Algorithms for NLP

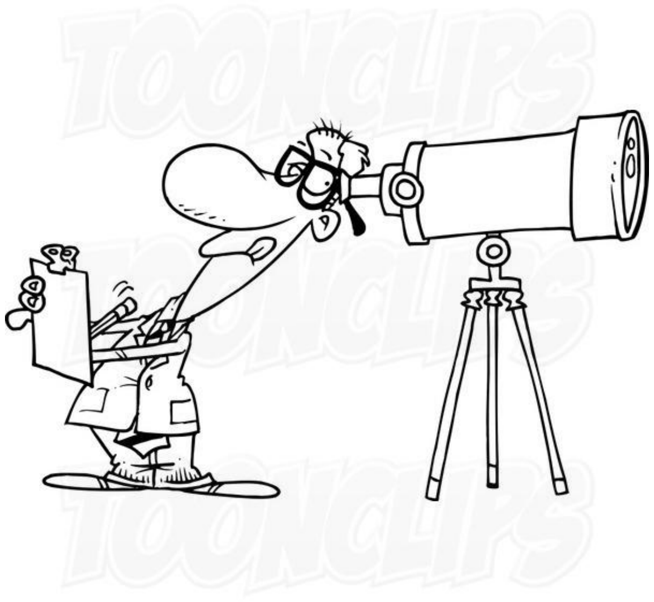
Parsing I

Yulia Tsvetkov – CMU

Slides: Ivan Titov – University of Edinburgh,
Taylor Berg-Kirkpatrick – CMU/UCSD, Dan Klein – UC Berkeley

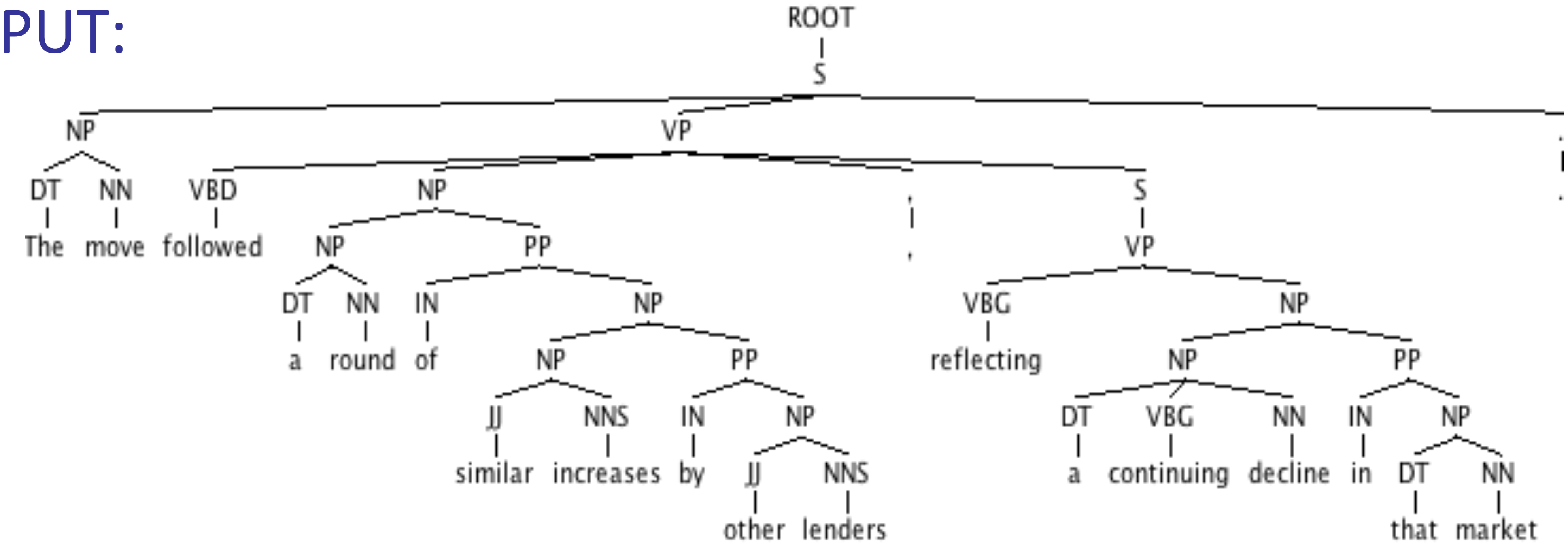
Ambiguity

- I saw a girl with a telescope



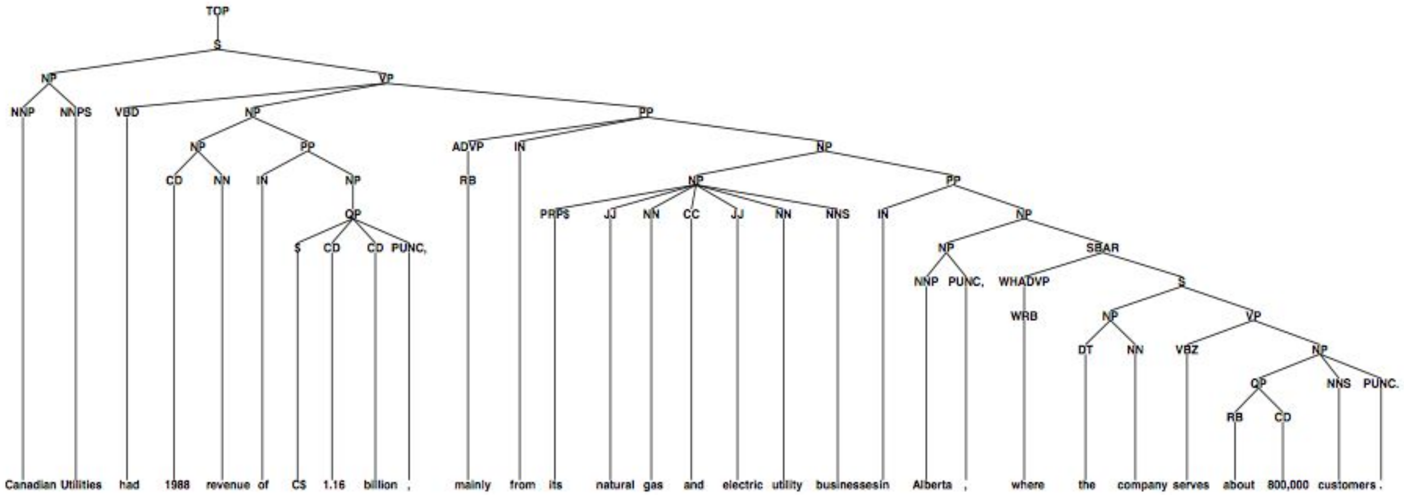
Parsing

- INPUT:
  - The move followed a round of similar increases by other lenders, reflecting a continuing decline in that market
- OUTPUT:



A Supervised ML Problem

- Data for parsing experiments:
  - Penn WSJ Treebank = 50,000 sentences with associated trees
  - Usual set-up: 40,000 training, 2,400 test



Canadian Utilities had 1988 revenue of $ 1.16 billion , mainly from its natural gas and electric utility businesses in Alberta , where the company serves about 800,000 customers*[from Michael Collins slides]*

Outline

- Syntax: intro, CFGs, PCFGs
- CFGs: Parsing
- PCFGs: Parsing
- Parsing evaluation

Syntax

Syntax

- The study of the patterns of formation of sentences and phrases from word

my dog	Pron N
the dog	Det N
the cat	Det N
the large cat	Det Adj N
the black cat	Det Adj N
ate a sausage	V Det N

Syntax

- The study of the patterns of formation of sentences and phrases from word
  - Borders with semantics and morphology sometimes blurred

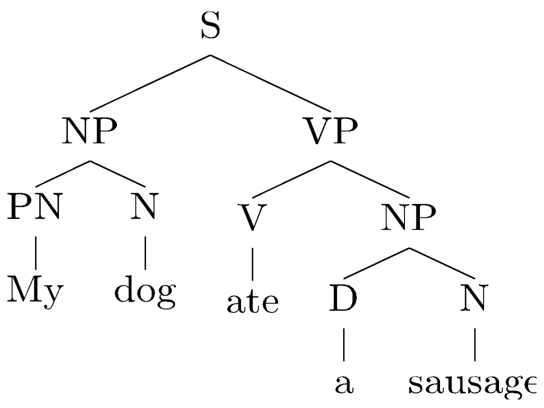
Afyonkarahisarlılaştırabildiklerimizdenmişsinizcesinee

in Turkish means

"as if you are one of the people that we thought to be originating from Afyonkarahisar" [wikipedia]

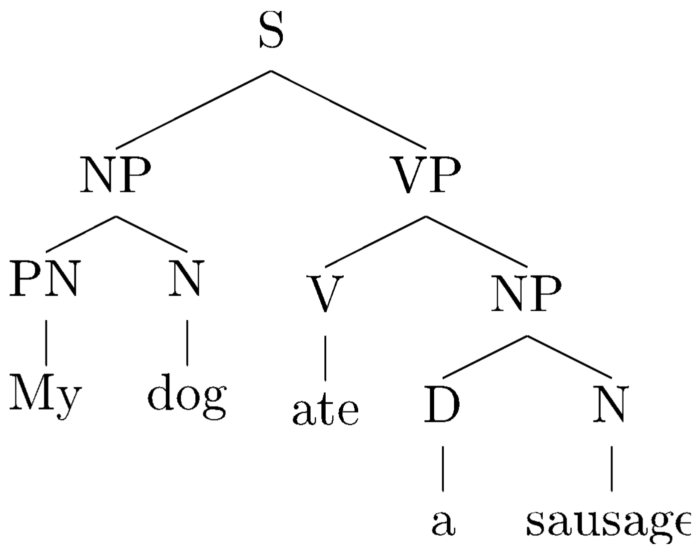
Parsing

- The process of predicting syntactic representations
- Syntactic Representations
  - Different types of syntactic representations are possible, for example:



Constuent (a.k.a. phrase-structure) tree

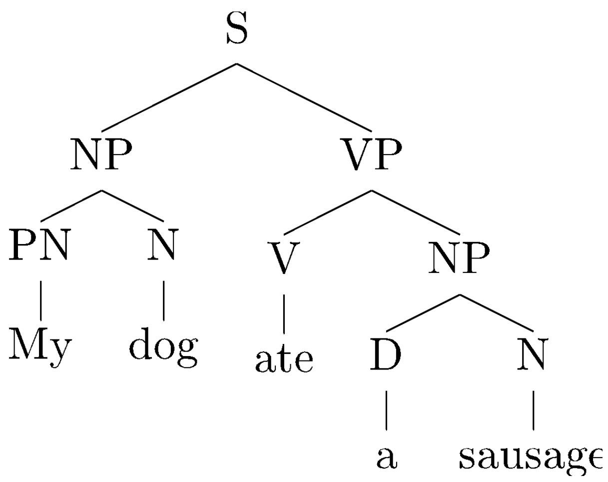
Constituent trees



- 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 (aka preterminals)
  - PN – pronoun
  - D – determiner
  - V – verb
  - N – noun
  - P – preposition

Bracketing notation



- It is often convenient to represent a tree as a bracketed sequence

(S

(NP (PN My) (N Dog))

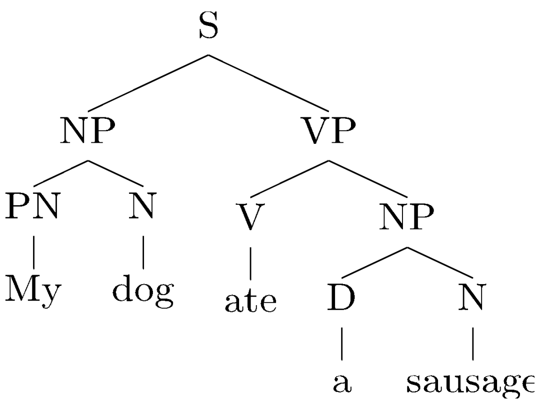
(VP (V ate)

(NP (D a) (N sausage))

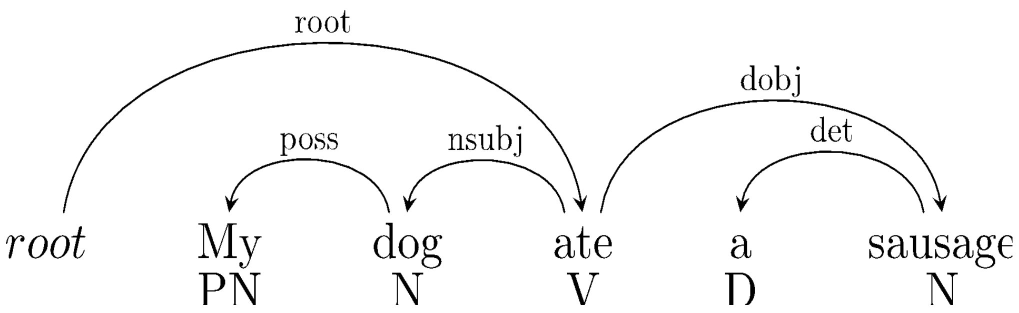
)

Parsing

- The process of predicting syntactic representations
- Syntactic Representations
  - Different types of syntactic representations are possible, for example:

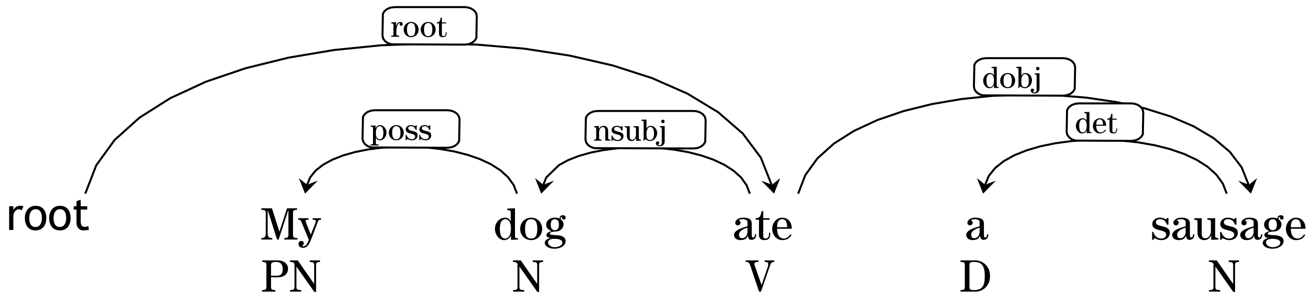


Constuent (a.k.a. phrase-structure) tree



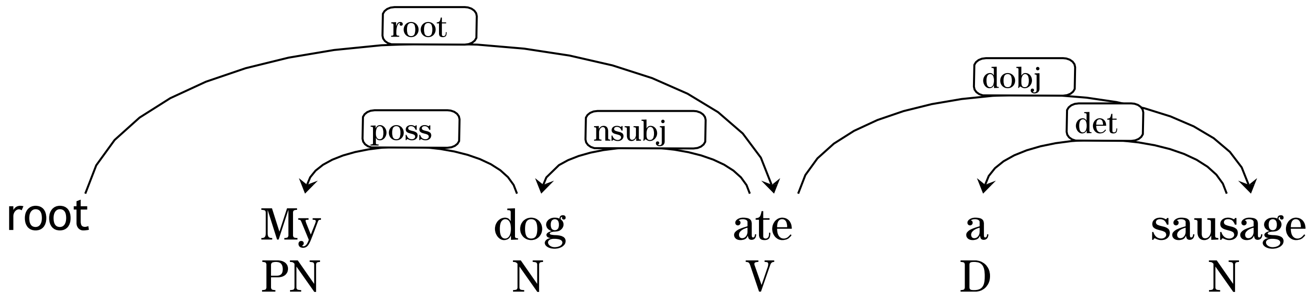
Dependency tree

Dependency trees



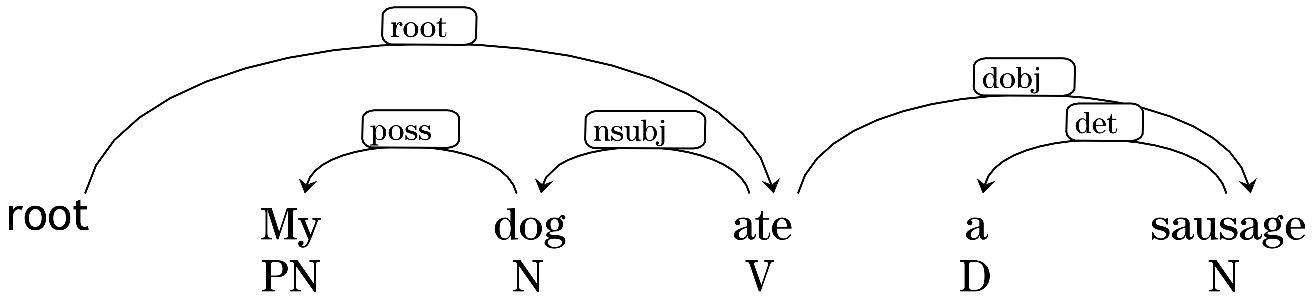
- Nodes are words (along with PoS tags)
- Directed arcs encode syntactic dependencies between them
- Labels are types of relations between the words
  - poss – possesive
  - dobj – direct object
  - nsub - subject
  - det - determiner

Recovering shallow semantics



- Some semantic information can be (approximately) derived from syntactic information
  - Subjects (nsubj) are (often) agents ("initiator / doers for an action")
  - Direct objects (dobj) are (often) patients ("affected entities")

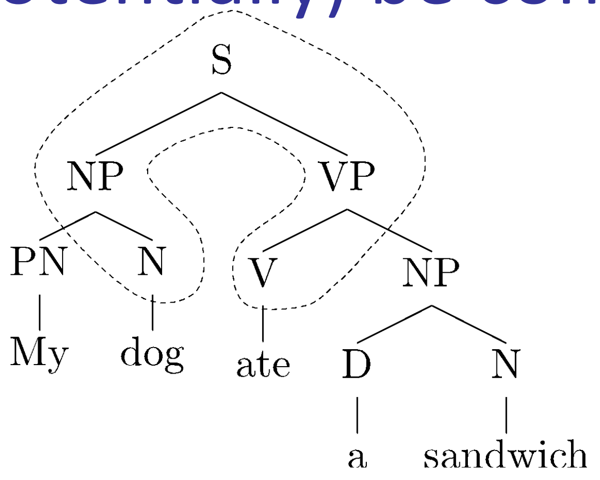
Recovering shallow semantics



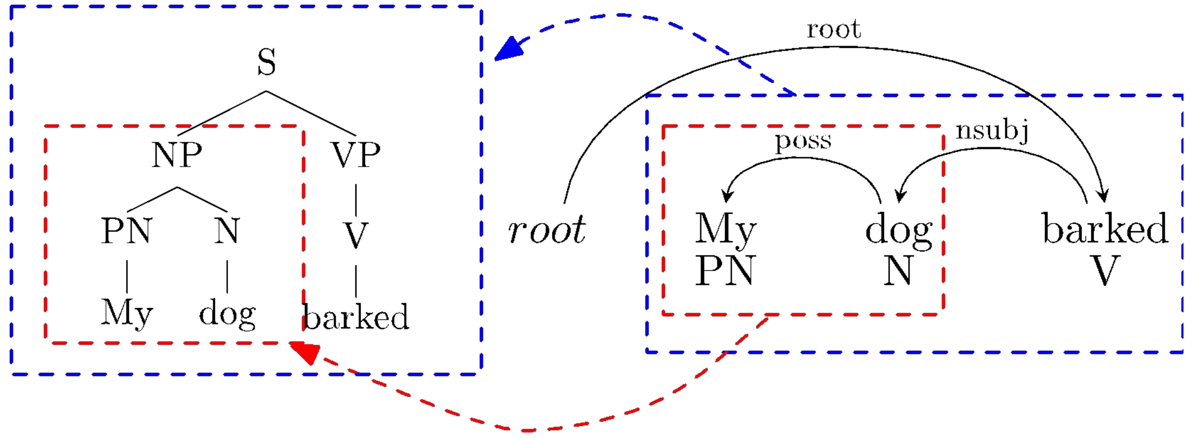
- Some semantic information can be (approximately) derived from syntactic information
  - Subjects (nsubj) are (often) agents ("initiator / doers for an action")
  - Direct objects (dobj) are (often) patients ("affected entities")
- But even for agents and patients consider:
  - Mary is baking a cake in the oven
  - A cake is baking in the oven
- In general it is not trivial even for the most shallow forms of semantics
  - E.g., consider prepositions: *in* can encode direction, position, temporal information, …

Constituent and dependency representations

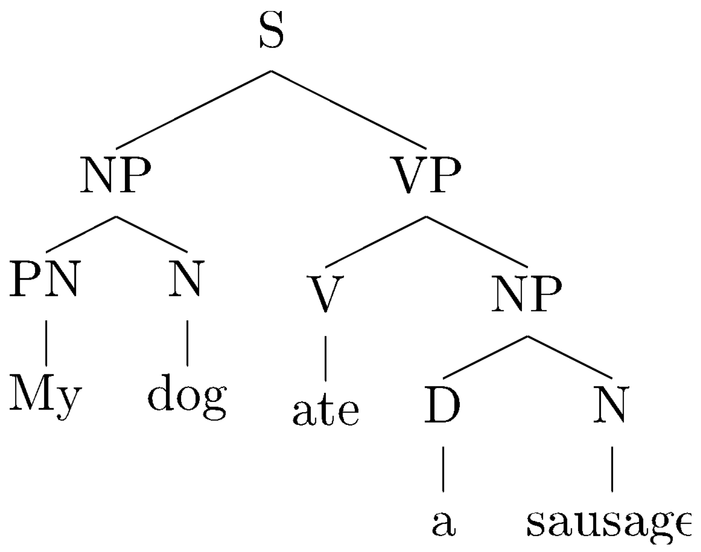
- Constituent trees can (potentially) be converted to dependency trees



- Dependency trees can (potentially) be converted to constituent trees



Constituent trees



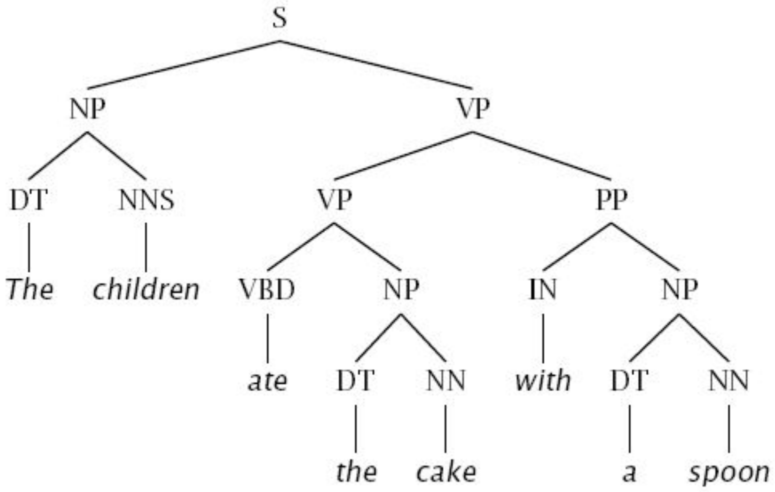
- Internal nodes correspond to phrases
  - S – a sentence
  - NP (Noun Phrase): My dog, a sandwich, lakes,..
  - VP (Verb Phrase): ate a sausage, barked, …
  - PP (Prepositional phrases): with a friend, in a car, …

- Nodes immediately above words are PoS tags (aka preterminals)
  - PN – pronoun
  - D – determiner
  - V – verb
  - N – noun
  - P – preposition

Constituency Tests

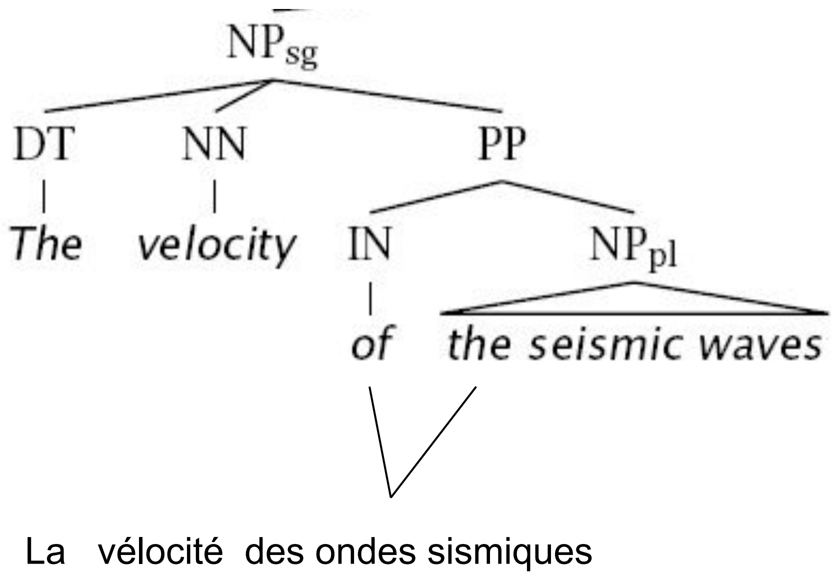
- How do we know what nodes go in the tree?

- Classic constituency tests:
  - Substitution by *proform*
  - Movement
    - Clefting
    - Preposing
    - Passive
  - Modification
  - Coordination/Conjunction
  - Ellipsis/Deletion



Conflicting Tests

- Constituency isn't always clear
  - Units of transfer:
    - think about ~ penser à
    - talk about ~ hablar de
  - Phonological reduction:
    - I will go → I'll go
    - I want to go → I wanna go
    - a le centre → au centre



CFGs

Context Free Grammar (CFG)

Grammar (CFG)		Lexicon
ROOT → S	NP → NP PP	NN → interest
S → NP VP	VP → VBP NP	NNS → raises
NP → DT NN	VP → VBP NP PP	VBP → interest
NP → NN NNS	PP → IN NP	VBZ → raises
		…

- Other grammar formalisms: LFG, HPSG, TAG, CCG…

Treebank Sentences

```
( (S (NP-SBJ The move)
     (VP followed
         (NP (NP a round)
             (PP of
                 (NP (NP similar increases)
                     (PP by
                         (NP other lenders))
                     (PP against
                         (NP Arizona real estate loans)))))
         ,
         (S-ADV (NP-SBJ *)
                (VP reflecting
                    (NP (NP a continuing decline)
                        (PP-LOC in
                                (NP that market))))))
     .))
```

CFGs

S

$$S \rightarrow NP\ \ VP$$

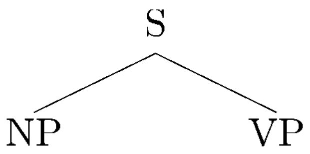
$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P\ \ NP$$

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

CFGs



$$S \rightarrow NP\ \ VP$$

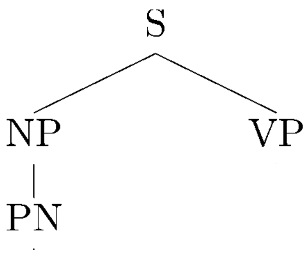
$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P\ \ NP$$

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

CFGs



$$S \rightarrow NP\ \ VP$$

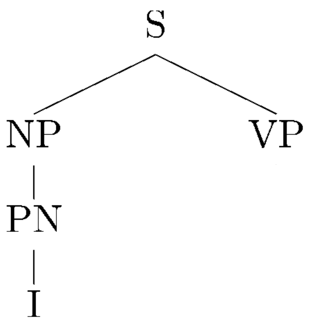
$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P\ \ NP$$

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

CFGs



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

$$VP \rightarrow V$$

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

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

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

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

$$NP \rightarrow PN$$

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

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

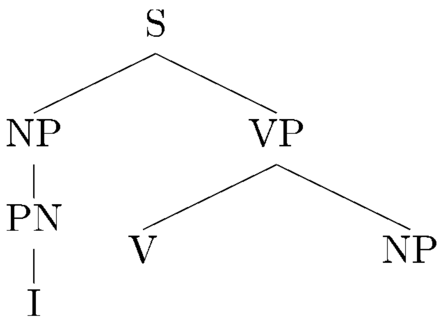
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



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

$$VP \rightarrow V$$

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

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

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

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

$$NP \rightarrow PN$$

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

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

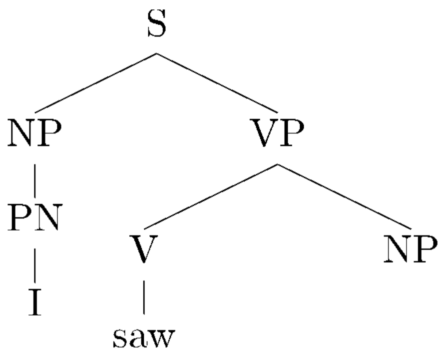
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



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

$$VP \rightarrow V$$

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

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

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

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

$$NP \rightarrow PN$$

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

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

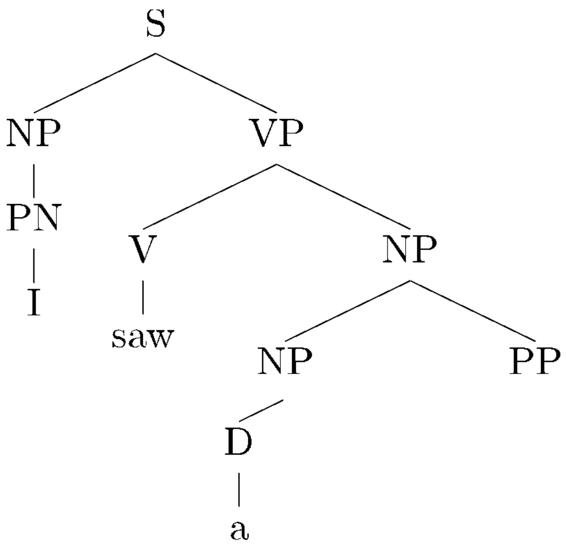
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



$$S \rightarrow NP\ \ VP$$

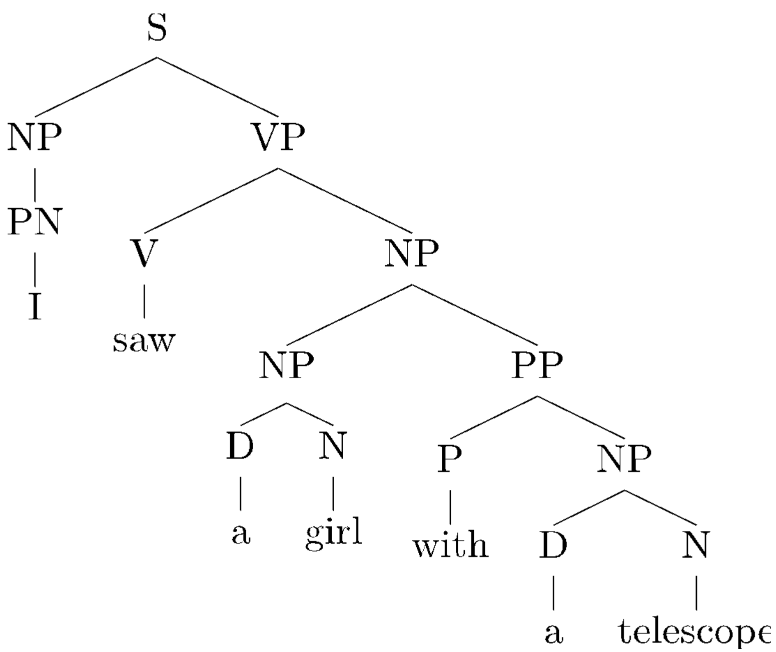
$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P\ \ NP$$

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

CFGs



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

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

$$NP \rightarrow NP \;\; PP$$
$$NP \rightarrow D \;\; N$$
$$NP \rightarrow PN$$

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

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

Context-Free Grammars

- A context-free grammar is a 4-tuple <*N, T, S, R*>
  - *N* : the set of non-terminals
    - Phrasal categories: S, NP, VP, ADJP, etc.
    - Parts-of-speech (pre-terminals): NN, JJ, DT, VB
  - *T* : the set of terminals (the words)
  - *S* : the start symbol
    - Often written as ROOT or TOP
    - *Not* usually the sentence non-terminal S
  - *R* : the set of rules
    - Of the form $X \rightarrow Y_1\ Y_2 \ldots Y_k$, with $X, Y_i \in N$
    - Examples: $S \rightarrow NP\ VP$, $VP \rightarrow VP\ CC\ VP$
    - Also called rewrites, productions, or local trees

An example grammar

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

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

$S = \{S\}$

$R:$

Called **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

$N \rightarrow girl$

$N \rightarrow telescope$

$N \rightarrow sandwich$

$PN \rightarrow I$

$V \rightarrow saw$

$V \rightarrow ate$

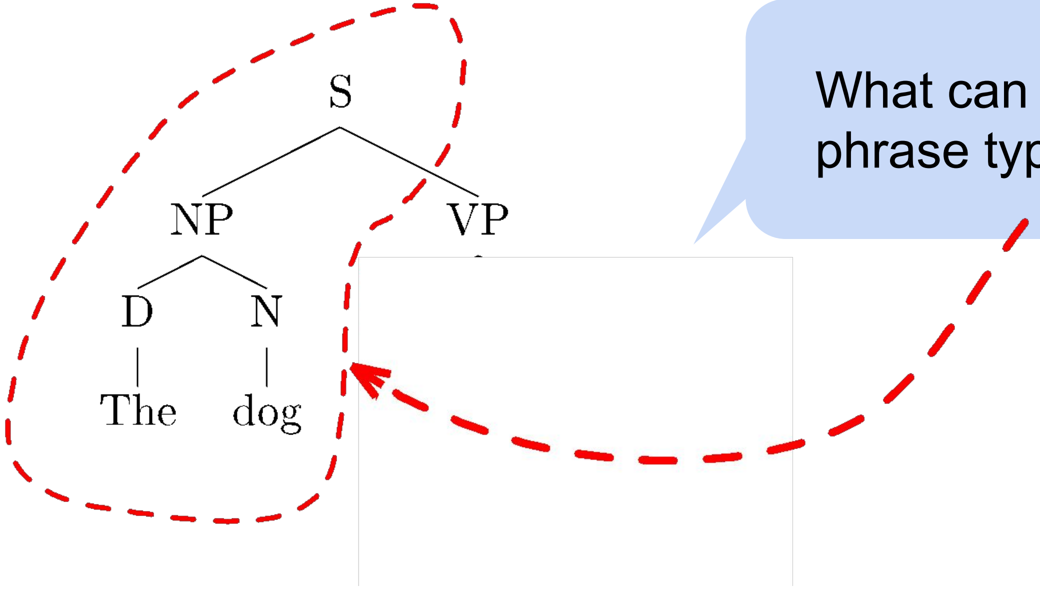
$P \rightarrow with$

$P \rightarrow in$

$D \rightarrow a$

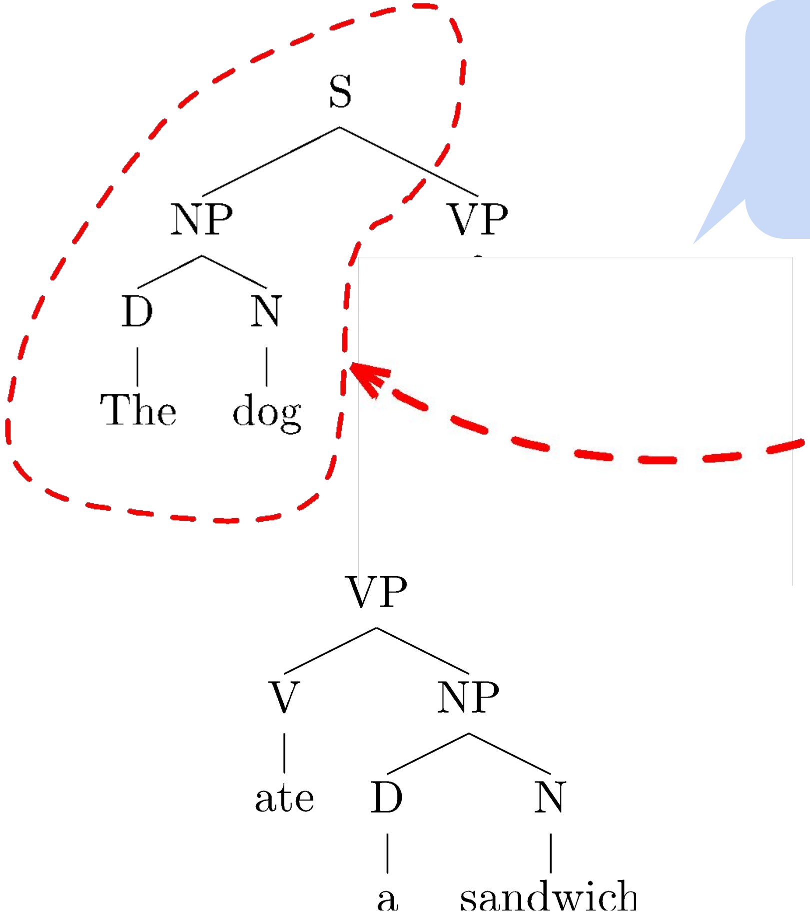
$D \rightarrow the$

Why context-free?

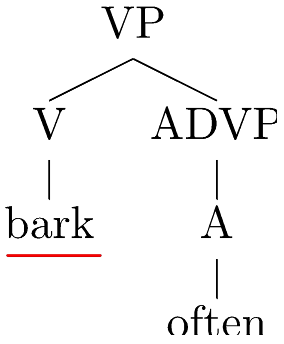


What can be a sub-tree is only affected by what the phrase type is (VP) but not the context

Why context-free?



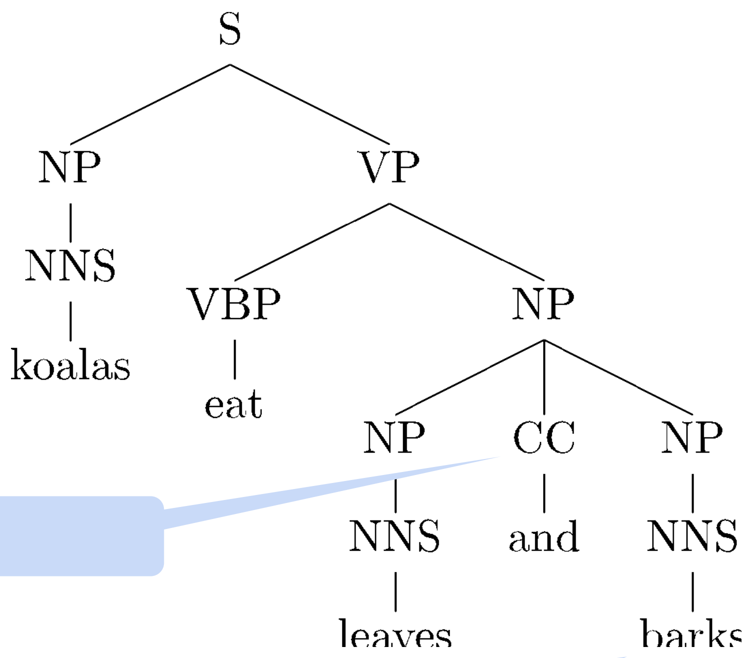
What can be a sub-tree is only affected by what the phrase type is (VP) but not the context



Not grammatical

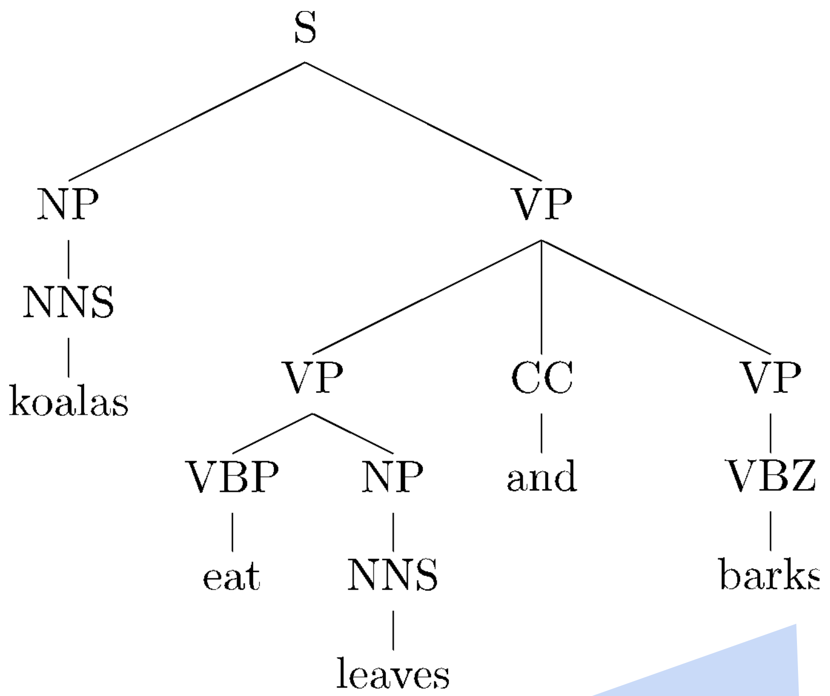
Coordination ambiguity

- Here, the coarse VP and NP categories cannot enforce subject-verb agreement in number resulting in the coordination ambiguity



Coordination

"Bark" can refer both to a noun or a verb

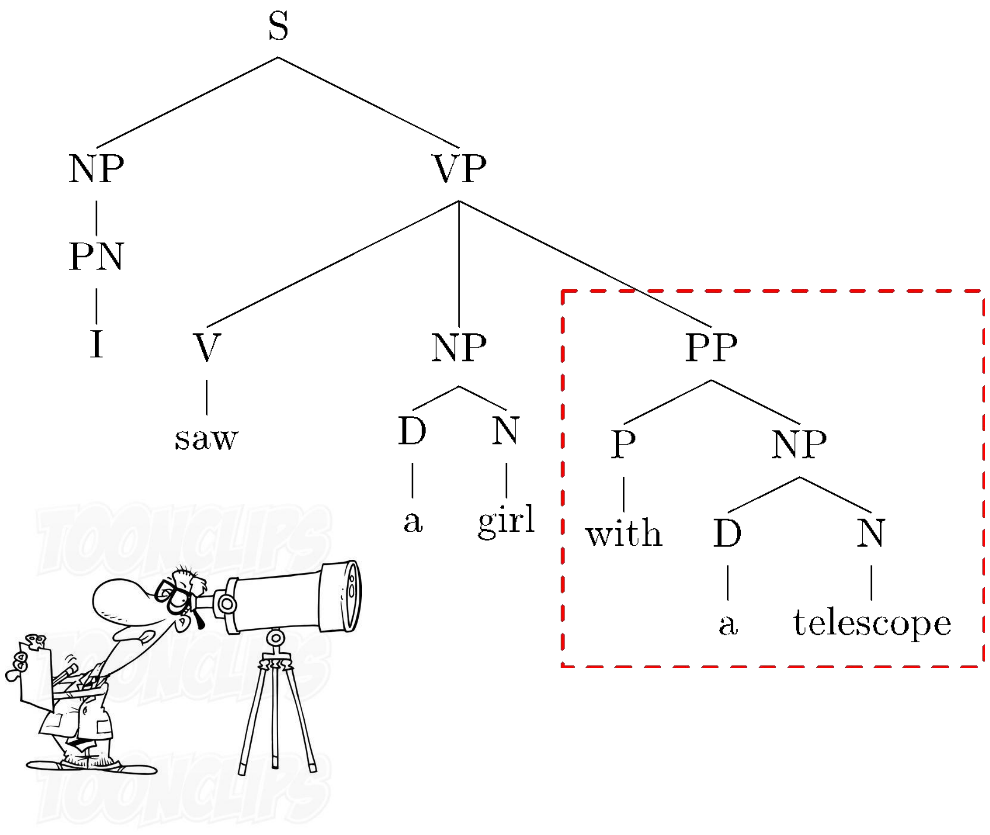


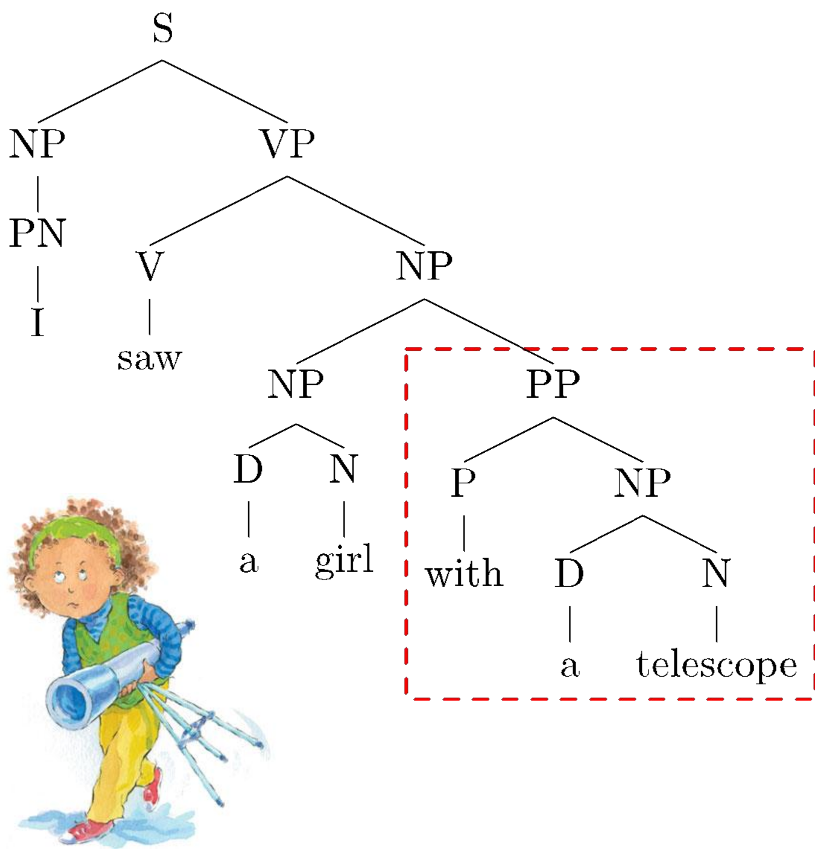
This tree would be ruled out if the context would be somehow captured (subject-verb agreement)

Ambiguities

Why parsing is hard? Ambiguity

- Prepositional phrase attachment ambiguity





PP Ambiguity

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

- 3 prepositional phrases, 5 interpretations:
  - Put the block **((in the box on the table) in the kitchen)**
  - Put the block (in the box (on the table in the kitchen))
  - Put ((the block in the box) on the table) in the kitchen.
  - Put (the block (in the box on the table)) in the kitchen.
  - Put **(the block in the box) (on the table in the kitchen)**

PP Ambiguity

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

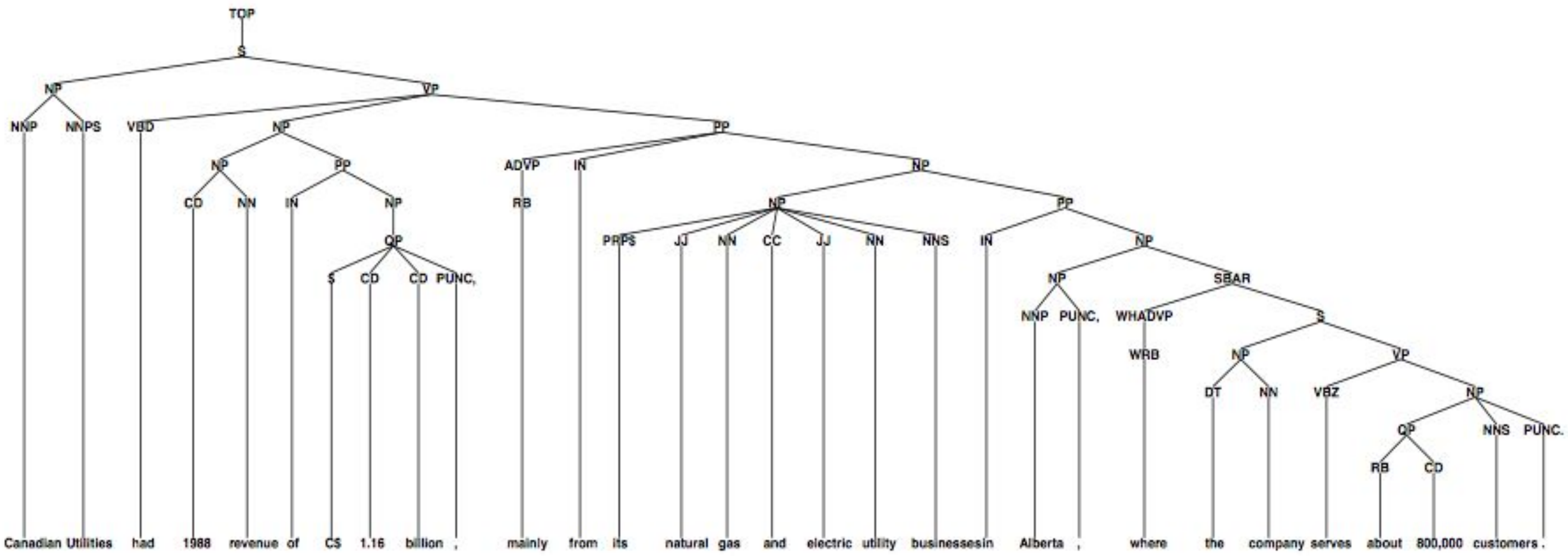
- 3 prepositional phrases, 5 interpretations:
  - Put the block ((in the box on the table) in the kitchen)
  - Put the block (in the box (on the table in the kitchen))
  - Put ((the block in the box) on the table) in the kitchen.
  - Put (the block (in the box on the table)) in the kitchen.
  - Put (the block in the box) (on the table in the kitchen)
- A general case:

$$Cat_n = \binom{2n}{n} - \binom{2n}{n-1} \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$$

Catalan numbers

$$1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, \ldots$$

A typical tree from a standard dataset (Penn treebank WSJ)



Canadian Utilities had 1988 revenue of $ 1.16 billion , mainly from its natural gas and electric utility businesses in Alberta , where the company serves about 800,000 customers .

[from Michael Collins slides]

Syntactic Ambiguities I

- Prepositional phrases:
  They cooked the beans in the pot on the stove with handles.
- Particle vs. preposition:
  The puppy tore up the staircase.
- Complement structures
  The tourists objected to the guide that they couldn't hear.
  She knows you like the back of her hand.
- Gerund vs. participial adjective
  Visiting relatives can be boring.
  Changing schedules frequently confused passengers.

Syntactic Ambiguities II

- Modifier scope within NPs
  impractical design requirements
  plastic cup holder

- Multiple gap constructions
  The chicken is ready to eat.
  The contractors are rich enough to sue.

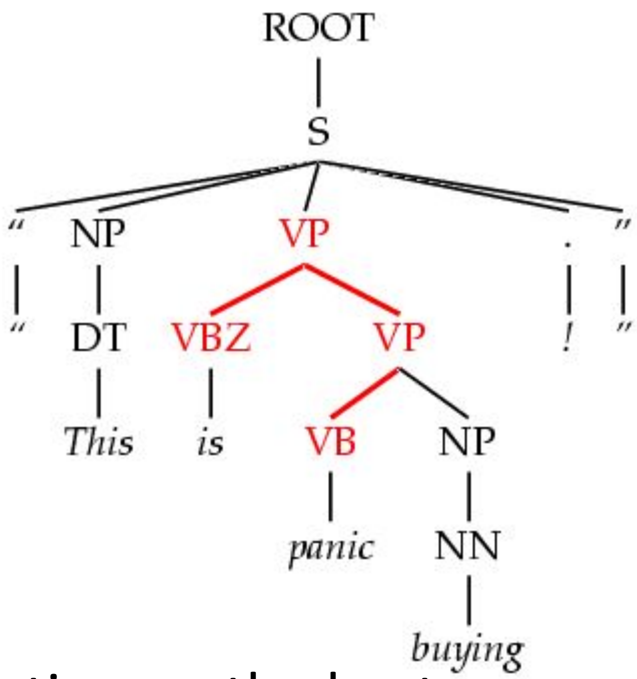
- Coordination scope:
  Small rats and mice can squeeze into holes or cracks in the wall.

Dark Ambiguities

- *Dark ambiguities*: most analyses are shockingly bad (meaning, they don't have an interpretation you can get your mind around)

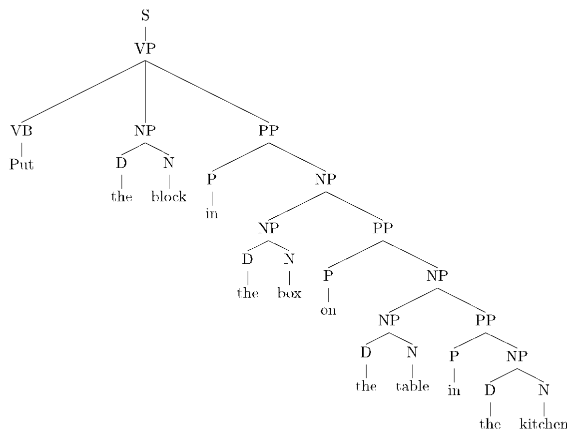
This analysis corresponds to the correct parse of

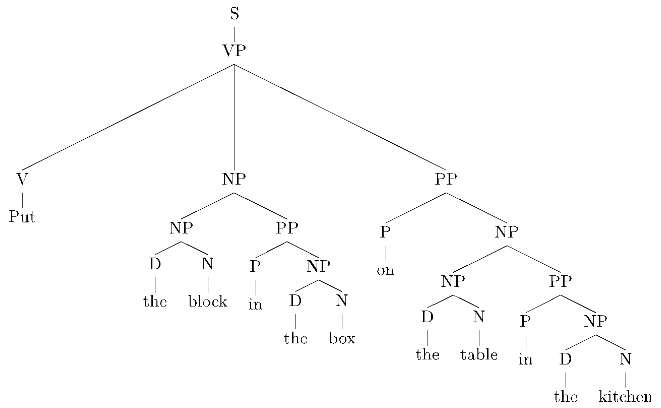
"This is panic buying ! "

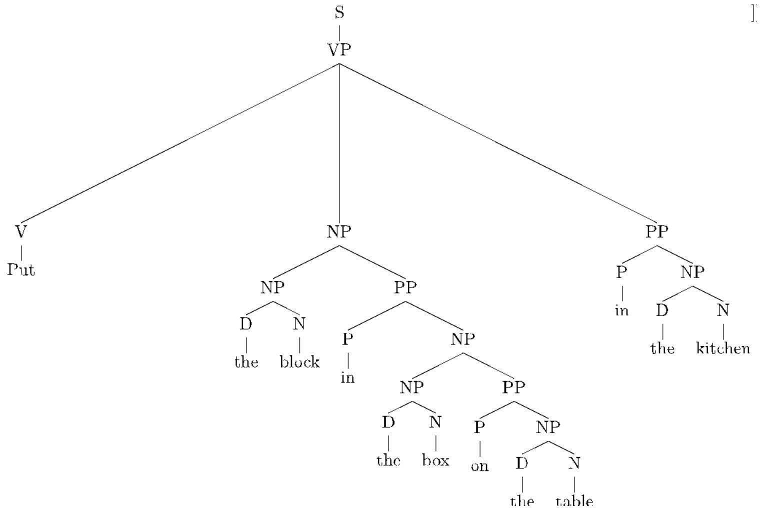


- Unknown words and new usages
- Solution: We need mechanisms to focus attention on the best ones, probabilistic techniques do this

How to Deal with Ambiguity?







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

PCFGs

Probabilistic Context-Free Grammars

- A context-free grammar is a tuple <*N, T, S, R*>
  - *N* : the set of non-terminals
    - Phrasal categories: S, NP, VP, ADJP, etc.
    - Parts-of-speech (pre-terminals): NN, JJ, DT, VB
  - *T* : the set of terminals (the words)
  - *S* : the start symbol
    - Often written as ROOT or TOP
    - *Not* usually the sentence non-terminal S
  - *R* : the set of rules
    - Of the form $X \rightarrow Y_1\ Y_2\ \ldots\ Y_k$, with $X, Y_i \in N$
    - Examples: $S \rightarrow NP\ VP$, $VP \rightarrow VP\ CC\ VP$
    - Also called rewrites, productions, or local trees

- A PCFG adds:
  - A top-down production probability per rule $P(Y_1\ Y_2\ \ldots\ Y_k \mid X)$

PCFGs

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

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

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

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

$S \to NP\ \ VP$	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)

$N \to girl$	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

PCFGs

$S \rightarrow NP\ VP$ 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

S

$p(T) =$

PCFGs

$S \rightarrow NP \;\; VP$ 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

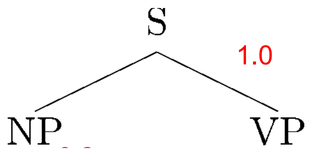
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



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

PCFGs

$S \rightarrow NP \;\; VP$ 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

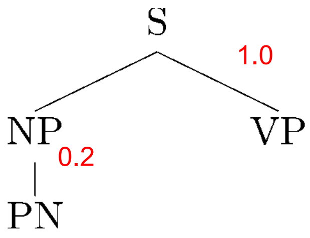
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



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

PCFGs

$S \rightarrow NP \;\; VP$ 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

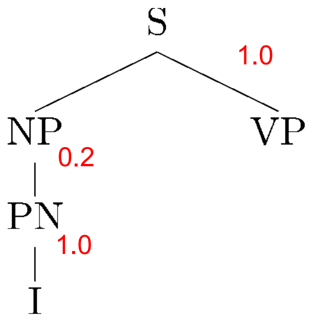
$V \rightarrow ate$ 0.5

$P \rightarrow with$ 0.6

$P \rightarrow in$ 0.4

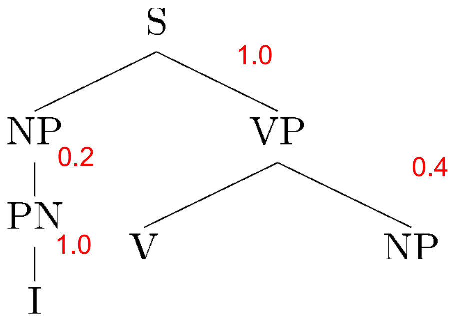
$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7



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

PCFGs



$S \rightarrow NP \;\; VP$ 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

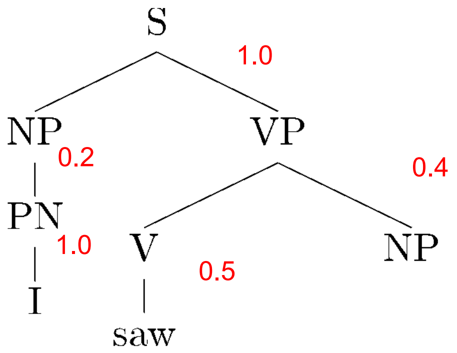
$P \rightarrow in$ 0.4

$D \rightarrow a$ 0.3

$D \rightarrow the$ 0.7

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

PCFGs



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

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

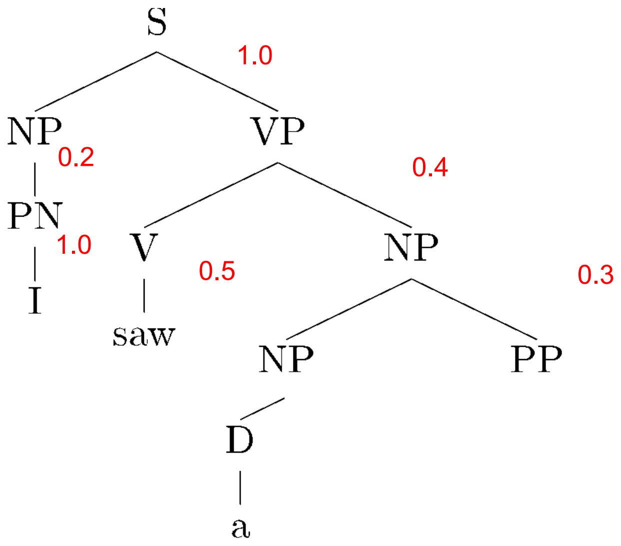
$$NP \rightarrow NP \;\; PP \quad 0.3$$
$$NP \rightarrow D \;\; N \quad 0.5$$
$$NP \rightarrow PN \quad 0.2$$

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

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

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

PCFGs



$S \rightarrow NP\ \ VP$ 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

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

PCFGs

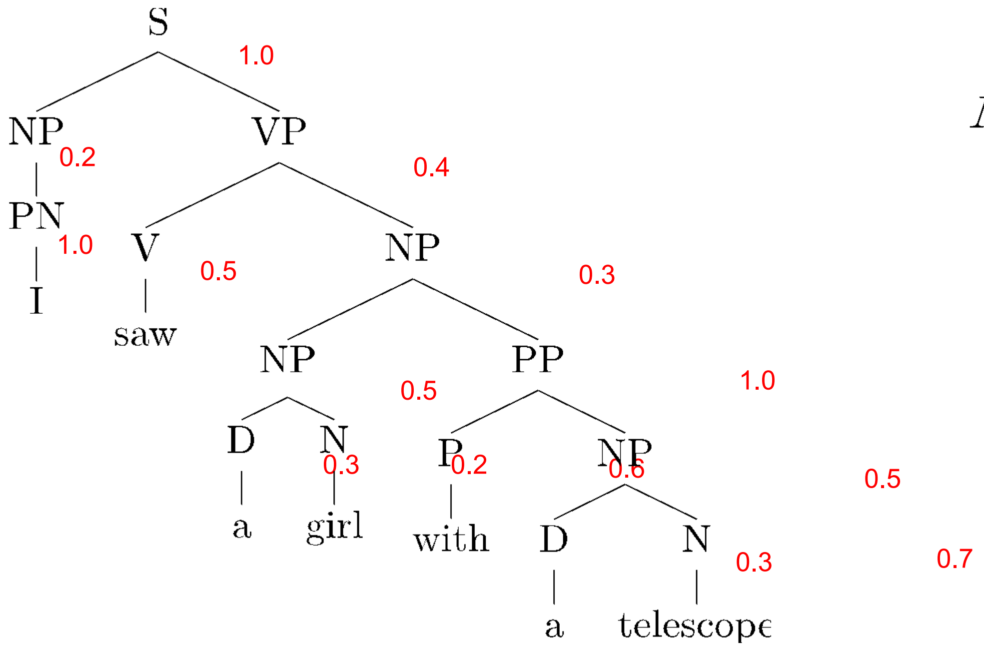
$$S \rightarrow NP \;\; VP \;\; 1.0$$

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

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

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

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

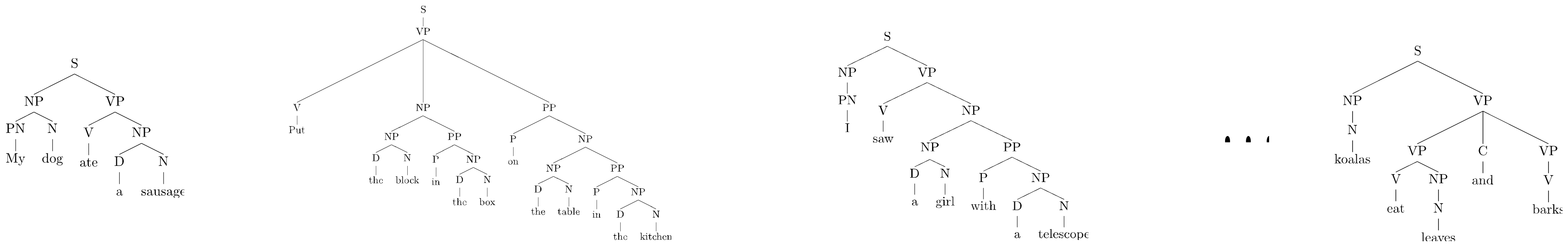


$$p(T) = 1.0 \times 0.2 \times 1.0 \times 0.4 \times 0.5 \times 0.3 \times$$
$$0.5 \times 0.3 \times 0.2 \times 1.0 \times 0.6 \times 0.5 \times 0.3 \times 0.7 = 2.26 \times 10^{-5}$$

PCFG Estimation

ML estimation

- A treebank: a collection sentences annotated with constituent trees



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

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

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

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: $\sum_T P(T) < 1$

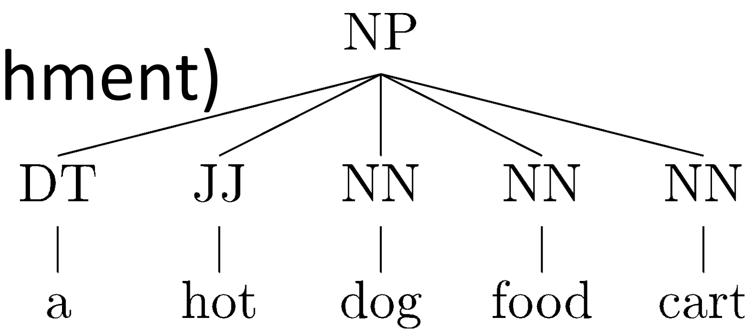
- Good news: any PCFG estimated with the maximum likelihood procedure are always proper (Chi and Geman, 98)

Penn Treebank: peculiarities

- Wall street journal: around 40, 000 annotated sentences, 1,000,000 words
  - Fine-grained 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)



CKY Parsing

Parsing

- Parsing is search through the space of all possible parses
  - e.g., we may want either any parse, all parses or the highest scoring parse (if PCFG):

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

- Bottom-up:
  - One starts from words and attempt to construct the full tree
- Top-down
  - Start from the start symbol and attempt to expand to get the sentence

CKY algorithm (aka CYK)

- Cocke-Kasami-Younger algorithm
  - Independently discovered in late 60s / early 70s

- An efficient bottom up parsing algorithm for (P)CFGs
  - can be used both for the recognition and parsing problems
  - Very important in NLP (and beyond)

- We will start with the non-probabilistic version

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)

$N \to telescope$ $D \to the$

$$C \to C_1 C_2$$

Binary inner rules $S \to NP\ VP$ $NP \to D\ N$

Constraints on the grammar

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

$$C \to x$$

$$C \to C_1 C_2$$

- 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 by defining such transformations that allows for easy reverse transformation

Transformation to CNF form

- What one need to do to convert to CNF form

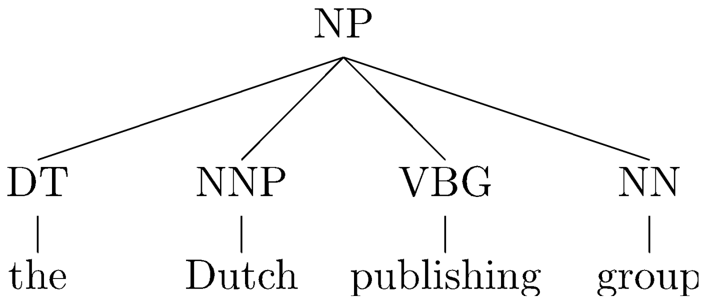
  - Get rid of unary rules: $C \to C_1$
  - Get rid of N-ary rules: $C \to C_1\, C_2 \ldots C_n \quad (n > 2)$

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

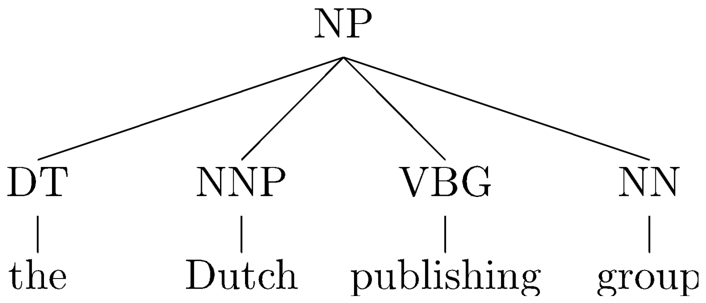
- Consider $NP \rightarrow DT\ \ NNP\ \ VBG\ \ NN$



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

Transformation to CNF form: binarization

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

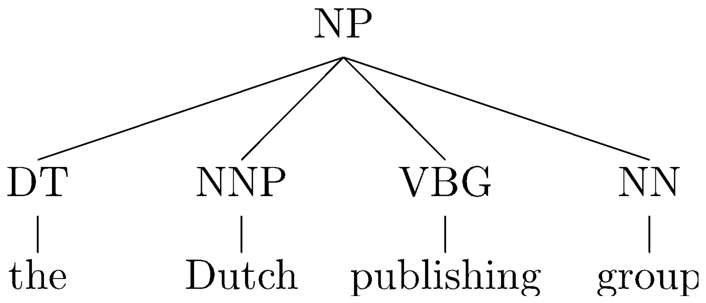


- How do we get a set of binary rules which are equivalent?

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

Transformation to CNF form: binarization

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



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

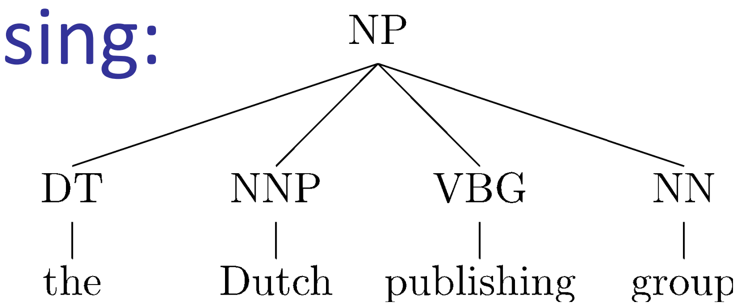
$$NP \rightarrow DT\ \ X$$
$$X \rightarrow NNP\ \ Y$$
$$Y \rightarrow VBG\ \ NN$$

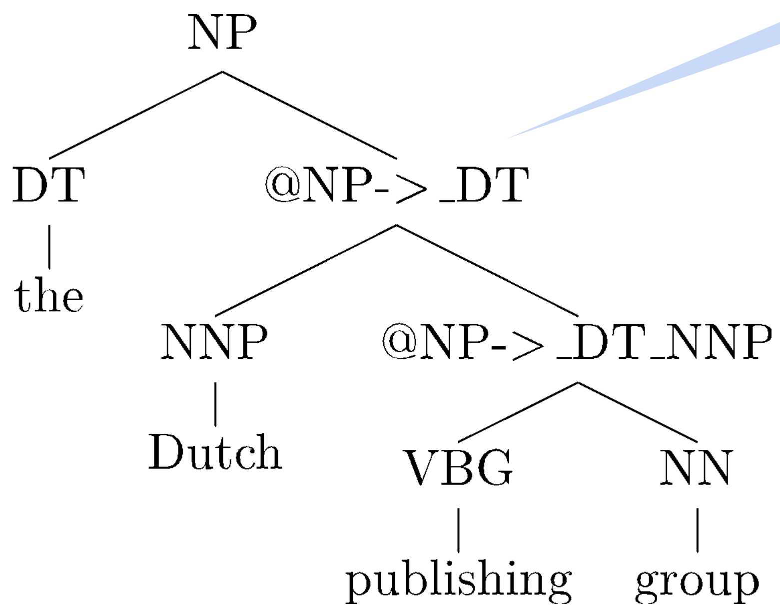
- A more systematic way to refer to new non-terminals

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

Transformation to CNF form: binarization

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





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

Can be easily reversed on postprocessing

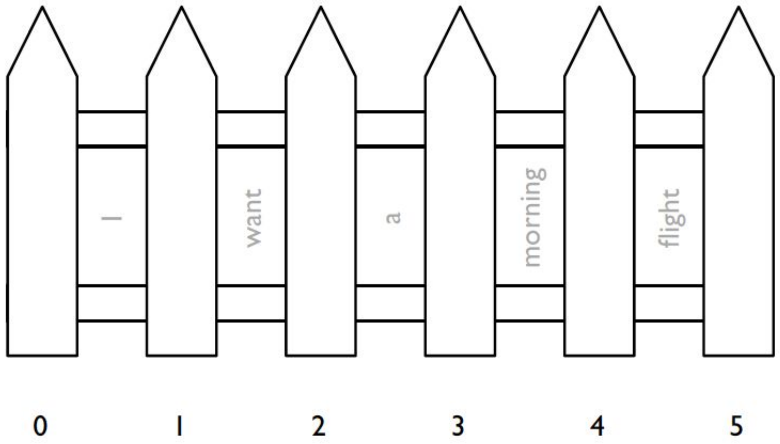
CKY: Parsing task

- We a given
  - a grammar <N, T, S, R>
  - a sequence of words $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$

- Our goal is to produce a parse tree for w

CKY: Parsing task

- We a given
  - a grammar <N, T, S, R>
  - a sequence of words $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$
- Our goal is to produce a parse tree for w
- We need an easy way to refer to substrings of w



indices refer to fenceposts

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

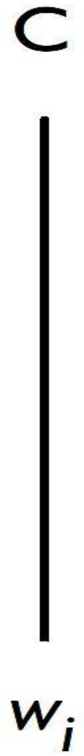
Parsing one word

$$C \to w_i$$

w_i

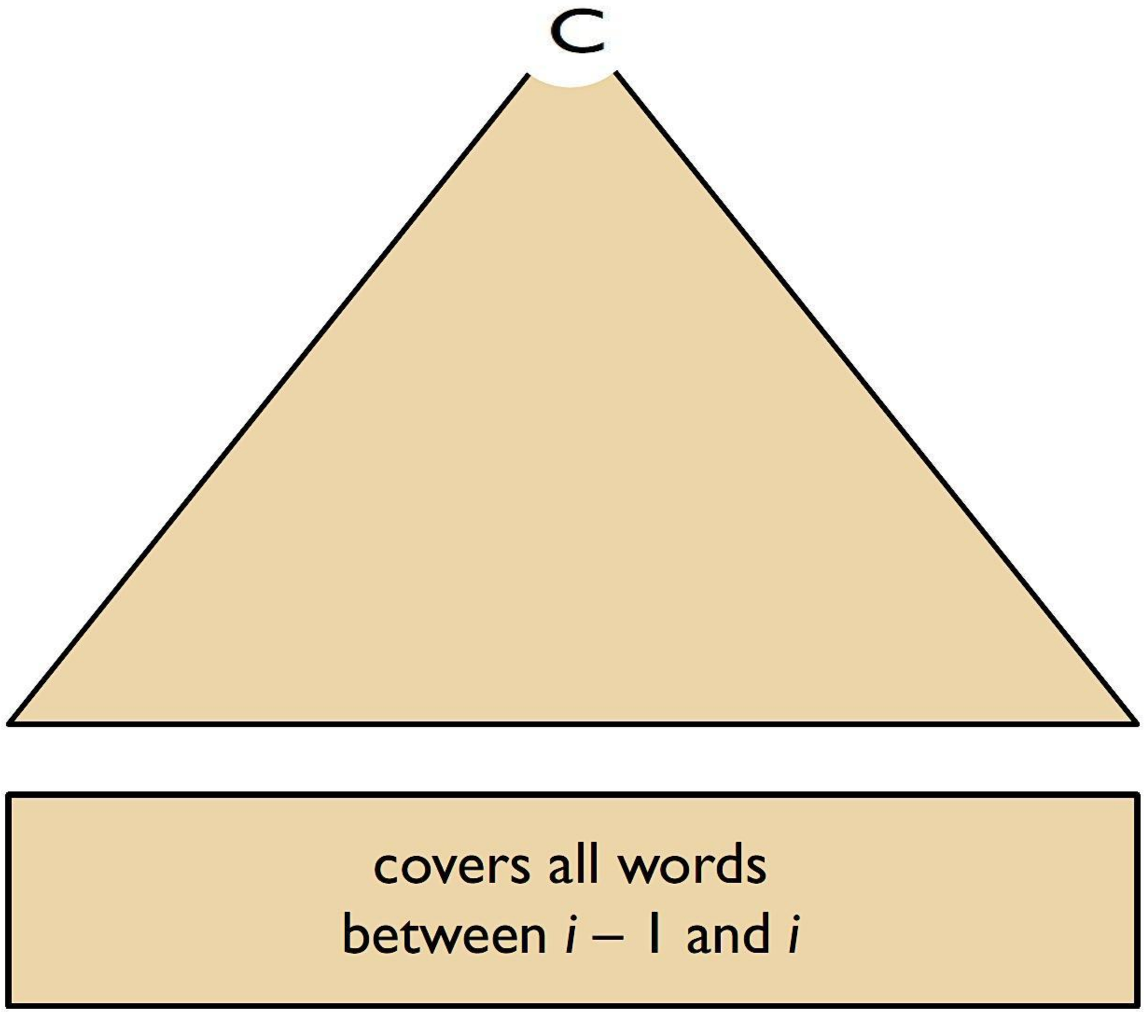
Parsing one word

$C \rightarrow w_i$



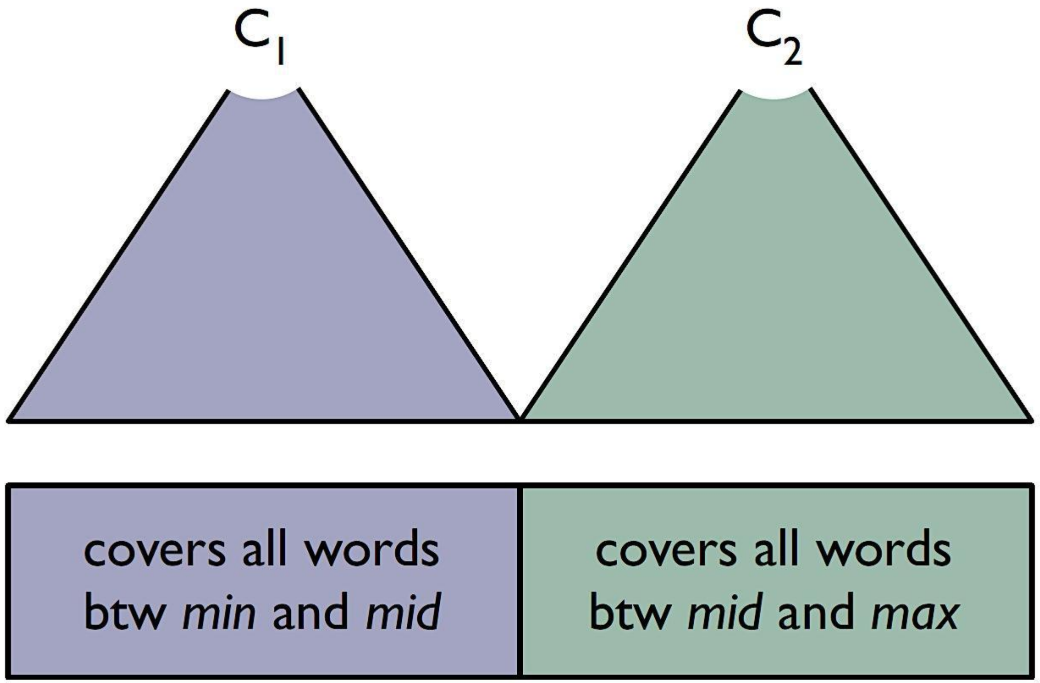
Parsing one word

$C \to w_i$



Parsing longer spans

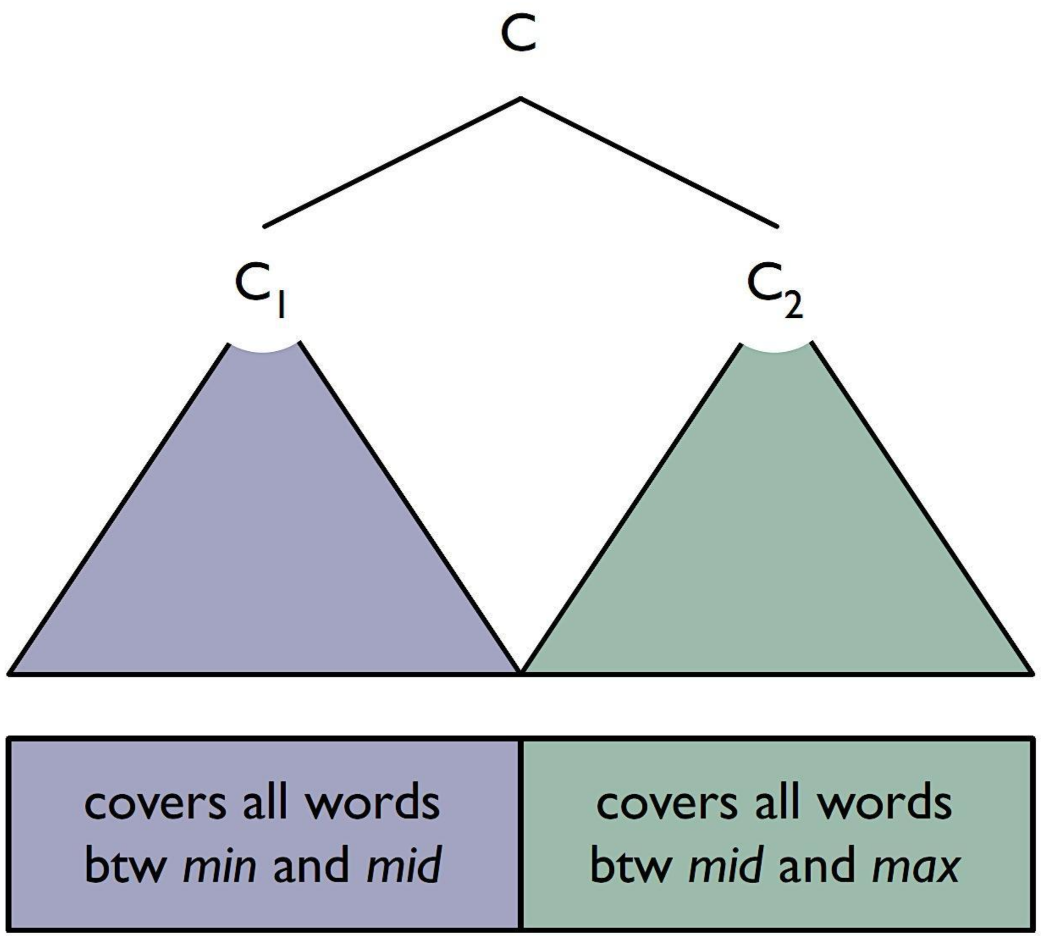
$C \rightarrow C_1 \;\; C_2$



Check through all C1, C2, mid

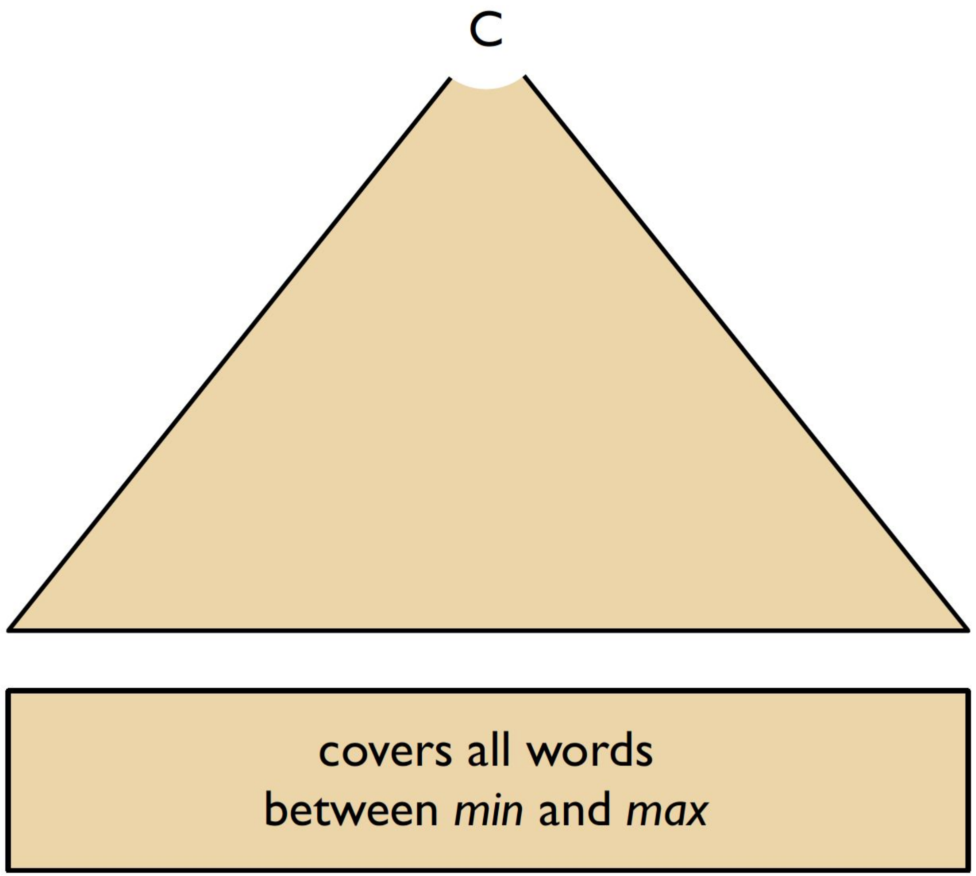
Parsing longer spans

$$C \to C_1 \ \ C_2$$



Check through all C1, C2, mid

Parsing longer spans



CKY in action

| lead | can | poison |

0 1 2 3

Inner rules

$$S \rightarrow NP\ VP$$

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

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

Preterminal rules

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

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

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

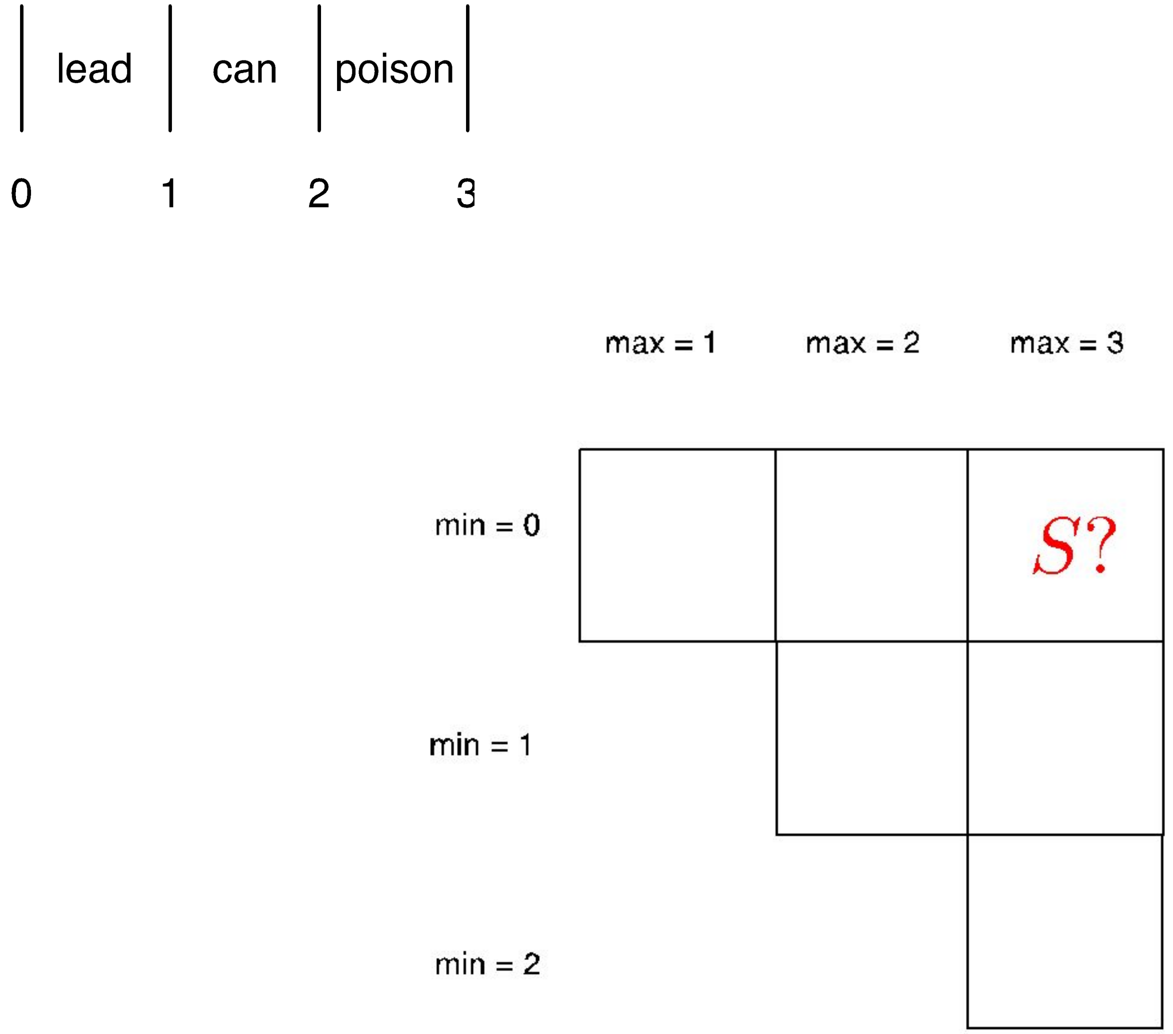


Chart (aka parsing triangle)

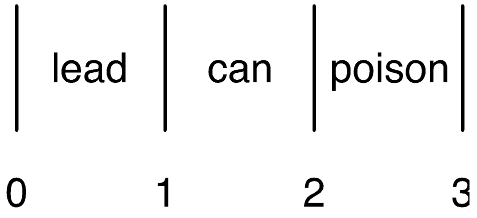
$$S \rightarrow NP \ VP$$

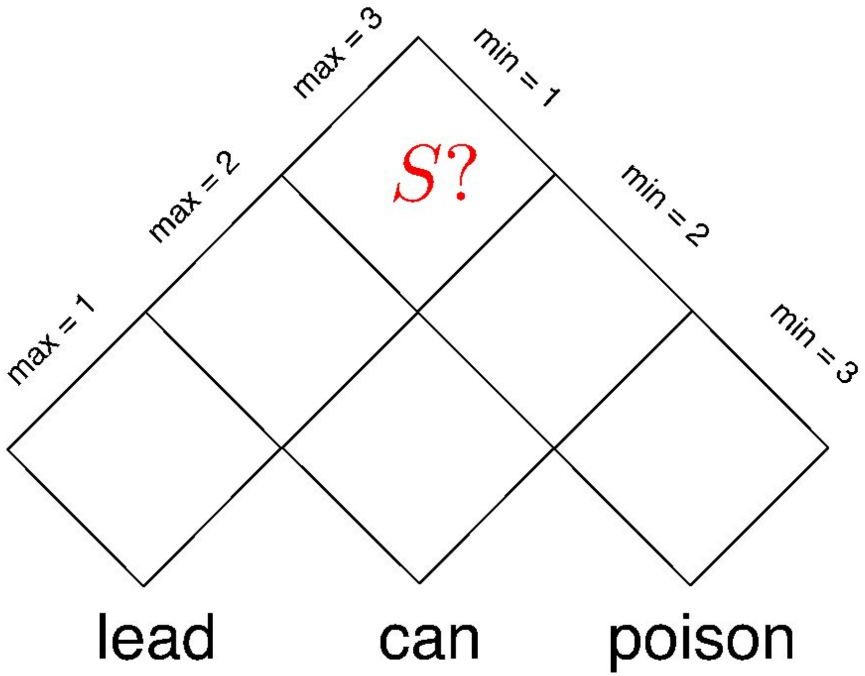
Inner rules

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

Preterminal rules

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





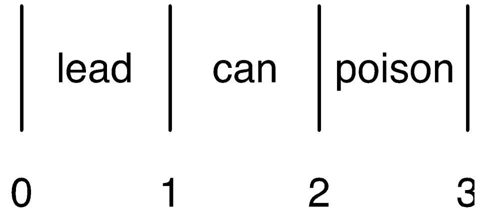
$$S \rightarrow NP \ VP$$

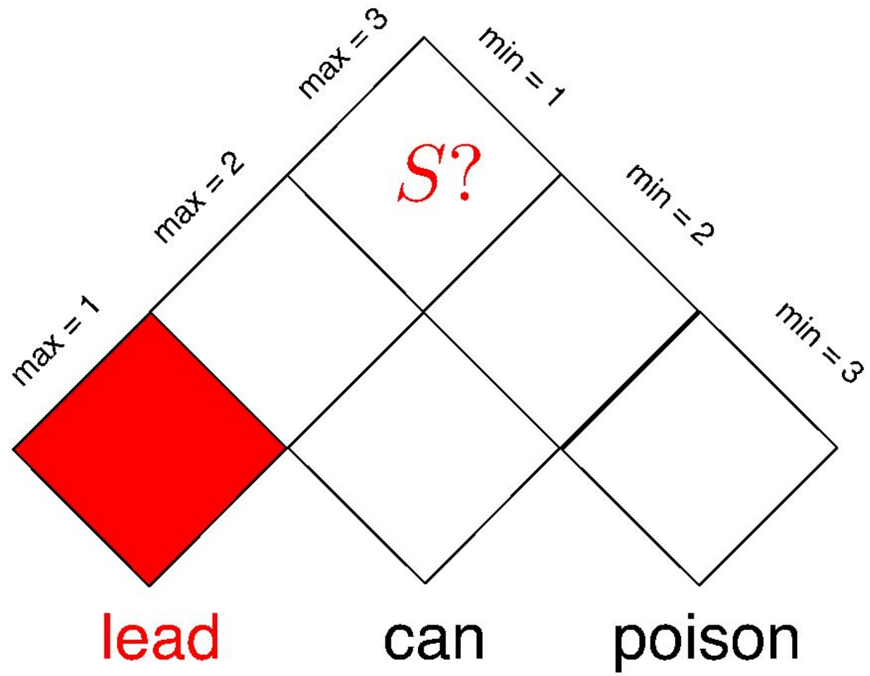
Inner rules

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

Preterminal rules

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





$$S \rightarrow NP\ \ VP$$

Inner rules

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

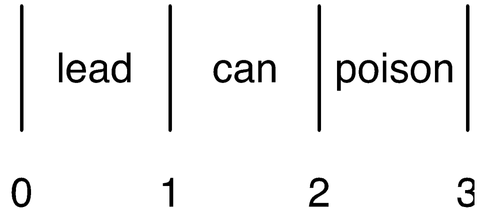
$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

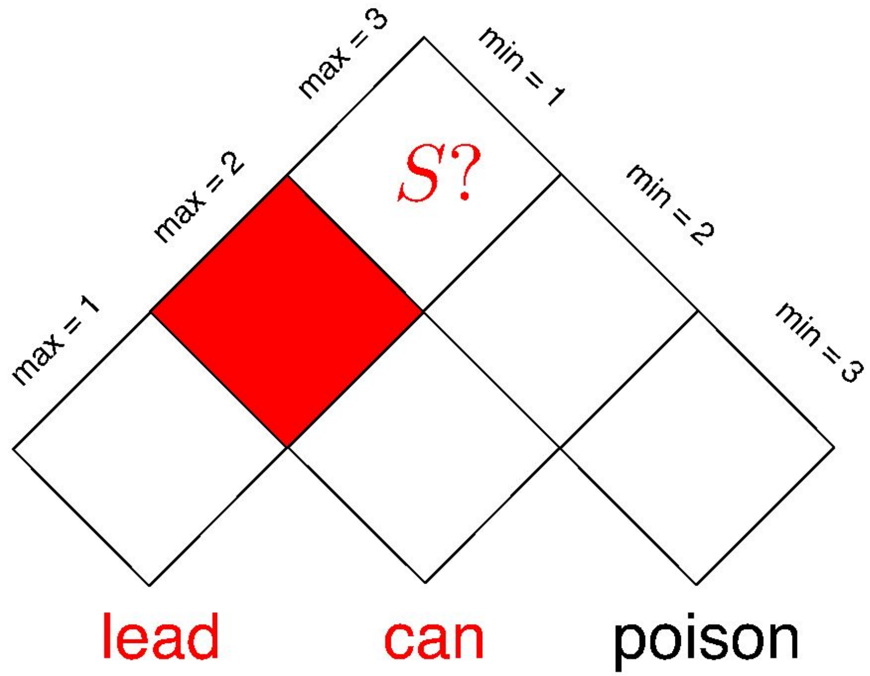
Preterminal rules

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

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

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





Inner rules

$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow M \ \ V$$

$$VP \rightarrow V$$

$$NP \rightarrow N$$

$$NP \rightarrow N \ NP$$

Preterminal rules

$$N \rightarrow can$$

$$N \rightarrow lead$$

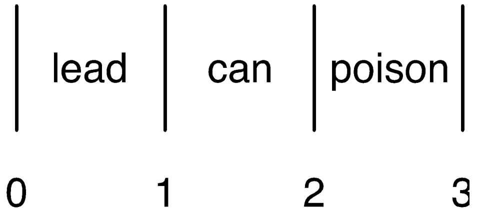
$$N \rightarrow poison$$

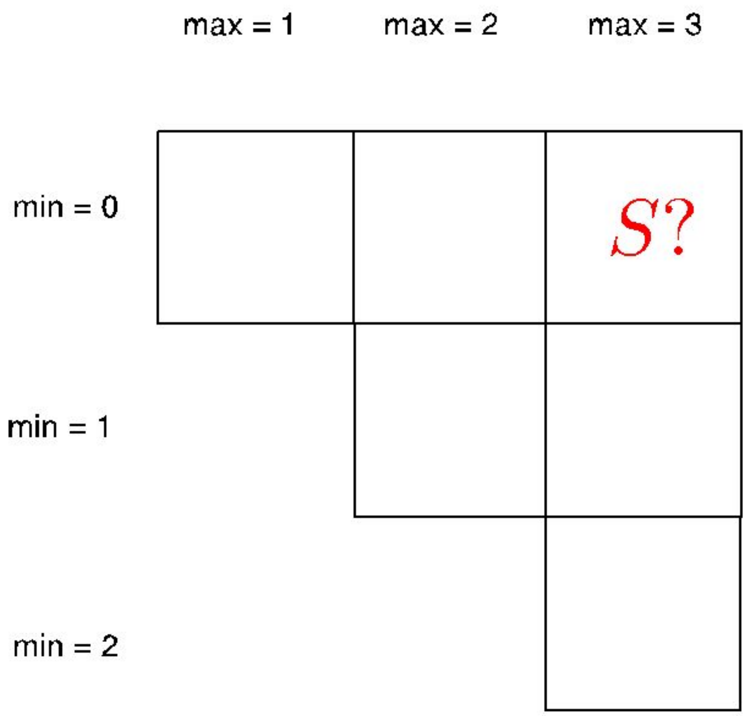
$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$





Inner rules

$$S \rightarrow NP \ \ VP$$

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

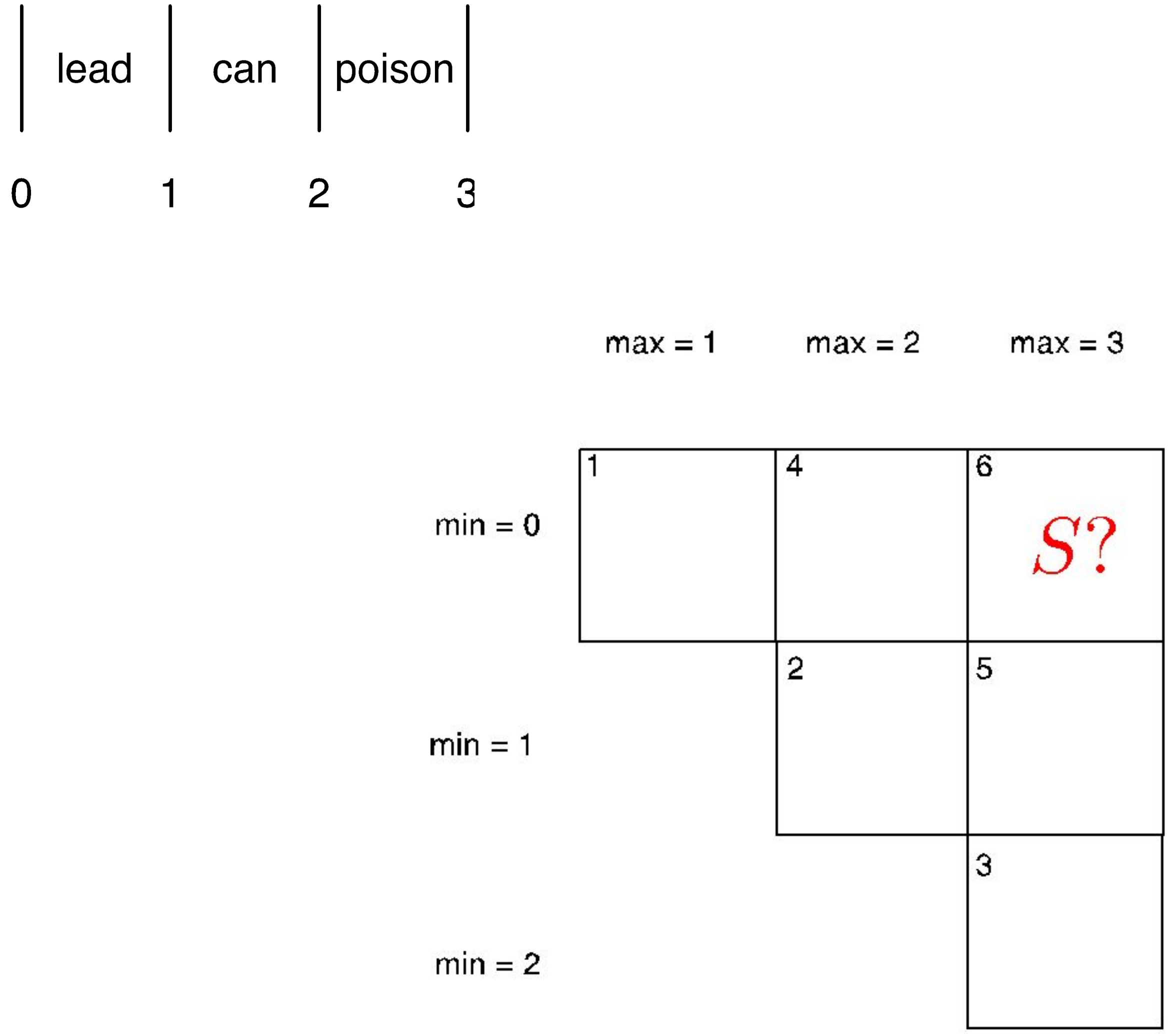
$$NP \rightarrow N$$
$$NP \rightarrow N \ NP$$

Preterminal rules

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

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

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



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

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

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

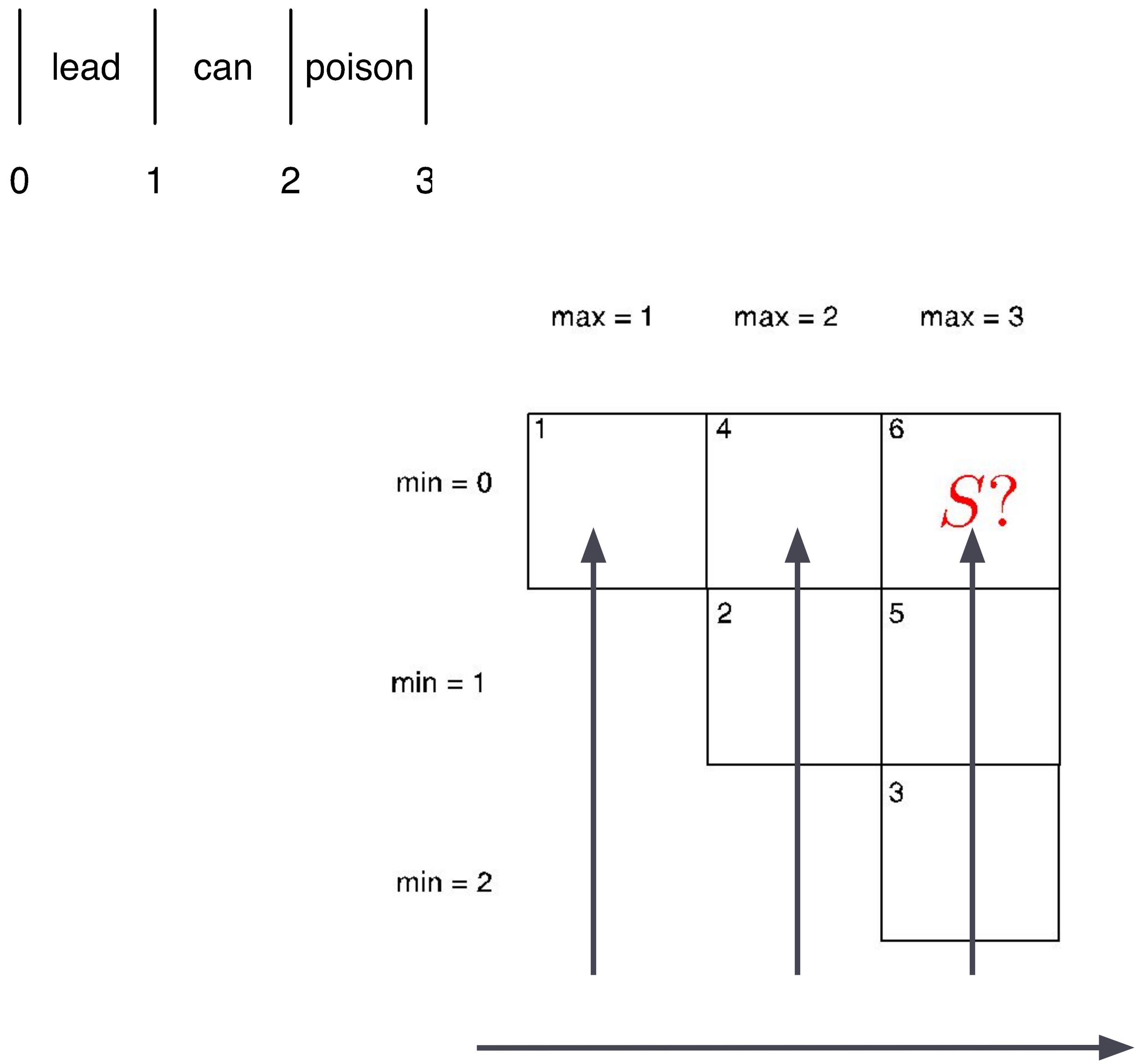
Inner rules

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

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

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

Preterminal rules



$$S \rightarrow NP\ VP$$

Inner rules

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

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

Preterminal rules

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

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

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

	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

$$N \rightarrow can$$

$$N \rightarrow lead$$

$$N \rightarrow poison$$

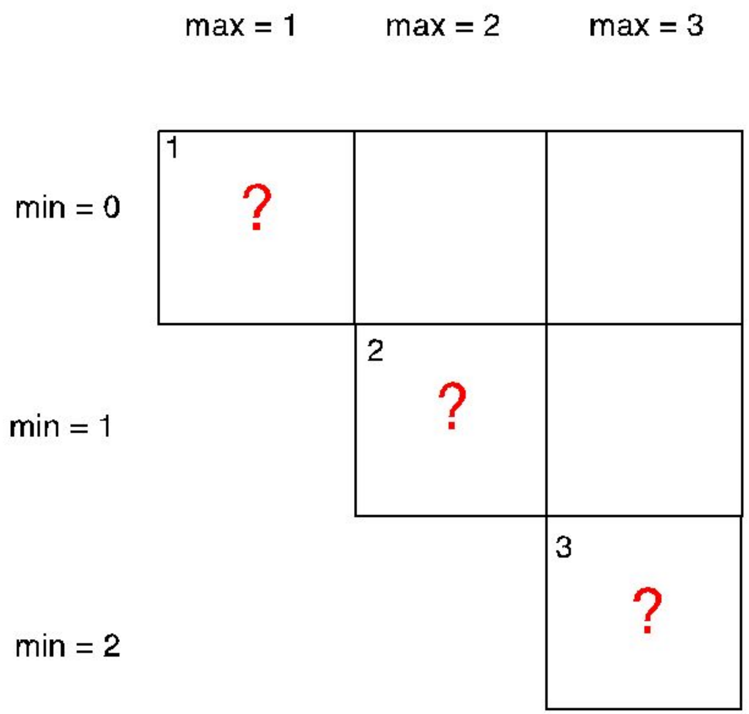
$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$

| lead | can | poison |
0 1 2 3



Inner rules

$$S \rightarrow NP\ VP$$

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

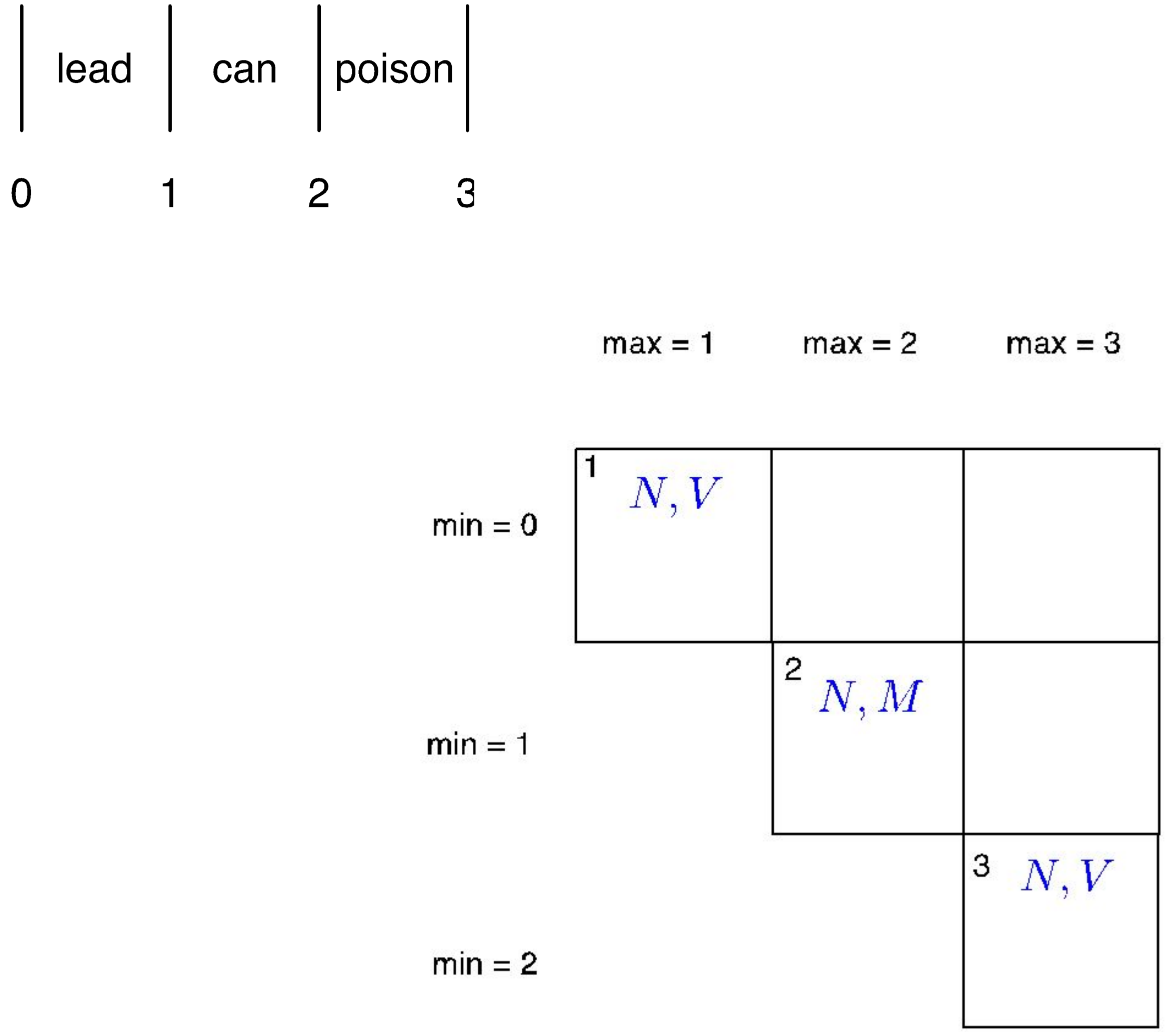
$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

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

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

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



Inner rules

$$S \rightarrow NP \ VP$$

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

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

Preterminal rules

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

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

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

lead	can	poison

0 1 2 3

Check about unary rules

	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

$$S \rightarrow NP\ VP$$

Inner rules

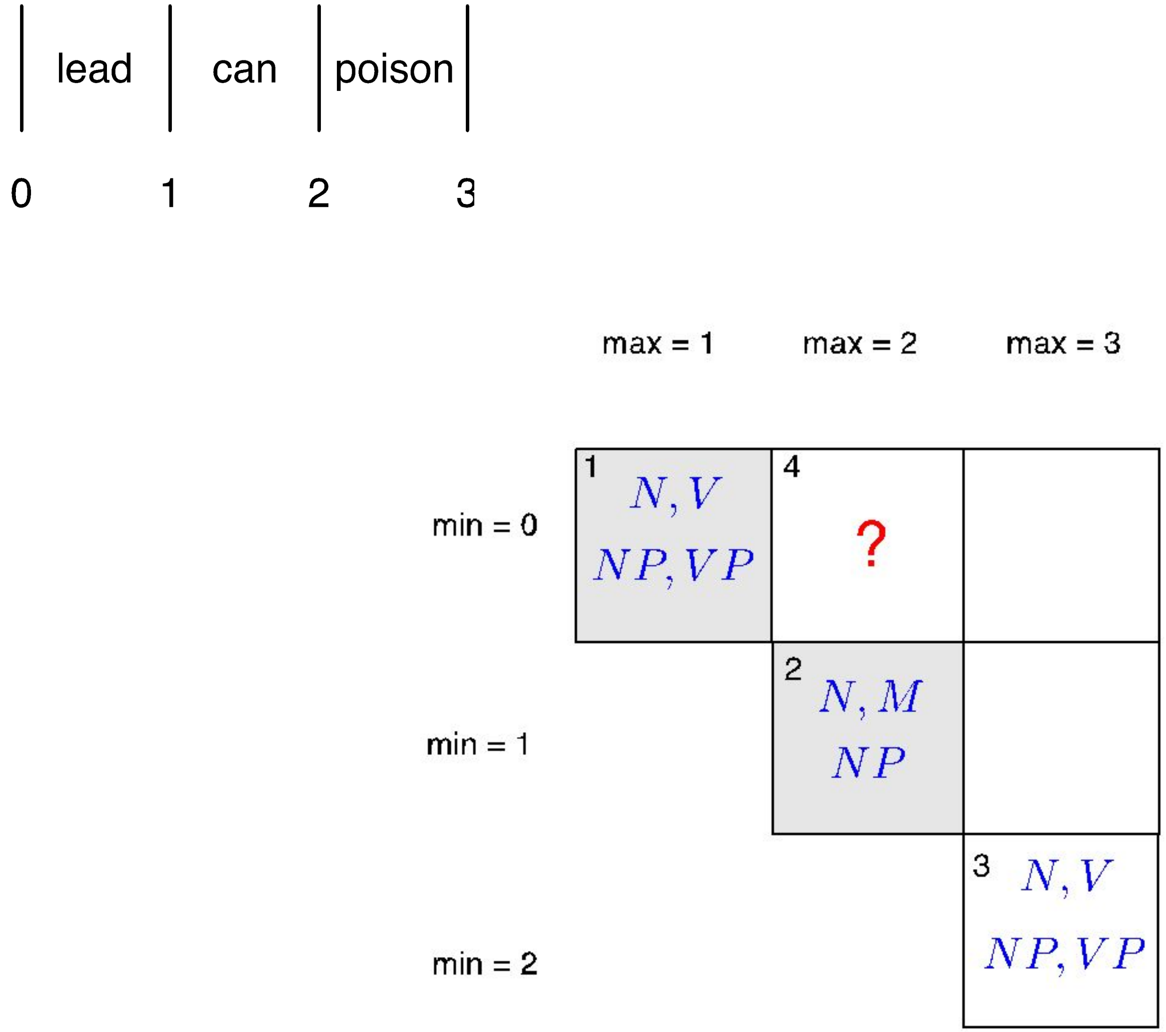
$$VP \rightarrow M\ V$$
$$VP \rightarrow V$$
$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

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

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

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



$$S \rightarrow NP\ VP$$

Inner rules

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

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

Preterminal rules

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

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

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

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

$$S \rightarrow NP\ VP$$

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

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

Preterminal rules

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

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

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

| 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

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

Inner rules

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

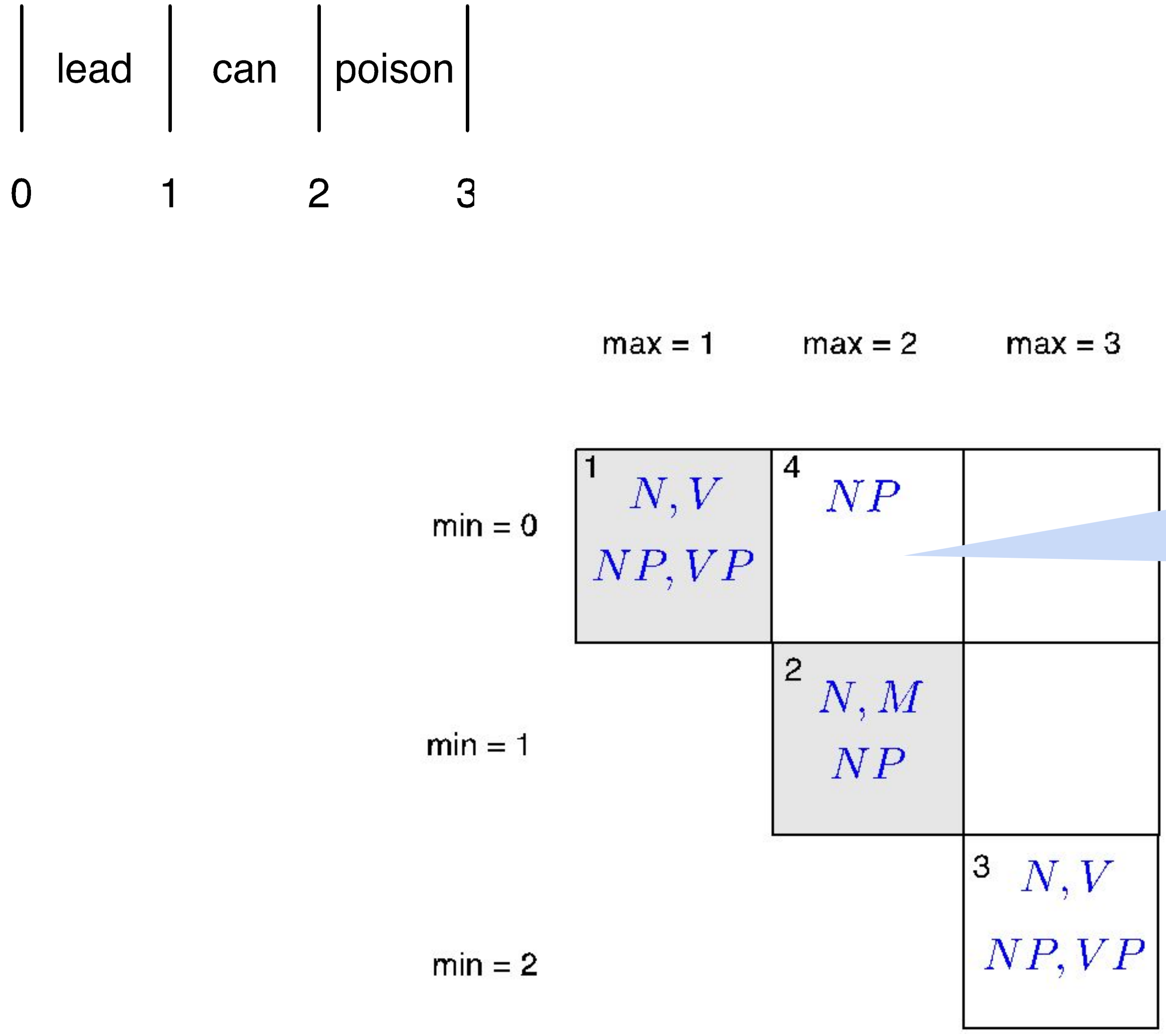
$$NP \rightarrow N$$
$$NP \rightarrow N \; NP$$

Preterminal rules

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

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

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



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

$$N \rightarrow can$$

$$N \rightarrow lead$$

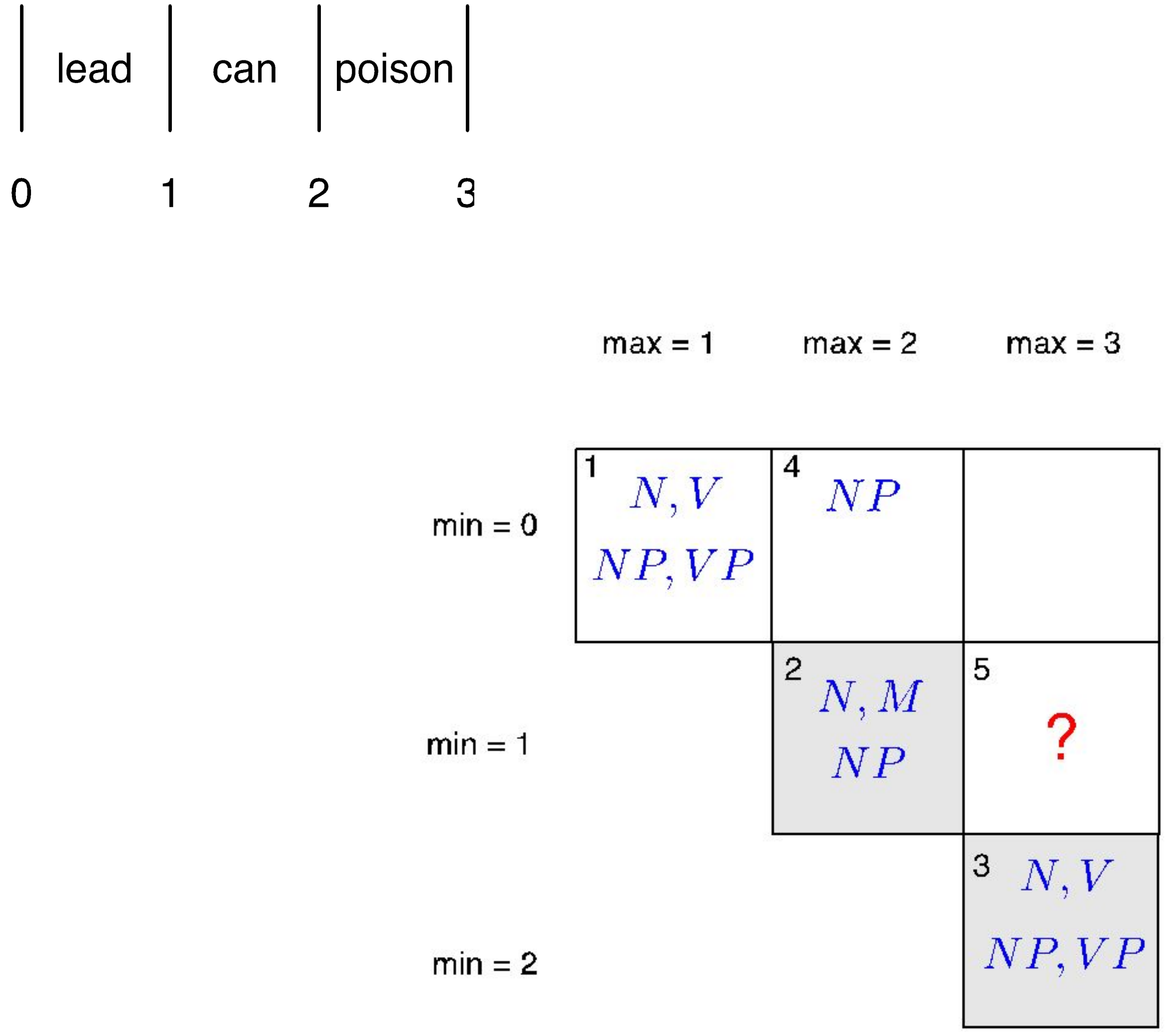
$$N \rightarrow poison$$

$$M \rightarrow can$$

$$M \rightarrow must$$

$$V \rightarrow poison$$

$$V \rightarrow lead$$



$$S \rightarrow NP\ VP$$

Inner rules

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

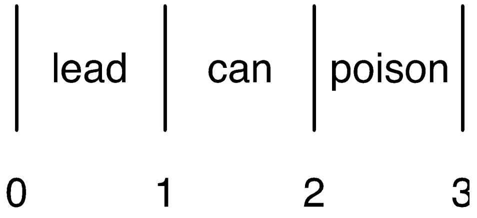
$$NP \rightarrow N$$
$$NP \rightarrow N\ NP$$

Preterminal rules

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

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

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



	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

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

Check about unary rules: no unary rules here

Inner rules

$$S \rightarrow NP\ VP$$

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

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

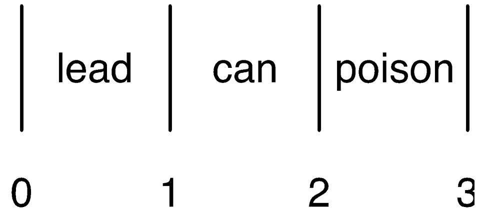
Preterminal rules

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

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

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

$$S \rightarrow NP\ VP$$



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

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

Inner rules

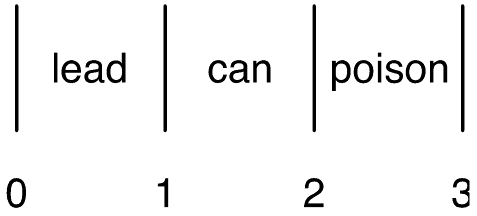
	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 ?
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

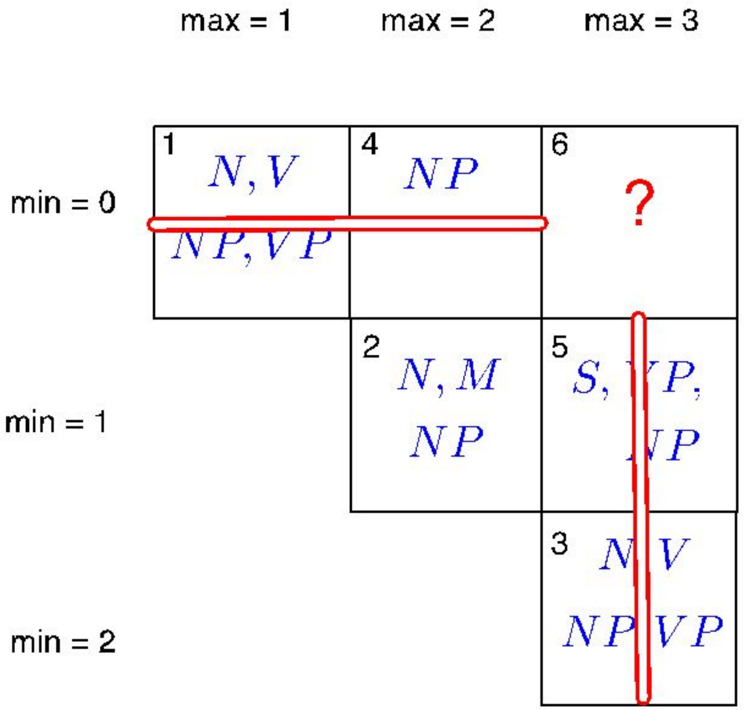
$$N \rightarrow can$$
$$N \rightarrow lead$$
$$N \rightarrow poison$$

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

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

Preterminal rules





$$S \rightarrow NP\ VP$$

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

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

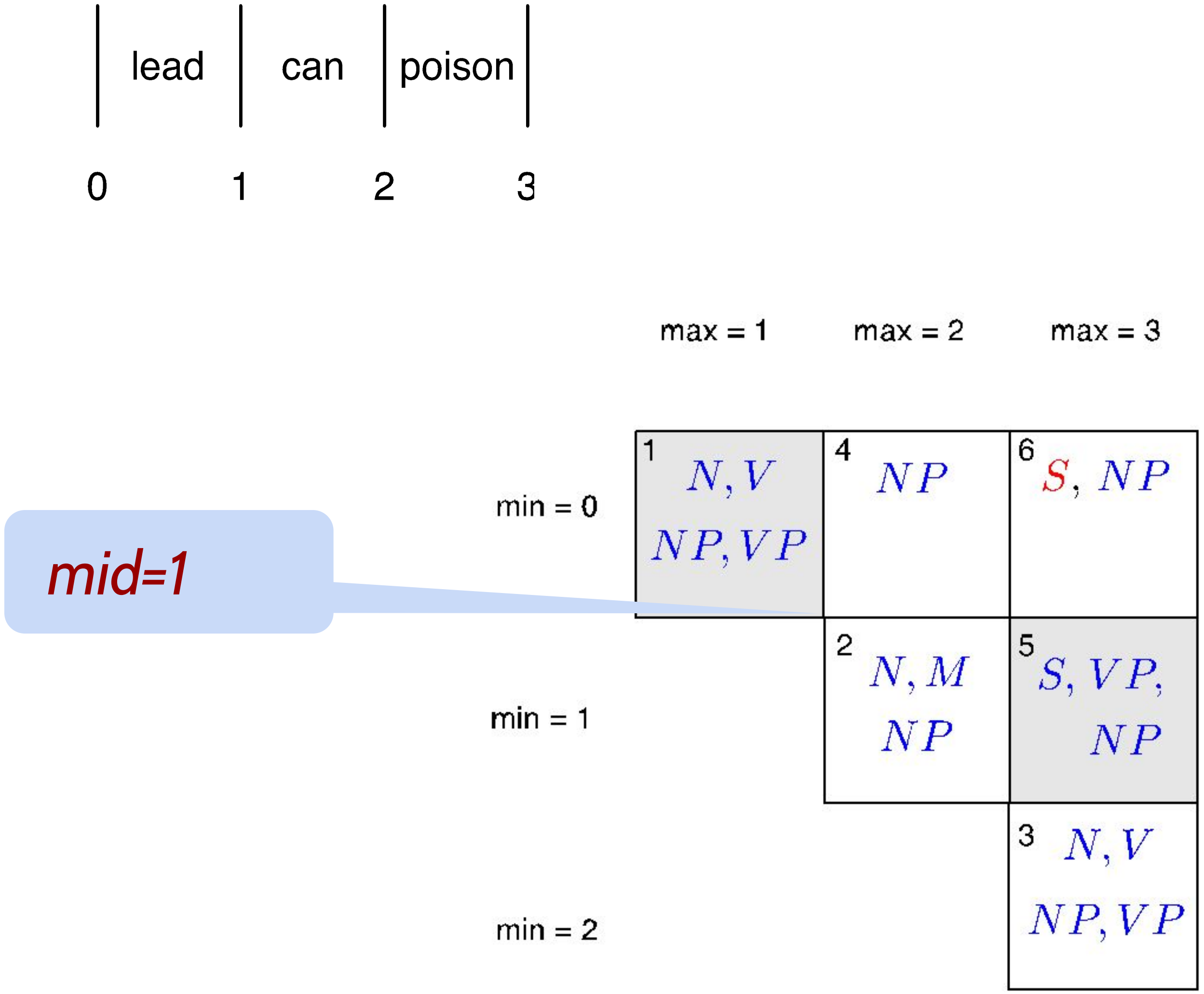
Inner rules

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

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

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

Preterminal rules



$S \to NP\ VP$

Inner rules

$VP \to M\ V$

$VP \to V$

$NP \to N$

$NP \to N\ NP$

Preterminal rules

$N \to can$

$N \to lead$

$N \to poison$

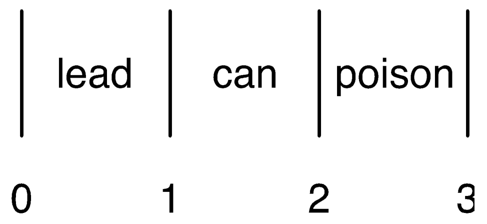
$M \to can$

$M \to must$

$V \to poison$

$V \to lead$

$S \rightarrow NP\ VP$



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

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

$M \rightarrow can$

$M \rightarrow must$

$V \rightarrow poison$

$V \rightarrow lead$

	lead		can		poison	
0		1		2		3

	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

Apparently the sentence is ambiguous for the grammar: (as the grammar overgenerates)

$S \rightarrow NP\ VP$

Inner rules

$VP \rightarrow M\ V$

$VP \rightarrow V$

$NP \rightarrow N$

$NP \rightarrow N\ NP$

Preterminal rules

$N \rightarrow can$

$N \rightarrow lead$

$N \rightarrow poison$

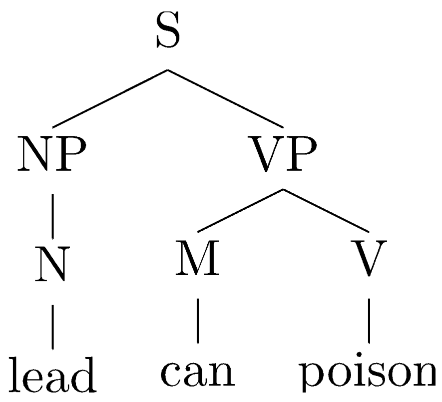
$M \rightarrow can$

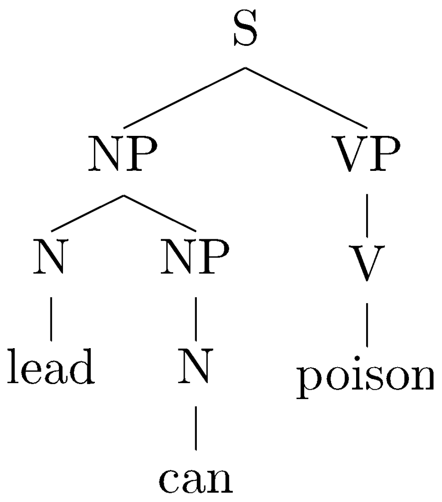
$M \rightarrow must$

$V \rightarrow poison$

$V \rightarrow lead$

Ambiguity

- 



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

CKY more formally

Here we assume that labels (C) are integer indices

Chart can be represented by a Boolean 3D array `chart[min][max][C]`

- Relevant entries have $0 < min < max \leq n$

`chart[min][max][C] = true` if the signature (min, max, C) is already added to the chart; `false` otherwise.

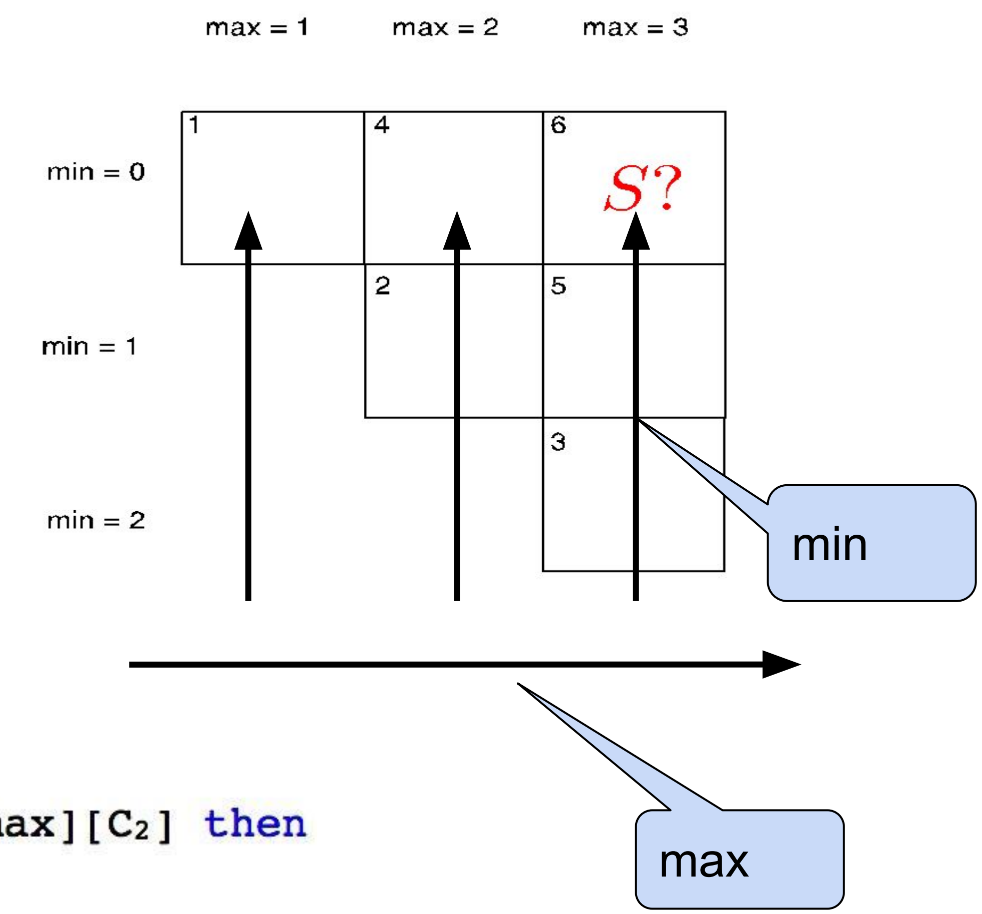
	max = 1	max = 2	max = 3
min = 0	1 N, V NP, VP	4 NP	6 $S, VP,$ NP
min = 1		2 N, M NP	5 $S, VP,$ NP
min = 2			3 N, V NP, VP

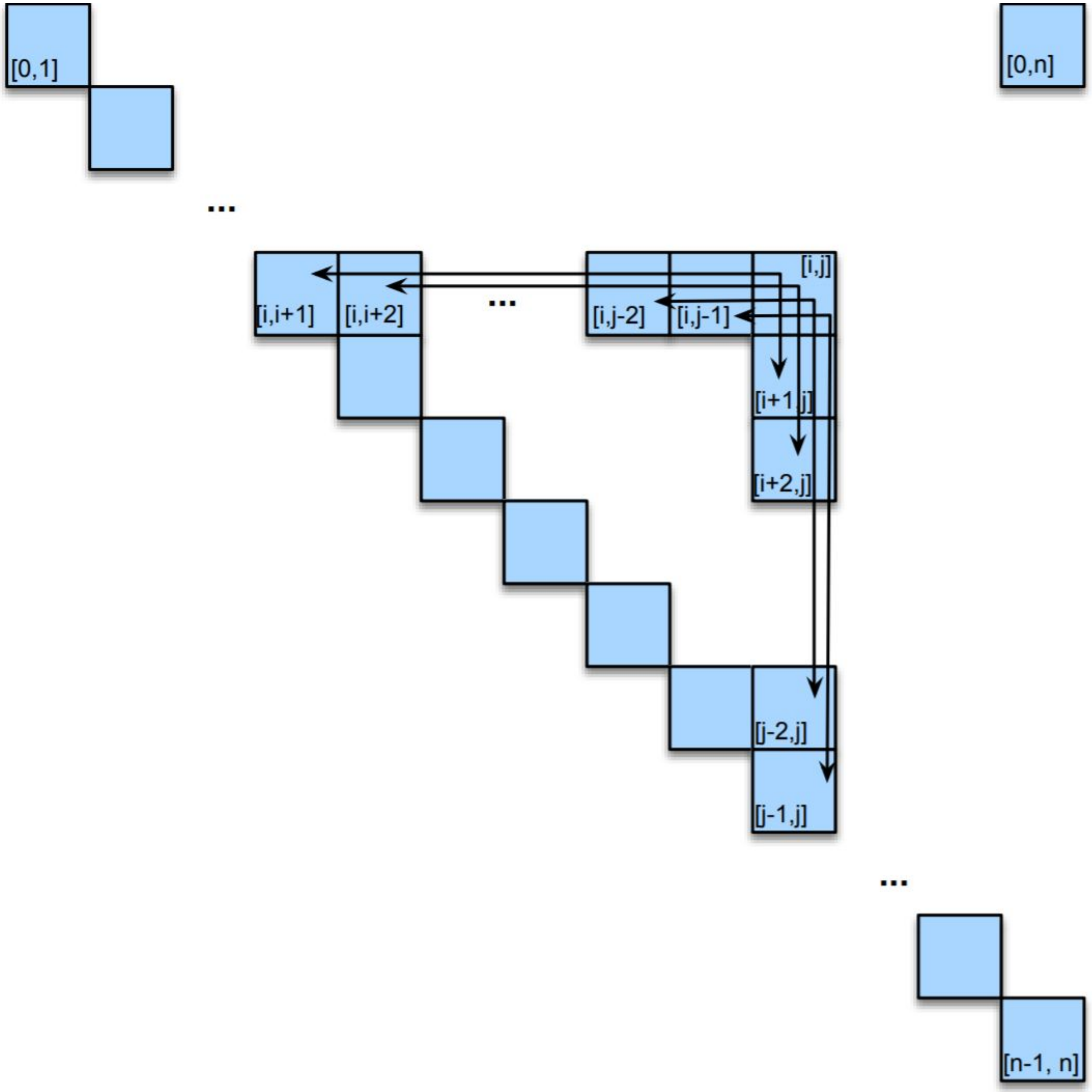
Implementation: preterminal rules

```
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
```





[0,1]
[0,n]
...
[i,j]
[i,i+1]
[i,i+2]
...
[i,j-2]
[i,j-1]
[i+1,j]
[i+2,j]
[j-2,j]
[j-1,j]
...
[n-1, n]

Unary rules

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

Unary rules

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

```
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!

Unary rules

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

```
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
```

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

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

Unary closure

- What if the grammar contained 2 rules:

$$A \to B$$
$$B \to C$$

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

$$A \to B \to C$$

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

$$\begin{array}{l} A \to B \\ B \to C \end{array} \quad \Rightarrow \quad \begin{array}{l} A \to B \\ B \to C \\ A \to C \end{array} \quad \begin{array}{l} A \to A \\ B \to B \\ C \to C \end{array}$$

Convenient for programming reasons in the PCFG case

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
```

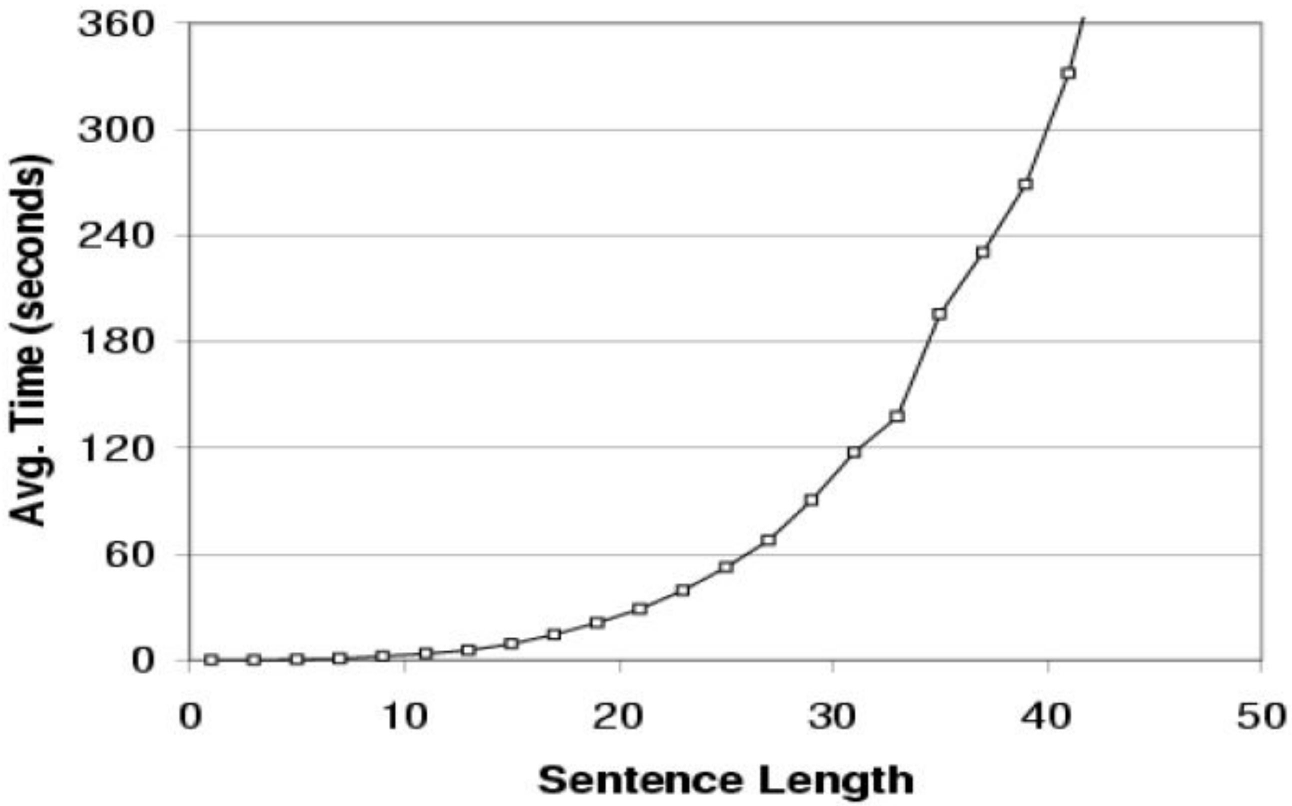
Algorithm analysis

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
```

$O(n^3|R|)$ where $|R|$ is is the number of rules in the grammar

Practical time complexity



$\sim n^{3.6}$

Probabilistic CKY

PCFGs

$S \to NP\ VP$ 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

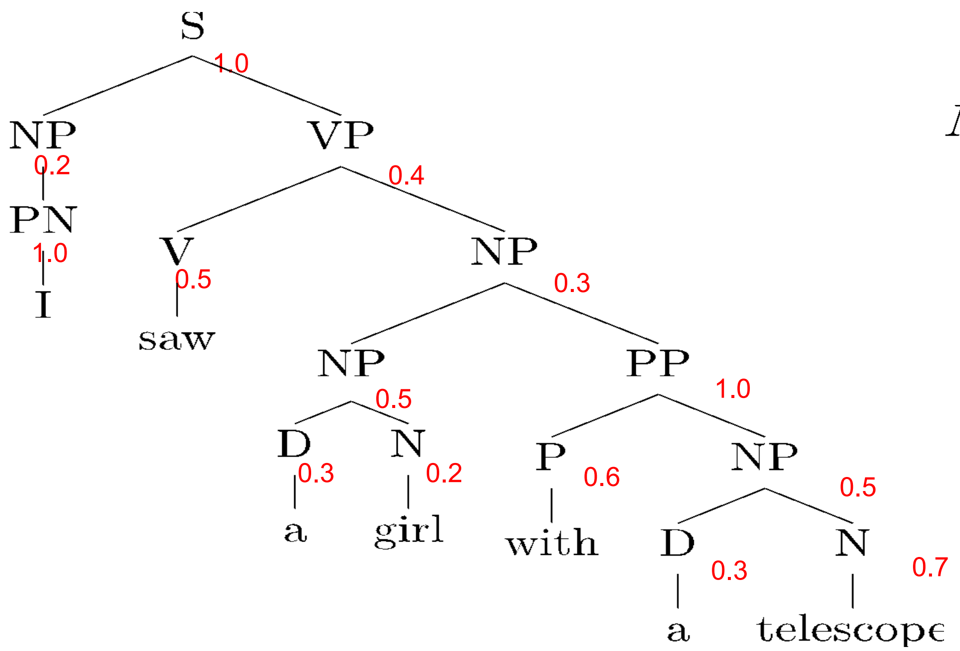
$V \to ate$ 0.5

$P \to with$ 0.6

$P \to in$ 0.4

$D \to a$ 0.3

$D \to the$ 0.7



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

CKY with PCFGs

- Chart is represented by a 3d array of floats

  `chart[min][max][label]`

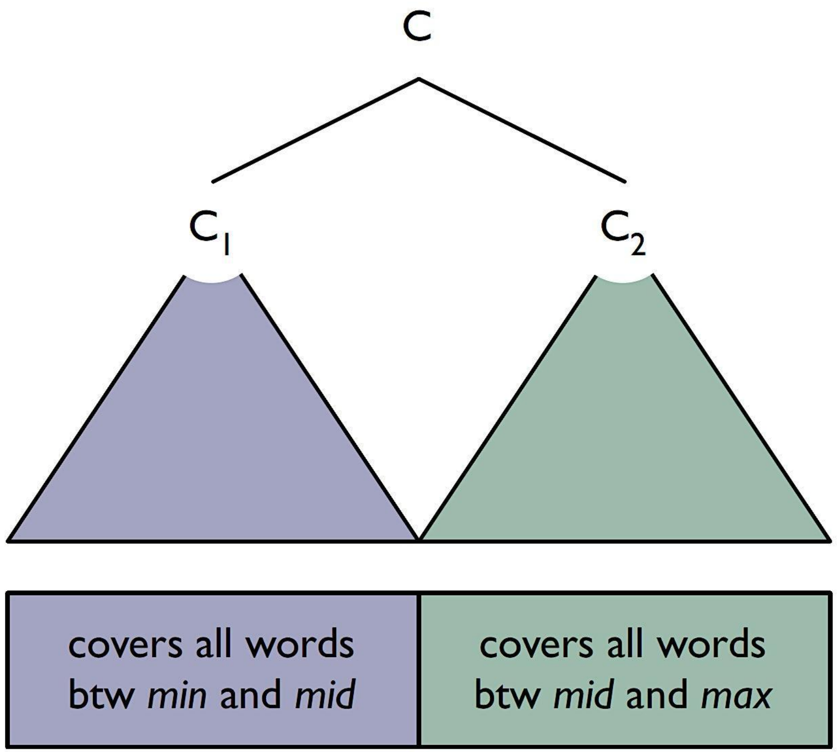
  - It stores probabilities for the most probable subtree with a given signature

- `chart[0][n][S]` will store the probability of the most probable full parse tree

-

Intuition

$C \rightarrow C_1 \ C_2$



For every C choose C_1, C_2 and mid such that

$$P(T_1) \times P(T_2) \times P(C \rightarrow C_1 C_2)$$

is maximal, where T_1 and T_2 are left and right subtrees.

Implementation: preterminal rules

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

Implementation: binary rules

```
for each max from 2 to n
  for each min from max - 2 down to 0
    for each syntactic category C
    double best = undefined
      for each binary rule C -> C₁ C₂
        for each mid from min + 1 to max - 1
          double t₁ = chart[min][mid][C₁]
          double t₂ = chart[mid][max][C₂]
          double candidate = t₁ * t₂ * p(C -> C₁ C₂)
          if candidate > best then
            best = candidate
    chart[min][max][C] = best
```

Unary rules

- Similarly to CFGs: after producing scores for signatures (c, i, j), try to improve the scores by applying unary rules (and rule chains)
  - If improved, update the scores

Unary (reflexive transitive) closure

$$\begin{array}{ll} A \rightarrow B & 0.1 \\ B \rightarrow C & 0.2 \\ \ldots & \end{array} \Rightarrow \begin{array}{lr} A \rightarrow B & 0.1 \\ B \rightarrow C & 0.2 \\ A \rightarrow C & 0.2 \times 0.1 \end{array} \quad \begin{array}{ll} A \rightarrow A & 1 \\ B \rightarrow B & 1 \\ C \rightarrow C & 1 \end{array}$$

$$\ldots$$

Note that this is not a PCFG anymore as the rules do not sum to 1 for each parent

Unary (reflexive transitive) closure

The fact that the rule is composite needs to be stored to recover the true tree

$$
\begin{array}{ll}
A \rightarrow B & 0.1 \\
B \rightarrow C & 0.2 \\
\ldots &
\end{array}
\quad \Rightarrow \quad
\begin{array}{lr}
A \rightarrow B & 0.1 \\
B \rightarrow C & 0.2 \\
A \rightarrow C & 0.2 \times 0.1
\end{array}
\quad
\begin{array}{ll}
A \rightarrow A & 1 \\
B \rightarrow B & 1 \\
C \rightarrow C & 1
\end{array}
$$

$$\ldots$$

Note that this is not a PCFG anymore as the rules do not sum to 1 for each parent

Unary (reflexive transitive) closure

The fact that the rule is composite needs to be stored to recover the true tree

$$\begin{array}{ll} A \to B & 0.1 \\ B \to C & 0.2 \\ \ldots & \end{array} \Rightarrow \begin{array}{ll} A \to B & 0.1 \\ B \to C & 0.2 \\ A \to C & 0.2 \times 0.1 \end{array} \quad \begin{array}{ll} A \to A & 1 \\ B \to B & 1 \\ C \to C & 1 \end{array} \quad \ldots$$

Note that this is not a PCFG anymore as the rules do not sum to 1 for each parent

$$\begin{array}{ll} A \to B & 0.1 \\ B \to C & 0.2 \\ A \to C & 1.e-5 \end{array} \Rightarrow \begin{array}{ll} A \to B & 0.1 \\ B \to C & 0.1 \\ A \to C & 0.02 \end{array} \quad \begin{array}{ll} A \to A & 1 \\ B \to B & 1 \\ C \to C & 1 \end{array}$$

What about loops, like: $A \to B \to A \to C$?

Recovery of the tree

- For each signature we store backpointers to the elements from which it was built (e.g., rule and, for binary rules, midpoint)
  - start recovering from [0, n, S]

- Be careful with unary rules
  - Basically you can assume that you always used an unary rule from the closure (but it could be the trivial one $C \rightarrow C$)

Speeding up the algorithm (approximate search)

Any ideas?

Speeding up the algorithm

- Basic pruning (roughly):
  - For every span (i,j) store only labels which have the probability at most N times smaller than the probability of the most probable label for this span
  - Check not all rules but only rules yielding subtree labels having non-zero probability
- Coarse-to-fine pruning
  - Parse with a smaller (simpler) grammar, and precompute (posterior) probabilities for each spans, and use only the ones with non-negligible probability from the previous grammar

Parsing evaluation

- Intrinsic evaluation:
  - Automatic: evaluate against annotation provided by human experts (gold standard) according to some predefined measure
  - Manual: … according to human judgment

- Extrinsic evaluation: score syntactic representation by comparing how well a system using this representation performs on some task
  - E.g., use syntactic representation as input for a semantic analyzer and compare results of the analyzer using syntax predicted by different parsers.

Standard evaluation setting in parsing

- Automatic intrinsic evaluation is used: parsers are evaluated against gold standard by provided by linguists
  - There is a standard split into the parts:
    - training set: used for estimation of model parameters
    - development set: used for tuning the model (initial experiments)
    - test set: final experiments to compare against previous work

Automatic evaluation of constituent parsers

- Exact match: percentage of trees predicted correctly
- Bracket score: scores how well individual phrases (and their boundaries) are identified
- Crossing brackets: percentage of phrases boundaries crossing

The most standard measure; we will focus on it

Brackets scores

Subtree signatures for CKY

- The most standard score is bracket score
- It regards a tree as a collection of brackets: $[min, max, C]$
- The set of brackets predicted by a parser is compared against the set of brackets in the tree annotated by a linguist
- Precision, recall and F1 are used as scores

Preview: F1 bracket score