# Syntactic Parsing

Prof. Sameer Singh

CS 295: STATISTICAL NLP

WINTER 2017

February 7, 2017

# Outline

- Syntactic Parsing
- Context Free Grammars
- Parsing: CKY Algorithm

# Outline

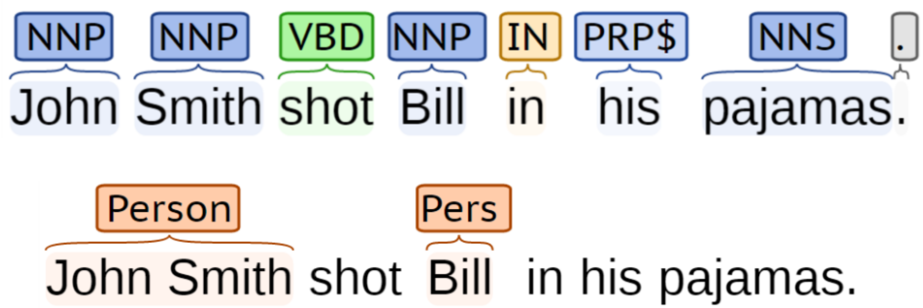
Syntactic Parsing

Context Free Grammars

Parsing: CKY Algorithm

# Limitations of Sequence Tags

John Smith shot Bill in his pajamas.




What happened?
Who shot who?
Who was wearing the pajamas?

# Constituents

- Constituent behave as a unit that can be rearranged:

  John talked [to the children] [about drugs].

  John talked [about drugs] [to the children].

  John talked drugs to the children about

- Or substituted/expanded:

  John talked [to the children taking the drugs] [about alcohol].

Harry the Horse
a high-class spot such as Mindy's
the broadway coppers
the reason he comes into the Hot Box
they
three parties from Brooklyn

X

arrive(s)
attract(s)
love(s)
sit(s)

"Noun phrases appear before verbs in English."

# Constituents and Grammars

**Grammar**

- Tells you how the constituents can be arranged
- Implicit knowledge for us (we often can't tell *why* something is wrong)
- Generate all, and only, the possible sentences of the language
- Different from meaning:

Colorless green ideas sleep furiously.

- The words are in the right order,
- And that ideas are green and colorless,
- And that ideas sleep,
- And that sleeping is done furiously,
- As opposed to: "sleep green furiously ideas colorless"

# Uses of Parsing

[ send [the text message from James] [to Sharon] ]

[ translate [the message] [from Hindi] [to English] ]

- Grammar checkers
- Dialog systems
- High precision question answering
- Named entity recognition
- Sentence compression
- Extracting opinions about products
- Improved interaction in computer games
- Helping linguists find data
- Machine translation
- Relation extraction systems

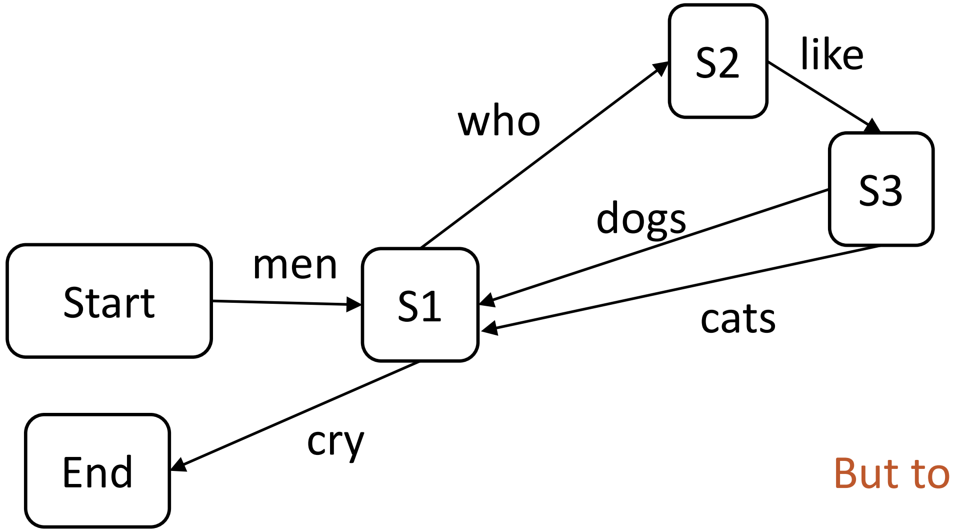
# Outline

- Syntactic Parsing
- **Context Free Grammars**
- Parsing: CKY Algorithm

# Basic Grammar: Regular Expr.

- You can capture individual words:
  - (man|dog|cat)
- Simple sentences:
  - (man|dog|cat)(ate|loves|consumed)(.|food|lunch)
- Infinite length? Yes!
  - men (who like (cats|dogs))* cry.

Finite State Machine




But too weak for English.

# Context-Free Grammars

"Constituents"

Grammar, G

Terminal Symbols, T

{man, dog, cat, likes ...}

Non-terminal Symbols, N

S, NP, VP, ...

S ↓ sentence

Noun, Verb, DET

Rules

n.t.

A → B C D ← seq of n.t.

A → w ← a single terminal

A → B A B

B A B

B B A B B

... B B A B B ...

$B^* A B^*$

Grammar applies rules recursively..

If we can construct the input sentence, it is in the grammar, otherwise not.

# Example CFG

$A \to w_1 | w_2$

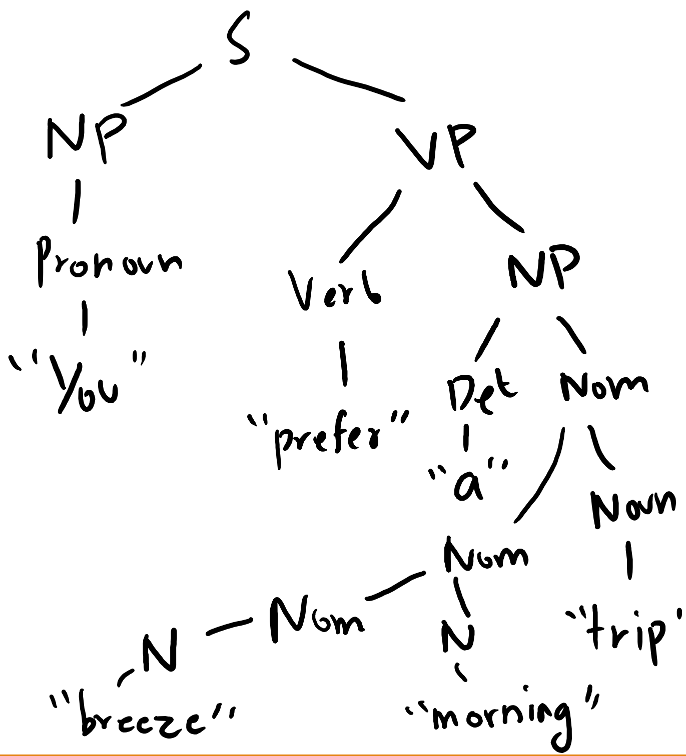
$A \to w_1$

$A \to w_2$

"Lexicon"

| | |
|---|---|
| *Noun* → | *flights* \| *breeze* \| *trip* \| *morning* |
| *Verb* → | *is* \| *prefer* \| *like* \| *need* \| *want* \| *fly* |
| *Adjective* → | *cheapest* \| *non-stop* \| *first* \| *latest* \| *other* \| *direct* |
| *Pronoun* → | *me* \| *I* \| *you* \| *it* |
| *Proper-Noun* → | *Alaska* \| *Baltimore* \| *Los Angeles* \| *Chicago* \| *United* \| *American* |
| *Determiner* → | *the* \| *a* \| *an* \| *this* \| *these* \| *that* |
| *Preposition* → | *from* \| *to* \| *on* \| *near* |
| *Conjunction* → | *and* \| *or* \| *but* |

| Grammar Rules | | | Examples |
|---|---|---|---|
| *S* | → | *NP VP* | I + want a morning flight |
| *NP* | → | *Pronoun* | I |
| | \| | *Proper-Noun* | Los Angeles |
| | \| | *Det Nominal* | a + flight |
| *Nominal* | → | *Nominal Noun* | morning + flight |
| | \| | *Noun* | flights |
| *VP* | → | *Verb* | do |
| | \| | *Verb NP* | want + a flight |
| | \| | *Verb NP PP* | leave + Boston + in the morning |
| | \| | *Verb PP* | leaving + on Thursday |
| *PP* | → | *Preposition NP* | from + Los Angeles |

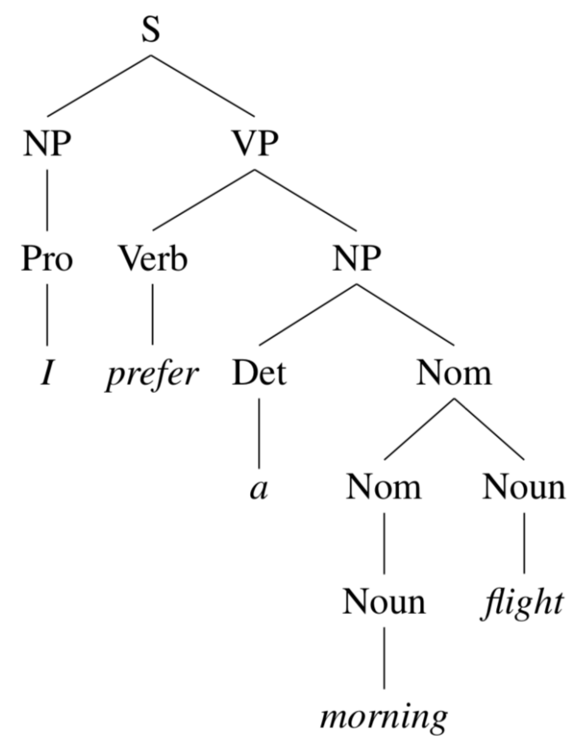


# Example Parse Tree

NP → A size A col NoAdjNP

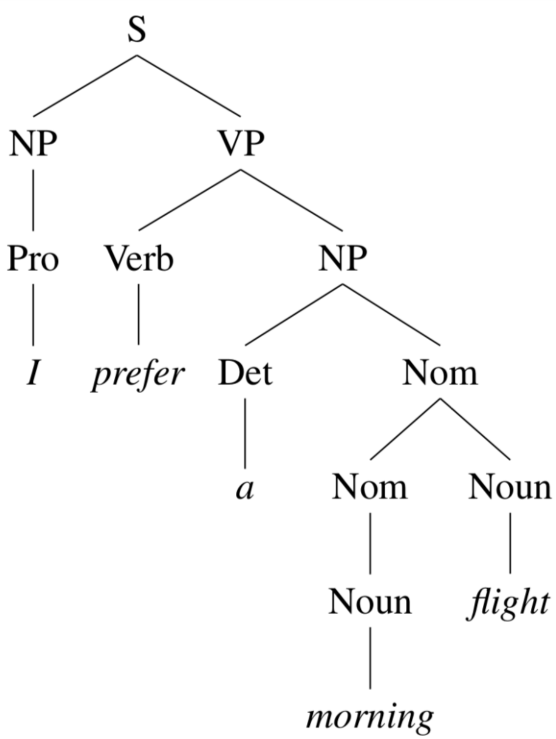
I prefer a morning flight.

| Grammar Rules | | | Examples |
|---|---|---|---|
| *S* | → | *NP VP* | I + want a morning flight |
| *NP* | → | *Pronoun* | I |
| | \| | *Proper-Noun* | Los Angeles |
| | \| | *Det Nominal* | a + flight |
| *Nominal* | → | *Nominal Noun* | morning + flight |
| | \| | *Noun* | flights |
| *VP* | → | *Verb* | do |
| | \| | *Verb NP* | want + a flight |
| | \| | *Verb NP PP* | leave + Boston + in the morning |
| | \| | *Verb PP* | leaving + on Thursday |
| *PP* | → | *Preposition NP* | from + Los Angeles |



# Example Parse Tree: Brackets

I prefer a morning flight.




$[_S [_{NP} [_{Pro} \text{I}]] [_{VP} [_{Verb} \text{prefer}] [_{NP} [_{Det} \text{a}] [_{Nom} \ldots]]]]$

# More details: Noun Phrases

**Simple Noun Phrases**

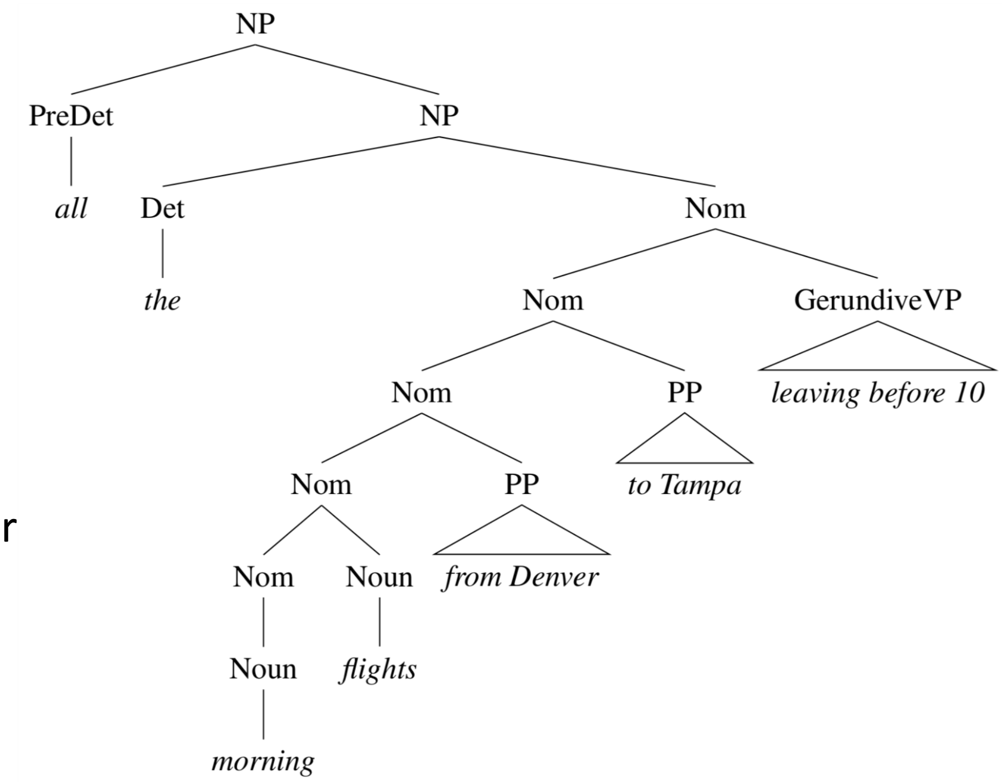
*NP → ProperNoun*

*NP → Det Nominal*

*Nominal → Noun | Noun Nominal*

**Complex Noun Phrases**

"all the morning flights from Denver to Tampa leaving before 10"



# Recursive Noun Phrases

this is the house

this is the house that Jack built

this is the cat that lives in the house that Jack built

this is the dog that chased the cat that lives in the house that Jack built

this is the flea that bit the dog that chased the cat that lives in the house the Jack built

this is the virus that infected the flea that bit the dog that chased the cat that lives in the house that Jack built

# More details: Verb Phrases

Simple Verb Phrases

| | |
|---|---|
| *VP → Verb* | disappear |
| *VP → Verb NP* | prefer a morning flight |
| *VP → Verb NP PP* | leave Boston in the morning |
| *VP → Verb PP* | leave in the morning |

But all verbs are not the same!
(this grammar overgenerates)

Solution: subcategorize!

| | |
|---|---|
| Sneezed: | John sneezed. |
| Find: | Please find a flight to NY. |
| Give: | Give me a cheaper fare. |
| Help: | Can you help me with a flight? |
| Prefer: | I prefer to leave earlier. |
| Told: | I was told United has a flight. |

# Types of Sentences

| | | |
|---|---|---|
| Declarative | $S \rightarrow NP\ VP$ | A plane left. |
| Imperative | $S \rightarrow VP$ | Show the plane. |
| Yes/no Questions | $S \rightarrow Aux\ NP\ VP$ | Did the plane leave?<br>Aux |
| Wh-Questions | $S \rightarrow WhNP\ Aux\ NP\ VP$ | When did the plane leave?<br>WhNP Aux |

# Source of Grammar?

Manual

Noam Chomsky

Write symbolic grammar (CFG or often richer) and lexicon

| | |
|---|---|
| S → NP VP | NN → *interest* |
| NP → (DT) NN | NNS → *rates* |
| NP → NN NNS | NNS → *raises* |
| NP → NNP | VBP → *interest* |
| VP → V NP | VBZ → *rates* |

Used grammar/proof systems to prove parses from words

*Fed raises interest rates 0.5% in effort to control inflation*

- Minimal grammar: 36 parses
- Simple 10 rule grammar: 592 parses
- Real-size broad-coverage grammar: millions of parses

# Source of Grammar?

From data!

The Penn Treebank

Building a treebank seems a lot slower and less useful than building a grammar

But a treebank gives us many things

- Reusability of the labor
    - Many parsers, POS taggers, etc.
    - Valuable resource for linguistics
- Broad coverage
- Frequencies and distributional information
- A way to evaluate systems

[Marcus et al. 1993, *Computational Linguistics*]

```
( (S
    (NP-SBJ (DT The) (NN move))
    (VP (VBD followed)
      (NP
        (NP (DT a) (NN round))
        (PP (IN of)
          (NP
            (NP (JJ similar) (NNS increases))
            (PP (IN by)
              (NP (JJ other) (NNS lenders)))
            (PP (IN against)
              (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
      (, ,)
      (S-ADV
        (NP-SBJ (-NONE- *))
        (VP (VBG reflecting)
          (NP
            (NP (DT a) (VBG continuing) (NN decline))
            (PP-LOC (IN in)
              (NP (DT that) (NN market)))))))
    (. .)))
```

# Some of the rules, with counts

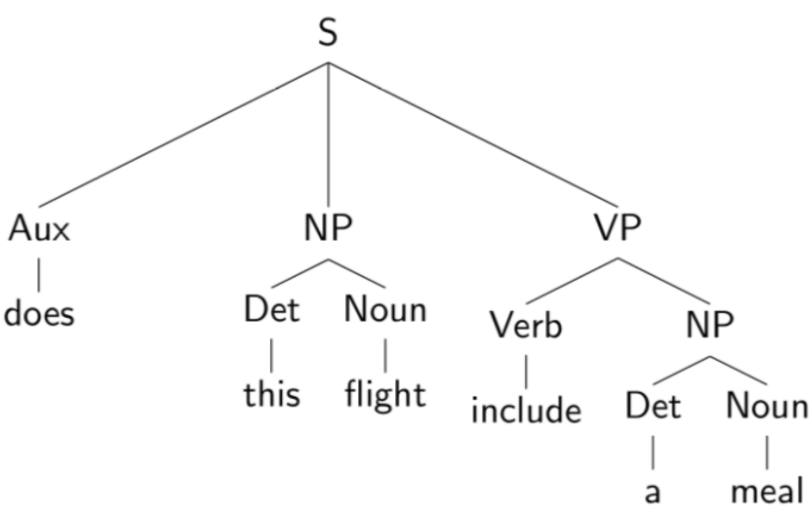
40717 PP → IN NP
33803 S → NP-SBJ VP
22513 NP-SBJ → -NONE-
21877 NP → NP PP
20740 NP → DT NN
14153 S → NP-SBJ VP .
12922 VP → TO VP
11881 PP-LOC → IN NP
11467 NP-SBJ → PRP
11378 NP → -NONE-
11291 NP → NN
...
989 VP → VBG S
985 NP-SBJ → NN
983 PP-MNR → IN NP
983 NP-SBJ → DT
969 VP → VBN VP

100 VP → VBD PP-PRD
100 PRN → : NP :
100 NP → DT JJS
100 NP-CLR → NN
99 NP-SBJ-1 → DT NNP
98 VP → VBN NP PP-DIR
98 VP → VBD PP-TMP
98 PP-TMP → VBG NP
97 VP → VBD ADVP-TMP VP
...
10 WHNP-1 → WRB JJ
10 VP → VP CC VP PP-TMP
10 VP → VP CC VP ADVP-MNR
10 VP → VBZ S , SBAR-ADV
10 VP → VBZ S ADVP-TMP

4500 rules for VP!

# Evaluating Parses

Each parse tree is represented by a list of tuples: $\{\langle t_i, s_i, e_i \rangle\}$




$\langle S, 0, 6 \rangle$ $\langle NP, 1, 3 \rangle$

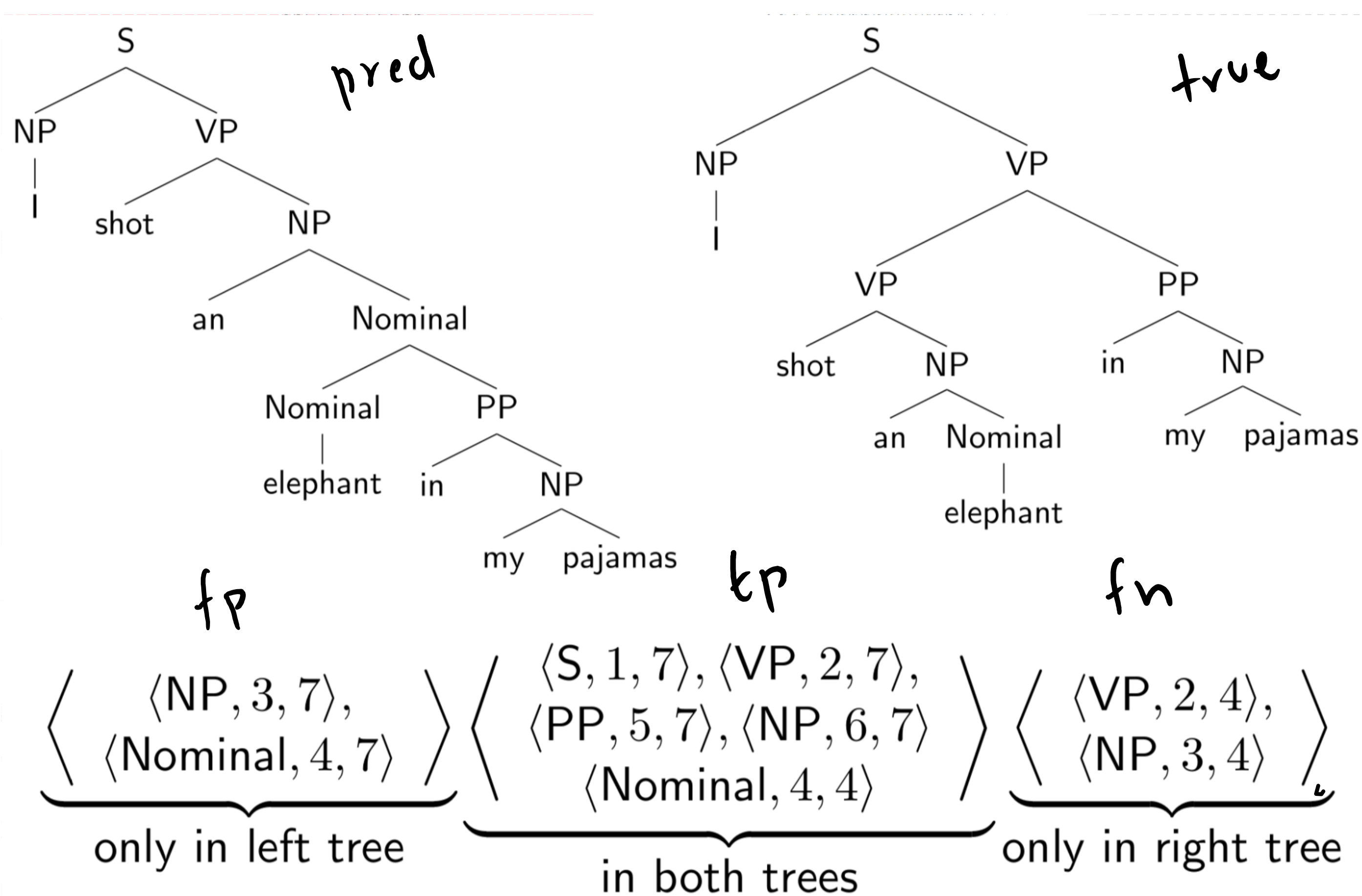
$\langle Aux, 0, 1 \rangle$ $\langle Det, 1, 2 \rangle$

$\langle Noun, 2, 3 \rangle$

$\langle NP, 4, 6 \rangle$

Use this to estimate precision/recall!

# Evaluating Parses: Example



# Outline

Syntactic Parsing

Context Free Grammars

Parsing: CKY Algorithm

# The Parsing Problem

Given sentence **x** and grammar **G**,

Recognition

Is sentence **x** in the grammar? If so, prove it.

"Proof" is a deduction, valid parse tree.

Parsing

Show one or more derivations for **x** in **G**.

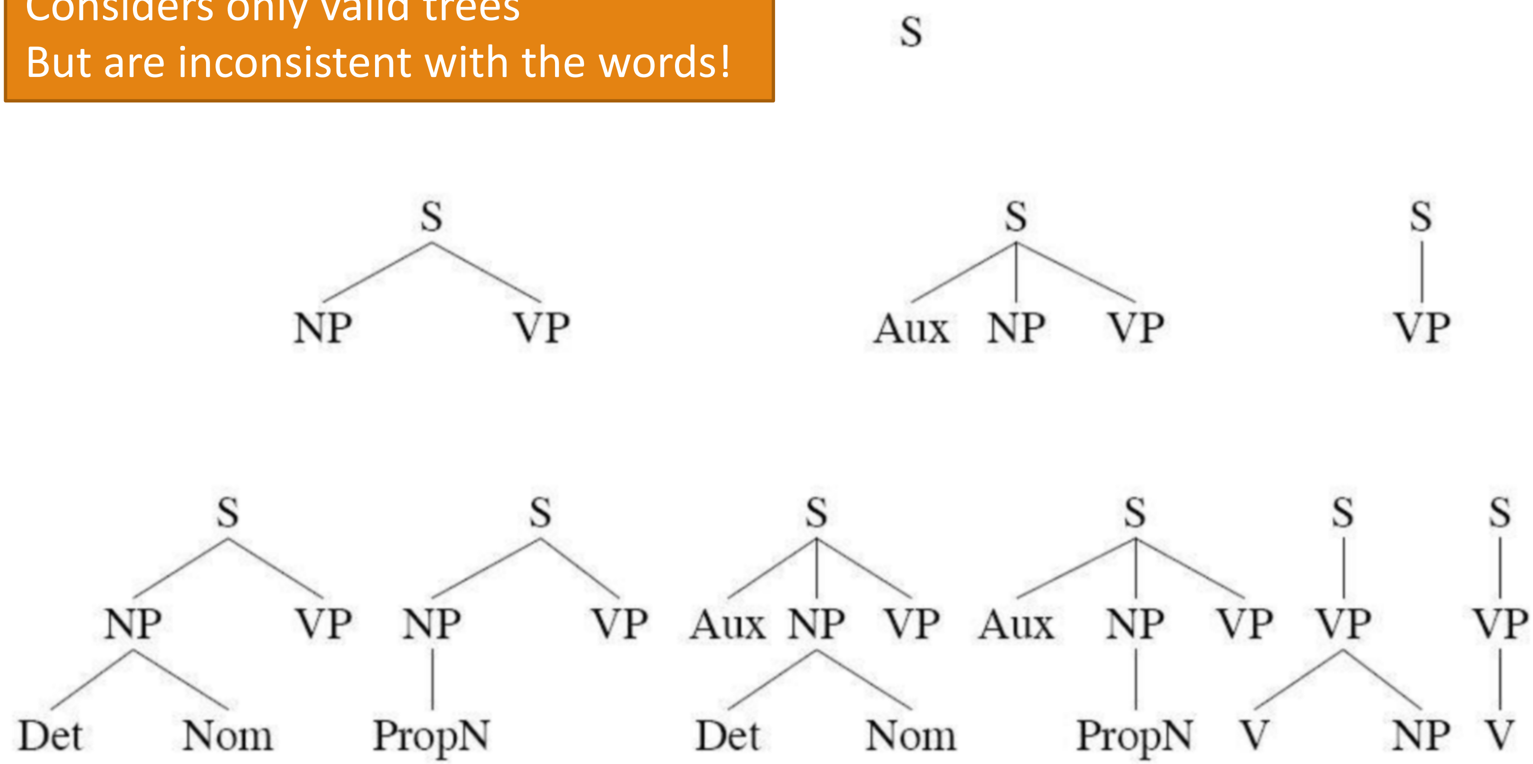
Even with small grammars, brute force grows exponentially!

"Book that flight"

# Top Down Parsing

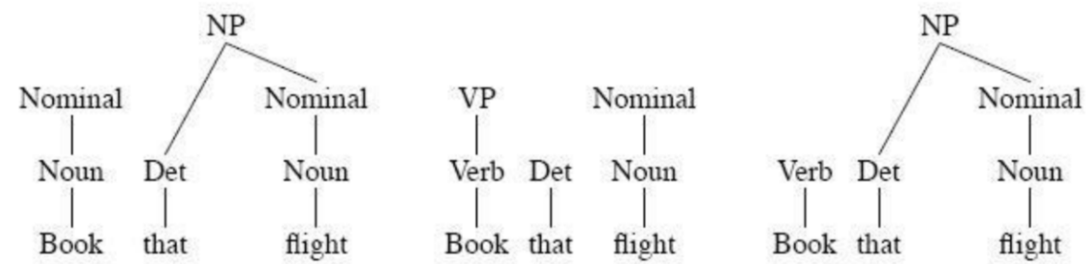
Considers only valid trees
But are inconsistent with the words!

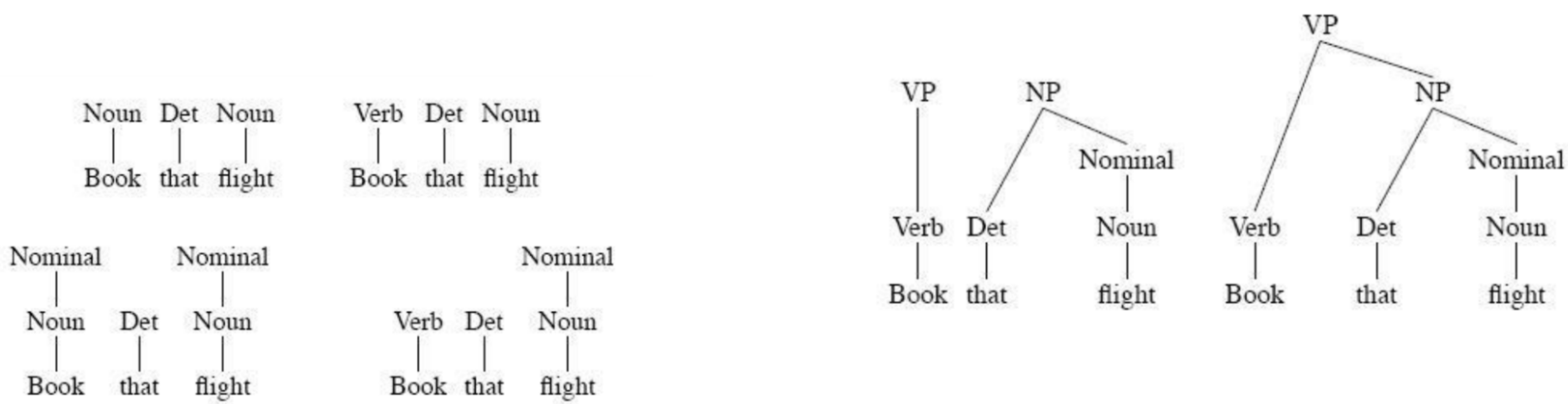
"Book that flight"



# Bottom-up Parsing

"Book that flight"




Builds only consistent trees
But most of them are invalid (don't go anywhere)!

# Chomsky Normal Form

Context free grammar where all non-terminals to go:
- 2 non-terminals, or
- A single terminal

$A \rightarrow B\ C$ $\qquad D \rightarrow w$

**Converting to CNF**

Case 1

$$A \rightarrow B \quad B \rightarrow C\ D \quad B \rightarrow w \quad \Rightarrow \quad A \rightarrow C\ D \quad A \rightarrow w$$

Case 2

$$A \rightarrow B\ C\ D\ E \quad \Rightarrow \quad A \rightarrow X\ E \quad X \rightarrow Y\ D \quad Y \rightarrow B\ C$$

| Original Grammar | Chomsky Normal Form |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

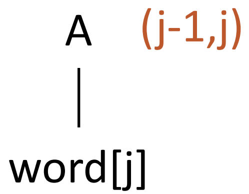
# Dynamic Programming

table[i,j] = Set of all valid non-terminals for the constituent span (i,j)

**Base case**

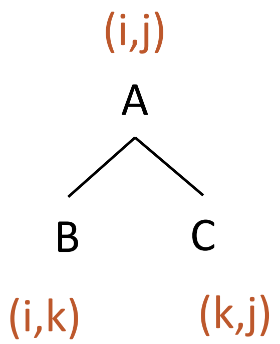
Rule: $A \rightarrow word[j]$

A should be in table[j-1,j]




**Recursion**

Rule: $A \rightarrow B\ C$



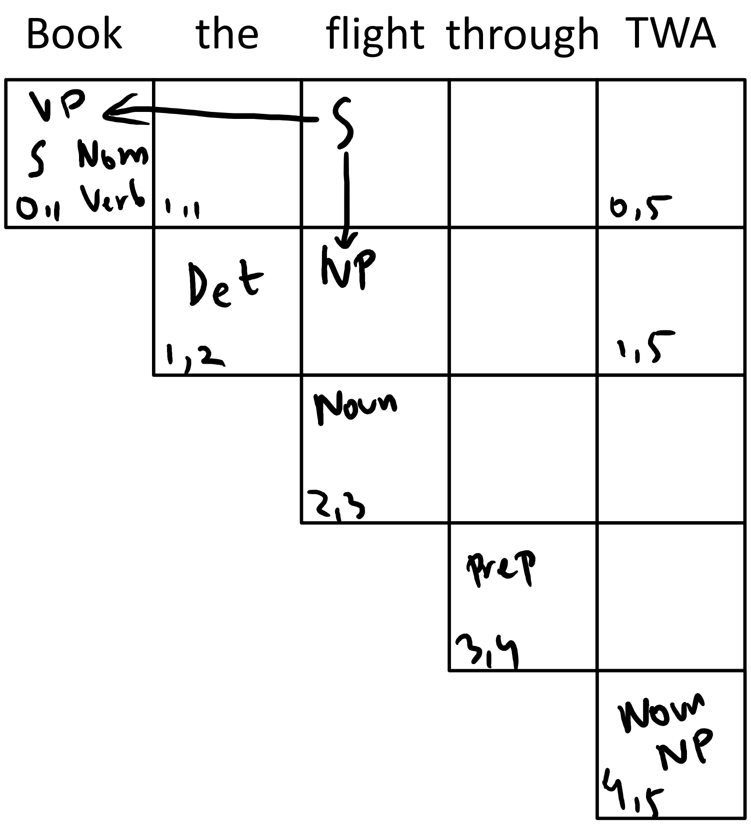

If you find a k such that

B is in table[i,k], and

C is in table[k,j], then A should be in table[i,j]

# CKY Algorithm

S → NP VP
S → X1 VP
X1 → Aux NP
S → book | include | prefer
S → Verb NP
S → X2 PP
S → Verb PP
S → VP PP
NP → I | she | me
NP → TWA | Houston
NP → Det Nominal
Nominal → book | flight | meal | money
Nominal → Nominal Noun
Nominal → Nominal PP
VP → book | include | prefer
VP → Verb NP
VP → X2 PP
X2 → Verb NP
VP → Verb PP
VP → VP PP
PP → Preposition NP



# Outline

Syntactic Parsing

Context Free Grammars

Parsing: CKY Algorithm

# Upcoming…

**Homework**

- Homework 2 is due in a week: February 13, 2017
- Homework 1 grades will be available tonight

**Project**

- Proposal is due on tonight
- Only 2 pages

**Summaries**

- Paper summaries: February 17, February 28, March 14
- Only 1 page each