Bottom-up Chart Parsing: the CKY algorithm

Data Structures and Algorithms for Computational Linguistics III (ISCL-BA-07)

Çağrı Çöltekin
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2021/22

version: 9cfc034 @2022-02-02

Parsing so far

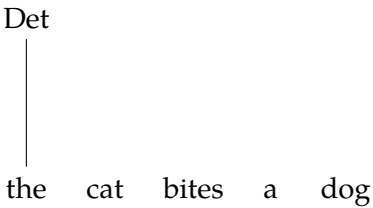
- Parsing is the task of automatic syntactic analysis
- For most practical purposes, context-free grammars are the most useful formalism for parsing
- We can formulate parsing as
  - Top-down: begin with the start symbol, try to *produce* the input string to be parsed
  - Bottom up: begin with the input, and try to *reduce* it to the start symbol
- Both strategies can be cast as search with backtracking
- Backtracking parsers are inefficient: they recompute sub-trees multiple times

Bottom-up parsing as search

the cat bites a dog

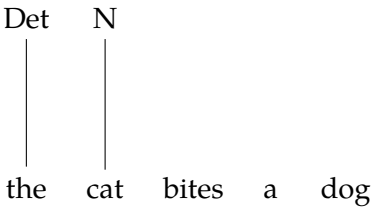
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



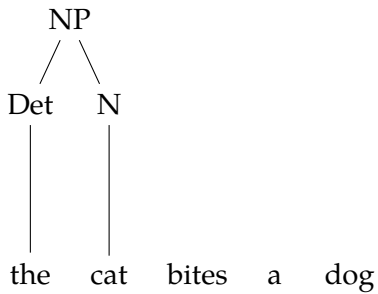
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow \text{the} \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



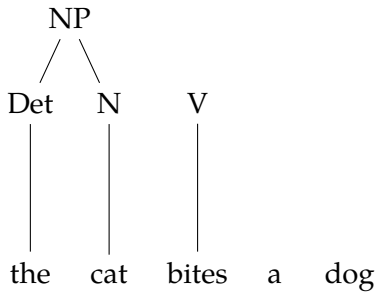
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



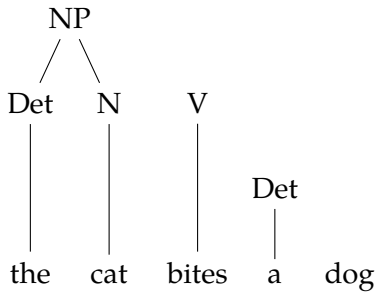
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



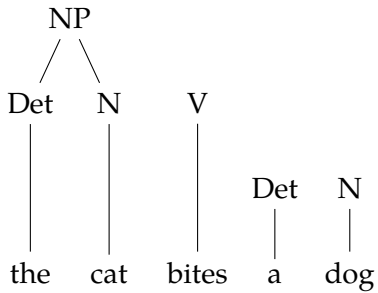
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



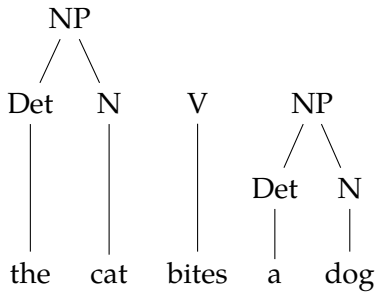
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



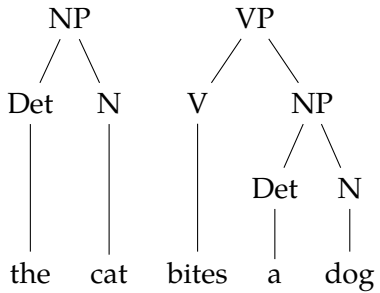
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



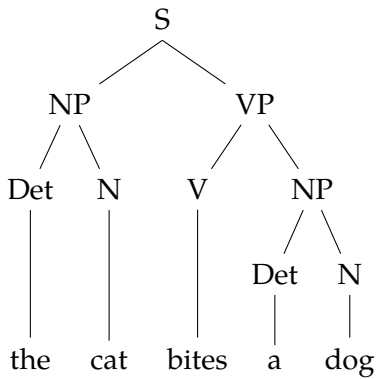
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
Det &\rightarrow a \\
Det &\rightarrow the \\
N &\rightarrow cat \\
N &\rightarrow dog \\
V &\rightarrow bites \\
N &\rightarrow bites
\end{aligned}
$$

Bottom-up parsing as search



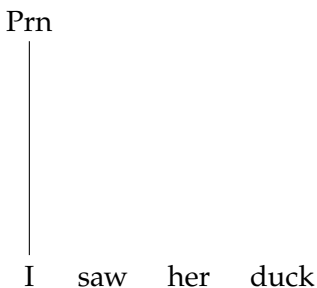
$$
\begin{aligned}
S &\rightarrow \text{NP VP} \\
NP &\rightarrow \text{Det N} \\
VP &\rightarrow \text{V NP} \\
VP &\rightarrow \text{V} \\
Det &\rightarrow \text{a} \\
Det &\rightarrow \text{the} \\
N &\rightarrow \text{cat} \\
N &\rightarrow \text{dog} \\
V &\rightarrow \text{bites} \\
N &\rightarrow \text{bites}
\end{aligned}
$$

Dealing with ambiguity

I saw her duck

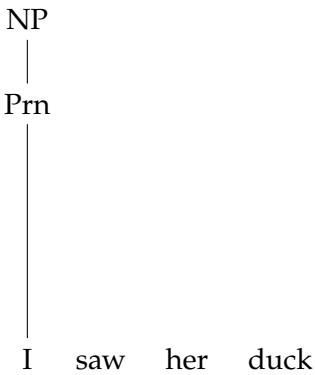
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I ←
Prn → she
Prn → her

Dealing with ambiguity



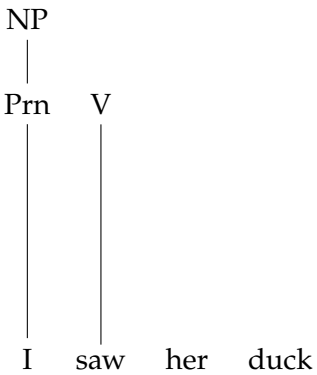
S → NP VP
NP → Prn N
NP → Prn ←
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



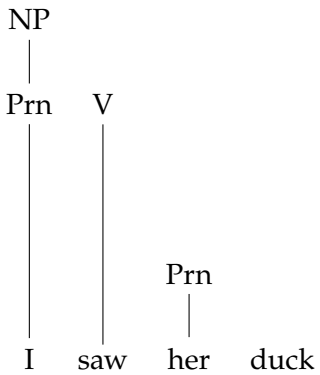
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw ←
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



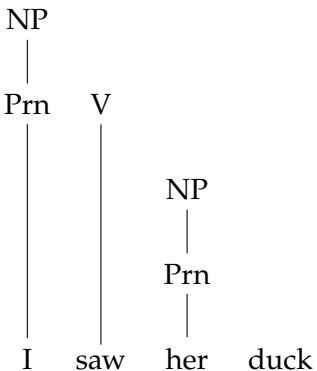
S	$\rightarrow$ NP VP	
NP	$\rightarrow$ Prn N	
NP	$\rightarrow$ Prn	
VP	$\rightarrow$ V NP	
VP	$\rightarrow$ V	
VP	$\rightarrow$ V S	
N	$\rightarrow$ duck	
V	$\rightarrow$ duck	
V	$\rightarrow$ saw	
Prn	$\rightarrow$ I	
Prn	$\rightarrow$ she	
Prn	$\rightarrow$ her	$\leftarrow$

Dealing with ambiguity



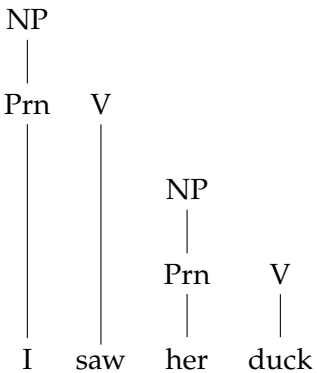
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
NP &\rightarrow Prn \qquad \leftarrow \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
VP &\rightarrow V\ S \\
N &\rightarrow \text{duck} \\
V &\rightarrow \text{duck} \\
V &\rightarrow \text{saw} \\
Prn &\rightarrow \text{I} \\
Prn &\rightarrow \text{she} \\
Prn &\rightarrow \text{her}
\end{aligned}
$$

Dealing with ambiguity



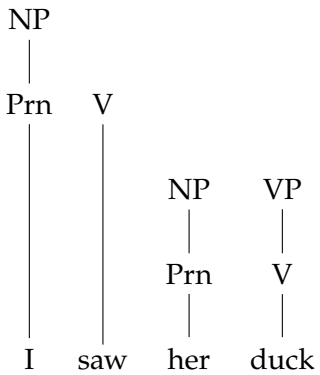
S → NP VP
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck ←
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



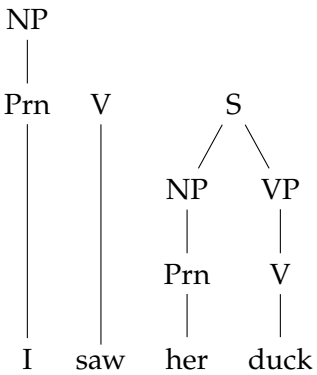
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
NP &\rightarrow Prn \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \qquad \leftarrow \\
VP &\rightarrow V\ S \\
N &\rightarrow duck \\
V &\rightarrow duck \\
V &\rightarrow saw \\
Prn &\rightarrow I \\
Prn &\rightarrow she \\
Prn &\rightarrow her
\end{aligned}
$$

Dealing with ambiguity



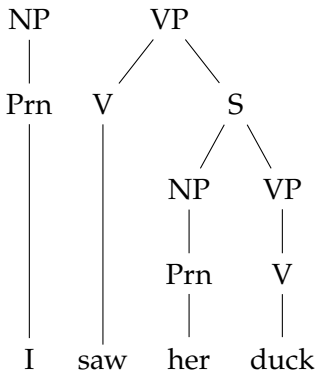
S → NP VP ←
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



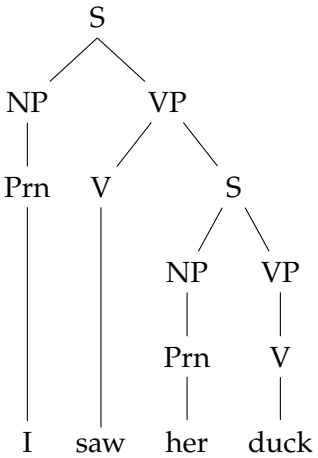
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
NP &\rightarrow Prn \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
VP &\rightarrow V\ S \quad \leftarrow \\
N &\rightarrow duck \\
V &\rightarrow duck \\
V &\rightarrow saw \\
Prn &\rightarrow I \\
Prn &\rightarrow she \\
Prn &\rightarrow her
\end{aligned}
$$

Dealing with ambiguity



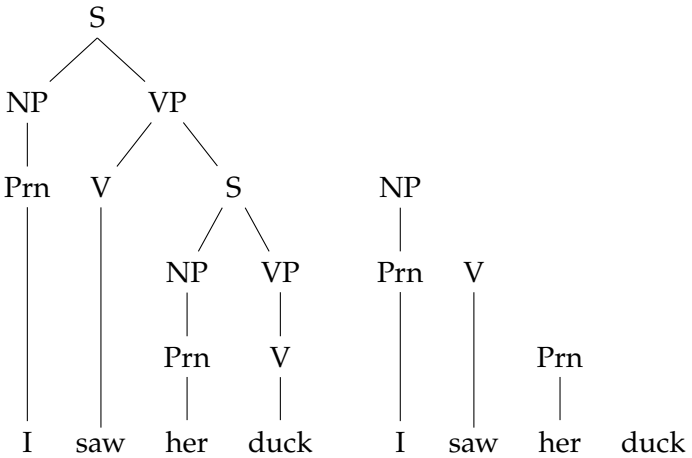
S → NP VP ←
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



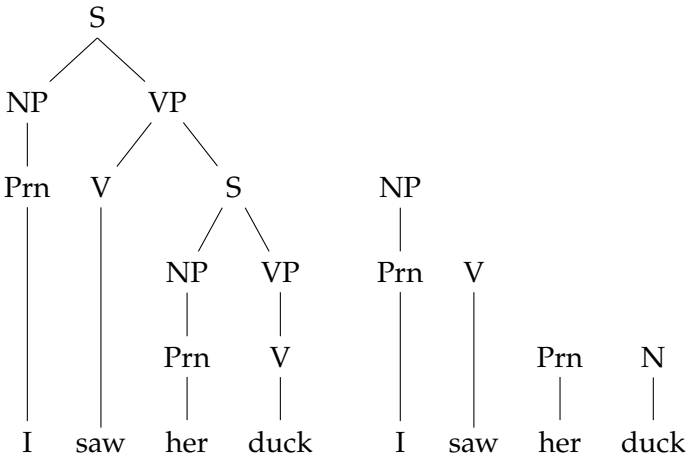
$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
NP &\rightarrow Prn \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
VP &\rightarrow V\ S \\
N &\rightarrow duck \\
V &\rightarrow duck \\
V &\rightarrow saw \\
Prn &\rightarrow I \\
Prn &\rightarrow she \\
Prn &\rightarrow her
\end{aligned}
$$

Dealing with ambiguity



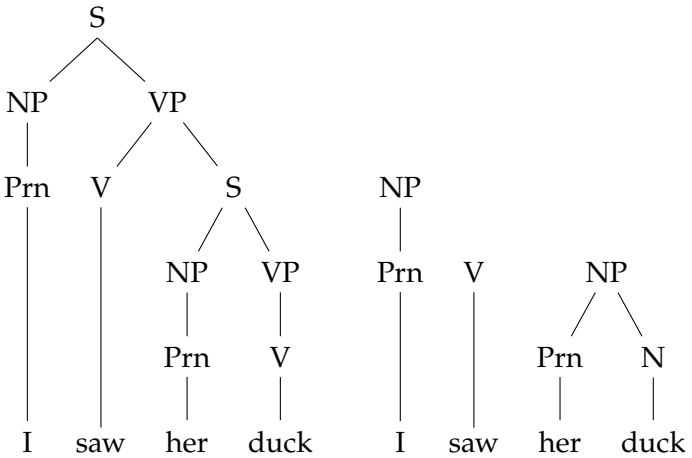
S	→ NP VP	
NP	→ Prn N	
NP	→ Prn	
VP	→ V NP	
VP	→ V	
VP	→ V S	
N	→ duck	←
V	→ duck	
V	→ saw	
Prn	→ I	
Prn	→ she	
Prn	→ her	

Dealing with ambiguity



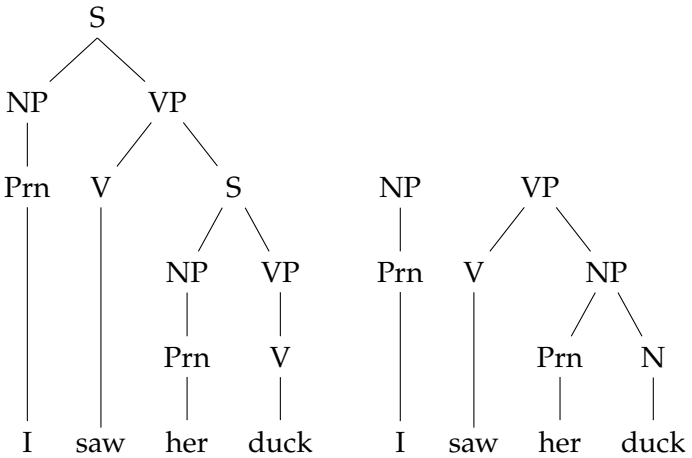
S → NP VP
NP → Prn N ←
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



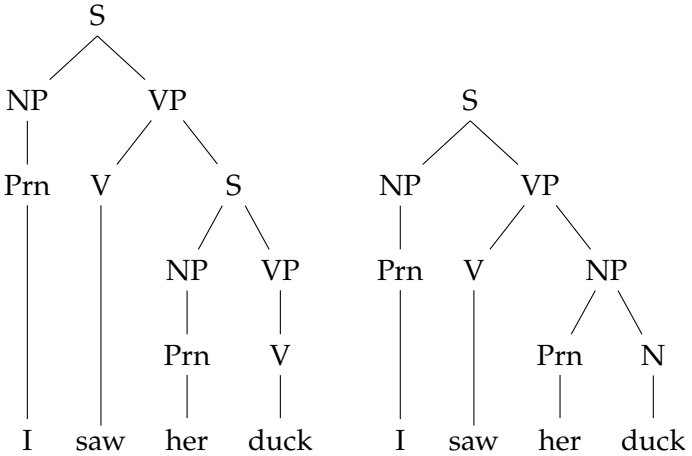
S → NP VP
NP → Prn N
NP → Prn
VP → V NP ←
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

Dealing with ambiguity



S → NP VP ←
NP → Prn N
NP → Prn
VP → V NP
VP → V
VP → V S
N → duck
V → duck
V → saw
Prn → I
Prn → she
Prn → her

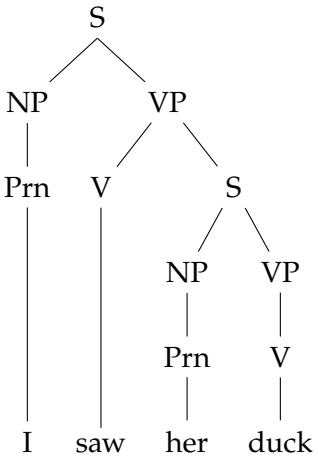
Dealing with ambiguity

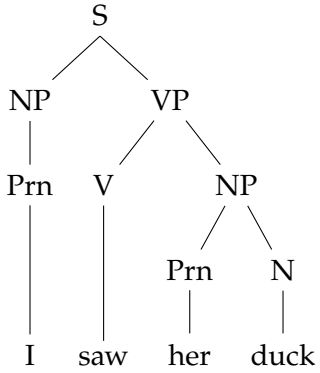


$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
NP &\rightarrow Prn \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
VP &\rightarrow V\ S \\
N &\rightarrow duck \\
V &\rightarrow duck \\
V &\rightarrow saw \\
Prn &\rightarrow I \\
Prn &\rightarrow she \\
Prn &\rightarrow her
\end{aligned}
$$

How to represent multiple parses

parse forest grammar





$$
\begin{aligned}
S_{0:4} &\rightarrow NP_{0:1}\ VP_{1:4} \\
NP_{0:1} &\rightarrow Prn_{0:1} \\
Prn_{0:1} &\rightarrow I_{0:1} \\
VP_{1:4} &\rightarrow V_{1:2}\ S_{2:4} \\
V_{1:2} &\rightarrow saw_{1:2} \\
S_{2:4} &\rightarrow Prn_{2:3}\ V_{3:4} \\
V_{3:4} &\rightarrow duck_{3:4} \\
\hline
VP_{1:4} &\rightarrow V_{1:2}\ NP_{2:4} \\
NP_{2:4} &\rightarrow Prn_{2:3}\ N_{3:4}
\end{aligned}
$$

CKY algorithm

- The CKY (Cocke–Kasami–Younger) parsing algorithm is a dynamic programming algorithm
- It processes the input *bottom up*, and saves the intermediate results on a *chart*
- Time complexity for *recognition* is $O(n^3)$
- Space complexity is $O(n^2)$
- It requires the CFG to be in *Chomsky normal form* (CNF) (can somewhat be relaxed, but not common)

Chomsky normal form (CNF)

- A CFG is in CNF, if the rewrite rules are in one of the following forms
  - $A \rightarrow B\ C$
  - $A \rightarrow a$

  where A, B, C are non-terminals and a is a terminal
- Any CFG can be converted to CNF
- Resulting grammar is *weakly equivalent* to the original grammar:
  - it generates/accepts the same language
  - but the derivations are different

Converting to CNF: example

$$
\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{S} &\rightarrow \text{Aux NP VP} \\
\text{NP} &\rightarrow \text{the N} \\
\text{VP} &\rightarrow \text{V NP} \\
\text{VP} &\rightarrow \text{V} \\
\text{N} &\rightarrow \text{cat} \\
\text{N} &\rightarrow \text{dog} \\
\text{V} &\rightarrow \text{bites} \\
\text{N} &\rightarrow \text{bites}
\end{aligned}
$$

Converting to CNF: example

S → NP VP
S → Aux NP VP
NP → the N
VP → V NP
VP → V
N → cat
N → dog
V → bites
N → bites

- S → Aux NP VP
  S → Aux NP VP ⇒ S → Aux X
  X → NP VP

Converting to CNF: example

$$
\begin{array}{lcl}
\text{S} & \rightarrow & \text{NP VP} \\
\text{S} & \rightarrow & \text{Aux NP VP} \\
\text{NP} & \rightarrow & \text{the N} \\
\text{VP} & \rightarrow & \text{V NP} \\
\text{VP} & \rightarrow & \text{V} \\
\text{N} & \rightarrow & \text{cat} \\
\text{N} & \rightarrow & \text{dog} \\
\text{V} & \rightarrow & \text{bites} \\
\text{N} & \rightarrow & \text{bites}
\end{array}
$$

- $\text{S} \rightarrow \text{Aux NP VP}$
  $\text{S} \rightarrow \text{Aux NP VP} \quad \Rightarrow \quad \text{S} \rightarrow \text{Aux X}$
  $\text{X} \rightarrow \text{NP VP}$
- $\text{NP} \rightarrow \text{the N}$
  $\text{NP} \rightarrow \text{the N} \quad \Rightarrow \quad \text{NP} \rightarrow \text{X N}$
  $\text{X} \rightarrow \text{the}$

Converting to CNF: example

S → NP VP
S → Aux NP VP
NP → the N
VP → V NP
VP → V
N → cat
N → dog
V → bites
N → bites

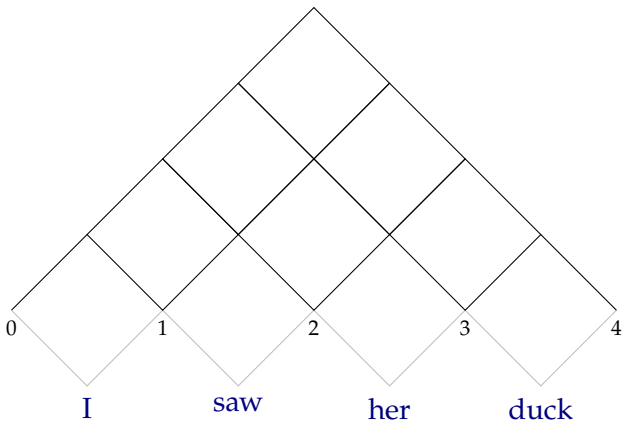
- S → Aux NP VP
  S → Aux NP VP ⇒ S → Aux X
  X → NP VP
- NP → the N
  NP → the N ⇒ NP → X N
  X → the
- VP → V
  VP → V ⇒ VP → bites

Converting to CNF

1. Eliminate the ϵ rules: if $A \rightarrow \epsilon$ is in the grammar
   - replace any rule $B \rightarrow \alpha\ A\ \beta$ with two rules
     $B \rightarrow \alpha\ \beta$
     $B \rightarrow \alpha\ A'\ \beta$
   - add $A' \rightarrow \alpha$ for all α (except ϵ) whose LHS is A
   - repeat the process for newly created ϵ rules
   - remove the rules with ϵ on the RHS (except $S \rightarrow \epsilon$)
2. Eliminate unit rules: for a rule $A \rightarrow B$
   - Replace the rule with $A \rightarrow \alpha_1 \mid ... \mid \alpha_n$, where α_1, ..., α_n are all RHS or rule B
   - Remove the rule $A \rightarrow B$
   - Repeat the process until no unit rules remain
3. Binarize all the non-binary rules with non-terminal on the RHS: for a rule $A \rightarrow X_1\ X_2\ ...\ X_n$:
   - Replace the rule with $A \rightarrow A_1\ X_2 ... X_n$, and add $A_1 \rightarrow X_1\ X_2$
   - Repeat the process until all new rules are binary

CKY demonstration

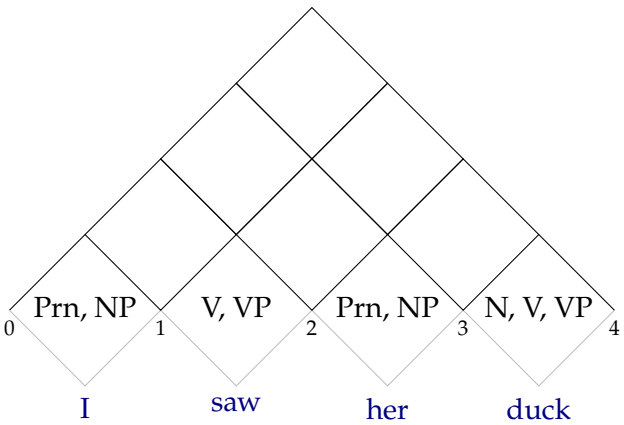
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

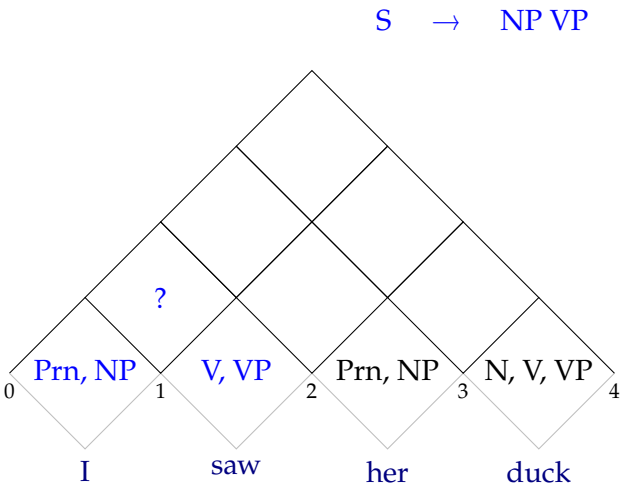
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

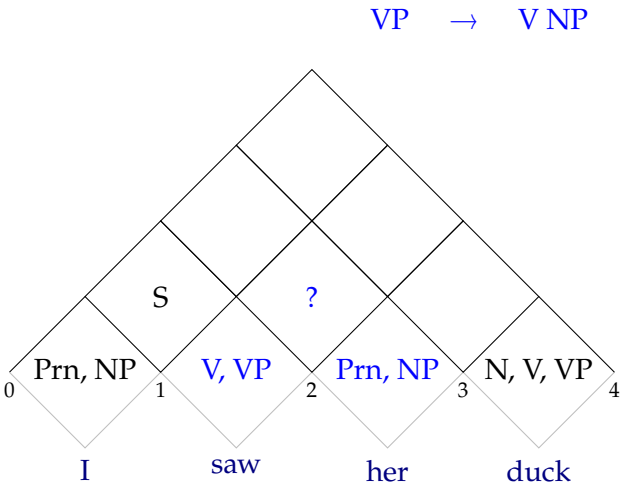
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

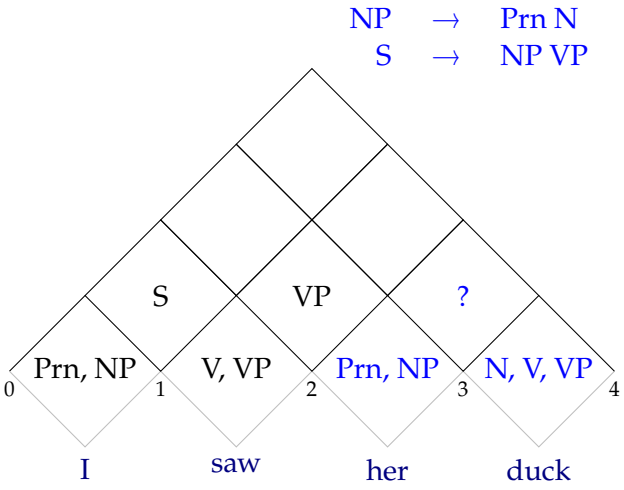
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

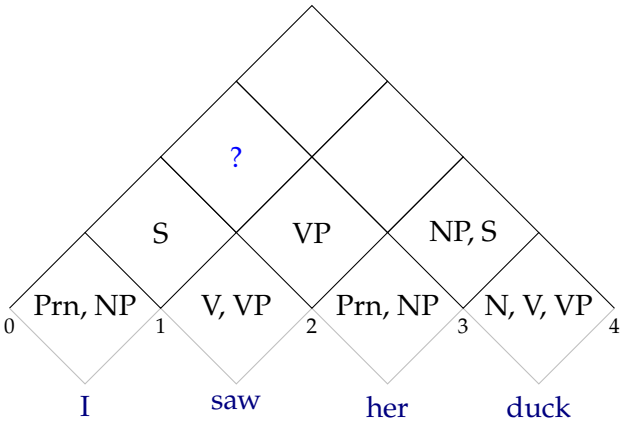
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

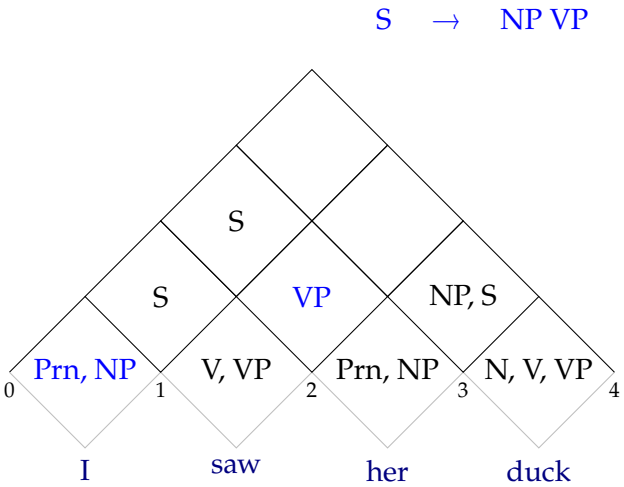
an ambiguous example



S $\rightarrow$ NP VP
NP $\rightarrow$ Prn N
VP $\rightarrow$ V NP
VP $\rightarrow$ V S
N $\rightarrow$ duck
VP $\rightarrow$ duck | saw
V $\rightarrow$ duck | saw
Prn $\rightarrow$ I | she | her
NP $\rightarrow$ I | she | her

CKY demonstration

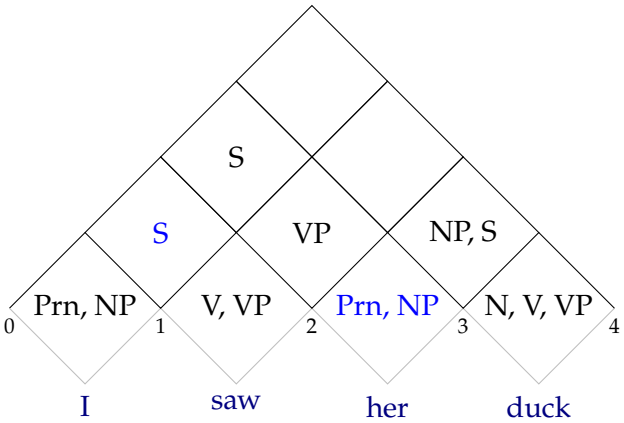
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

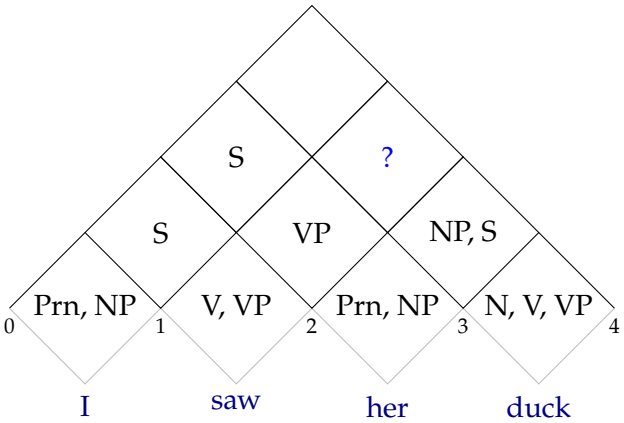
an ambiguous example



S $\rightarrow$ NP VP
NP $\rightarrow$ Prn N
VP $\rightarrow$ V NP
VP $\rightarrow$ V S
N $\rightarrow$ duck
VP $\rightarrow$ duck | saw
V $\rightarrow$ duck | saw
Prn $\rightarrow$ I | she | her
NP $\rightarrow$ I | she | her

CKY demonstration

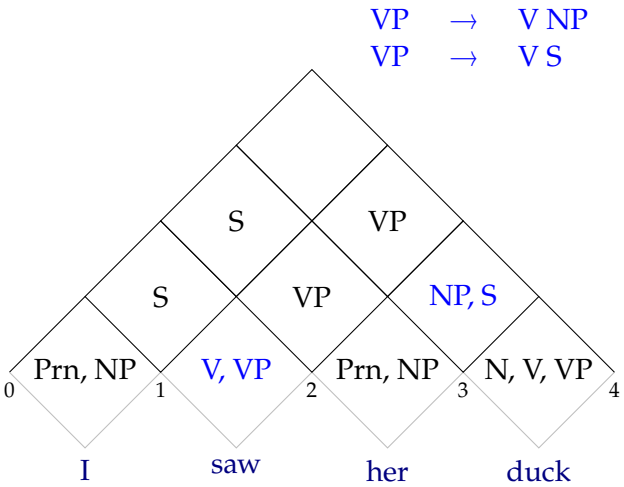
an ambiguous example



S $\rightarrow$ NP VP
NP $\rightarrow$ Prn N
VP $\rightarrow$ V NP
VP $\rightarrow$ V S
N $\rightarrow$ duck
VP $\rightarrow$ duck | saw
V $\rightarrow$ duck | saw
Prn $\rightarrow$ I | she | her
NP $\rightarrow$ I | she | her

CKY demonstration

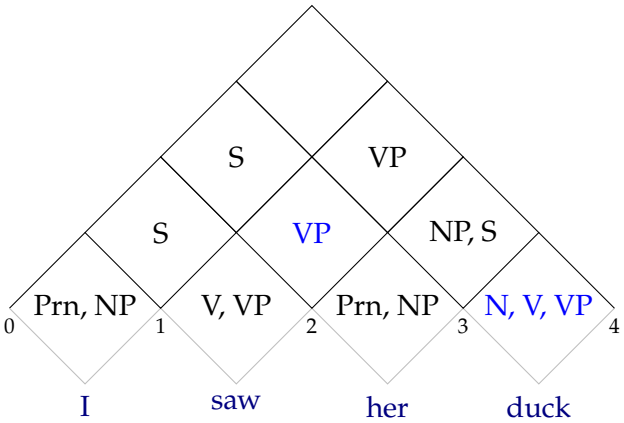
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

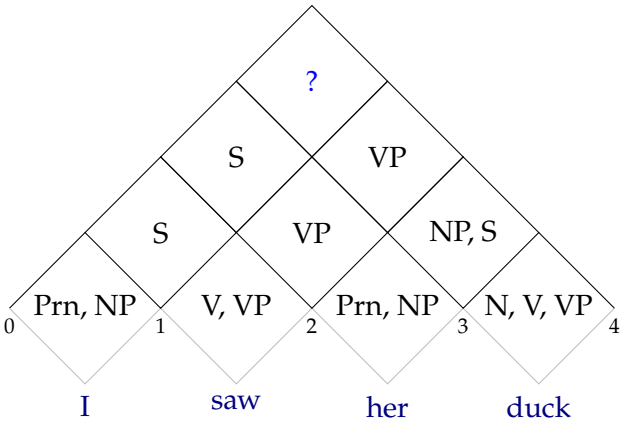
an ambiguous example



S $\rightarrow$ NP VP
NP $\rightarrow$ Prn N
VP $\rightarrow$ V NP
VP $\rightarrow$ V S
N $\rightarrow$ duck
VP $\rightarrow$ duck | saw
V $\rightarrow$ duck | saw
Prn $\rightarrow$ I | she | her
NP $\rightarrow$ I | she | her

CKY demonstration

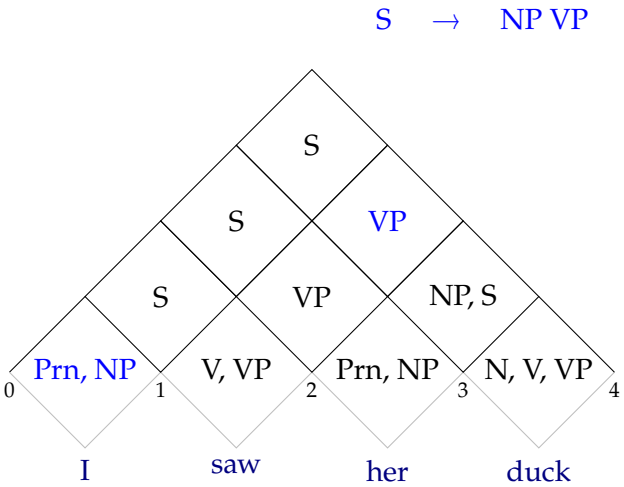
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

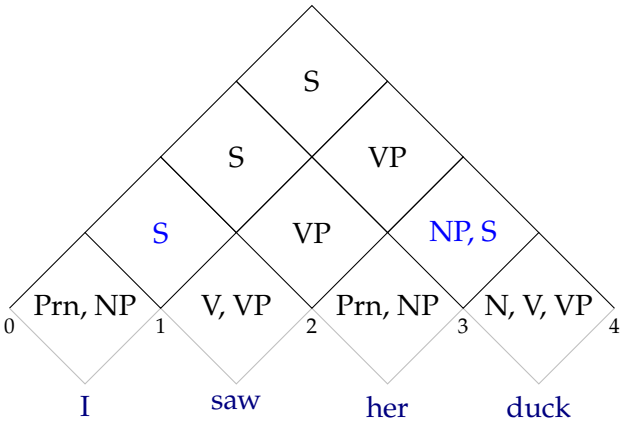
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

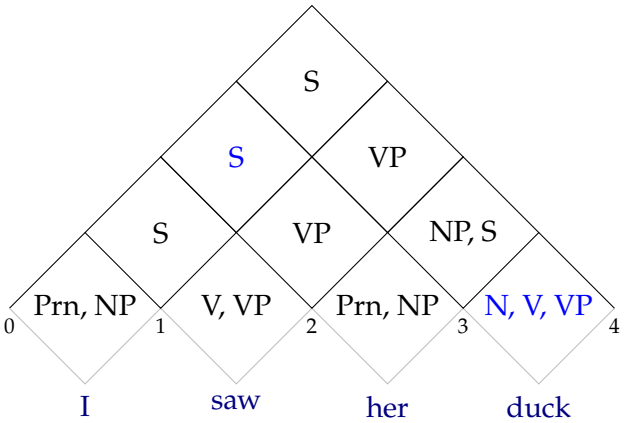
an ambiguous example



S → NP VP
NP → Prn N
VP → V NP
VP → V S
N → duck
VP → duck | saw
V → duck | saw
Prn → I | she | her
NP → I | she | her

CKY demonstration

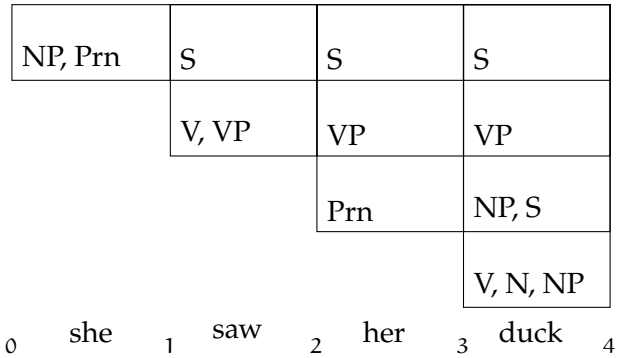
an ambiguous example



$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Prn\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V\ S \\
N &\rightarrow duck \\
VP &\rightarrow duck \mid saw \\
V &\rightarrow duck \mid saw \\
Prn &\rightarrow I \mid she \mid her \\
NP &\rightarrow I \mid she \mid her
\end{aligned}
$$

CKY demonstration: the chart

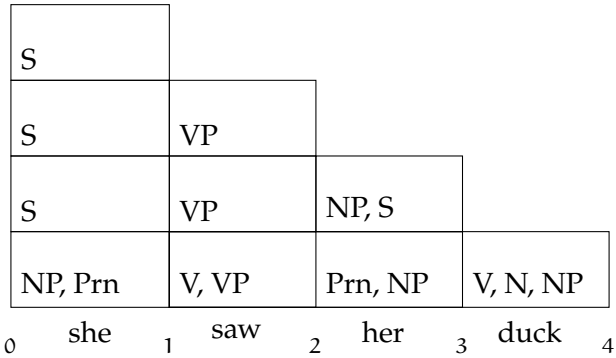
our chart is a 2D array



Space complexity is $O(n^2)$.

CKY demonstration: the chart

our chart is a 2D array – this is more convenient for programming



Space complexity is $O(n^2)$.

Parsing vs. recognition

- We went through a recognition example
- Note that the algorithm is not directional: it takes the complete input
- Recognition accepts or rejects a sentence based on a grammar
- For parsing, we want to know the derivations that yielded a correct parse
- To recover parse trees, we
  - follow the same procedure as recognition
  - add back links to keep track of the derivations

Chart parsing example (CKY parsing)

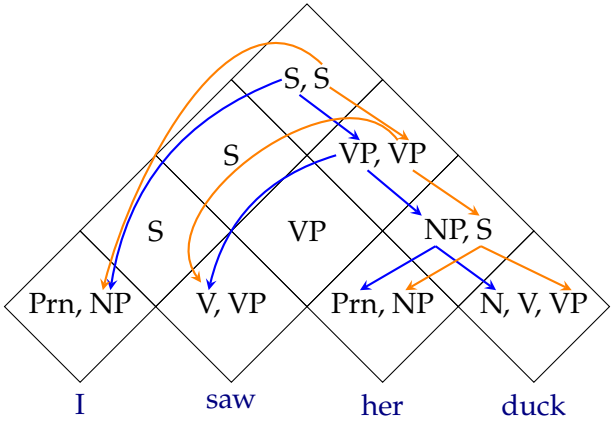
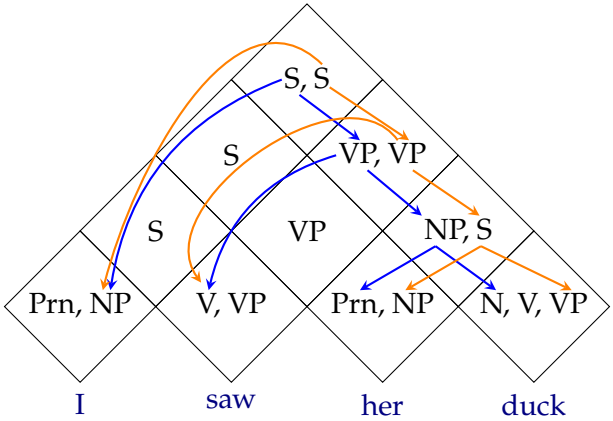


Chart parsing example (CKY parsing)



The chart stores a *parse forest* efficiently.

Summary

+ CKY avoids re-computing the analyses by storing the earlier analyses (of sub-spans) in a table
− It still computes lower level constituents that are not allowed by the grammar
− CKY requires the grammar to be in CNF
- CKY has $O(n^3)$ recognition complexity
- For parsing we need to keep track of backlinks
- CKY can efficiently store all possible parses in a chart
- Enumerating all possible parses have exponential complexity (worst case)
- Suggested reading: Jurafsky and Martin (2009, draft 3rd ed, section 13.2)

Summary

+ CKY avoids re-computing the analyses by storing the earlier analyses (of sub-spans) in a table
− It still computes lower level constituents that are not allowed by the grammar
− CKY requires the grammar to be in CNF
- CKY has $O(n^3)$ recognition complexity
- For parsing we need to keep track of backlinks
- CKY can efficiently store all possible parses in a chart
- Enumerating all possible parses have exponential complexity (worst case)
- Suggested reading: Jurafsky and Martin (2009, draft 3rd ed, section 13.2)

Next:

- Top-down chart parsing: Earley algorithm
- Suggested reading:
  - Jurafsky and Martin (2009, section 13.2.4)
  - Grune and Jacobs (2007, section 7.2)

Acknowledgments, references, additional reading material

Grune, Dick and Ceriel J.H. Jacobs (2007). *Parsing Techniques: A Practical Guide*. second. Monographs in Computer Science. The first edition is available at `http://dickgrune.com/Books/PTAPG_1st_Edition/BookBody.pdf`. Springer New York. ISBN: 9780387689548.

Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. second edition. Pearson Prentice Hall. ISBN: 978-0-13-504196-3.