

# Context-free grammars

## ICS 482 Natural Language Processing

---

Lecture 11: Syntax and  
Context-free grammars

Husni Al-Muhtaseb

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# ICS 482 Natural Language Processing

---

Lecture 11: Syntax and  
Context-free grammars

Husni Al-Muhtaseb

# Syntax and Context-free grammars

ICS 482 Natural Language

Processing

**Lecture 11:**

**Husni Al-Muhtaseb**

# NLP Credits and Acknowledgment

These slides were adapted from  
presentations of the Authors of  
the book

**SPEECH and LANGUAGE PROCESSING:**  
**An Introduction to Natural Language Processing,**  
**Computational Linguistics, and Speech Recognition**

and some modifications from  
presentations found in the WEB  
by several scholars including the  
following

# NLP Credits and Acknowledgment

---

If your name is missing please contact me  
muhtaseb  
At  
Kfupm.  
Edu.  
sa

# NLP Credits and Acknowledgment

Husni Al-Muhtaseb

James Martin  
Jim Martin  
Dan Jurafsky  
Sandiway Fong  
Song young in  
Paula Matuszek  
Mary-Angela  
Papalaskari  
Dick Crouch  
Tracy Kin  
L. Venkata  
Subramaniam  
Martin Volk  
Bruce R. Maxim  
Jan Hajič  
Srinath Srinivasa  
Simeon Ntafos  
Paolo Pirjanian  
Ricardo Vilalta  
Tom Lenaerts

Heshaam Feili

Björn Gambäck  
Christian Korthals  
Thomas G.  
Dietterich  
Devika  
Subramanian  
Duminda  
Wijesekera  
Lee McCluskey  
David J. Kriegman  
Kathleen McKeown  
Michael J. Ciaraldi  
David Finkel  
Min-Yen Kan  
Andreas Geyer-  
Schulz  
Franz J. Kurfess  
Tim Finin  
Nadjet Bouayad  
Kathy McCoy  
Hans Uszkoreit  
Azadeh Maghsoodi

Khurshid Ahmad

Staffan Larsson  
Robert Wilensky  
Feiyu Xu  
Jakub Piskorski  
Rohini Srihari  
Mark Sanderson  
Andrew Elks  
Marc Davis  
Ray Larson  
Jimmy Lin  
Marti Hearst  
Andrew  
McCallum  
Nick Kushmerick  
Mark Craven  
Chia-Hui Chang  
Diana Maynard  
James Allan

Martha Palmer  
julia hirschberg  
Elaine Rich  
Christof Monz  
Bonnie J. Dorr  
Nizar Habash  
Massimo Poesio  
David Goss-  
Grubbs  
Thomas K Harris  
John Hutchins  
Alexandros  
Potamianos  
Mike Rosner  
Latifa Al-Sulaiti  
Giorgio Satta  
Jerry R. Hobbs  
Christopher  
Manning  
Hinrich Schütze  
Alexander Gelbukh  
Gina-Anne Levow  
Guitao Gao  
Qing Ma  
Zeynep Altan

# Previous Lectures

---

- ❑ Pre-start questionnaire
- ❑ Introduction and Phases of an NLP system
- ❑ NLP Applications - Chatting with Alice
- ❑ Finite State Automata & Regular Expressions & languages
- ❑ Morphology: Inflectional & Derivational
- ❑ Parsing and Finite State Transducers
- ❑ Stemming & Porter Stemmer
- ❑ 20 Minute Quiz
- ❑ Statistical NLP – Language Modeling
- ❑ N Grams
- ❑ Smoothing and NGram: Add-one & Witten-Bell
- ❑ Return Quiz1
- ❑ Parts of Speech
- ❑ Arabic Parts of Speech

# Today's Lecture

---

- **C**ontext **F**ree **G**rammar (**CFG**)
  - Syntax and Grammar
  - Derivation & Parsing
  - Recursion
  - Agreement
  - Subcategorization



# Administration

---

## □ Quiz 2

- When? 3<sup>rd</sup> April 2007 or 10<sup>th</sup> April 2007?
- Class time
- Covered material
  - Textbook: Ch 6, 8, 9 + External Material + ...
  - We are not covering Speech related material

# Syntax

---

- ❑ Syntax: 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
- ❑ Isn't it the kind of stuff you were later taught in school?

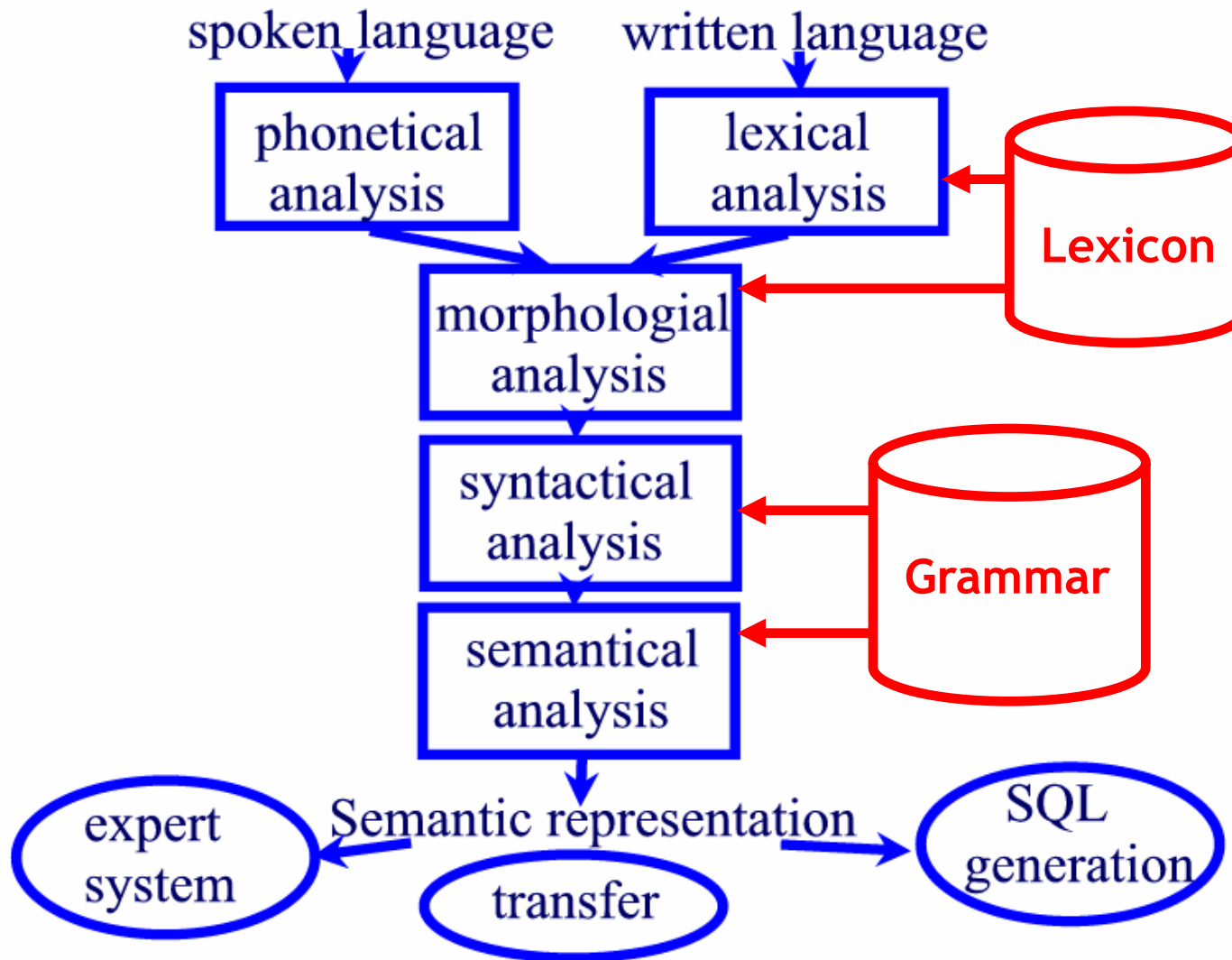
# Syntax

---

## □ Applications

- Grammar checkers
- Question answering
- Information extraction
- Machine translation

# General NLP System Architecture



# Analysis of Natural Languages

---

## □ Syntax

actual structure of a sentence

## □ Parsing

best possible way to make an analysis of a sentence

## □ Semantics

representation of the meaning of a sentence

## Grammar

the rule set used in the analysis

# NL Understanding

---

- Understanding written text
  - Which books are bestsellers?
  - Who wrote them?
- Stages
  - Morphology: analyze word inflection
  - Syntax: determine grammatical structure
  - Semantics: convert to a form that is meaningful to a computer
    - eg, SQL query
  - Pragmatics and discourse: influence of context
    - eg, what *them* refers to

# Example

---

- ❑ Original: Who wrote them?
- ❑ Morphology: who write/past them
- ❑ Grammar: [verb=write, subject=who, object=them]
- ❑ Semantics: Select title, firstname, lastname from [X]
- ❑ Discourse & pragmatics:
  - Select title, firstname, lastname from books Where salesthisyear >10000

# Grammar: Definition

---

- A grammar defines syntactically legal sentences
  - Ahmad ate an apple (syntactically legal)
  - Ahmad ate apple (not syntactically legal)
  - Ahmad ate a building (syntactically legal)
- Sentences may be grammatically OK but not acceptable (acceptability?)
  - The sleepy table eats the green idea.
  - تأكل المنضدة الناعسة الفكرة الخضراء. (تركيب صحيح لكن هل الجملة مقبولة؟)



# Context-Free Grammars (CFG)

---

## □ Capture constituency and ordering

### ■ Ordering is easy

What are the rules that govern the ordering of words and bigger units in the language

### ■ What's constituency?

How do words group into units and what we say about how the various kinds of units behave

# CFG Examples

---

- $S \rightarrow NP VP$
- $NP \rightarrow Det NOMINAL$
- $NOMINAL \rightarrow Noun$
- $VP \rightarrow Verb$
- $Det \rightarrow a$
- $Noun \rightarrow flight$
- $Verb \rightarrow left$

# CFGs

---

## □ $S \rightarrow NP VP$

- This says that there are units called S, NP, and VP in this language
- That an S consists of an NP followed immediately by a VP
- Doesn't say that that's the only kind of S
- Nor does it say that this is the only place that NPs and VPs occur

# Generativity

---

- As with FSAs and FSTs you can view these rules as either analysis or synthesis machines
  - Generate strings in the language
  - Reject strings not in the language
  - Impose structures (trees) on strings in the language

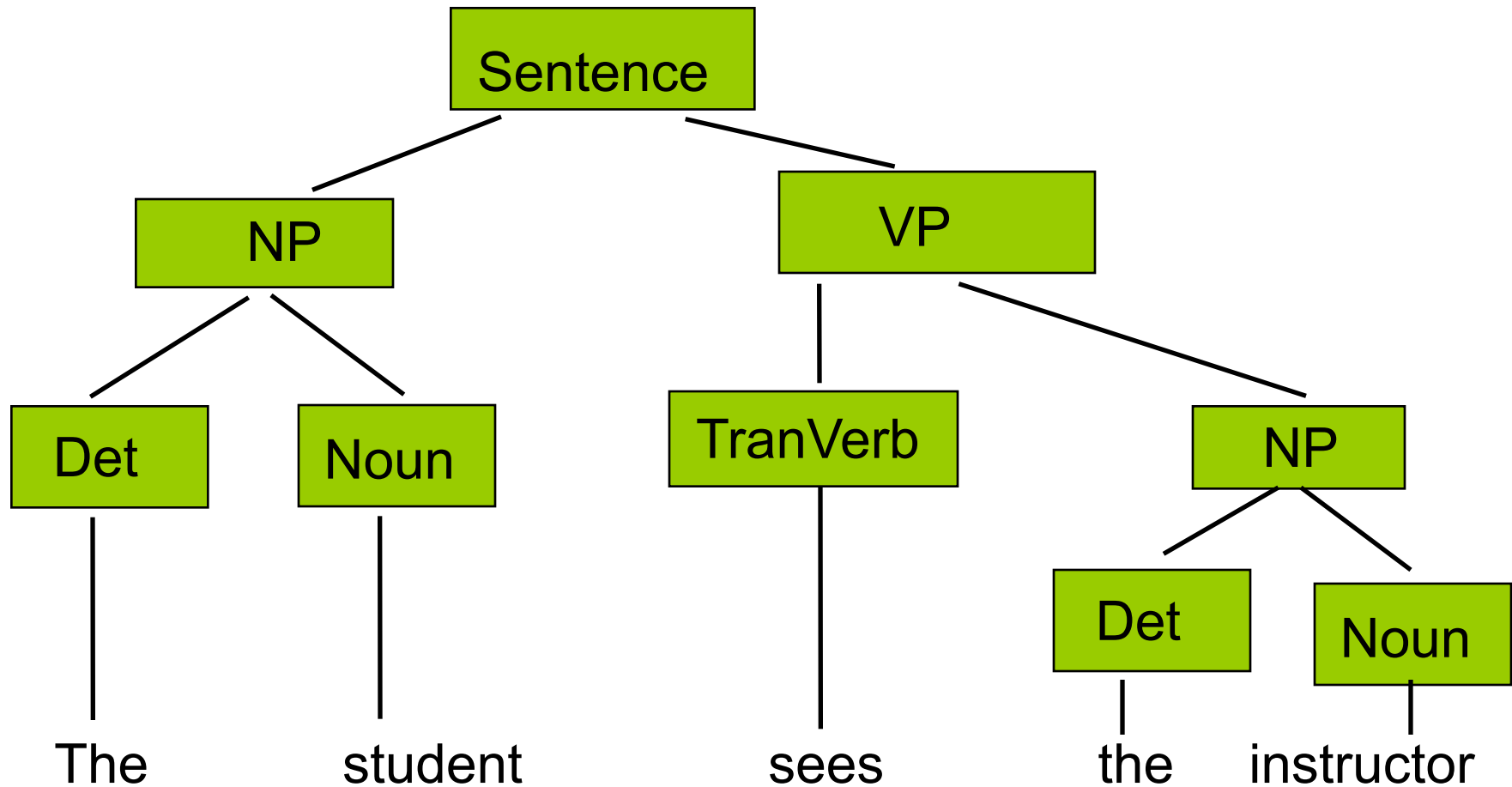
# Derivations

---

- A derivation is a sequence of rules applied to a string that accounts for that string
  - Covers all the elements in the string
  - Covers only the elements in the string

# Derivations as Trees

The student sees the instructor



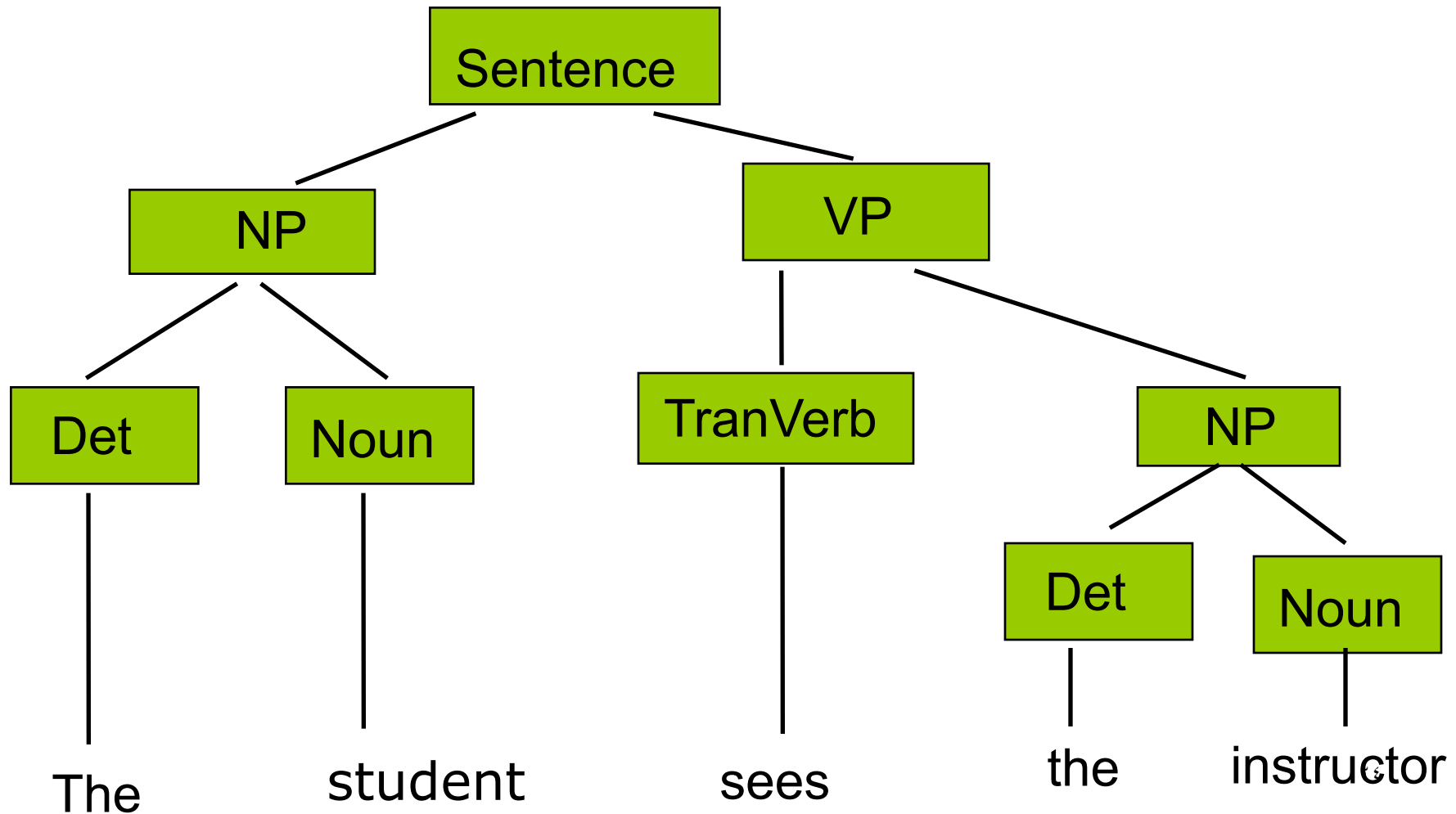
# Parsing

---

- Parsing is the process of taking a string and a grammar and returning a (many?) parse tree(s) for that string
- It is equivalent to running a finite-state transducer with a tape
  - Its just more powerful

# One Parsing Tree

---





# Context?

---

- 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 is out there all by itself

$A \rightarrow B C$

Means that I can rewrite an A as a B followed by a C regardless of the context in which A is found

# Key Constituents (English)

---

- Sentences
- Noun phrases
- Verb phrases
- Prepositional phrases

# Sentence-Types

---

□ Declaratives: A plane left

$S \rightarrow NP VP$

□ Imperatives: Leave!

$S \rightarrow VP$

□ Yes-No Questions: Did the plane leave?

$S \rightarrow Aux NP VP$

□ WH Questions: When did the plane leave?

$S \rightarrow WH Aux NP VP$

# Recursion

---

- We'll have to deal with rules such as the following where the non-terminal on the left also appears somewhere on the right (directly).

NP → NP PP     [[The flight] [to Jeddah]]

VP → VP PP     [[departed Riyadh] [at noon]]

# Recursion

---

- This is what makes syntax interesting
  - flights from Dammam
  - Flights from Dammam to Riyadh
  - Flights from Dammam to Riyadh in February
  - Flights from Dammam to Riyadh in February on a Friday
  - Flights from Dammam to Riyadh in February on a Friday under SR300
  - Flights from Dammam to Riyadh in February on a Friday under SR300 with lunch

# Recursion

---

- This is what makes syntax interesting

[[flights] [from Dammam]]

[[[Flights] [from Dammam]] [to Riyadh]]

[[[[Flights] [from Dammam]] [to Riyadh]] [in  
February]]

[[[[[Flights] [from Dammam]] [to Riyadh]] [in  
February]] [on a Friday]]

Etc.

# Recursion

---

- This is what makes syntax interesting

[NP PP]

[[NP PP] PP]

[[[NP PP] PP] PP]

[[[[NP PP] PP] PP] PP]

Etc.

# Context Free

---

- If we have a rule like
  - $VP \rightarrow V NP$
  - It only cares that the thing after the verb (V) is a Noun Phrase (NP). It doesn't have to know about the internal affairs of that NP



# NP internally might be different

---

□ VP → V NP

□ I

■ like

flights from Dammam

Flights from Dammam to Riyadh

Flights from Dammam to Riyadh in February

Flights from Dammam to Riyadh in February on a Friday

Flights from Dammam to Riyadh in February on a Friday  
under SR300

Flights from Dammam to Riyadh in February on a Friday  
under SR300 with lunch

# Conjunctive Constructions

---

- $S \rightarrow S \text{ and } S$ 
  - Ahmad went to Jeddah and Ali followed him
- $NP \rightarrow NP \text{ and } NP$
- $VP \rightarrow VP \text{ and } VP$
- ...
- In fact the right rule for English is  
 $X \rightarrow X \text{ and } X$

# Problems

---

- Agreement
- Subcategorization

# Agreement

---

□ This boy

□ Those boys

□ This boy walks

□ Those boys walk

□ \*This boys

□ \*Those boy

□ \*This boy walk

□ \*Those boys walks

# Agreement

---

- 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.

# Possible CFG Solution

---

- $S \rightarrow NP VP$
  - $NP \rightarrow Det Nominal$
  - $VP \rightarrow V NP$
  - ...
- 
- Sg for singular
  - Pl for Plural
- $SgS \rightarrow SgNP SgVP$
  - $PIS \rightarrow PINp PIVP$
  - $SgNP \rightarrow SgDet SgNom$
  - $PINP \rightarrow PIDet PINom$
  - $PIVP \rightarrow PIV NP$
  - $SgVP \rightarrow SgV NP$
  - ...

# Agreement

---

- We need similar rules for pronouns, also for number agreement, etc.
  - 3SgNP → (Det) (Card) (Ord) (Quant) (AP) SgNominal
  - Non3SgNP → (Det) (Card) (Ord) (Quant) (AP) PlNominal
  - SgNominal → SgNoun | SgNoun SgNoun
  - etc.

**Card:** two friends

**Ord:** First person

**Quant:** Many people

**AP:** Adjective Phrase: **longest** sentence

# Notation

---

- Predet: Pre-determiner – all
- Det: Determiner – the a, an
- Card: Cardinal number – one two
- Ord: Ordinal number – first, second, other
- Quant: Quantifier – many, several
- AP is the adjective phrase. AP can have an adverb before the adjective.
  - AP → (Adv) Adj e.g. the least expensive fare



# Subcategorization

---

- Sneeze: Mazen [sneezed]
- Find: Please find [a flight to Jeddah]<sub>NP</sub>
- Give: Give [me]<sub>NP</sub> [a cheaper fare]<sub>NP</sub>
- Help: Can you help [me]<sub>NP</sub> [with a flight]<sub>PP</sub>
- Prefer: I prefer [to leave earlier]<sub>TO-VP</sub>
- Told: I was told [Saudia has a flight]<sub>S</sub>
- ...

# Subcategorization

---

- \*Mazen sneezed the book
- \*I found to fly to Jeddah
- \*Give with a flight
  
- Subcategorization expresses the constraints that a predicate (verb for now) places on the number and type of the argument it wants to take

# Subcategorization

---

## □ Frames: (around the verb)

- 0: eat, sleep
- NP: prefer, find, leave
- NP NP: show, give
- PP<sub>from</sub> PP<sub>to</sub>: fly, travel
- NP PP<sub>with</sub>: help, load
- VP<sub>to</sub>: prefer, want, need
- VP<sub>bare-stem</sub>: can, would, might
- S: mean

## □ What do we notice? Are we still in pure syntax?

# Towards Semantics

---

- It turns out that verb subcategorization facts will provide a key element for semantic analysis (determining who did what to who in an event).

# Subcategorization and generation

---

- The various rules for VPs over-generate.
  - They permit the presence of strings containing verbs and arguments that don't go together
  - For example
  - $VP \rightarrow V NP$  therefore  
**Sneezed the book** is a VP since "sneeze" is a verb and "the book" is a valid NP

# Possible CFG Solution

---

- $VP \rightarrow V$
- $VP \rightarrow V NP$
- $VP \rightarrow V NP PP$
- ...
- $VP \rightarrow \text{IntransV}$
- $VP \rightarrow \text{TransV NP}$
- $VP \rightarrow \text{TransV NP PP}$
- ...

Intrans: Intransitive

Trans: Transitive

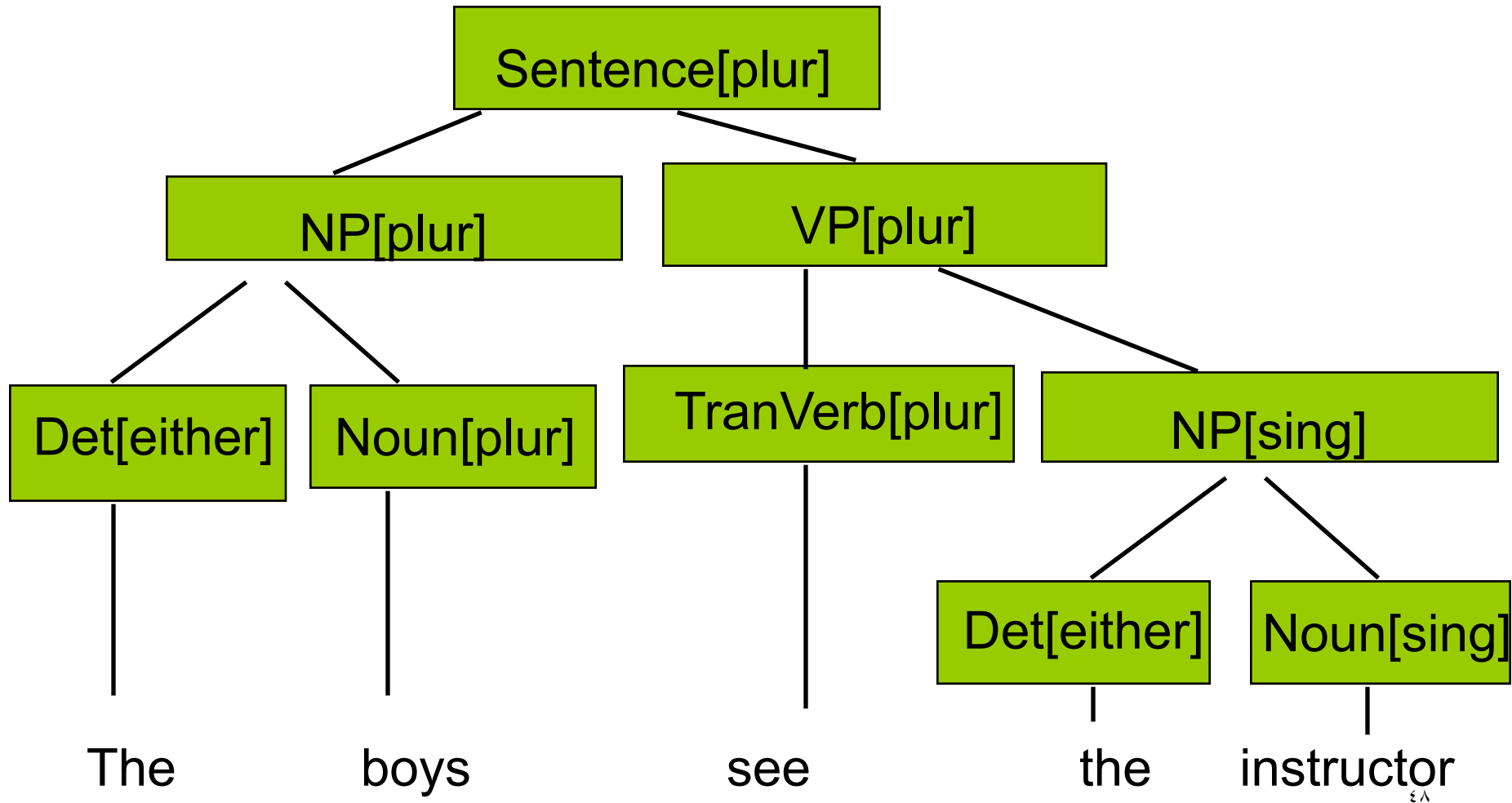
# Auxiliaries subcategories

---

- Modals: can, could, may, might
  - VP – Head verb bare stem
- Perfect: have
  - VP – Head verb past participle
- Progressive: be
  - VP – Head verb gerundive participle
- Passive: be
  - VP – Head verb past participle
- Multiple auxiliaries appear in particular order
  - modal < perfect < progressive < passive

# Parameterization with feature: Get a feeling

---





# Some features

---

- Number (singular, plural)
- Person (I, you, him)
- Tense (past, present, future)
- Gender (feminine, masculine)
- Polarity (positive, negative)
- ...

Thank you

---

السلام عليكم ورحمة الله