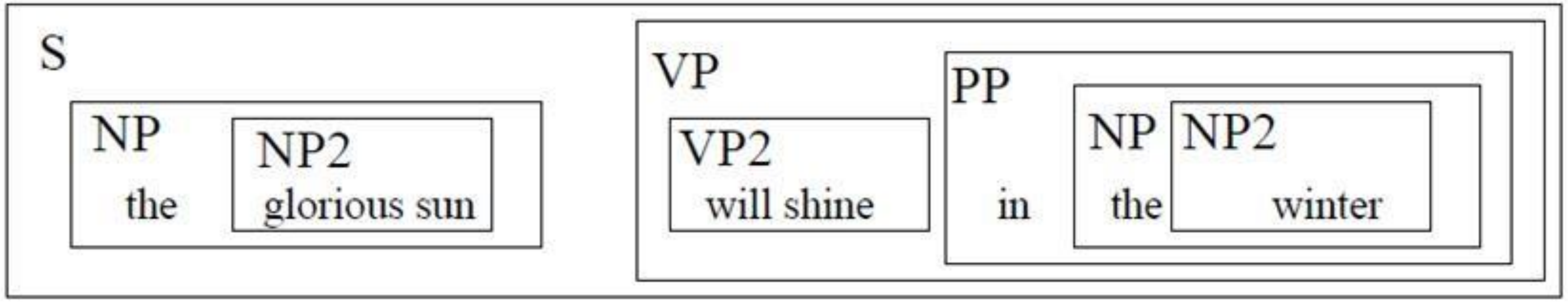
DT → *the* | ...
NN → *man* | apple | … (add words)
VB → *eats* | ...

Notations for (constituents) syntactic Structure

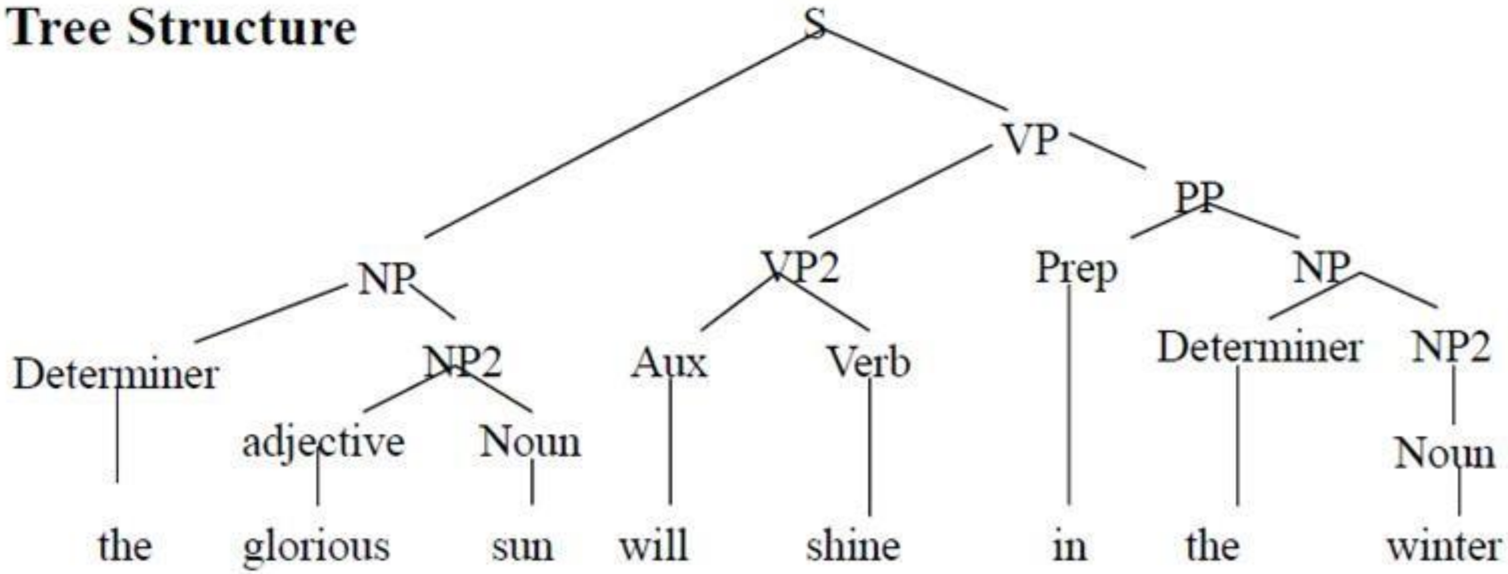
Bracketed text

[s[NP the [NP2 glorious sun]]   [VP [VP2 will shine] [PP in [NP the [NP2 winter]]]]]

Nested Boxes



Tree Structure



Generativity vs Parsing

- You can view these rules as either synthesis or analysis machines
  - Generate strings in the language
  - Reject strings not in the language
  - Impose structures (trees) on strings in the language
- The latter two are the analysis tasks of parsing
  - Parsing is the process of finding a derivation (i. e. sequence of productions) leading from the START symbol to a TERMINAL symbol (or TERMINALS to START symbol)
  - Shows how a particular sentence ***could be*** generated by the rules of the grammar
  - If sentence is structurally ambiguous, more than one possible derivation is produced

Key Constituents for English

- English has headed phrase structure
  - X-bar theory: in natural languages, phrases are headed by particular kinds of word that has modifiers and qualifiers around them (specifiers, adjuncts, and complements)
- Verb Phrases VP → … VB* …
- Noun Phrases NP → … NN* …
- Adjective Phrases ADJP → … JJ* …
- Adverb Phrases ADVP → … RB* …
- Sentences (and clauses): SBAR → S | SINV | SQ …
  - Sentences, inverted sentences, direct questions, … can also appear in larger clause structure SBAR where sentence is preceded by *that*
- Plus minor phrase types:
  - QP (quantifier phrase) in NP, PP (prepositional phrase), CONJP (multi word constructions: *as well as*), INTJ (interjections), etc.

Sentences

- Sentences
  - Declaratives: A plane left (Information, word order is subject then verb)

    S -> NP VP
  - Imperatives: Leave! (Give a command, Give Instructions)

    S -> VP
  - Yes-No Questions: Did the plane leave?

    S -> Aux NP VP
  - WH Questions: When did the plane leave?

    S -> WH Aux NP VP

Exercise

$S- > NP, VP$	$Adj- > angry \vert big \vert larger$
$VP- > Vbe, Adj$	$P- > at \vert on \vert under$
$NP- > Det, N$	$Det- > a \vert an \vert the$
$N- > Adj, N$	$Vbe- > is$
$Adj- > Adj, PP$	$N- > table \vert bull \vert snake$
$PP- > P, NP$	

Write down three (03) structurally different, grammatical sentences generated by this grammar.

Noun Phrases

- Noun phrases have a head noun with pre and post-modifiers
  - Determiners, Cardinals, Ordinals, Quantifiers and Adjective Phrases are all optional, indicated here with parentheses

    NP -> (DT) (Card) (Ord) (Quan) (AP) Noun

    Noun -> NN | NP | NPS | NNS
  - Post-modifiers include prepositional phrases, gerundive phrases, and relative clauses

    the man [from Moscow]

    any flights [arriving after 11pm] (gerundive)

    the spy[who came in from the cold] (relative clause)

Some examples on these slides are from the Jurafsky and Martin text and from Jim Martin's online course materials.

Recursive Rules

- One type of Noun phrase is a Noun Phrase followed by a Prepositional phrase

  NP -> NP PP

  PP -> Prep NP

- Of course, this is what makes syntax interesting

  flights from Denver

  flights from Denver to Miami

  flights from Denver to Miami in February

  flights from Denver to Miami in February on a Friday

  flights from Denver to Miami in February on a Friday under $300

  flights from Denver to Miami in February on a Friday under $300 with lunch

  – Syntax trees for these examples also need rules for NP -> Noun, etc.

Verb Phrases

- Simple Verb phrases

VP -> Verb	*leave*
| Verb NP	*leave Boston*
| Verb NP PP	*leave Boston in the morning*
| Verb PP	*leave in the morning*

- Verbs may also be followed by a clause

  VP -> Verb S

  I think I would like to take a 9:30 flight

- The phrase or clause following a verb is sometimes called the complementizer

Conjunctive Constructions

- S -> S and S
  - John went to NY and Mary followed him
- NP -> NP and NP
- VP -> VP and VP
- …
- In fact the right rule for English is

  X -> X and X

Problems

- Context-Free Grammars can represent many parts of natural language adequately
- Here are some of the problems that are difficult to represent in a CFG:
  - Agreement
  - Subcategorization
  - Movement (for want of a better term)

Agreement

- This dog
- Those dogs

- *This dogs
- *Those dog

- This dog eats
- Those dogs eat

- *This dog eat
- *Those dogs eats

- In English,
  - subjects and verbs have to agree in person and number
  - Determiners and nouns have to agree in number
- Many languages have agreement systems that are far more complex than this.
- Solution can be either to add rules with agreement or to have a layer on the grammar called the features

Subcategorization

- Subcategorization expresses the constraints that a particular verb (sometimes called the predicate) places on the number and syntactic types of arguments it wants to take (occur with).
  - Sneeze: John sneezed
  - Find: Please find [a flight to NY]$_{NP}$
  - Give: Give [me]$_{NP}$[a cheaper fare]$_{NP}$
  - Help: Can you help [me]$_{NP}$[with a flight]$_{PP}$
  - Prefer: I prefer [to leave earlier]$_{TO\text{-}VP}$
  - Told: I was told [United has a flight]$_{S}$

Subcategorization

- Should these be correct?
  - John sneezed the book
  - I prefer United has a flight
  - Give with a flight
- The various rules for VPs *overgenerate*.
  - They permit the presence of strings containing verbs and arguments that don't go together
  - For example VP -> V NP therefore

    Sneezed the book is a VP since "sneeze" is a verb and "the book" is a valid NP
- Now *overgeneration* is a problem for a generative approach.
  - The grammar should represent all and only the strings in a language
- From a practical point of view... not so clear that there's a problem

Movement

- Consider the verb "booked" in the following example:
  - [[My travel agent]$_{NP}$ [booked [the flight]$_{NP}$]$_{VP}$]$_{S}$
- i.e. "book" is a straightforward transitive verb. It expects a single NP arg within the VP as an argument, and a single NP arg as the subject.

Example

```
import nltk
import nltk.grammar


grammar1 = nltk.CFG.fromstring("""
    S -> NP VP
    VP -> V NP | V NP PP
    PP -> P NP
    NP -> "John" | "Mary" | "Bob" | Det N | Det N PP | P NP

    V -> "saw" | "ate" | "walked"
    Det -> "a" | "an" | "the" | "my"
    N -> "man" | "dog" | "cat" | "telescope" | "park"
    P -> "in" | "on" | "with" |"by"
    ProN -> "John" | "Mary" | "Bob"
    """)


#grammar1 = nltk.data.load('file:simple.cfg')
sent = "Mary saw Bob with the telescope".split()
rd_parser = nltk.RecursiveDescentParser(grammar1)
for tree in rd_parser.parse(sent):
    print tree

#NP -> Det N | Det N PP | P NP | ProN
#NP -> "John" | "Mary" | "Bob" | Det N | Det N PP | P NP
# Mary saw Bob with the telescope
```

(1) a. this dog

b. *these dog

(2) a. these dogs

b. *this dogs

(3) a. the dog runs

b. *the dog run

(4) a. the dogs run

b. *the dogs runs

```
S    ->   NP VP
NP   ->   Det N
VP   ->   V

Det   ->  'this'
N     ->  'dog'
V     ->  'runs'
```

Questions!

- **Consider the following fragment of English grammar.**
- $S \rightarrow NP\ VP$ — $D \rightarrow a \mid the$
- $NP \rightarrow D\ N$ — $N \rightarrow boy \mid rabbit \mid bird \mid cat \mid tree$
- $VP \rightarrow V \mid V\ NP \mid V\ PP$ — $V \rightarrow saw \mid gave \mid flew \mid ran$
- $PP \rightarrow P\ NP$ — $P \rightarrow with \mid into \mid from \mid at$

- What additional rule(s) would you include to accommodate the following sentences?

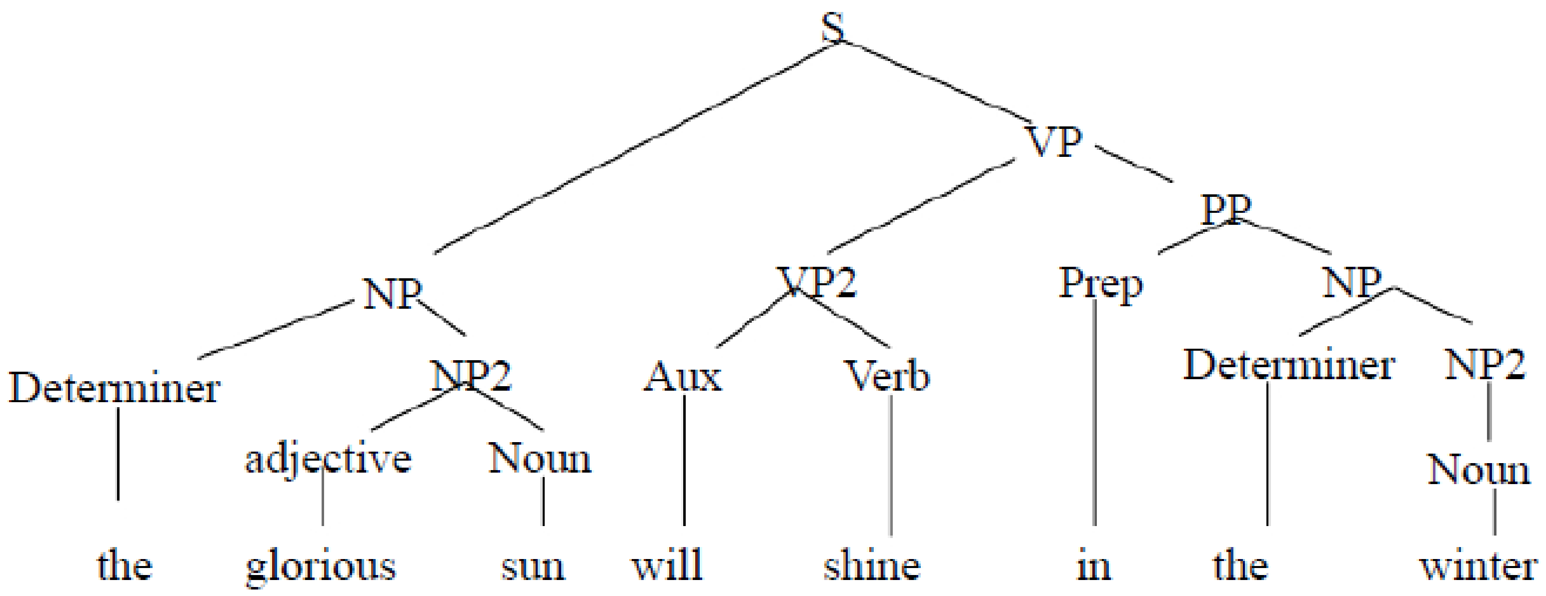
1. John saw Marry
2. The man said the dog chased the cat

Dependency Grammars

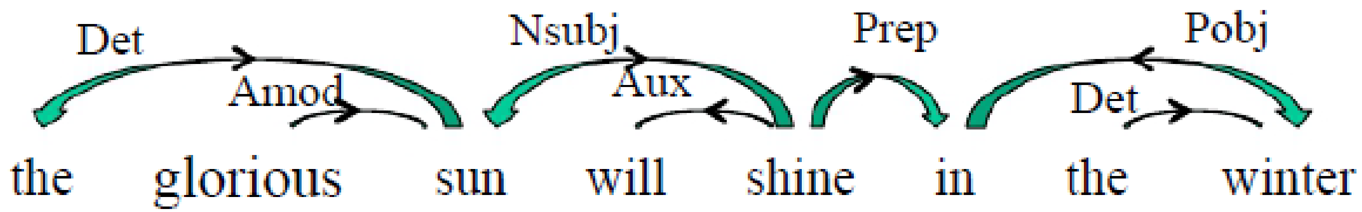
- Dependency grammars offer a different way to represent syntactic structure
  - CFGs represent constituents in a parse tree that can derive the words of a sentence
  - Dependency grammars represent syntactic dependency relations between words that show the syntactic structure
  - Typed dependency grammars label those relations as to what the syntactic structure is
- Syntactic structure is the set of relations between a word (aka the head word) and its dependents.

Examples

- Context Free Grammar Tree Structure



- Dependency Relation Structure



Projective vs. Non-Projective

- In the dependency graph as depicted in the previous example, with the words in sentence order, if no arcs cross, then it is a projective tree
- If there are crossing arcs, then it is a non-projective tree