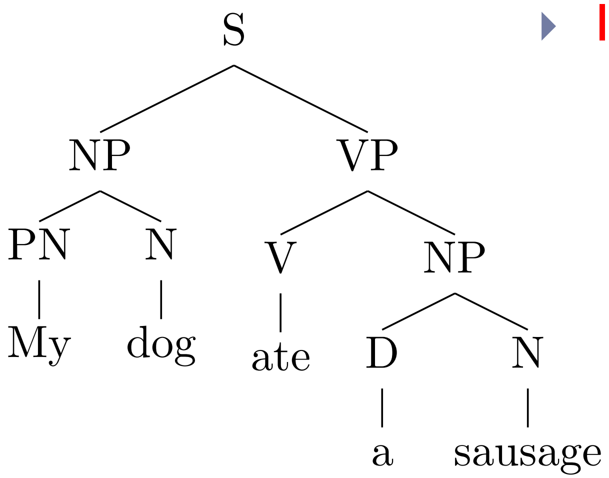
Foundations for Natural Language Processing
Lecture 15
Syntax and Parsing (part 2)

Ivan Titov

Last time

- We discussed syntax and ambiguity
- Context free grammars
- Classes of parsing algorithms
- Today:
  - CKY algorithm
  - Probabilistic CFGs, and CKY for PCFGs

Recap: Constituent trees



- Internal nodes correspond to phrases

S – a sentence

NP (Noun Phrase): My dog, a sandwich, lakes, ..

VP (Verb Phrase): ate a sausage, barked, …

PP (Prepositional phrases): with a friend, in a car, …

- Nodes immediately above words are PoS tags

PN – pronoun
D – determiner
V – verb
N – noun
P – preposition

Recap: An example grammar

$$V = \{S, VP, NP, PP, N, V, PN, P\}$$

$$\Sigma = \{girl, telescope, sandwich, I, saw, ate, with, in, a, the\}$$

$$S = \{S\}$$

$$R:$$

Inner rules

$S \rightarrow NP \ \ VP$ (NP A girl) (VP ate a sandwich)

$VP \rightarrow V$

$VP \rightarrow V \ \ NP$ (V ate) (NP a sandwich)

$VP \rightarrow VP \ \ PP$ (VP saw a girl) (PP with a telescope)

$NP \rightarrow NP \ \ PP$ (NP a girl) (PP with a sandwich)

$NP \rightarrow D \ \ N$ (D a) (N sandwich)

$NP \rightarrow PN$

$PP \rightarrow P \ \ NP$ (P with) (NP with a sandwich)

Preterminal rules

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$PN \rightarrow I$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

CKY algorithm (aka CYK)

- Cocke-Kasami-Younger algorithm
  - Independently discovered in late 60s / early 70s
- An efficient bottom-up parsing algorithm for CFGs
  - can be used both for the recognition and parsing problems
- Very important in NLP (and beyond)

- As we will see, it is generalizable to probabilistic modeling / PCFGs

Constraints on the grammar

- The basic CKY algorithm supports only rules in the *Chomsky Normal Form* (CNF):

$$C \to x$$

Unary preterminal rules, generation of words given PoS tags
$D \to the \quad N \to telescope$

$$C \to C_1 C_2$$

Binary inner rules (e.g., $S \to NP\ VP,\ NP \to D\ \ N$)

Constraints on the grammar

- The basic CKY algorithm supports only rules in the *Chomsky Normal Form* (CNF):

$$C \to x$$

Unary preterminal rules, generation of words given PoS tags
$D \to the \quad N \to telescope$

$$C \to C_1 C_2$$

Binary inner rules (e.g., $S \to NP\ VP,\ NP \to D\ \ N$)

- Any CFG can be converted to an equivalent CNF
  - Equivalent means that they define the same language
  - However (syntactic) trees will look differently
  - It is possible to address it but defining such transformations that allows for easy reverse transformation

Transformation to CNF form

- What one need to do to convert to CNF form
  - Get rid of empty (aka epsilon) productions: $C \rightarrow \epsilon$
  - Get rid of unary rules: $C \rightarrow C_1$
  - N-ary rules: $C \rightarrow C_1\ C_2 \ldots C_n \quad (n > 2)$

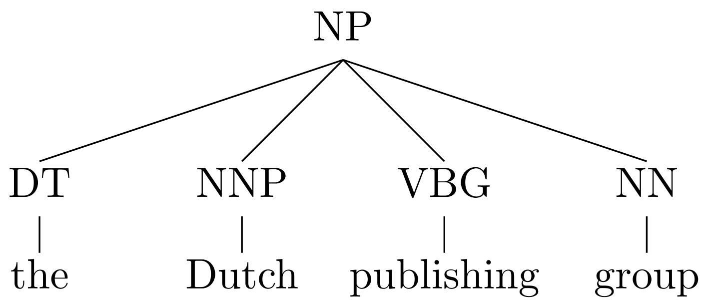
Generally not a problem as there are no empty production in the standard treebanks (or they can be disregarded)

Not a problem, as our CKY algorithm will support unary rules

Crucial to process them, as required for efficient parsing

Transformation to CNF form: binarization

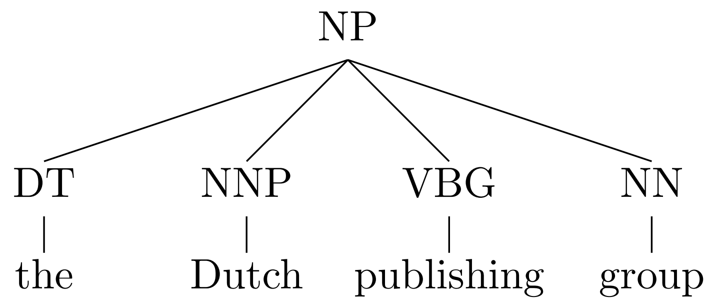
- Consider $NP \rightarrow DT \;\; NNP \;\; VBG \;\; NN$



- How do we get a set of binary rules which are equivalent?

Transformation to CNF form: binarization

- Consider $NP \rightarrow DT \;\; NNP \;\; VBG \;\; NN$



- How do we get a set of binary rules which are equivalent?

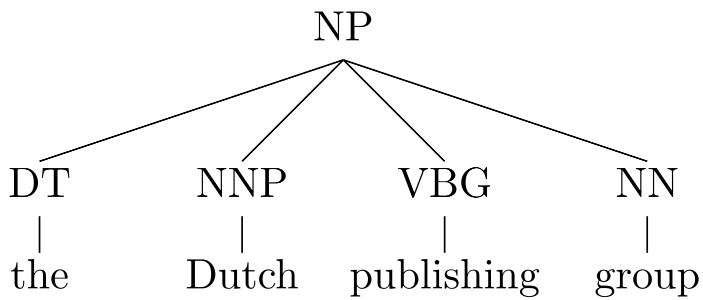
$$NP \rightarrow DT \;\; X$$

$$X \rightarrow NNP \;\; Y$$

$$Y \rightarrow VBG \;\; NN$$

Transformation to CNF form: binarization

- Consider $NP \rightarrow DT \;\; NNP \;\; VBG \;\; NN$



- How do we get a set of binary rules which are equivalent?

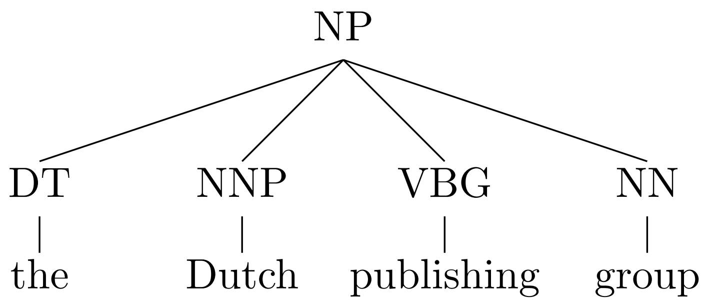
$$NP \rightarrow DT \;\; X$$
$$X \rightarrow NNP \;\; Y$$
$$Y \rightarrow VBG \;\; NN$$

- A more systematic way to refer to new non-terminals

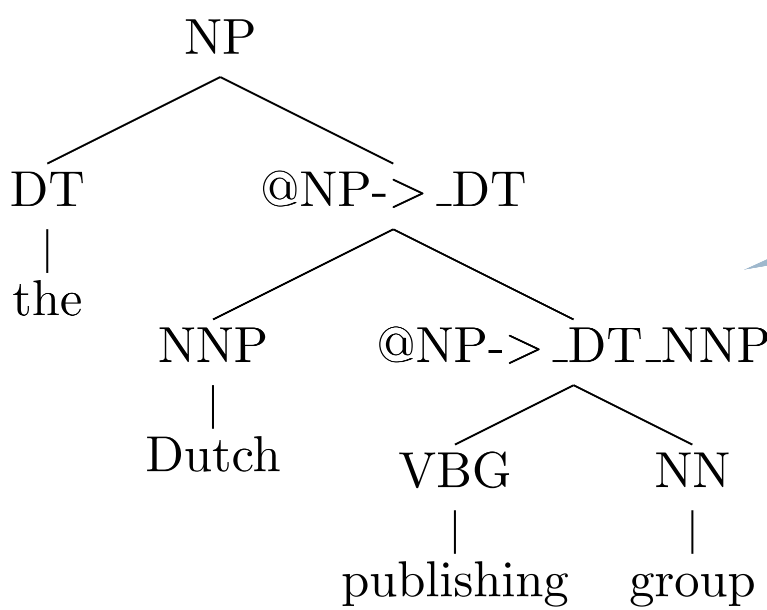
$$NP \rightarrow DT \;\; @NP|DT$$
$$@NP|DT \rightarrow NNP \;\; @NP|DT\_NNP$$
$$@NP|DT\_NNP \rightarrow VBG \;\; NN$$

Transformation to CNF form: binarization

- Instead of binarizing tules we can binarize trees on preprocessing:



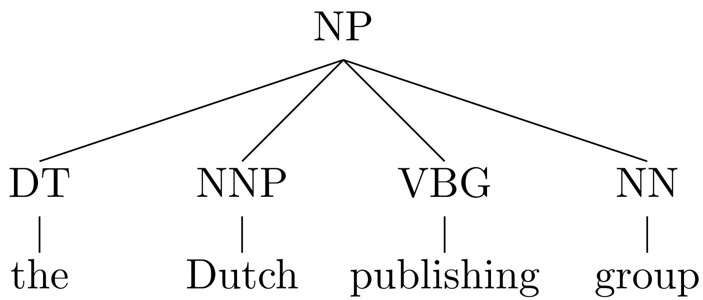
Also known as **lossless Markovization** in the context of PCFGs

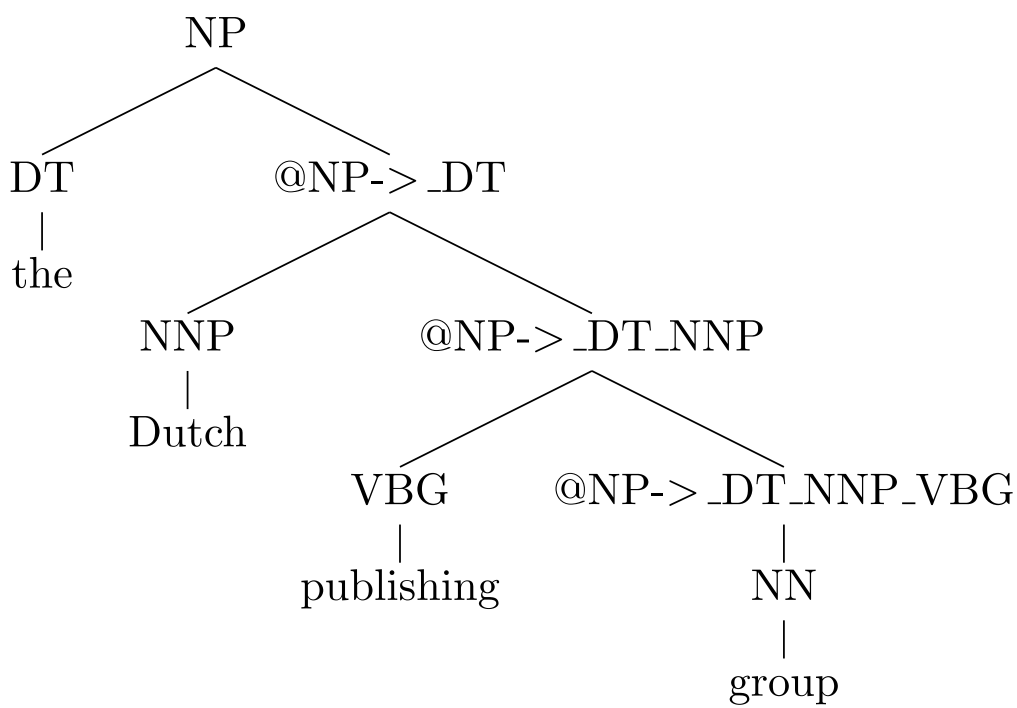


Can be easily reversed on postprocessing

Transformation to CNF form: binarization

- Instead of binarizing tules we can binarize trees on preprocessing:





CKY: Parsing task

start symbol

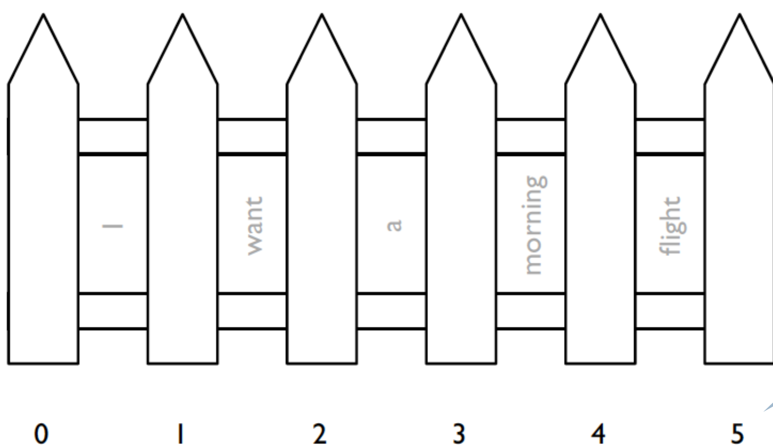
- We a given
  - a grammar $G = (V, \Sigma, R, S)$
  - a sequence of words $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$
- Our goal is to produce a parse tree for $\boldsymbol{w}$

CKY: Parsing task

start symbol

- We a given
  - a grammar $G = (V, \Sigma, R, S)$
  - a sequence of words $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$
- Our goal is to produce a parse tree for $\boldsymbol{w}$

- We need an easy way to refer to substrings of $\boldsymbol{w}$



indices refer to fenceposts

span (i, j) refers to words between fence posts *i* and *j*

Recall -- Key problems

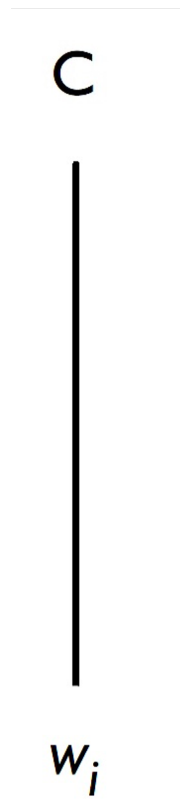
- Recognition problem: does the sentence belong to the language defined by CFG?
  - Is there a derivation which yields the sentence?
- Parsing problem: what is a derivation (tree) corresponding the sentence?
  - Probabilistic parsing: what is the most probable tree for the sentence?

Parsing one word

$$C \rightarrow w_i$$

w_i

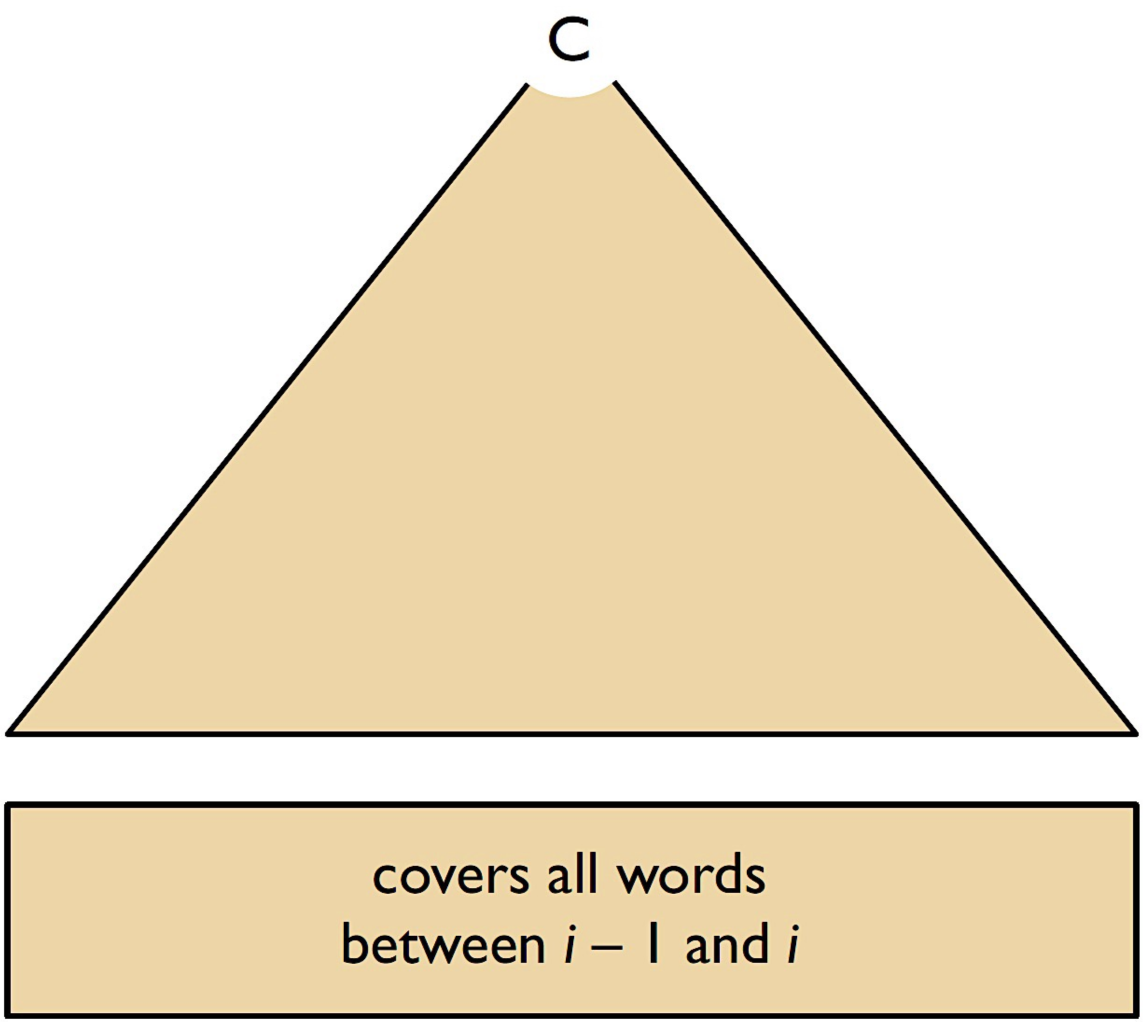
Parsing one word



$$C \to w_i$$

Parsing one word

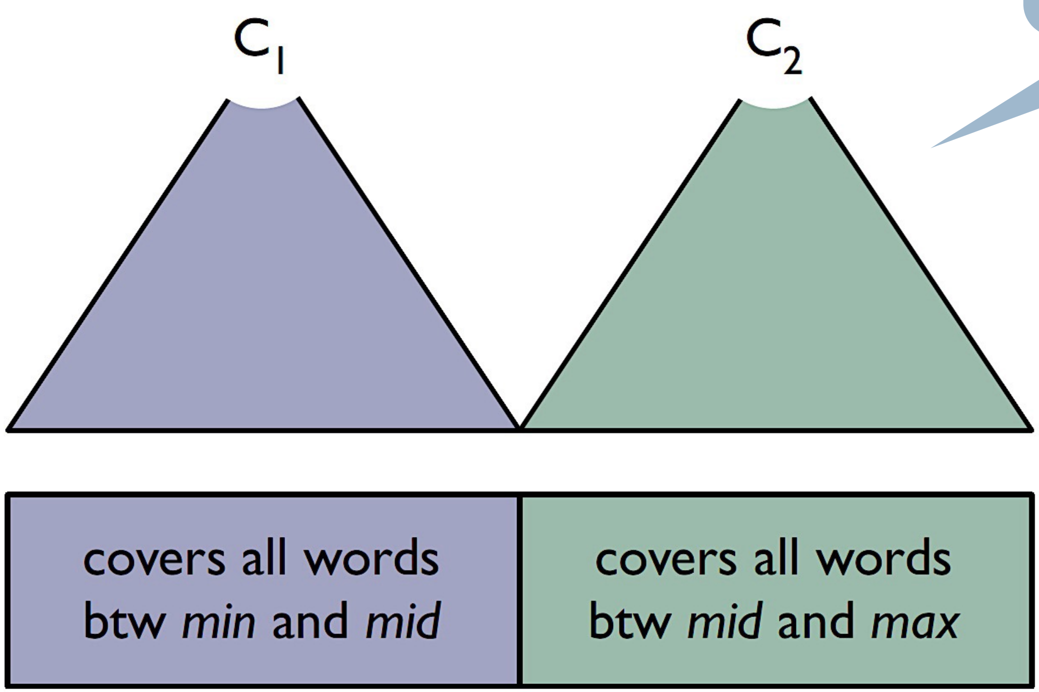
$$C \to w_i$$



Parsing longer spans

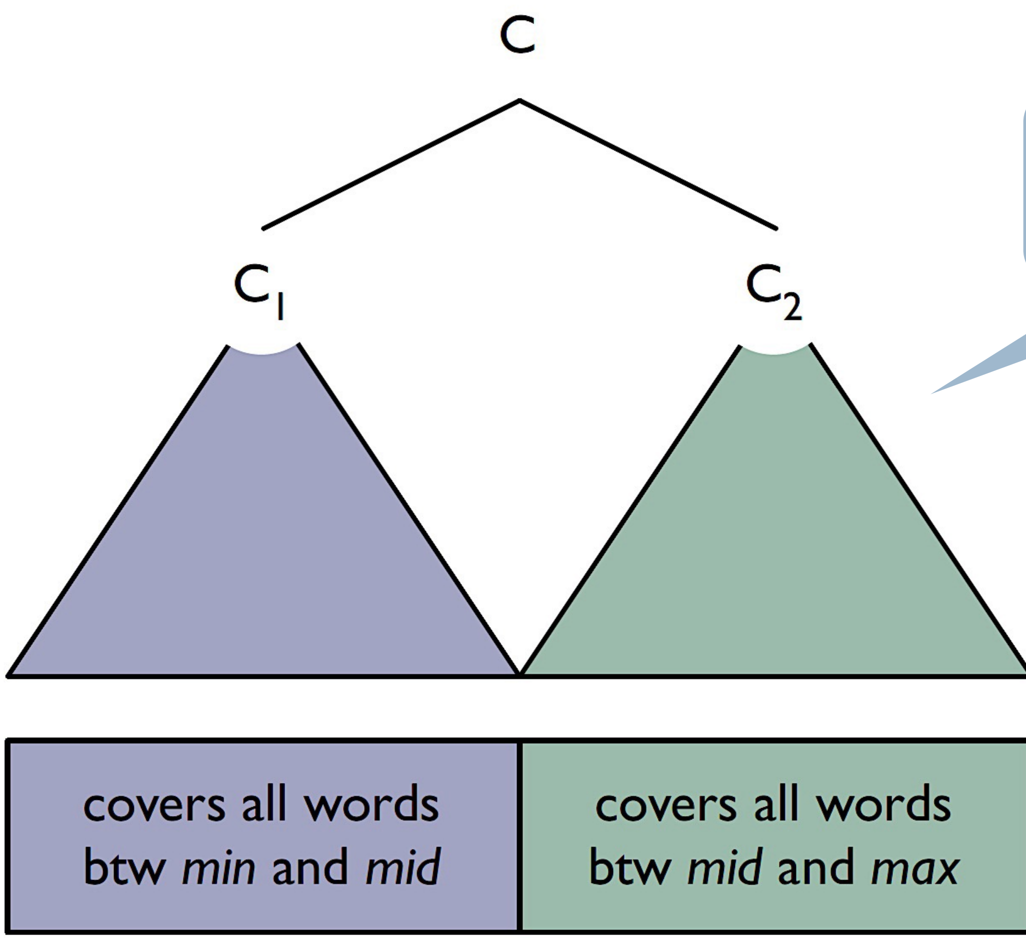
$$C \rightarrow C_1 \;\; C_2$$

Check through all C_1, C_2, mid



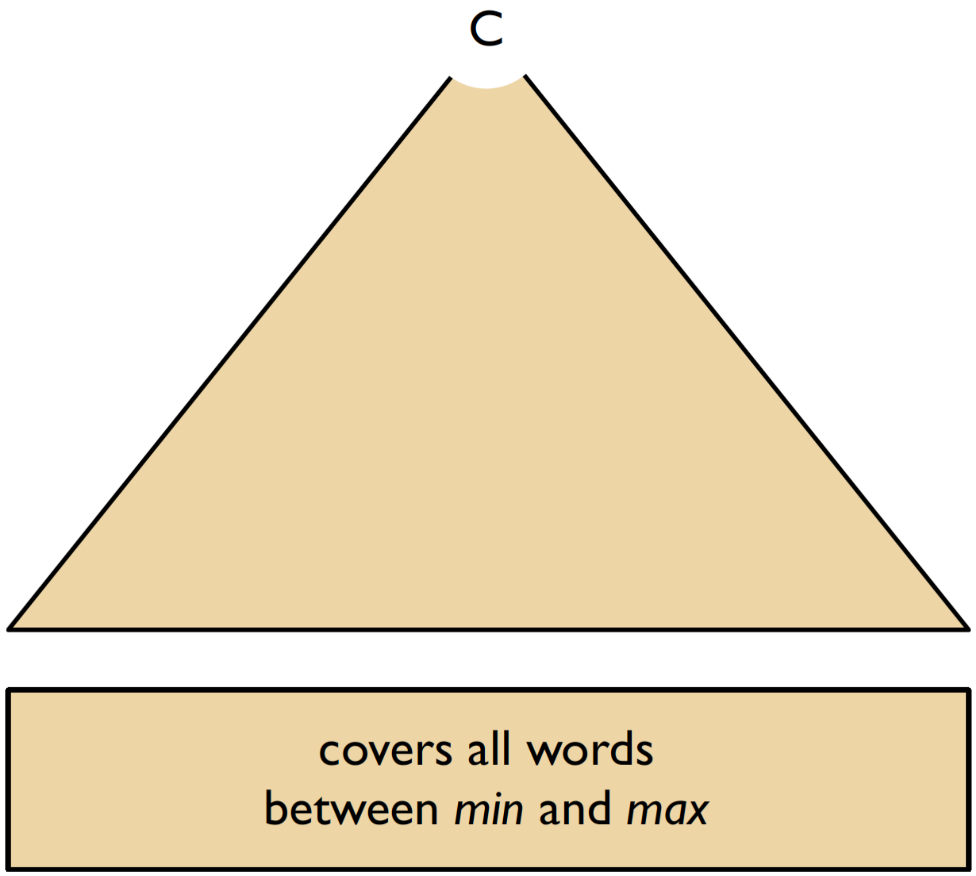
Parsing longer spans

$$C \rightarrow C_1 \;\; C_2$$



Check through all C_1, C_2, mid

Parsing longer spans



Signatures

- Applications of rules is independent of inner structure of a parse tree
- We only need to know the corresponding span and the root label of the tree
  - Its signature $[min, max, C]$

Also known as an *edge*

CKY idea

- Compute for every span a set of admissible labels (may be empty for some spans)
  - Start from small trees (single words) and proceed to larger ones
- When done, check if *S* is among admissible labels for the whole sentence, if yes – the sentence belong to the language
  - That is if a tree with signature *[0, n, S]* exists
- Unary rules?

CKY in action

lead	can	poison

0 1 2 3

$$S \rightarrow NP \ \ VP$$

Inner rules

$$VP \rightarrow M \ \ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N \ NP$$

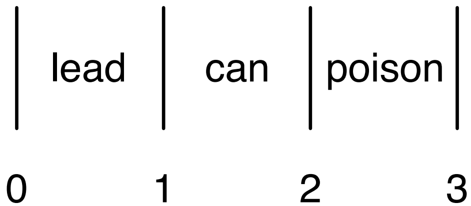
Preterminal rules

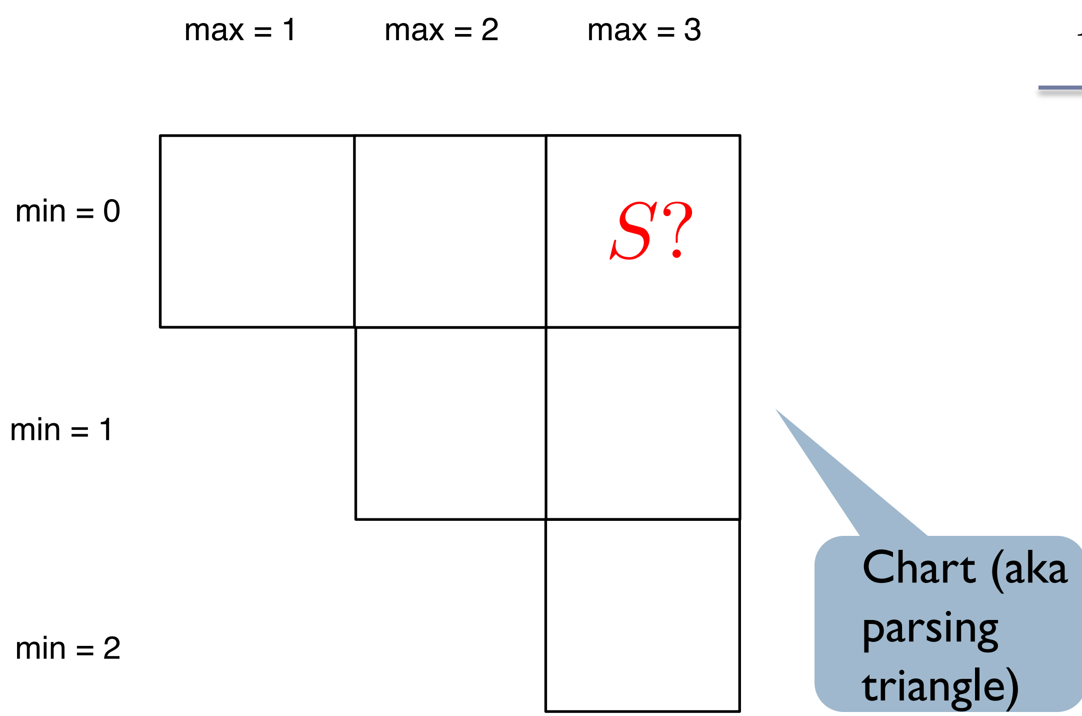
$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action





Inner rules

$$S \to NP\ VP$$

$$VP \to M\ V$$

$$VP \to V$$

$$NP \to N$$

$$NP \to N\ NP$$

Preterminal rules

$$N \to can$$

$$N \to lead$$

$$N \to poison$$

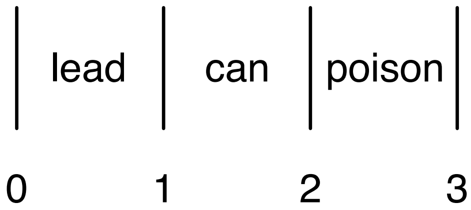
$$M \to can$$

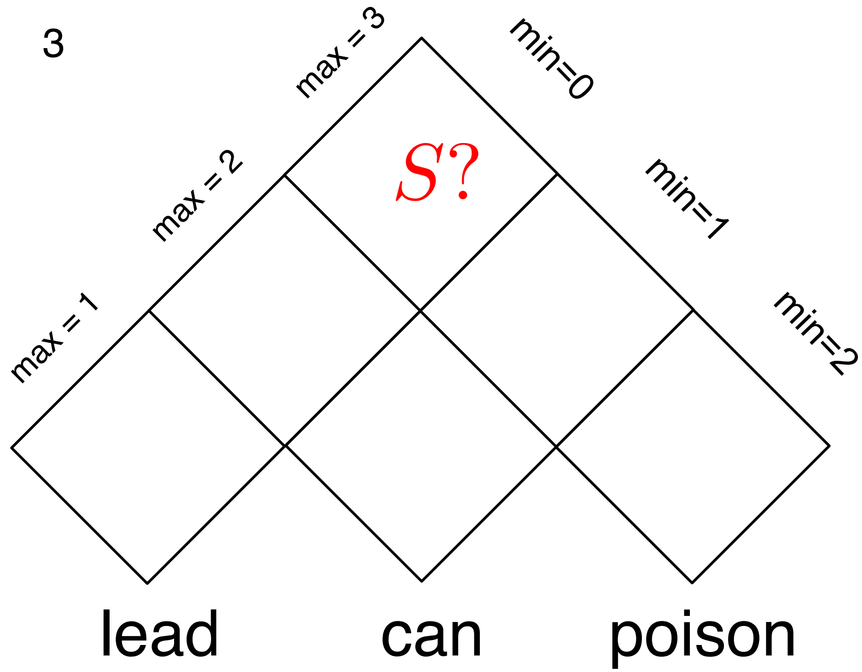
$$M \to must$$

$$V \to poison$$

$$V \to lead$$

CKY in action





Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

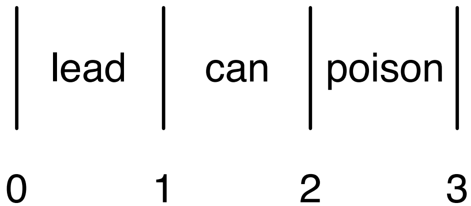
Preterminal rules

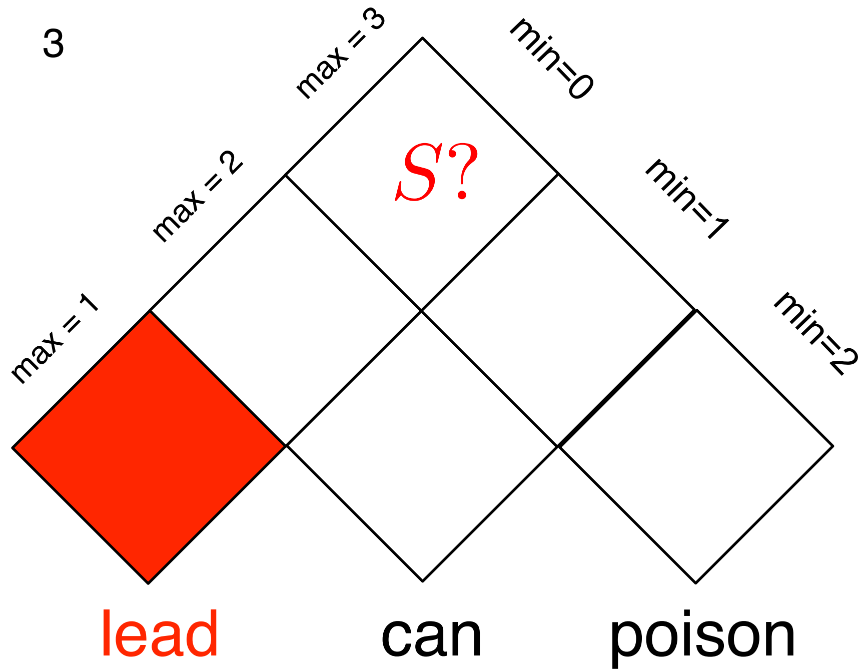
$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action





$S \rightarrow NP\ VP$

Inner rules

$VP \rightarrow M\ \ V$

$VP \rightarrow V$

$NP \rightarrow N$

$NP \rightarrow N\ NP$

Preterminal rules

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

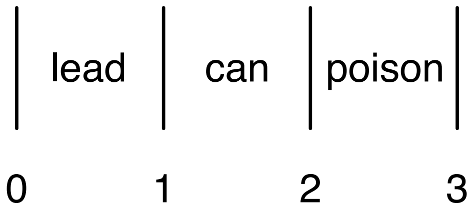
$M \rightarrow can$

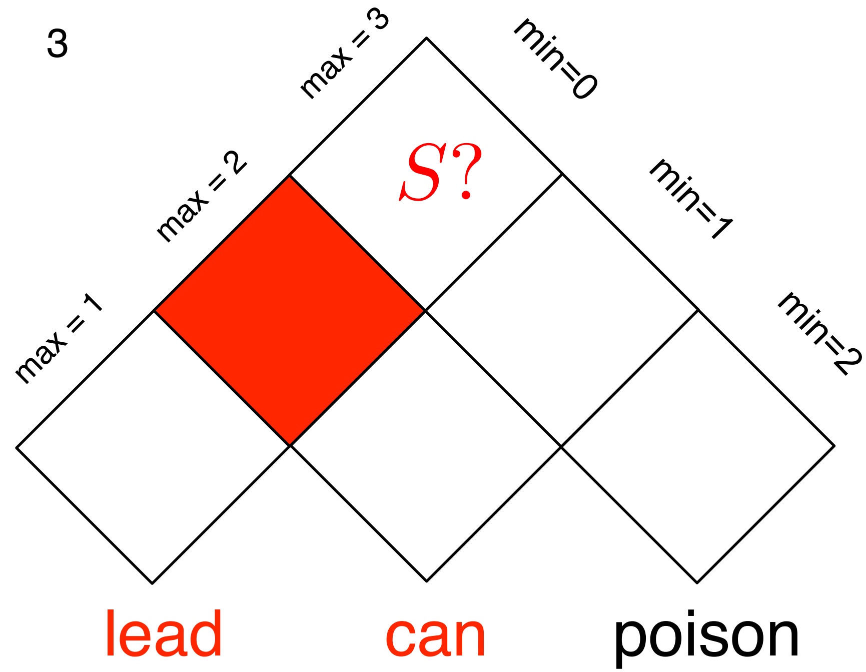
$M \rightarrow must$

$V \rightarrow poison$

$V \rightarrow lead$

CKY in action





$$S \rightarrow NP\ \ VP$$

Inner rules

$$VP \rightarrow M\ \ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

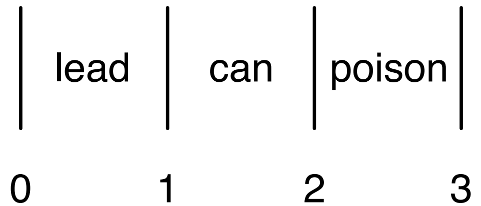
Preterminal rules

$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action



	max = 1	max = 2	max = 3
min = 0			$S?$
min = 1			
min = 2			

Inner rules

$$S \to NP\ VP$$

$$VP \to M\ V$$

$$VP \to V$$

$$NP \to N$$

$$NP \to N\ NP$$

Preterminal rules

$$N \to can$$

$$N \to lead$$

$$N \to poison$$

$$M \to can$$

$$M \to must$$

$$V \to poison$$

$$V \to lead$$

CKY in action

	lead		can		poison	
0		1		2		3

	max = 1	max = 2	max = 3
min = 0	1	4	6 $S?$
min = 1		2	5
min = 2			3

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 ?		
min = 1		2 ?	
min = 2			3 ?

Inner rules

$$S \rightarrow NP\ \ VP$$

$$VP \rightarrow M\ \ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 ?		
min = 1		2 ?	
min = 2			3 ?

Inner rules

$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow M \ \ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N \ NP$$

Preterminal rules

$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V		
min = 1		2 N, M	
min = 2			3 N, V

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

lead	can	poison

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP		
min = 1		2 N, M NP	
min = 2			3 N, V NP, VP

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$

$$\boxed{VP \rightarrow V}$$

$$\boxed{NP \rightarrow N}$$

$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 ?	
min = 1		2 N, M NP	
min = 2			3 N, V NP, VP

Inner rules

$$S \to NP\ \ VP$$

$$VP \to M\ \ V$$

$$VP \to V$$

$$NP \to N$$

$$NP \to N\ NP$$

Preterminal rules

$$N \to can$$

$$N \to lead$$

$$N \to poison$$

$$M \to can$$

$$M \to must$$

$$V \to poison$$

$$V \to lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 ?	
min = 1		2 N, M NP	
min = 2			3 N, V NP, VP

$$S \to NP\ \ VP$$

Inner rules

$$VP \to M\ \ V$$
$$VP \to V$$

$$NP \to N$$
$$\boxed{NP \to N\ NP}$$

Preterminal rules

$$N \to can$$
$$N \to lead$$
$$N \to poison$$

$$M \to can$$
$$M \to must$$

$$V \to poison$$
$$V \to lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	
min = 1		2 N, M NP	
min = 2			3 N, V NP, VP

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$\boxed{NP \rightarrow N\ NP}$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

lead	can	poison

0 1 2 3

	max = 1	max = 2	
min = 0	1 N, V NP, VP	4 NP	
min = 1		2 N, M NP	
min = 2			3 N, V NP, VP

Check about unary rules: no unary rules here

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	
min = 1		2 N, M NP	5 ?
min = 2			3 N, V NP, VP

Inner rules

$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow M \ \ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N \ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

$\boxed{S \rightarrow NP\ VP}$

Inner rules

$\boxed{VP \rightarrow M\ V}$

$VP \rightarrow V$

$NP \rightarrow N$

$\boxed{NP \rightarrow N\ NP}$

Preterminal rules

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

$M \rightarrow can$

$M \rightarrow must$

$V \rightarrow poison$

$V \rightarrow lead$

CKY in action

	lead		can		poison	
0		1		2		3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Check about unary rules: no unary rules here

Inner rules

$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow M \ \ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N \ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

lead	can	poison

0 1 2 3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 ?
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Inner rules

$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow M \ \ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N \ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

CKY in action

	lead		can		poison	
0		1		2		3

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 ?
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Inner rules

$$S \rightarrow NP\ VP$$

$$VP \rightarrow M\ V$$
$$VP \rightarrow V$$

$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

$$M \rightarrow can$$
$$M \rightarrow must$$

$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action

| lead | can | poison |

0 1 2 3

$\text{mid} = 1$

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 S, NP
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Inner rules

$\boxed{S \rightarrow NP\ VP}$

$VP \rightarrow M\ \ V$

$VP \rightarrow V$

$NP \rightarrow N$

$\boxed{NP \rightarrow N\ NP}$

Preterminal rules

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

$M \rightarrow can$

$M \rightarrow must$

$V \rightarrow poison$

$V \rightarrow lead$

CKY in action

| lead | can | poison |

0 1 2 3

$$S \rightarrow NP\ VP$$

mid = 2

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 S, NP $S(?!)$
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Inner rules

$$VP \rightarrow M\ V$$
$$VP \rightarrow V$$
$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$
$$M \rightarrow can$$
$$M \rightarrow must$$
$$V \rightarrow poison$$
$$V \rightarrow lead$$

CKY in action

lead	can	poison

0 1 2 3

$S \rightarrow NP\ VP$

Inner rules

$VP \rightarrow M\ \ V$

$VP \rightarrow V$

$NP \rightarrow N$

$NP \rightarrow N\ NP$

Preterminal rules

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

$M \rightarrow can$

$M \rightarrow must$

$V \rightarrow poison$

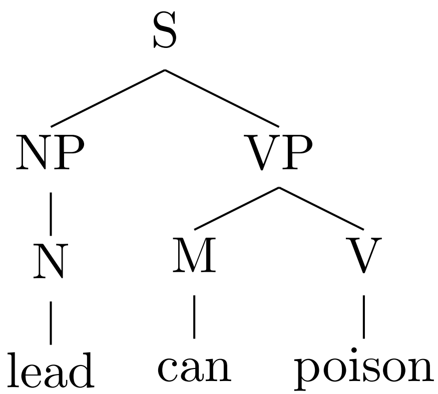
$V \rightarrow lead$

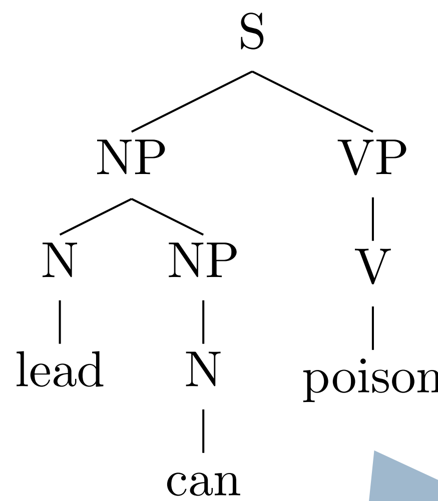
$\mathrm{mid} = 2$

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 S, NP $S(?!)$
min = 1		2 N, M NP	5 $S, VP,$ NP
			3 N, V

Apparently the sentence is ambiguous with the grammar

Ambiguity





No subject-verb agreement, and *poison* used as an intransitive verb

Apparently the sentence is ambiguous with the grammar

CKY more formally

Here we assume that labels (C) are integer indices

- Chart can be represented by a Boolean array `chart[min][max][C]`
  - Relevant entries have $0 \leq$ `min < max` $\leq$ `n`
- `chart[min][max][C] = true` if the signature (min, max, C) is already added to the chart; `false` otherwise.

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 $S, VP,$ NP
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

CKY more formally

Here we assume that labels (C) are integer indices

- Chart can be represented by a Boolean array `chart[min][max][C]`
  - Relevant entries have $0 \leq$ `min` < `max` $\leq$ `n`
- `chart[min][max][C] = true` if the signature (min, max, C) is already added to the chart; `false` otherwise.

	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 $S, VP,$ NP
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

Implementation: preterminal rules

```
for each wᵢ from left to right
  for each preterminal rule C -> wᵢ
    chart[i - 1][i][C] = true
```

Implementation: binary rules

```
for each max from 2 to n
  for each min from max - 2 down to 0
    for each syntactic category C
      for each binary rule C -> C1 C2
        for each mid from min + 1 to max - 1
          if chart[min][mid][C1] and chart[mid][max][C2] then
            chart[min][max][C] = true
```

Unary rules

- How to integrate unary rules $C \rightarrow C_1$?

Implementation: unary rules

```
for each max from 1 to n
  for each min from max - 1 down to 0
    // First, try all binary rules as before.
    ...
    // Then, try all unary rules.
    for each syntactic category C
      for each unary rule C -> C1
        if chart[min][max][C1] then
          chart[min][max][C] = true
```

new bounds!

Implementation: unary rules

```
for each max from 1 to n
  for each min from max - 1 down to 0
    // First, try all binary rules as before.
    ...
    // Then, try all unary rules.
    for each syntactic category C
      for each unary rule C -> C1
        if chart[min][max][C1] then
          chart[min][max][C] = true
```

new bounds!

But we forgot something!

Unary closure

- What if the grammar contained 2 rules:

$$A \rightarrow B$$
$$B \rightarrow C$$

- But *C* can be derived from *A* by a chain of rules:

$$A \rightarrow B \rightarrow C$$

- One could support chains in the algorithm but it is easier to extend the grammar, to get the transitive closure

$$\begin{array}{c} A \rightarrow B \\ B \rightarrow C \end{array} \quad \Rightarrow \quad \begin{array}{c} A \rightarrow B \\ B \rightarrow C \\ A \rightarrow C \end{array}$$

Implementation: skeleton

```
// int n = number of words in the sequence

// int m = number of syntactic categories in the grammar

// int s = the (number of the) grammar's start symbol

boolean[][][] chart = new boolean[n + 1][n + 1][m]

// Recognize all parse trees built with with preterminal rules.

// Recognize all parse trees built with inner rules.

return chart[0][n][s]
```

Algorithm analysis

- Time complexity?

Algorithm analysis

- Time complexity?

```
for each max from 2 to n
  for each min from max - 2 down to 0
    for each syntactic category C
      for each binary rule C -> C1 C2
        for each mid from min + 1 to max - 1
```

Algorithm analysis

- Time complexity?

```
for each max from 2 to n
  for each min from max - 2 down to 0
    for each syntactic category C
      for each binary rule C -> C1 C2
        for each mid from min + 1 to max - 1
```

A few seconds for sentences under < 20 words for a non-optimized parser

- $\theta(n^3|R|)$, where $|R|$ is the number of rules in the grammar

Algorithm analysis

- Time complexity?

```
for each max from 2 to n
  for each min from max - 2 down to 0
    for each syntactic category C
      for each binary rule C -> C1 C2
        for each mid from min + 1 to max - 1
```

A few seconds for sentences under < 20 words for a non-optimized parser

  - $\theta(n^3|R|)$, where $|R|$ is the number of rules in the grammar

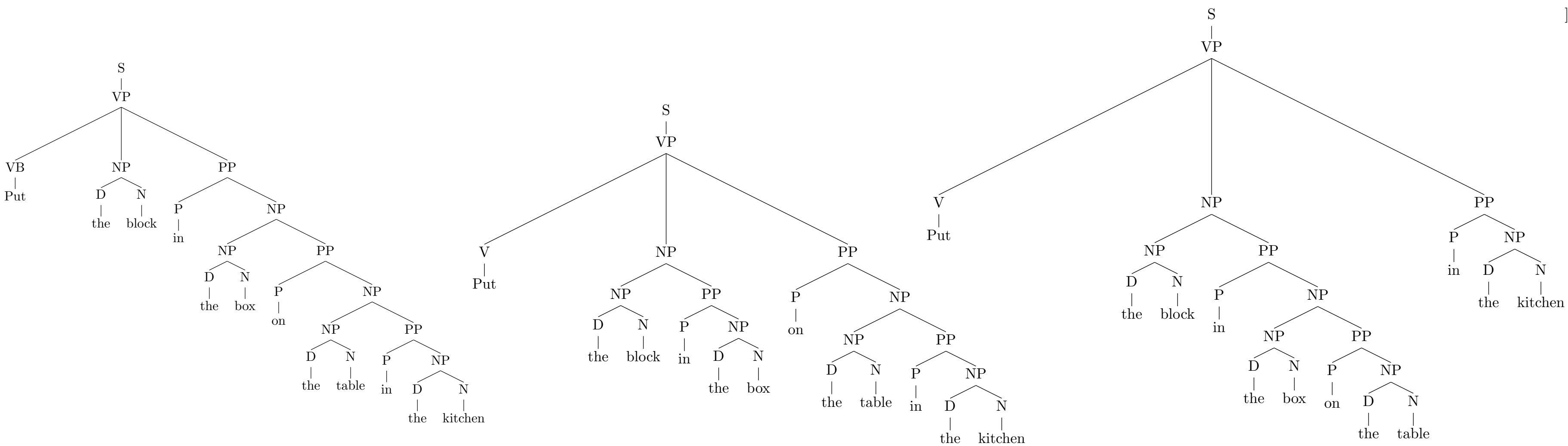
- There exist algorithms with better asymptotical time complexity but the `constant' makes them slower in practice (in general)

Today

- CKY for the recognition problem
- Probabilistic PCFGs
- CKY for PCFGs

How to deal with ambiguity?

- There are (exponentially) many derivation for a typical sentence



……

Put the block in the box on the table in the kitchen

- We want to score all the derivations to encode how plausible they are

An example probabilistic CFG

Associate probabilities with the rules $p(X \to \alpha)$:

$$\forall\, X \to \alpha \in R: \quad 0 \le p(X \to \alpha) \le 1$$

$$\forall X \in N: \quad \sum_{\alpha: X \to \alpha \in R} p(X \to \alpha) = 1$$

Now we can score a tree as a product of probabilities corresponding to the used rules

Rule	Prob.	Example
$S \to NP\ VP$	1.0	(NP A girl) (VP ate a sandwich)
$VP \to V$	0.2	
$VP \to V\ NP$	0.4	(VP ate) (NP a sandwich)
$VP \to VP\ PP$	0.4	(VP saw a girl) (PP with …)
$NP \to NP\ PP$	0.3	(NP a girl) (PP with ….)
$NP \to D\ N$	0.5	(D a) (N sandwich)
$NP \to PN$	0.2	
$PP \to P\ NP$	1.0	(P with) (NP with a sandwich)

Rule	Prob.
$N \to girl$	0.2
$N \to telescope$	0.7
$N \to sandwich$	0.1
$PN \to I$	1.0
$V \to saw$	0.5
$V \to ate$	0.5
$P \to with$	0.6
$P \to in$	0.4
$D \to a$	0.3
$D \to the$	0.7

CFGs

$S \to NP\ \ VP$ 1.0

$VP \to V$ 0.2

$VP \to V\ \ NP$ 0.4

$VP \to VP\ \ PP$ 0.4

$NP \to NP\ \ PP$ 0.3

$NP \to D\ \ N$ 0.5

$NP \to PN$ 0.2

$PP \to P\ \ NP$ 1.0

$N \to girl$ 0.2

$N \to telescope$ 0.7

$N \to sandwich$ 0.1

$PN \to I$ 1.0

$V \to saw$ 0.5

$V \to ate$ 0.5

$P \to with$ 0.6

$P \to in$ 0.4

$D \to a$ 0.3

$D \to the$ 0.7

S

$p(T) =$

CFGs

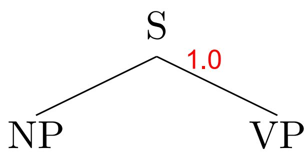
$$\boxed{S \rightarrow NP\ \ VP \quad 1.0}$$

$$VP \rightarrow V \quad 0.2$$
$$VP \rightarrow V\ \ NP \quad 0.4$$
$$VP \rightarrow VP\ \ PP \quad 0.4$$

$$NP \rightarrow NP\ \ PP \quad 0.3$$
$$NP \rightarrow D\ \ N \quad 0.5$$
$$NP \rightarrow PN \quad 0.2$$

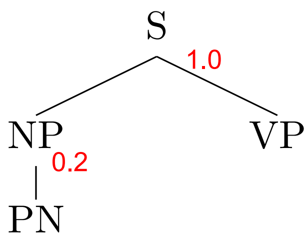
$$PP \rightarrow P\ \ NP \quad 1.0$$

$$N \rightarrow girl \quad 0.2$$
$$N \rightarrow telescope \quad 0.7$$
$$N \rightarrow sandwich \quad 0.1$$
$$PN \rightarrow I \quad 1.0$$
$$V \rightarrow saw \quad 0.5$$
$$V \rightarrow ate \quad 0.5$$
$$P \rightarrow with \quad 0.6$$
$$P \rightarrow in \quad 0.4$$
$$D \rightarrow a \quad 0.3$$
$$D \rightarrow the \quad 0.7$$



$$p(T) = 1.0 \times$$

CFGs



$$S \rightarrow NP \ \ VP \ \ 1.0$$

$$VP \rightarrow V \ \ 0.2$$

$$VP \rightarrow V \ \ NP \ \ 0.4$$

$$VP \rightarrow VP \ \ PP \ \ 0.4$$

$$NP \rightarrow NP \ \ PP \ \ 0.3$$

$$NP \rightarrow D \ \ N \ \ 0.5$$

$$\boxed{NP \rightarrow PN \ \ 0.2}$$

$$PP \rightarrow P \ \ NP \ \ 1.0$$

$$N \rightarrow girl \ \ 0.2$$

$$N \rightarrow telescope \ \ 0.7$$

$$N \rightarrow sandwich \ \ 0.1$$

$$PN \rightarrow I \ \ 1.0$$

$$V \rightarrow saw \ \ 0.5$$

$$V \rightarrow ate \ \ 0.5$$

$$P \rightarrow with \ \ 0.6$$

$$P \rightarrow in \ \ 0.4$$

$$D \rightarrow a \ \ 0.3$$

$$D \rightarrow the \ \ 0.7$$

$$p(T) = 1.0 \times 0.2 \times$$

CFGs

$S \rightarrow NP \ \ VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V \ \ NP$ 0.4

$VP \rightarrow VP \ \ PP$ 0.4

$NP \rightarrow NP \ \ PP$ 0.3

$NP \rightarrow D \ \ N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P \ \ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$\boxed{PN \rightarrow I \ \text{1.0}}$

$V \rightarrow saw$ 0.5

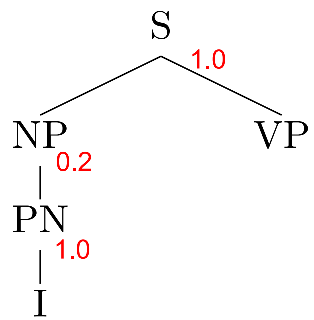
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



$$p(T) = 1.0 \times 0.2 \times 1.0 \times$$

CFGs

$S \rightarrow NP \ \ VP$ 1.0

$VP \rightarrow V$ 0.2

$\boxed{VP \rightarrow V \ \ NP \ \ 0.4}$

$VP \rightarrow VP \ \ PP$ 0.4

$NP \rightarrow NP \ \ PP$ 0.3

$NP \rightarrow D \ \ N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P \ \ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$V \rightarrow saw$ 0.5

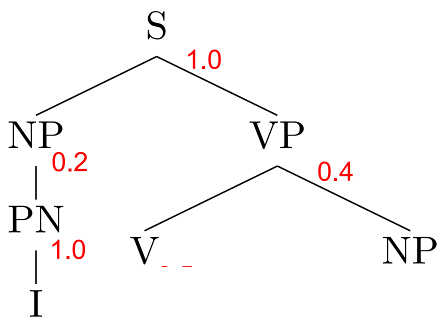
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

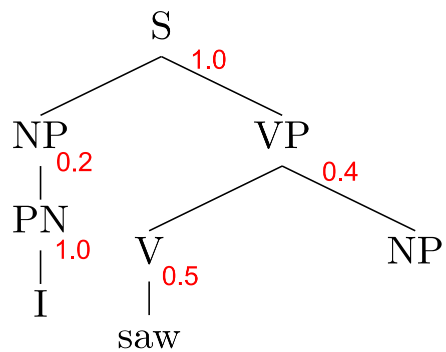
$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times$$

CFGs



$S \rightarrow NP \;\; VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V \;\; NP$ 0.4

$VP \rightarrow VP \;\; PP$ 0.4

$NP \rightarrow NP \;\; PP$ 0.3

$NP \rightarrow D \;\; N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P \;\; NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$\boxed{V \rightarrow saw \;\; 0.5}$

$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7

$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times$$

CFGs

$S \rightarrow NP\ \ VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V\ \ NP$ 0.4

$VP \rightarrow VP\ \ PP$ 0.4

$\boxed{NP \rightarrow NP\ \ PP \text{ 0.3}}$

$NP \rightarrow D\ \ N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P\ \ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$V \rightarrow saw$ 0.5

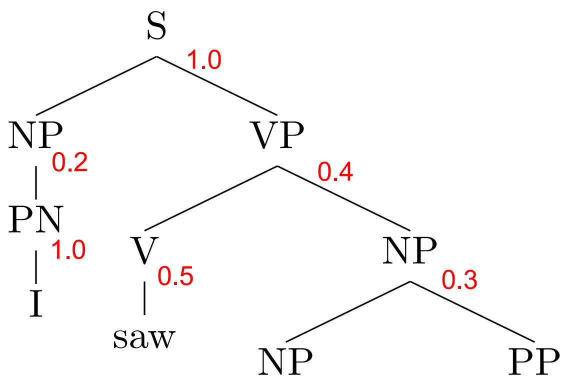
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

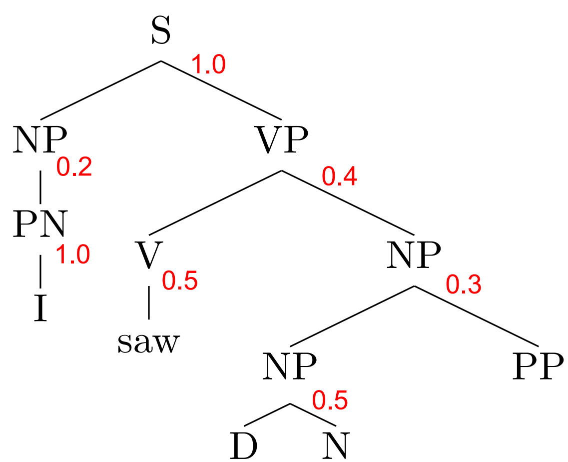
$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times$$

CFGs



$S \rightarrow NP\ VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V\ NP$ 0.4

$VP \rightarrow VP\ PP$ 0.4

$NP \rightarrow NP\ PP$ 0.3

$\boxed{NP \rightarrow D\ N \text{ 0.5}}$

$NP \rightarrow PN$ 0.2

$PP \rightarrow P\ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$V \rightarrow saw$ 0.5

$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7

$$
\begin{aligned}
p(T) = &1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times \\
&0.5 \times
\end{aligned}
$$

CFGs

$S \rightarrow NP\ \ VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V\ \ NP$ 0.4

$VP \rightarrow VP\ \ PP$ 0.4

$NP \rightarrow NP\ \ PP$ 0.3

$NP \rightarrow D\ \ N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P\ \ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$V \rightarrow saw$ 0.5

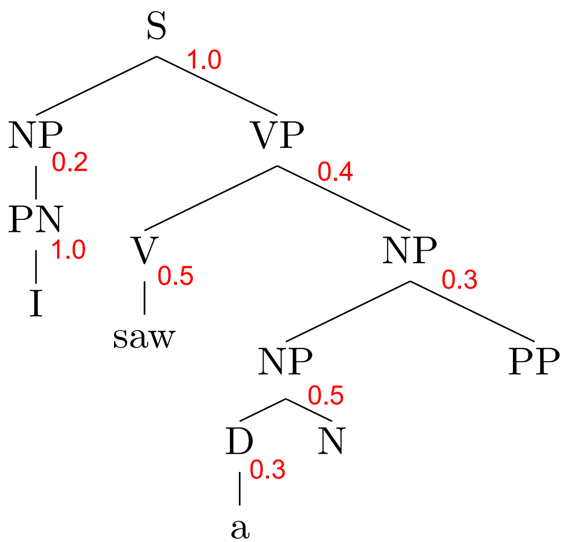
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

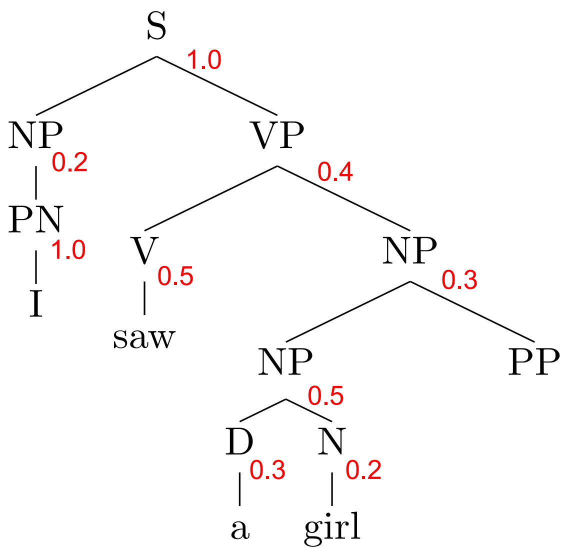
$\boxed{D \rightarrow a \text{ 0.3}}$

$D \rightarrow the$ 0.7



$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times$$
$$0.5 \times 0.3 \times$$

CFGs



$$S \rightarrow NP \;\; VP \;\; 1.0$$

$$VP \rightarrow V \;\; 0.2$$
$$VP \rightarrow V \;\; NP \;\; 0.4$$
$$VP \rightarrow VP \;\; PP \;\; 0.4$$

$$NP \rightarrow NP \;\; PP \;\; 0.3$$
$$NP \rightarrow D \;\; N \;\; 0.5$$
$$NP \rightarrow PN \;\; 0.2$$

$$PP \rightarrow P \;\; NP \;\; 1.0$$

$$\boxed{N \rightarrow girl \;\; 0.2}$$
$$N \rightarrow telescope \;\; 0.7$$
$$N \rightarrow sandwich \;\; 0.1$$
$$PN \rightarrow I \;\; 1.0$$
$$V \rightarrow saw \;\; 0.5$$
$$V \rightarrow ate \;\; 0.5$$
$$P \rightarrow with \;\; 0.6$$
$$P \rightarrow in \;\; 0.4$$
$$D \rightarrow a \;\; 0.3$$
$$D \rightarrow the \;\; 0.7$$

$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times 0.5 \times 0.3 \times 0.2$$

CFGs

$S \rightarrow NP\ \ VP$ 1.0

$VP \rightarrow V$ 0.2

$VP \rightarrow V\ \ NP$ 0.4

$VP \rightarrow VP\ \ PP$ 0.4

$NP \rightarrow NP\ \ PP$ 0.3

$NP \rightarrow D\ \ N$ 0.5

$NP \rightarrow PN$ 0.2

$PP \rightarrow P\ \ NP$ 1.0

$N \rightarrow girl$ 0.2

$N \rightarrow telescope$ 0.7

$N \rightarrow sandwich$ 0.1

$PN \rightarrow I$ 1.0

$V \rightarrow saw$ 0.5

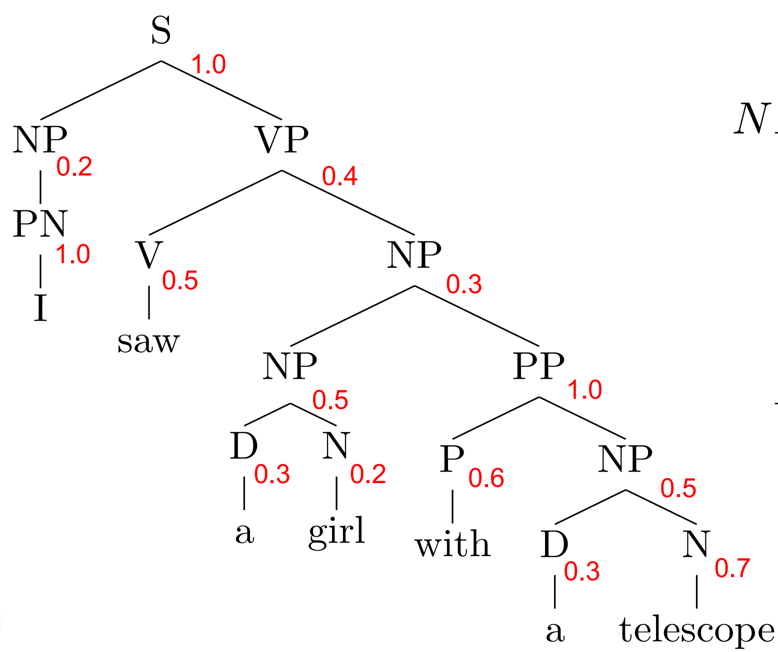
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times$$
$$0.5 \times 0.3 \times 0.2 \times 1.0 \times 0.6 \times 0.5 \times 0.3 \times 0.7$$
$$= 2.26 \times 10^{-5}$$

Distribution over trees

- We defined a distribution over production rules for each nonterminal
- Our goal was to define a distribution over parse trees

Unfortunately, not all PCFGs give rise to a proper distribution over trees, i.e. the sum over probabilities of all trees the grammar can generate may be less than 1: $\sum_T P(T) < 1$

Distribution over trees

- We defined a distribution over production rules for each nonterminal
- Our goal was to define a distribution over parse trees

Unfortunately, not all PCFGs give rise to a proper distribution over trees, i.e. the sum over probabilities of all trees the grammar can generate may be less than 1: $\sum_T P(T) < 1$

- Good news: any PCFG estimated with the maximum likelihood procedure are always proper [Chi and Geman, 98]

Distribution over trees

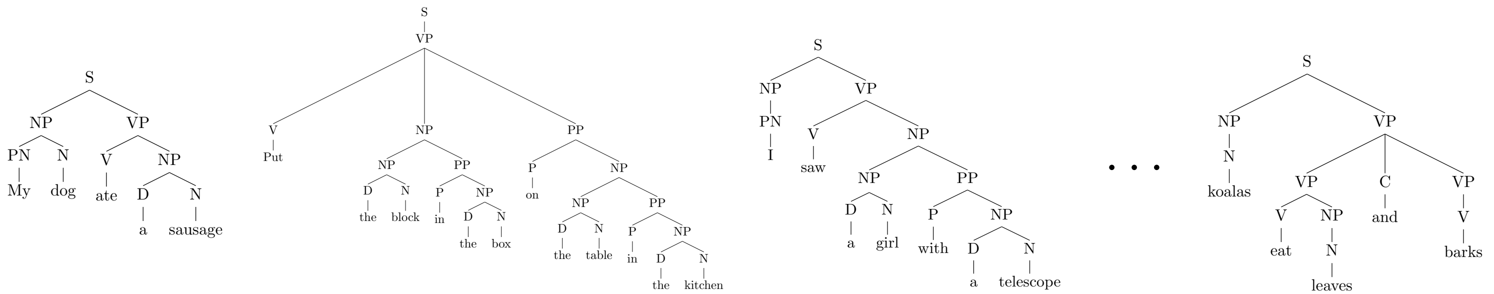
- Let us denote by $G(x)$ the set of derivations for the sentence x
- The probability distribution defines the scoring $P(T)$ over the trees $T \in G(x)$
- Finding the best parse for the sentence according to PCFG:

$$\arg\max_{T \in G(x)} P(T)$$

We will look at this later today

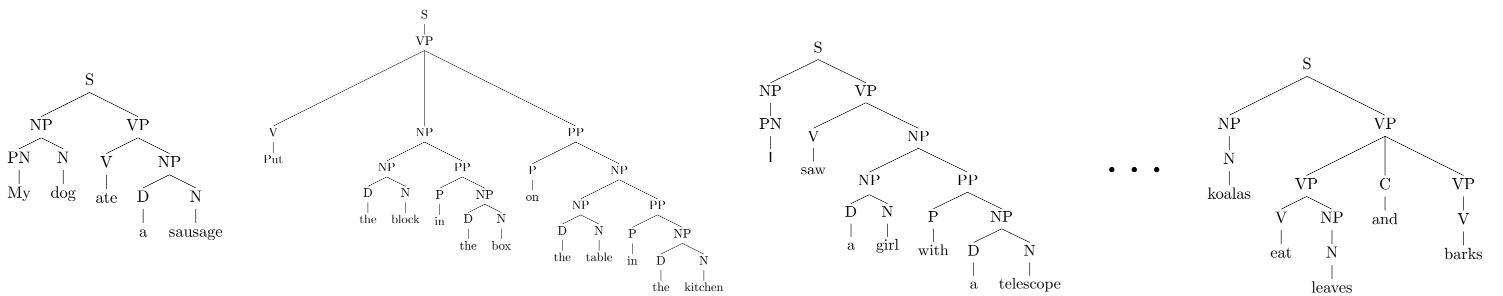
ML estimation

- A treebank: a collection sentences annotated with constituent trees



ML estimation

- A treebank: a collection sentences annotated with constituent trees



- An estimated probability of a rule (maximum likelihood estimates)

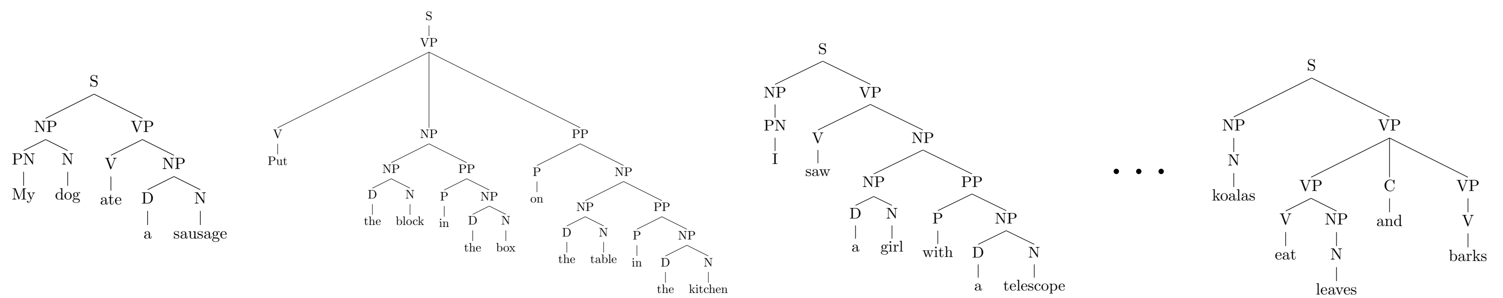
$$p(X \to \alpha) = \frac{C(X \to \alpha)}{C(X)}$$

The number of times the rule used in the corpus

The number of times the nonterminal X appears in the treebank

ML estimation

- A treebank: a collection sentences annotated with constituent trees



- An estimated probability of a rule (maximum likelihood estimates)

$$p(X \to \alpha) = \frac{C(X \to \alpha)}{C(X)}$$

The number of times the rule used in the corpus

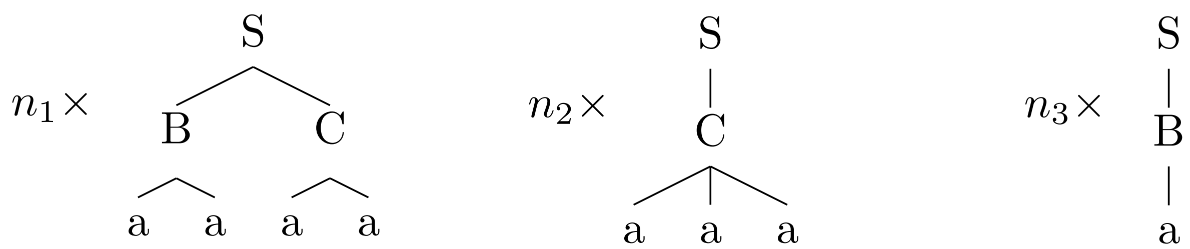
The number of times the nonterminal X appears in the treebank

- Smoothing is helpful
  - Especially important for preterminal rules, i.e. generation of words (= the task model in PoS tagging)
  - The same smoothing techniques as studied before can be used (e.g., GT or add 1 smoothing)

ML estimation: an example

[Ex. from Collins IWPT 01]

- A toy treebank:



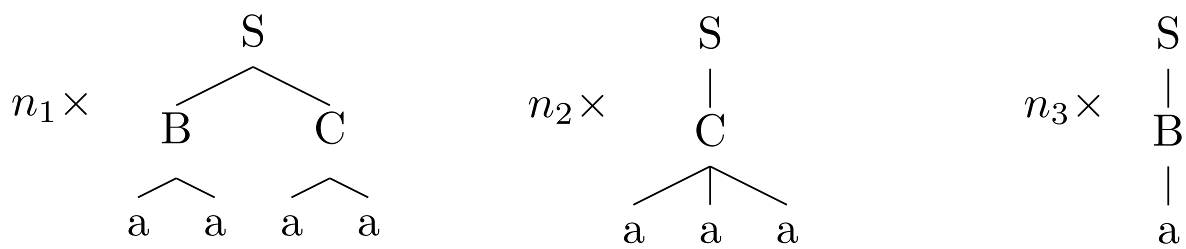
- Without smoothing:

Rule	Count	Prob. estimate
$S \to B\ C$		
$S \to C$		
$S \to B$		
$B \to a\ a$		
$B \to a$		
$C \to a\ a$		
$C \to a\ a\ a$		

ML estimation: an example

[Ex. from Collins IWPT 01]

- A toy treebank:



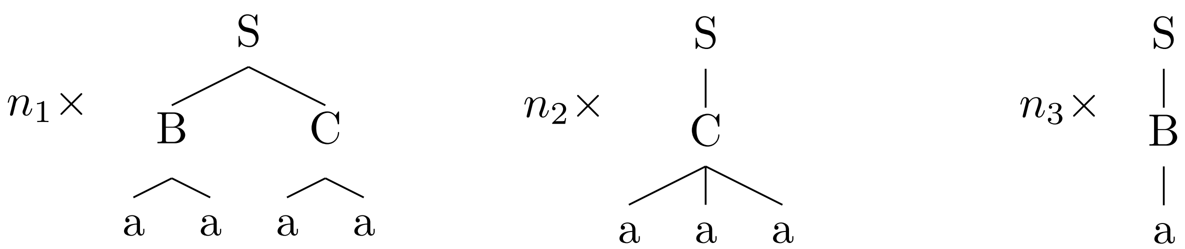
- Without smoothing:

Rule	Count	Prob. estimate
$S \rightarrow B\ C$	n_1	$n_1/(n_1+n_2+n_3)$
$S \rightarrow C$		
$S \rightarrow B$		
$B \rightarrow a\ a$		
$B \rightarrow a$		
$C \rightarrow a\ a$		
$C \rightarrow a\ a\ a$		

ML estimation: an example

[Ex. from Collins IWPT 01]

- A toy treebank:



- Without smoothing:

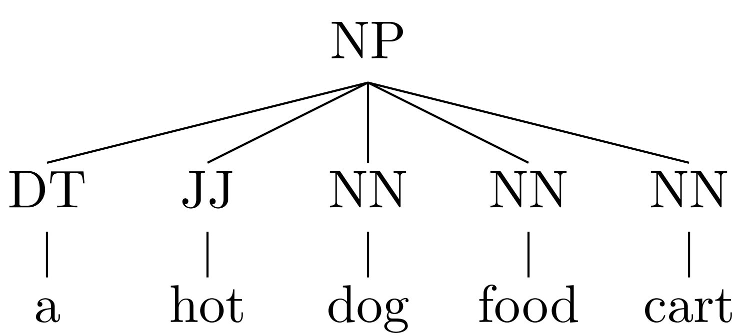
Rule	Count	Prob. estimate
$S \to B\ C$	n_1	$n_1/(n_1+n_2+n_3)$
$S \to C$	n_2	$n_2/(n_1+n_2+n_3)$
$S \to B$	n_3	$n_3/(n_1+n_2+n_3)$
$B \to a\ a$	n_1	$n_1/(n_1+n_3)$
$B \to a$	n_3	$n_3/(n_1+n_3)$
$C \to a\ a$	n_1	$n_1/(n_1+n_2)$
$C \to a\ a\ a$	n_2	$n_2/(n_1+n_2)$

Penn Treebank: peculiarities

- Wall street journal: around 40, 000 annotated sentences, 1,000,000 words
- Fine-grain part of speech tags (45), e.g., for verbs

VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBP	Verb, present (non-3rd person singular)
VBZ	Verb, present (3rd person singular)
MD	Modal

- Flat NPs (no attempt to disambiguate NP attachment)



Today

- CKY for the recognition problem
- Probabilistic PCFGs
- CKY for PCFGs

Summary

- CKY is an important tool, used in many applications
- PCFGs for statistical parsing
- Next time:
  - CKY for PCFGs,
  - ‘Vanilla’ PCFGs weakness and how to address them