Computational Grammar

Sarmad Hussain, Tafseer Ahmed, Nayyara Karamat, Shanza Nayyer
Center for Research in Urdu Language Processing
National University of Computer and Emerging Sciences
Lahore, Pakistan
www.crulp.org
sarmad.hussain@nu.edu.pk

How to develop a computational grammar?

1. Define a basic POS set
2. Identify simple constituents: phrases and sentences
3. Develop phrasal and sentence level context-free rules
4. Add additonal information
    i. Agreement
    ii. Grammatical relations
    iii. Sub-categorization frames
5. Repeat with more complex structures until analysis is stable

Part-of-Speech (POS)/Word Class

- Open Classes
  - Nouns
  - Verbs
  - Adjectives
  - Adverbs
- Closed Classes
  - Prepositions
  - Determiners
  - Pronouns
  - Conjunctions
  - Auxiliary Verbs
  - Particles
  - Numerals
  - Case Markers

How many POS tags?

- Penn Tree Bank = 45
- Brown Corpus = 87
- C5 tagset = 61
- C7 tagset = 146

- Requirement dependent on the application
- Define/refine tags as per requirement

Constituency

- Words normally group together into phrases
- Phrases group together to give larger phrases and sentences
- How to identify similar phrases or constituents
  - Similar syntactic environment
  - Movement

Similar Syntactic Environment

- He sits
- The boy sits
- The naughty boy sits
- The three young men sit
- The sit
- Three sit
- Young sit

Movement

- I am meeting with the nice old man
- With the nice old man, I am meeting
- With, I am meeting, the nice old man
- With the, I am meeting, nice old man
- With the nice, I am meeting, old man

Rule Definition using CFG

- Context-Free Grammar
  - A set of non-terminal symbols, N
  - A set of terminal symbols, Σ
  - A set of productions, P, such that each production is of the form A → α, where A belongs to N and α is a string from (Σ U N)*
  - A start symbol S

    - S → A
    - A → aA | a
      - a, aa, aaa, aaaa, …

Grammar Development

Noun Phrase

- [**John**] went
- [**She**] came
- [*The big brown loyal* ***dog***] is missing
- [*Some of the* ***books***] have got ruined by the rain
- The child tried [*ten ice-cream* ***flavors***]
- I saw [*the naughty* ***girls*** *planning mischief in the corner*].
- [*The* ***man*** *who took the secret packet from me*], vanished

Noun Phrase

[The **book** on the table] is good.

- Head = book
- Pre-Nominal Phrase = The
- Post Nominal Phrase = on the table

[*The **man** who took the secret packet*] vanished

- Head = man
- Pre-Nominal Phrase = The
- Post Nominal Phrase = who took the secret packet

NP – Grammar Rules

$NP \rightarrow NP_{pronoun}$

$NP \rightarrow NP_{noun}$

$NP_{noun} \rightarrow (PreNomP)\ n\ \ (PostNomP)$

Pre-Nominal Phrase

A good book

The nice girl

Few boys

First three hours

The three old mice

PreNomP – Grammar Rules

PreNomP → (DetP) (NumberP) (AdjP) (NounModP)

DetP

a, the , an

NumberP

some, few , three, third, first three

AdjP

nice, good, white, cloudy, warm

NounModP

cricket(in “cricket team”), grammar (in “grammar book”)

PreNomP - Adjectives

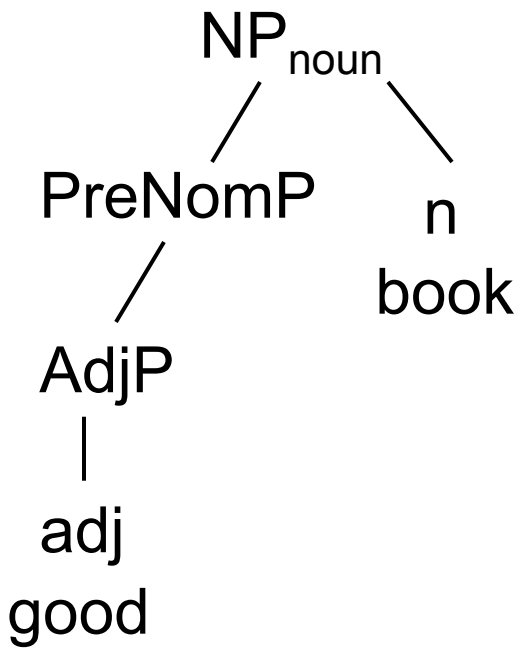
NPnoun → (PreNomP) n (PostNomP)

PreNomP → (DetP) (NumberP) (AdjP) (NounModP)

AdjP → adj

n: book

adj: good



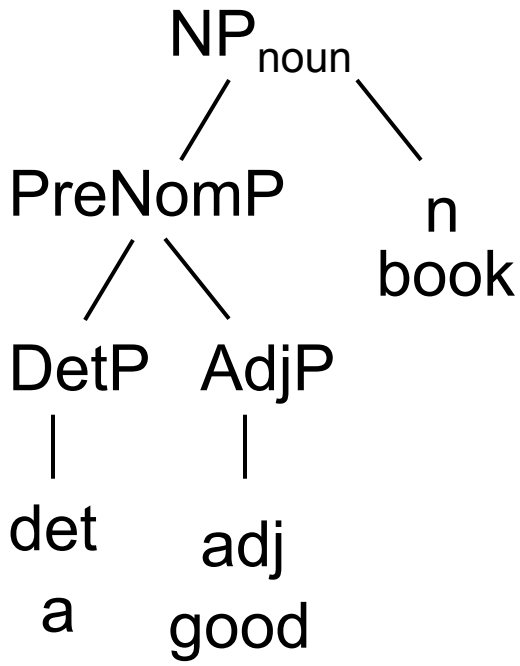
PreNomP - Determiner

NPnoun → (PreNomP) n (PostNomP)
PreNomP → (DetP) (NumberP) (AdjP) (NounModP)
AdjP → adj

DetP → det

n: book
adj: good

det: a



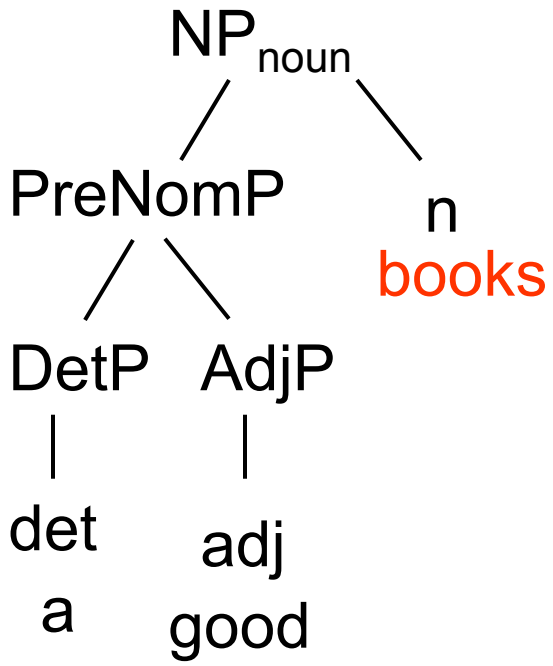
Over-Generation

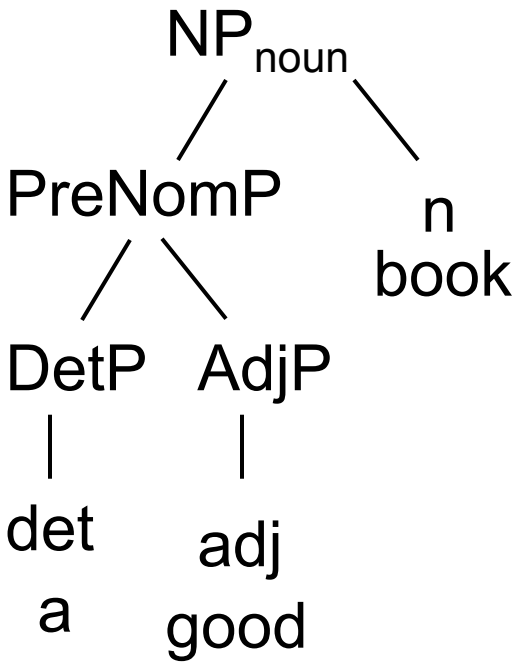
NPnoun → (PreNomP) n (PostNomP)
PreNomP → (DetP) (NumberP) (AdjP) (NounModP)
AdjP → adj

DetP → det

n: book
adj: good
det: a

n: books





Agreement Problem

A book

A books

The book

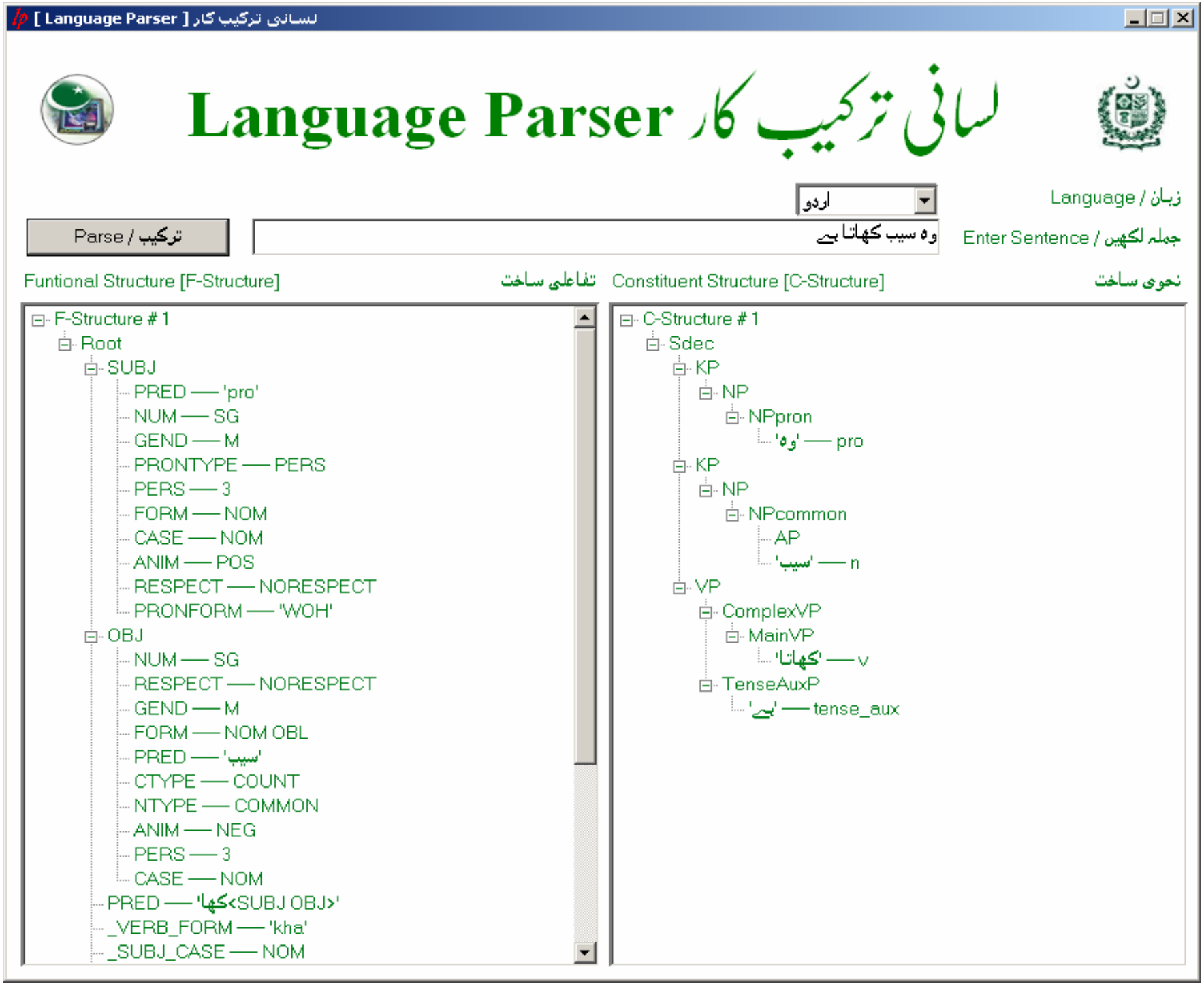
The books

Solution

- CFG is not sufficient
- Need to augment it
- Many frameworks available
- We look at Lexical Functional Grammar

Lexical Functional Grammar

- Defines two different syntactic structures
  - **C(Constituent) Structure** represents linear and hierarchical organization of words into phrases
  - **F(Functional) Structure** represents abstract functional syntactic organization of the sentence, representing syntactic predicate-argument structure and functional relations like subject and object


[Language Parser] لسانی ترکیب کار
Language Parser
لسانی ترکیب کار
Language / زبان
اردو
Enter Sentence / جملہ لکھیں
وہ سیب کھاتا ہے
Parse / ترکیب
Funtional Structure [F-Structure]
تفاعلی ساخت
Constituent Structure [C-Structure]
نحوی ساخت
F-Structure # 1
Root
SUBJ
PRED — 'pro'
NUM — SG
GEND — M
PRONTYPE — PERS
PERS — 3
FORM — NOM
CASE — NOM
ANIM — POS
RESPECT — NORESPECT
PRONFORM — 'WOH'
OBJ
NUM — SG
RESPECT — NORESPECT
GEND — M
FORM — NOM OBL
PRED — 'سیب'
CTYPE — COUNT
NTYPE — COMMON
ANIM — NEG
PERS — 3
CASE — NOM
PRED — 'کھا<SUBJ OBJ>'
_VERB_FORM — 'kha'
_SUBJ_CASE — NOM
C-Structure # 1
Sdec
KP
NP
NPpron
'وہ' — pro
KP
NP
NPcommon
AP
'سیب' — n
VP
ComplexVP
MainVP
'کھاتا' — v
TenseAuxP
'ہے' — tense_aux

Grammatical Functions

- **SUBJ**ect, **OBJ**ect, **OBJ2**,
  - He(SUBJ) made a cake(OBJ)
  - He(SUBJ) made me(OBJ) a cake(OBJ2)
- **COMP**lement, **XCOMP**lement
  - COMP is a closed function that contain an internal SUBJ phrase
    - David complained that Chris yawned. (COMP)
  - XCOMP is an open function that does not contain an internal SUBJ phrase
    - David seemed to yawn. (XCOMP)

Grammatical Functions

- **ADJ**unct, **XADJ**unct
  - ADJ is optional
    - He works in the morning. (ADJ)
  - XADJ is an adjunct having a verb without subject (subject outside the clause)
    - Stretching his arms, David yawned. (XADJ).
- **OBL**ique
  - Required arguments, normally attached with verbs
    - David gave the book to Chris. (OBL)

Grammatical Functions

- **Governable Grammatical Functions**

  These are subcategorized (required) by predicate

  SUBJ, OBJ, OBJ2, COMP, XCOMP, OBL

- **Modifying Grammatical Functions**

  These are not subcategorized (required) by predicate

  ADJ, XADJ

Well-formedness

- **Completeness**
  - F-Structure contains all the governable grammatical functions that its predicate governs: eat<SUBJ,OBJ>
  - They eat an apple
  - They eat
- **Coherence**
  - F-Structure disallows extra governable grammatical functions that its predicate does not governs
  - run<SUBJ>
    - They run
    - They run stadium
- **Consistency**
  - An Attribute in F-Structure can have at most one value
    - A boy runs
    - A boys runs

Sample F-Structures

Good book

$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{SG} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'good'} \end{bmatrix} \end{bmatrix}$$

Good books

$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{PL} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'good'} \end{bmatrix} \end{bmatrix}$$

PreNomP - Adjectives

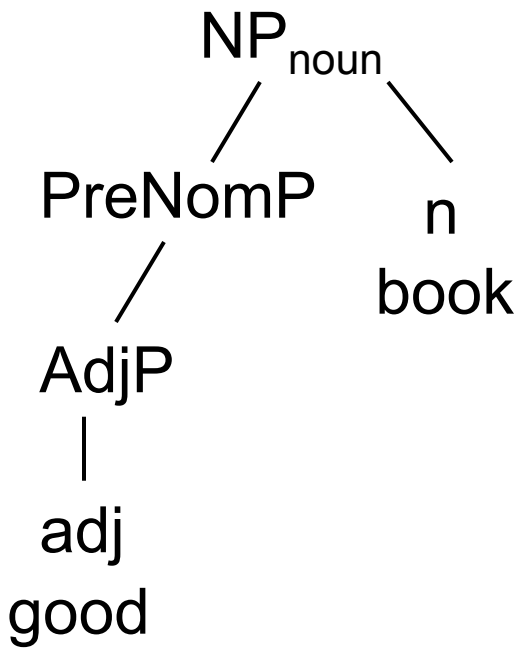
NPnoun → (PreNomP) n (PostNomP)

PreNomP → (DetP) (NumberP) (AdjP) (NounModP)

AdjP → adj

n: book

adj: good



Lexicon

good: adj, ↑ PRED = ‘good’

book: noun, ↑PRED = ‘book’, ↑ NUM = SG, ↑CASE = {NOM,ACC}

books: noun, ↑PRED = ‘book’, ↑ NUM = PL, ↑CASE = {NOM,ACC}

NP – Grammar Rules

Simple Syntax:

NP → NP_{noun}
NP_{noun} → (PreNomP) n (PostNomP)

With Functional Annotation:

NP → NP_{noun}:↑=↓;
NP_{noun} → (PreNomP:↑=↓;) n:↑=↓; (PostNomP:↑=↓;)
PreNomP → (AdjP:↑ ADJUNCT = ↓;) (NumberP) (AdjP) (NounModP)
AdjP → adj:↑ = ↓;

Unification

- Unification is a (partial) operation on feature structures that combines two feature structures such that the new feature structure contains all the information of the original two, and nothing more.
- If there is no smallest feature structure that is subsumed by both then we say that the unification is undefined.

Unification Examples

$$[\text{NUM} \quad \text{SG}] \text{ and } [\text{PERS} \quad 3] = \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{PERS} & 3 \end{bmatrix}$$

$$[\text{NUM} \quad \text{SG,PL}] \text{ and } [\text{NUM} \quad \text{SG}] = [\text{NUM} \quad \text{SG}]$$

$$[\text{NUM} \quad \text{SG}] \text{ and } [\text{NUM} \quad \text{PL}] = \textit{UNDEFINED}$$

Sample F-Structures

Good book

$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{SG} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'good'} \end{bmatrix} \end{bmatrix}$$

Good books

$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{PL} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{'good'} \end{bmatrix} \end{bmatrix}$$

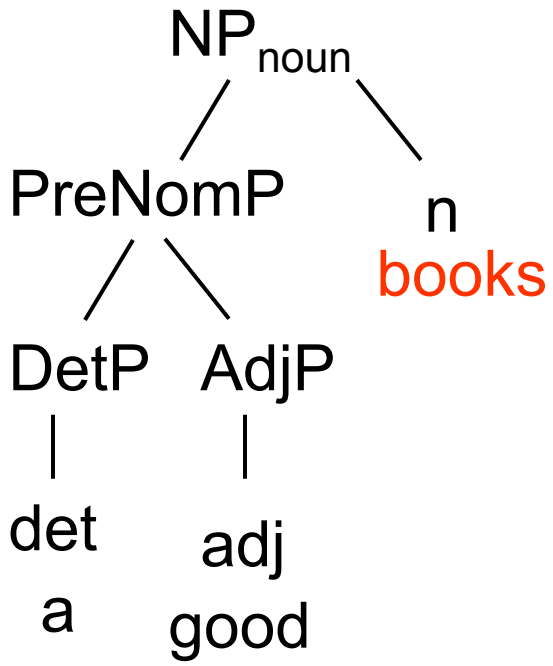
Over-Generation

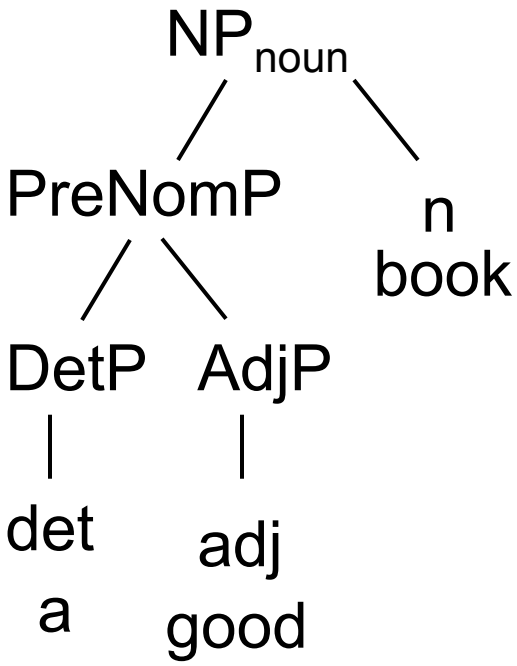
NPnoun → (PreNomP) n (PostNomP)
PreNomP → (DetP) (NumberP) (AdjP) (NounModP)
AdjP → adj

DetP → det

n: book
adj: good
det: a

n: books





NP – Lexicon and Grammar (2)

a: det, ↑ DEFINITE = NEG, ↑ NUM = SG.

the: det, ↑ DEFINITE = POS, ↑ NUM = {SG,PL}.

PreNomP → (DetP: ↑ SPEC = ↓, ↑NUM =c ↓NUM;)
(AdjP:↑ ADJUNCT = ↓;)

DetP → det:↑=↓; .

Agreement

- A book

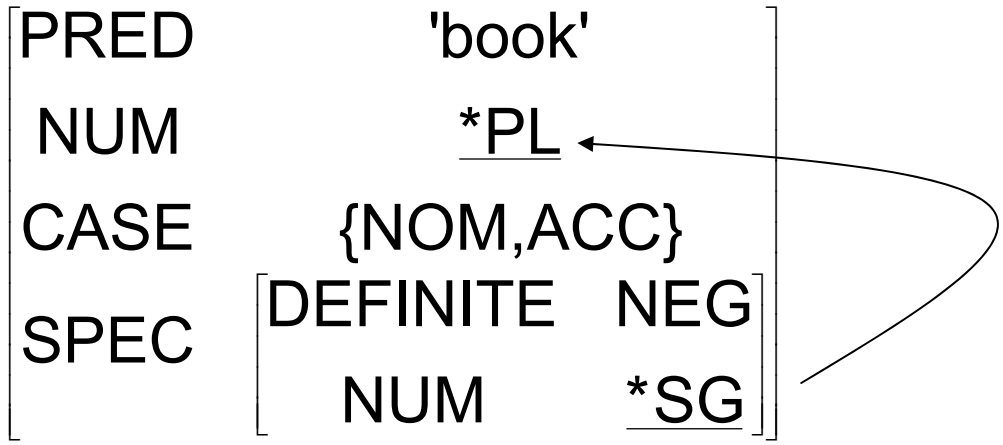
$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{SG} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{SPEC} & \begin{bmatrix} \text{DEFINITE} & \text{NEG} \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix}$$

- The books

$$\begin{bmatrix} \text{PRED} & \text{'book'} \\ \text{NUM} & \text{PL} \\ \text{CASE} & \{\text{NOM,ACC}\} \\ \text{SPEC} & \begin{bmatrix} \text{DEFINITE} & \text{POS} \\ \text{NUM} & \{\text{SG,PL}\} \end{bmatrix} \end{bmatrix}$$

Agreement

A books



Sentence

John drives the car

John sleeps

S → NP VP

VP → v (NP)

Sentence with F-Descriptions

S → NP: ↑ SUBJ= ↓; VP:↑=↓; .

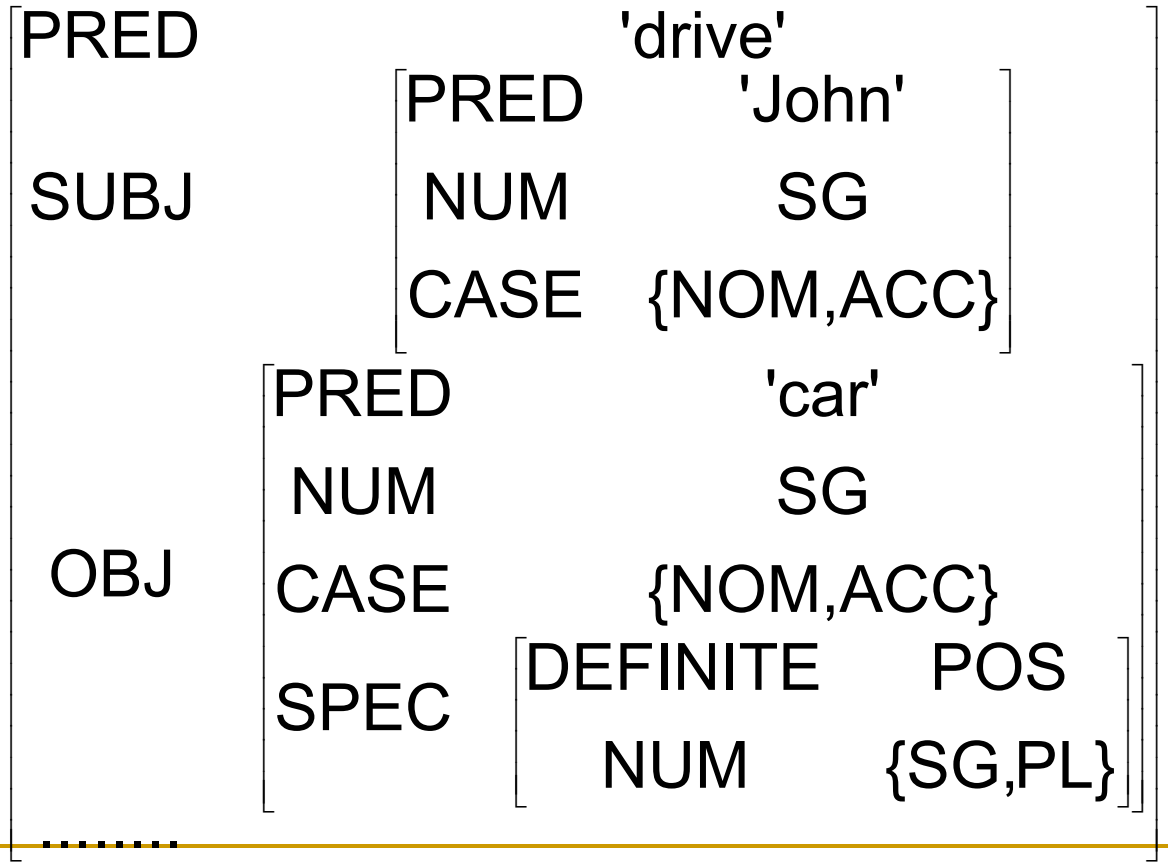
VP → v:↑ =↓; (NP: ↑ OBJ=↓;) .

sleeps: v, ↑ PRED = ‘sleep’, …

drives: v ↑ PRED = ‘drive’ , …

Sentence – Sample F-Structure

John drives the car.



Problem

John drives

John sleeps the car

Solution: Sub-Categorization Frame

sleeps: v, ↑ PRED = ‘sleep<SUBJ>’, …

drives: v ↑ PRED = ‘drive<SUBJ,OBJ>’ , …

likes:v, ↑ PRED = ‘like<SUBJ,OBJ>, …

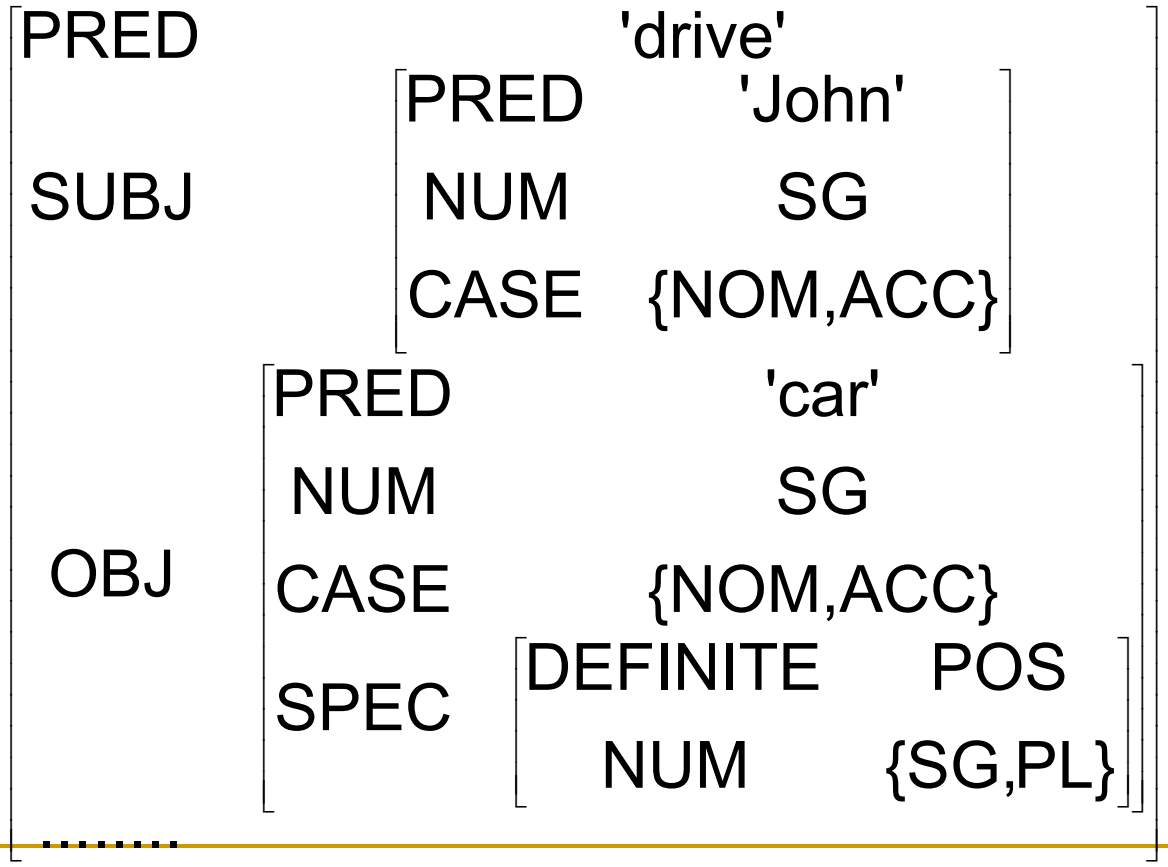
likes:v, ↑ PRED = ‘like<SUBJ,XCOMP>, …

John likes the car.

John likes to drive the car.

Sentence – Sample F-Structure

John drives the car.



Pronouns and Cases

$NP \rightarrow NP_{pronoun}$

$NP_{pronoun} \rightarrow$ pronoun

He likes her

John likes Mary

Her likes he

Mary likes John

She Likes him

Pronouns : Lexicon

he: pronoun, ↑ PRED = ‘pro’, ↑ PERS = 3, ↑ NUM = SG, ↑ GEND = M, ↑ CASE = NOM

she: pronoun, ↑ PRED = ‘pro’, ↑ PERS = 3, ↑ NUM = SG, ↑ GEND = F, ↑ CASE = NOM

I: pronoun, ↑ PRED = ‘pro’, ↑ PERS = 3, ↑ NUM = SG, ↑ GEND = {M,F}, ↑ CASE = NOM

him: pronoun, ↑ PRED = ‘pro’, ↑ PERS = 3, ↑ NUM = SG, ↑ GEND = M, ↑ CASE = ACC

her: pronoun, ↑ PRED = ‘pro’, ↑ PERS = 3, ↑ NUM = SG, ↑ GEND = F, ↑ CASE = ACC

Pronoun Case

Personal Pronouns (PERS)		
Singular (SG)	**Subjective** (NOM)	**Objective** (ACC/DAT)
1st person	I we	me us
2nd person	you	you
3rd person	he/ she/it they	him/her/it them

Sentence – Revised Grammar Rules

S -> NP: ↑ SUBJ= ↓, ↓ CASE =c NOM;
VP:↑=↓;

VP -> v:↑ =↓; (NP: ↑ OBJ=↓, ↓ CASE =c ACC;)

Other Phrases

- Verbal Phrase with tense and aspect
- Prepositional Phrase
- Adverbial Phrase
- Conjunctions
- …

Urdu Grammar – Some Remarks

- Agreements
  - within Noun Phrase
- Case Marking
- Free Order Language

Agreements within NP: Gender and Number

اچھا لڑکا(achcha larka)

اچھی لڑکی (achchi larki)

اچھے لڑکے (achche larke)

* اچھی لڑکا (achchi larka)

* اچھا لڑکی (achcha larki)

* اچھے لڑکا (achche larka)

NP - Urdu Lexicon and Grammar

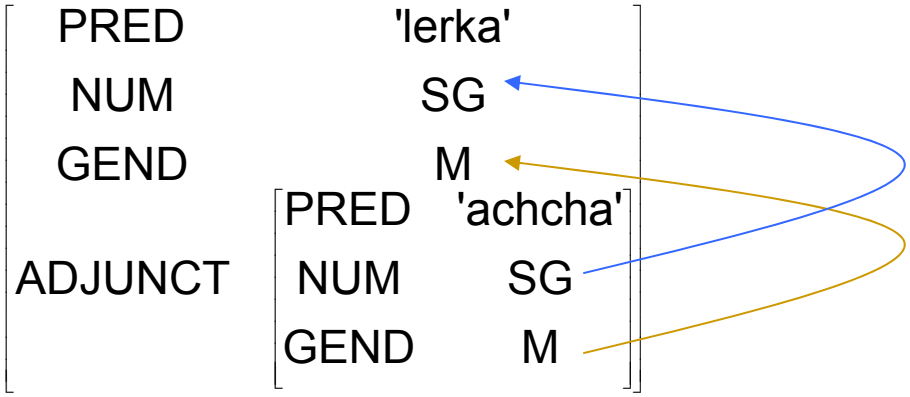
اچھا: verb, ↑ PRED = ‘اچھا’, ↑ NUM = SG, ↑
GEND = M.
اچھی : verb, ↑ PRED = ‘اچھا’, ↑ NUM = SG, ↑
GEND = F.

لڑکا: verb, ↑ PRED = ‘لڑکا’, ↑ NUM = SG, ↑
GEND = M.
لڑکی: verb, ↑ PRED = ‘لڑکی’, ↑ NUM = SG, ↑
GEND = F.

NP - Urdu Lexicon and Grammar (2)

NP_{noun} -> (AdjP: ↑ ADJUNCT =↓, ↑ NUM= ↓NUM, ↑ GEND = ↓ GEND,…;) n:↑=↓; .

اچھا لڑکا (achcha larka)



Case Marking

Eight (Morphological) Cases of Sanskrit

Case	**Represents**
Nominative	Subject
Accusative	Direct Object
Dative	Indirect Object
Instrumental	Cause
Locative	position
Ablative	comparison
Genitive	relation
Vocative	Addressing/Calling

Case Phrase and Case Markers

[لڑکا] [آم] کھاتا ہے۔

[larka] [aam] khata hay

[شکاری نے] [شیر کو] [[افریقہ کے] جنگل میں] [بندوق سے] مارا

[shikari ne] [shair ko] [[africa ke] jungle mein] [bandooq se] mara.

[لڑکے نے] [شکاری کو] [بندوق] دی۔

[larke ne] [shikari ko] [bandooq] di.

[لڑکے نے] [اسے] [بندوق] دی۔

[larke ne] [usse] [bandooq] di.

Case Markers

Urdu has case markers to represent cases

نے:cm, ↑ CASE = ERG

کو: cm, ↑ CASE = {ACC,DAT}

سے: cm, ↑ CASE = INST

میں: cm, ↑ CASE = LOC

Some pronouns have case through morphology

اسے:pronoun, ↑ PRED = 'pro', ↑ PERS = 3, NUM = SG,
GEND ={M,F}, ↑ CASE = {ACC,DAT}

Case Phrase

KP → NP_{noun} : ↑=↓, ↑CASE = NOM; .

KP → NP_{noun} : ↑=↓; cm : ↑=↓; .

شکاری نے (shikari ne) جنگل میں (jungle mein)

$$\begin{bmatrix} \text{PRED} & \text{'shikari'} \\ \text{NUM} & \text{SG} \\ \text{GEND} & \text{M} \\ \text{CASE} & \text{ERG} \end{bmatrix} \qquad \begin{bmatrix} \text{PRED} & \text{'jungle'} \\ \text{NUM} & \text{SG} \\ \text{GEND} & \text{M} \\ \text{CASE} & \text{LOC\_MEN} \end{bmatrix}$$

Word Order

[شکاری نے] [شیر کو] دیکھا۔

[shikari ne] [shair ko] dekha.

[شیر کو] [شکاری نے] دیکھا۔

[shair ko] [shikari ne] daikha.

[شکاری نے] دیکھا [شیر کو]۔

[shikari ne] daikha [shair ko].

[شیر نے] [شکاری کو] دیکھا۔

[shair ne] [shikari ko] dekha.

[شکاری کو] [شیر نے] دیکھا۔

[shikari ko] [shair ne] daikha.

Word Order and Shuffle Operator

S → KP:↑=SUBJ=↓, ↓ CASE =c NOM;
KP:↑=OBJ=↓, ↓ CASE =c ACC;
VerbalP: ↑=↓, …;

S → [KP:↑=SUBJ=↓, ↓ CASE =c NOM;
% KP:↑=OBJ=↓, ↓ CASE =c ACC;
% VerbalP: ↑=↓, …;
]

F-Structure of "He walks"

Pred	walk < Subject >	
Tense	Present	
Prog	False	
Perf	False	
Subject	Pred	Pronoun
	Number	Singular
	Gender	Masculine
	Person	Third
	Form	He
	Animated	True
	Case	Nominative

Parsing Algorithms

- Parsing as search
- Chart Parsing

Parsing as Search

- Top Down Algorithm
  - Only builds trees that are rooted in S
  - Wastes time building trees that don't match the input
- Bottom Up Algorithm
  - Only builds trees that match the input
  - Wastes time building trees that never lead to S

Problems in Search Algorithm

- Left Recursion

  NP → NP PP

  VP → VP PP

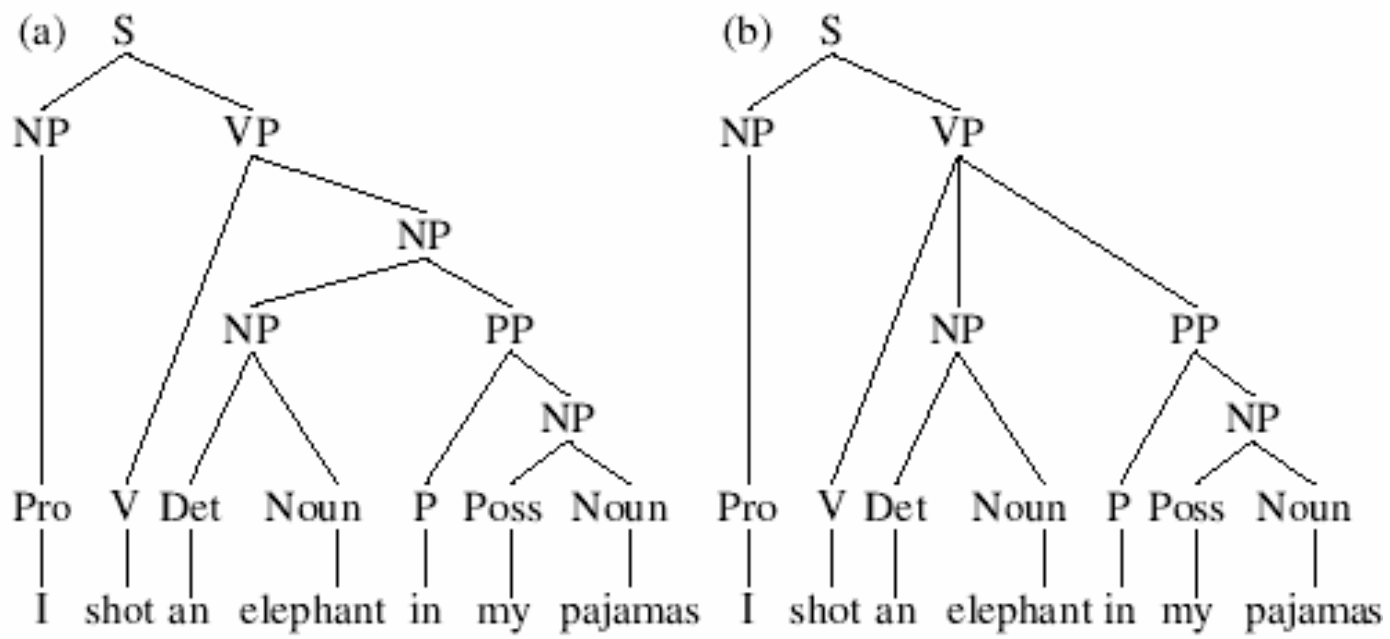
  S → S and S

Left Recursion can be removed from productions of the CFG, but the transformed grammar gives different parse structures

Problems in Search Algorithm

Repeated Parsing

Ambiguous Grammar results in repeated parses of same structures.



The phrases “I”, “an elephant”, “in my pajamas” are parsed twice in two different parses

Chart Parsing

- Chart parsing is an example of dynamic programming.
- Dynamic Programming solves problem by finding solutions of sub-problems, and then combining them to form complete solution.
- In Chart parsing, Sub-trees are built, that are used by more than one parse trees.
- Chart parsing avoids Left Recursion and Repeated parsing problems.

Chart Parsing Algorithm

A chart consists of N + 1 states (where N is the number of words in the input).

The algorithm goes through the states from left to right and processes each state in order applying one of the following operators to the state:

- Predictor
- Completer
- Scanner

Predictor

Predictor creates new states from top-down application of rules.

S -> .VP, [0,0]

applying Predictor on above results in adding the following new states on the chart.

VP -> .verb, [0,0]

VP -> .verb NP, [0,0]

Completer

The completer is applied to a state when its dot has reached the right end of a rule. (whole phrase is parsed.)

The completer takes old states and combines them into new states.

S -> . NP VP [0,0]

NP -> det noun. [0,2]

Applying Cpmpleter on above results in adding the following new state on the chart.

S -> NP . VP [0,2]

Scanner

The dot advances and new state is formed if there is POS Category at right of the dot, and this category is also predicted by a state (by current word of input sentence).

VP -> . verb NP [0, 1]

scanner checks that current word is a verb and adds new state.

VP -> verb . NP [0, 2]

Thank You

References and Further Readings

- Jurafsky, Daniel. (2000), Speech and language Processing, Prentice Hall, New Jersey.
- Butt, Miraiam. (1999), A Grammar Writer's Cookbook. CSLI Publications, Stanford, CA.
- http://cslu.cse.ogi.edu/HLTsurvey/ch3node5.html
- http://www.coli.uni-saarland.de/~kris/nlp-with-prolog/html/node83.html
- http://emsah.uq.edu.au/linguistics/Working%20Papers/ananda_ling/LFG_Summary.htm
- http://www.stanford.edu/group/nasslli/courses/as-cr-da/apbook-nasslli.pdf
- http://www.sfu.ca/person/dearmond/222/222.x-bar.htm