Computational Psycholinguistics, Lecture 2: Core ideas and algorithms

Roger Levy

UC San Diego
Department of Linguistics

Linguistics/CSE 256: Statistical NLP
17 February 2009

What is “sentence processing”?

- *Sentence processing* is the study of how humans comprehend and produce sentences (and words within sentences, and sequences of sentences, etc.) in real time.

Theoretical Desiderata

Realistic models of human sentence processing must account for:

- Robustness to arbitrary input
- Accurate disambiguation
- Inference on basis of incomplete input (Tanenhaus et al., 1995; Altmann and Kamide, 1999; Kaiser and Trueswell, 2004)
- Processing difficulty is *differential* and *localized*

Robustness

Real linguistic input is not always totally well-formed. . .

> *I think when she finally came to the realization that, you know, no, I can not, I can not take care of myself.*
> *. . .*
> *I mean, for somebody who is, you know, for most of their life has, has, uh, not just merely had a farm but had ten children had a farm, ran everything because her husband was away in the coal mines.*
> *And, you know, facing that situation, it's, it's quite a dilemma.*

Robustness

Real linguistic input is not always totally well-formed. . .

I think when she finally came to the realization that, you know, no, I can not, I can not take care of myself.
. . .
I mean, for somebody who is, you know, for most of their life has, has, uh, not just merely had a farm but had ten children had a farm, ran everything because her husband was away in the coal mines.
And, you know, facing that situation, it's, it's quite a dilemma.
(The woman is facing being put in a resting home.)

Robustness

Real linguistic input is not always totally well-formed. . .

> *I think when she finally came to the realization that, you know, no, I can not, I can not take care of myself.*
> *. . .*
> *I mean, for somebody who is, you know, for most of their life has, has, uh, not just merely had a farm but had ten children had a farm, ran everything because her husband was away in the coal mines.*
> *And, you know, facing that situation, it's, it's quite a dilemma.*
> *(The woman is facing being put in a resting home.)*

. . . but usually we come to understand it pretty well anyway.

Accurate disambiguation

Most sentences are ambiguous in ways we do not even notice:

Mary forgot the pitcher. . .

Accurate disambiguation

Most sentences are ambiguous in ways we do not even notice:

Mary forgot the pitcher. . .

Accurate disambiguation

Most sentences are ambiguous in ways we do not even notice:

Mary forgot the pitcher of water sitting near the stove.

Accurate disambiguation

Most sentences are ambiguous in ways we do not even notice:

Mary forgot the pitcher of water sitting near the stove.

Accurate disambiguation

Most sentences are ambiguous in ways we do not even notice:

Mary forgot the pitcher of water sitting near the stove.

That’s probably not what you were thinking of...

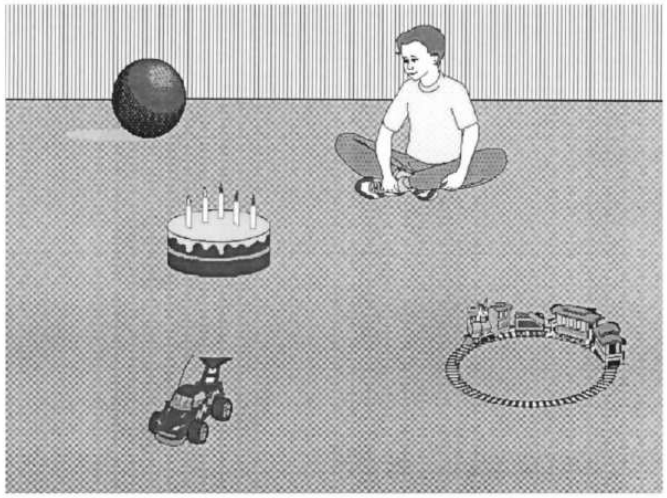
Inference on the basis of incomplete input

Comprehenders do not wait until the whole sentence has been heard to make inferences about what it means or will wind up meaning:

(Altmann and Kamide, 1999)

Inference on the basis of incomplete input

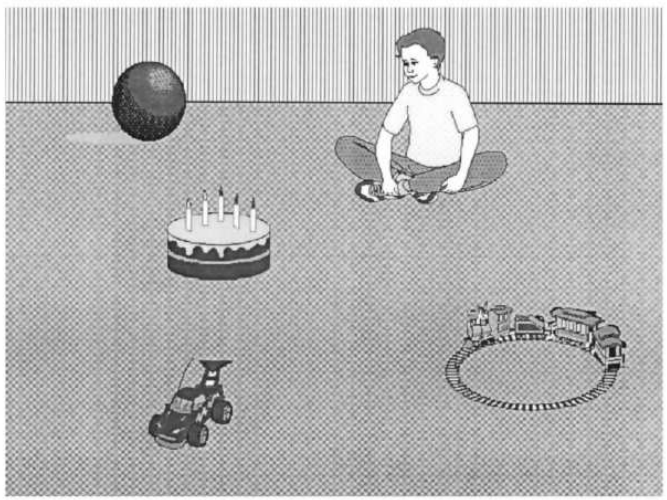
Comprehenders do not wait until the whole sentence has been heard to make inferences about what it means or will wind up meaning:



(Altmann and Kamide, 1999)

Inference on the basis of incomplete input

Comprehenders do not wait until the whole sentence has been heard to make inferences about what it means or will wind up meaning:

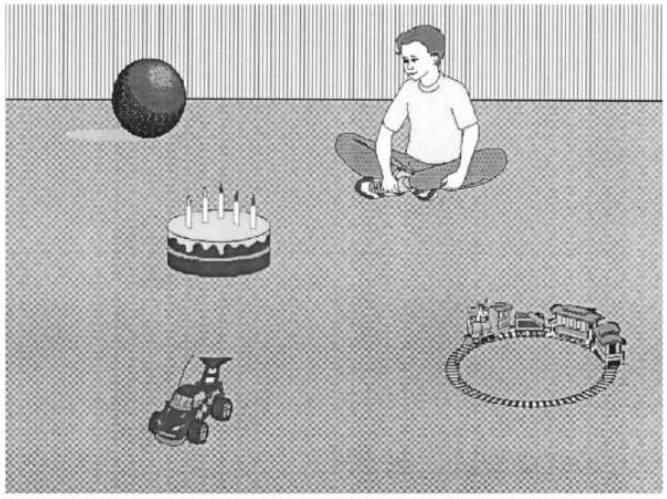


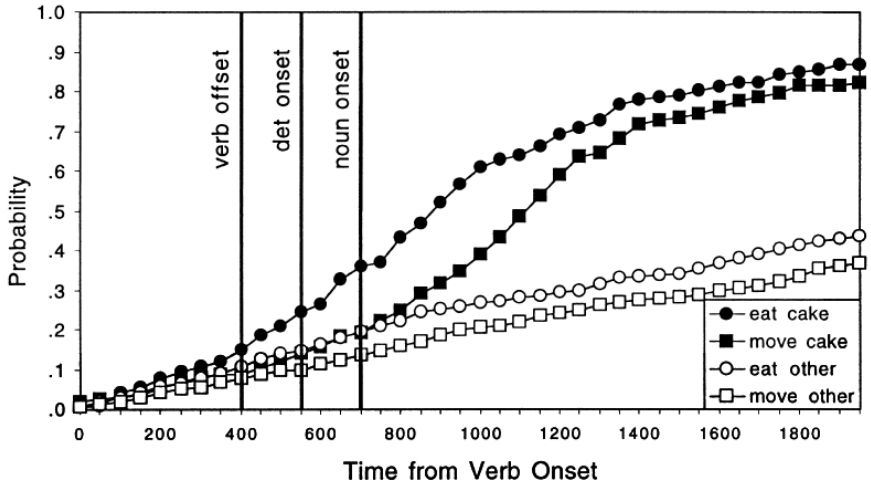
"The boy will ***eat*****/*****move*** the cake. . . "

(Altmann and Kamide, 1999)

Inference on the basis of incomplete input

Comprehenders do not wait until the whole sentence has been heard to make inferences about what it means or will wind up meaning:



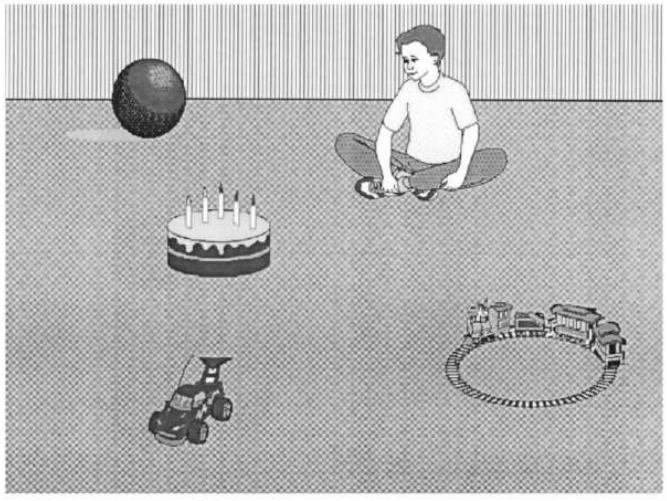


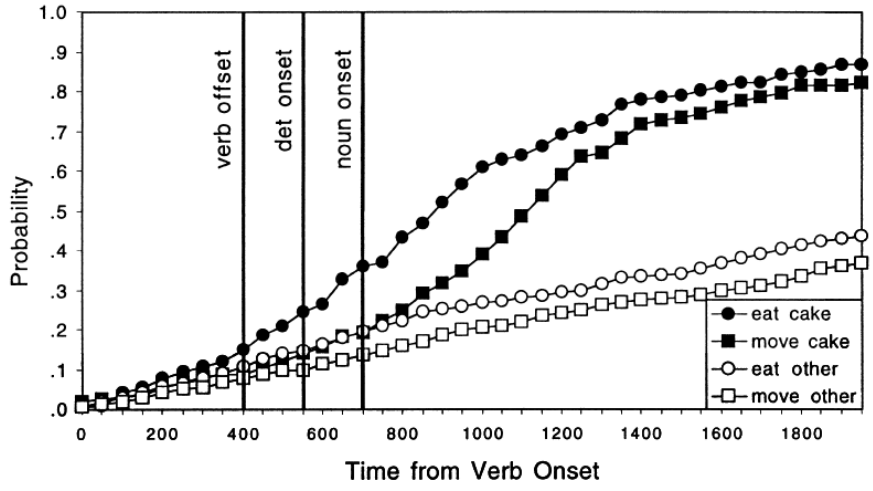
"The boy will ***eat***/***move*** the cake. . ."

(Altmann and Kamide, 1999)

Inference on the basis of incomplete input

Comprehenders do not wait until the whole sentence has been heard to make inferences about what it means or will wind up meaning:





"The boy will ***eat/move*** the cake. . ."

That is, comprehension is *incremental*

(Altmann and Kamide, 1999)

Processing difficulty is differential

Using multiple relative clauses in a sentence can make processing difficult:

> *This is the malt that the rat that the cat that the dog worried killed ate.*

Processing difficulty is differential

Using multiple relative clauses in a sentence can make processing difficult:

> *This is the malt that the rat that the cat that the dog worried killed ate.*

It's not the meaning of the sentence, or the use of relative clauses, that makes it hard:

Processing difficulty is differential

Using multiple relative clauses in a sentence can make processing difficult:

> *This is the malt that the rat that the cat that the dog worried killed ate.*

It's not the meaning of the sentence, or the use of relative clauses, that makes it hard:

> *This is the malt that was eaten by the rat that was killed by the cat that was worried by the dog.*

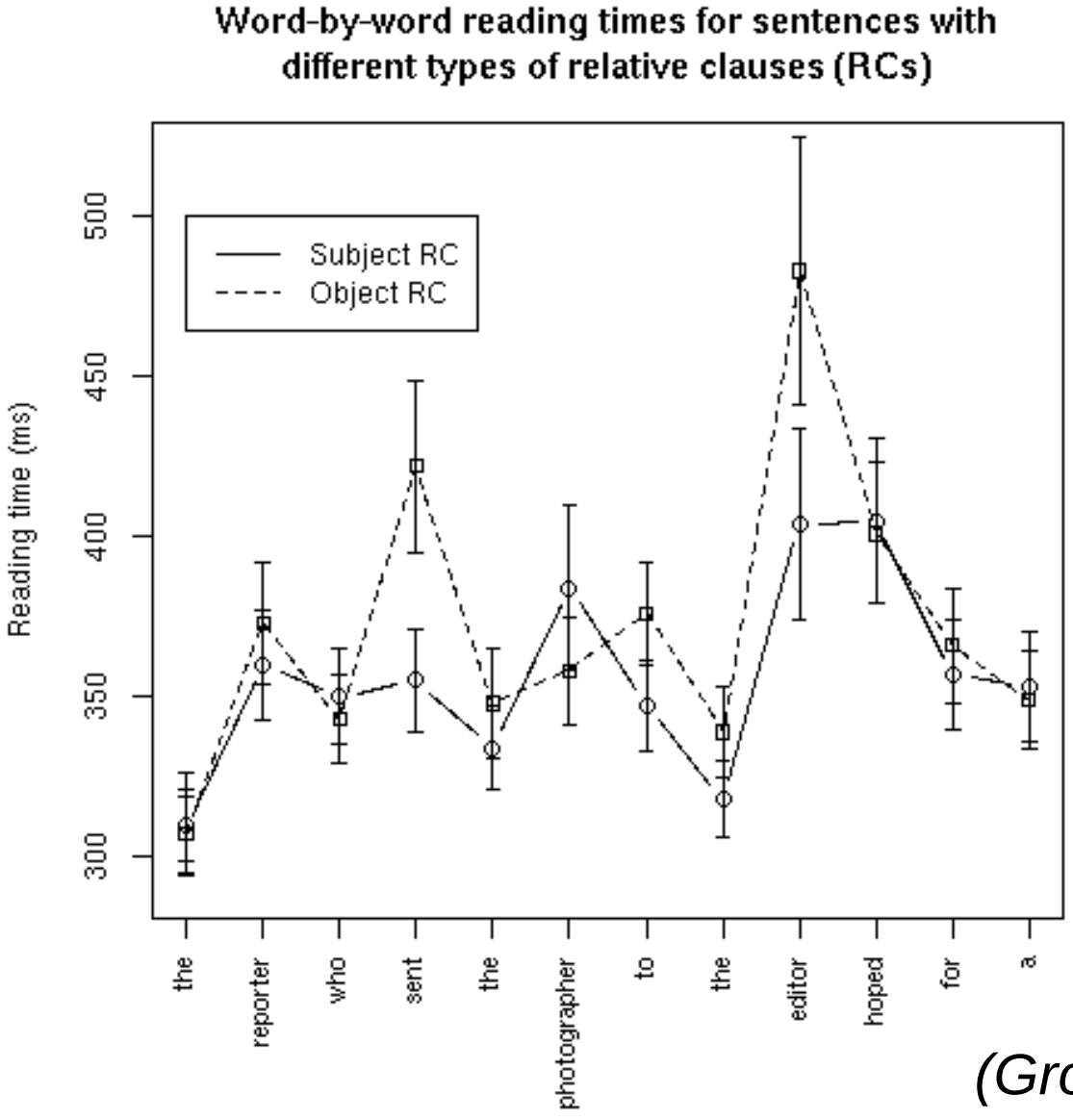
Processing difficulty is localized

[self-paced reading demo, Example1]

(Grodner and Gibson, 2005)

Processing difficulty is localized

[self-paced reading demo, Example1]



(Grodner and Gibson, 2005)

Goal of modeling sentence processing

- We need to account for these properties (robustness, disambiguation, inference from incomplete input, and differential difficulty) together

Goal of modeling sentence processing

- We need to account for these properties (robustness, disambiguation, inference from incomplete input, and differential difficulty) together
- Probabilistic models are an interesting way to do this

Goal of modeling sentence processing

- We need to account for these properties (robustness, disambiguation, inference from incomplete input, and differential difficulty) together
- Probabilistic models are an interesting way to do this
- You are probably already convinced about robustness, disambiguation, and inference!

Goal of modeling sentence processing

- We need to account for these properties (robustness, disambiguation, inference from incomplete input, and differential difficulty) together
- Probabilistic models are an interesting way to do this
- You are probably already convinced about robustness, disambiguation, and inference!
- I'll talk largely about some interesting tie-ins with differential difficulty

Try to guess the next word in the sentence

Try to guess the next word in the sentence

My brother came inside to. . .

Try to guess the next word in the sentence

My brother came inside to. . . chat? get warm? talk? eat? rest?

Try to guess the next word in the sentence

My brother came inside to. . . chat? get warm? talk? eat? rest?
The children went outside to. . .

Try to guess the next word in the sentence

My brother came inside to. . . chat? get warm? talk? eat? rest?

The children went outside to. . . play

Try to guess the next word in the sentence

My brother came inside to. . . chat? get warm? talk? eat? rest?
The children went outside to. . . play

- Empirically, it's been shown that more highly predictable words are read more quickly (Ehrlich and Rayner, 1981)

Try to guess the next word in the sentence

My brother came inside to. . . chat? get warm? talk? eat? rest?
The children went outside to. . . play

- Empirically, it's been shown that more highly predictable words are read more quickly (Ehrlich and Rayner, 1981)
- Why would this be the case?

Surprisal as a possible metric for processing difficulty

An event's surprisal is simply its negative log conditional probability

$$\log \frac{1}{P(x|\text{Context})}$$

Intuitively, this is a measure of the amount of information contained in the event

Four proposals for surprisal as a measure of processing difficulty/time:

- Surprisal of a word as primitive measure of processing (Mandelbrot, 1953; Attneave, 1959; Hale, 2001)
- Kullback-Leibler divergence (*relative entropy*) as size of update that the word induces for distribution over interpretations of input (Levy, 2005, 2008)
  - independently proposed as a measure of surprise in visual scene perception (Itti and Baldi, 2005)
- Surprisal as optimal perceptual discrimination (Norris, 2006)
- Surprisal as an optimal solution to the speed/resource tradeoff in language comprehension (Smith, 2006)

Probabilistic grammars for estimating surprisal

- Comprehenders' expectations about upcoming words should refect structural distributional regularities of the language
- Hence, probabilistic grammars are a good candidate
- We'll use *probabilistic context-free grammars* (PCFGs) as a model of language users' grammatical knowledge

Context-free Grammars

A context-free grammar (CFG) consists of a tuple (N, V, S, R) such that:

- N is a finite set of non-terminal symbols;
- V is a finite set of terminal symbols;
- S is the start symbol;
- R is a finite set of rules of the form $X \to \alpha$ where $X \in N$ and α is a sequence of symbols drawn from $N \cup V$.

A CFG *derivation* is the recursive expansion of non-terminal symbols in a string by rules in R, starting with S, and a *derivation tree* T is the history of those rule applications.

Context-free Grammars: an example

Let our grammar (the rule-set R) be

S	$\rightarrow$NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$Det N	N	$\rightarrow$ dog
NP	$\rightarrow$NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$P NP	P	$\rightarrow$ near
VP	$\rightarrow$V	V	$\rightarrow$ growled

The nonterminal set N is $\{S, NP, VP, Det, N, P, V\}$, the terminal set V is $\{the, dog, cat, near, growled\}$, and our start symbol S is S.

Context-free Grammars: an example II

S $\rightarrow$ NP VP	Det $\rightarrow$ the
NP $\rightarrow$ Det N	N $\rightarrow$ dog
NP $\rightarrow$ NP PP	N $\rightarrow$ cat
PP $\rightarrow$ P NP	P $\rightarrow$ near
VP $\rightarrow$ V	V $\rightarrow$ growled

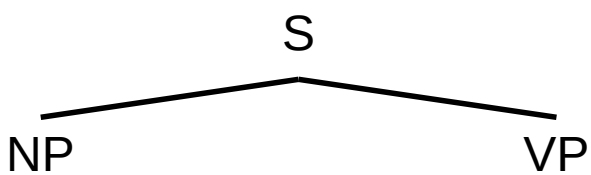
Here is a *derivation* and the resulting *derivation tree*:

S

Context-free Grammars: an example II

S	$\rightarrow$NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$Det N	N	$\rightarrow$ dog
NP	$\rightarrow$NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$P NP	P	$\rightarrow$ near
VP	$\rightarrow$V	V	$\rightarrow$ growled

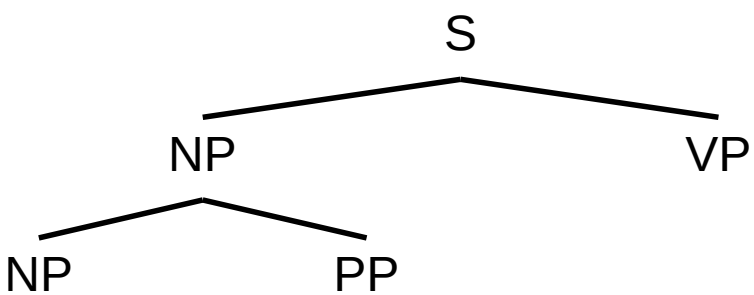
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S	$\rightarrow$NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$Det N	N	$\rightarrow$ dog
NP	$\rightarrow$NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$P NP	P	$\rightarrow$ near
VP	$\rightarrow$V	V	$\rightarrow$ growled

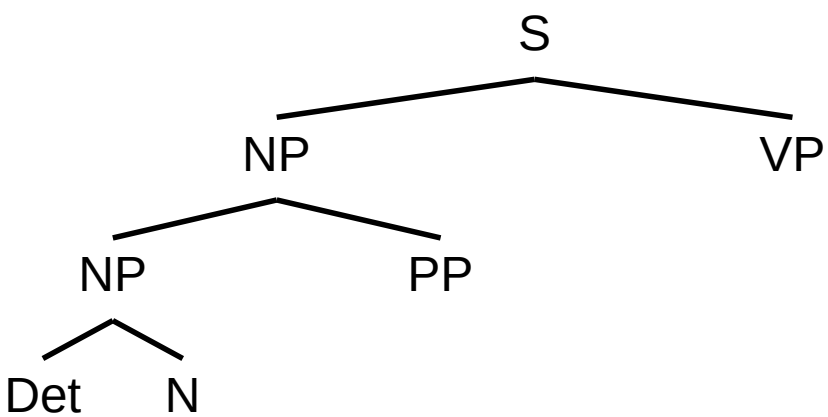
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S $\rightarrow$ NP VP	Det $\rightarrow$ the
NP $\rightarrow$ Det N	N $\rightarrow$ dog
NP $\rightarrow$ NP PP	N $\rightarrow$ cat
PP $\rightarrow$ P NP	P $\rightarrow$ near
VP $\rightarrow$ V	V $\rightarrow$ growled

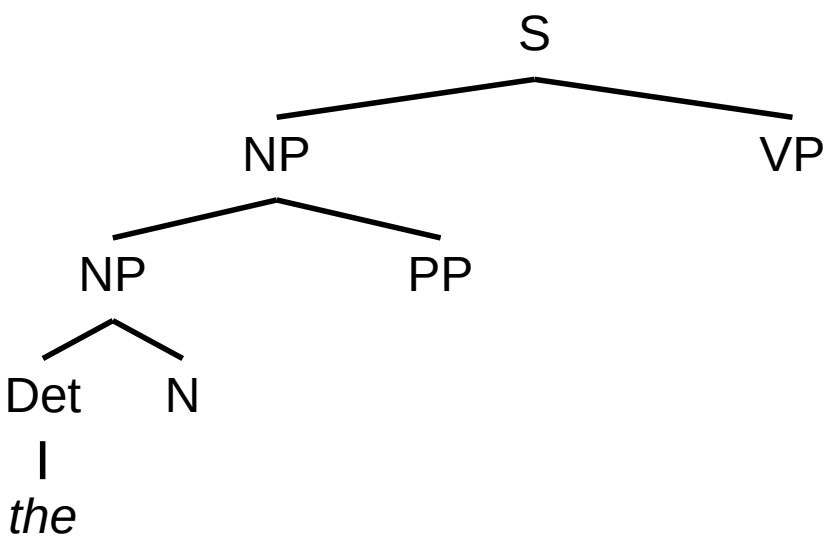
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S	→ NP VP	Det	→ the
NP	→ Det N	N	→ dog
NP	→ NP PP	N	→ cat
PP	→ P NP	P	→ near
VP	→ V	V	→ growled

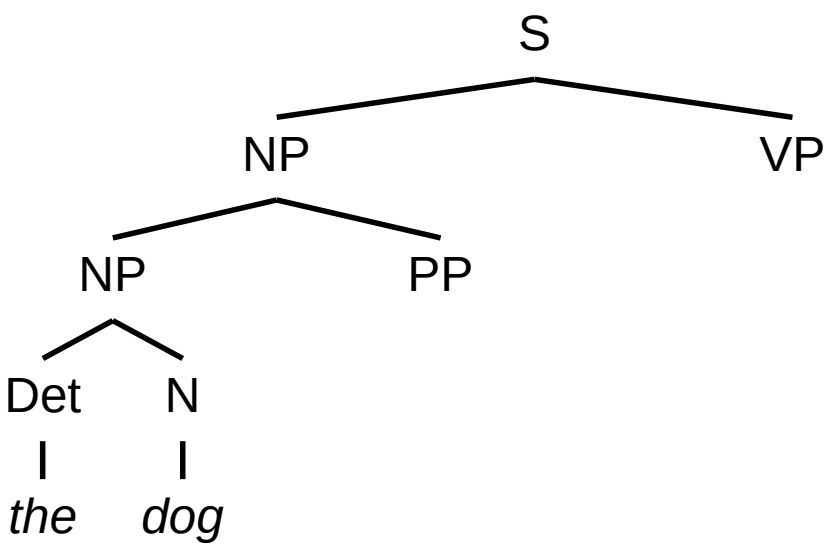
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S	$\rightarrow$ NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$ Det N	N	$\rightarrow$ dog
NP	$\rightarrow$ NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$ P NP	P	$\rightarrow$ near
VP	$\rightarrow$ V	V	$\rightarrow$ growled

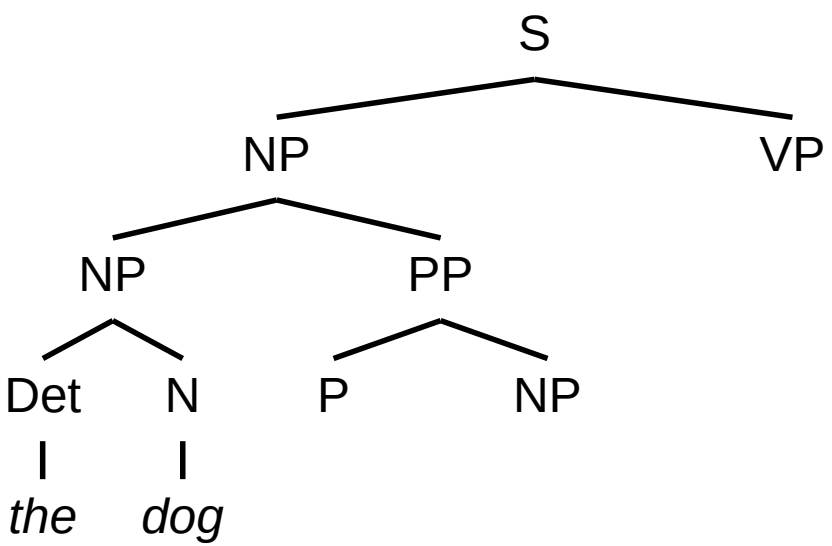
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S	$\rightarrow$ NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$ Det N	N	$\rightarrow$ dog
NP	$\rightarrow$ NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$ P NP	P	$\rightarrow$ near
VP	$\rightarrow$ V	V	$\rightarrow$ growled

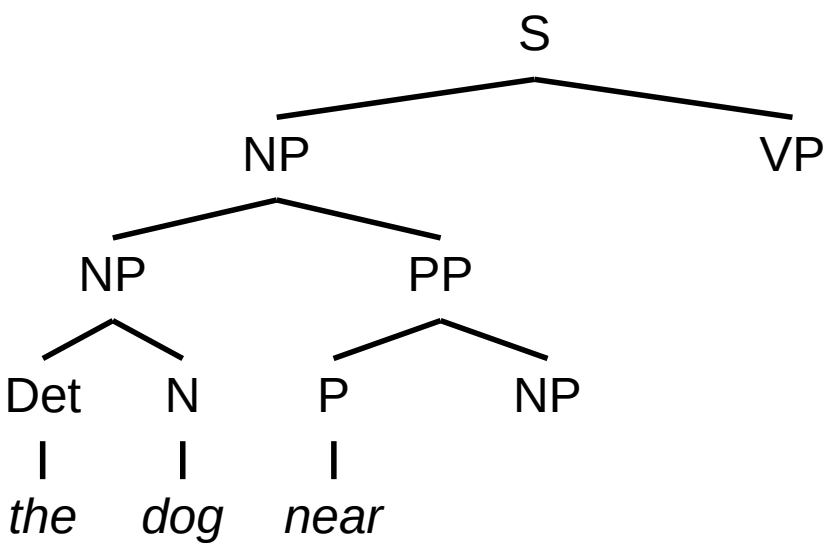
Here is a *derivation* and the resulting *derivation tree*:



Context-free Grammars: an example II

S	$\rightarrow$ NP VP	Det	$\rightarrow$ the
NP	$\rightarrow$ Det N	N	$\rightarrow$ dog
NP	$\rightarrow$ NP PP	N	$\rightarrow$ cat
PP	$\rightarrow$ P NP	P	$\rightarrow$ near
VP	$\rightarrow$ V	V	$\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:

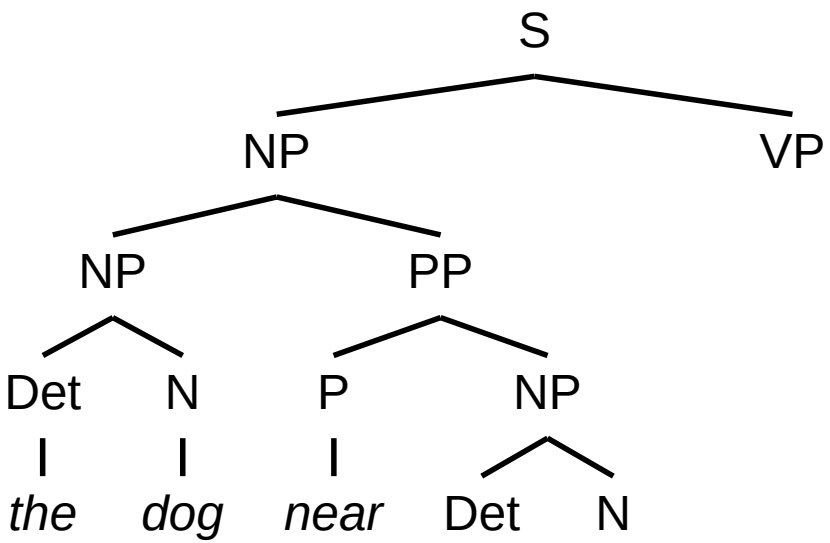


Context-free Grammars: an example II

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
VP $\rightarrow$ V

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:

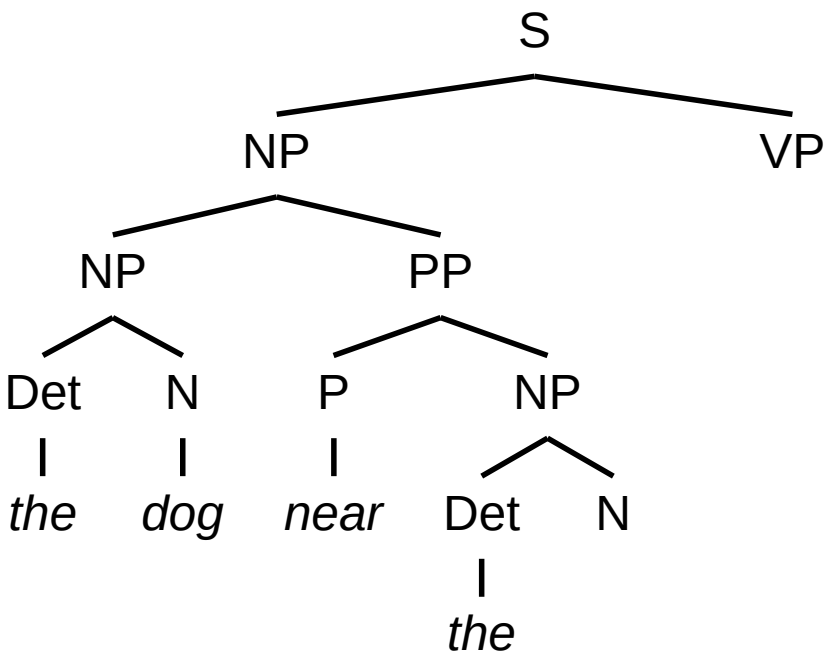


Context-free Grammars: an example II

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
VP $\rightarrow$ V

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:

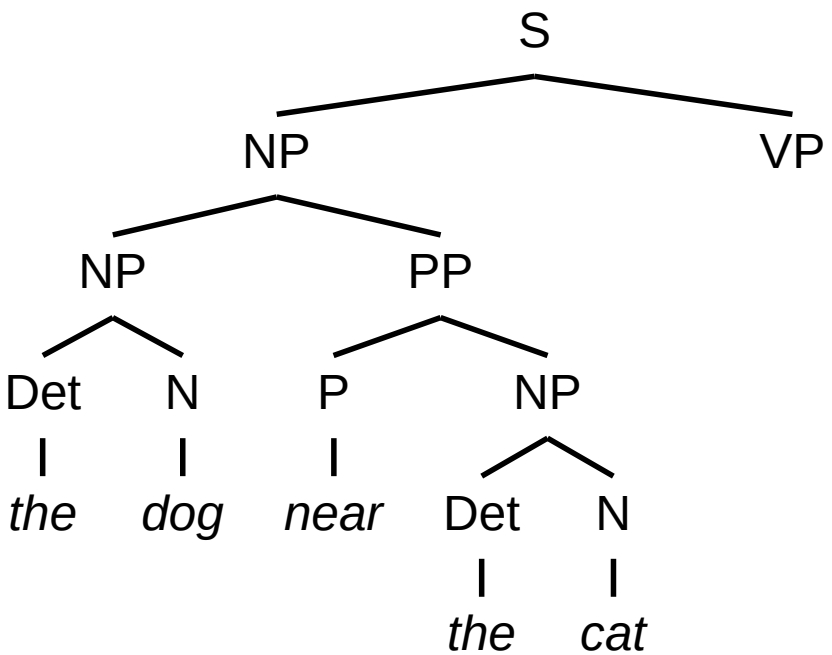


Context-free Grammars: an example II

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
VP $\rightarrow$ V

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:

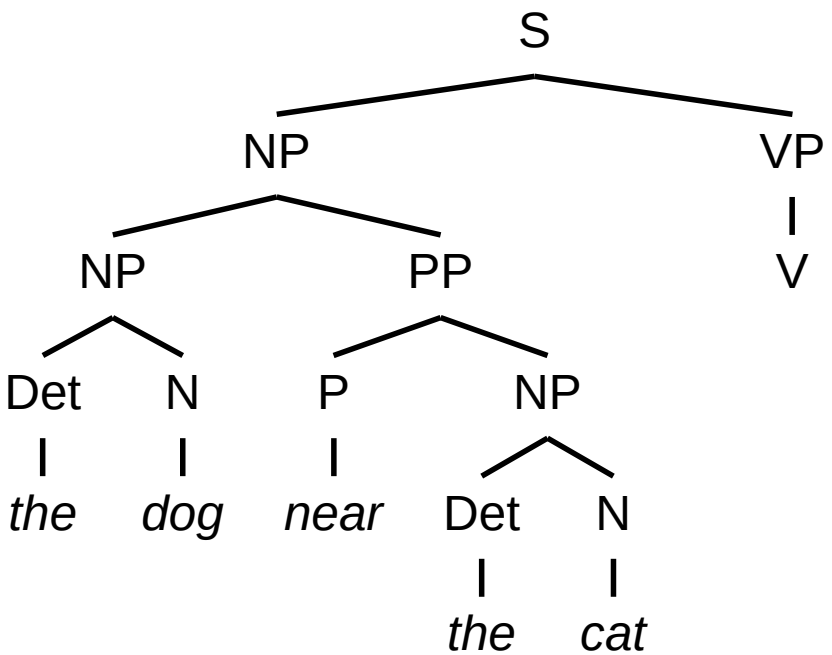


Context-free Grammars: an example II

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
VP $\rightarrow$ V

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:

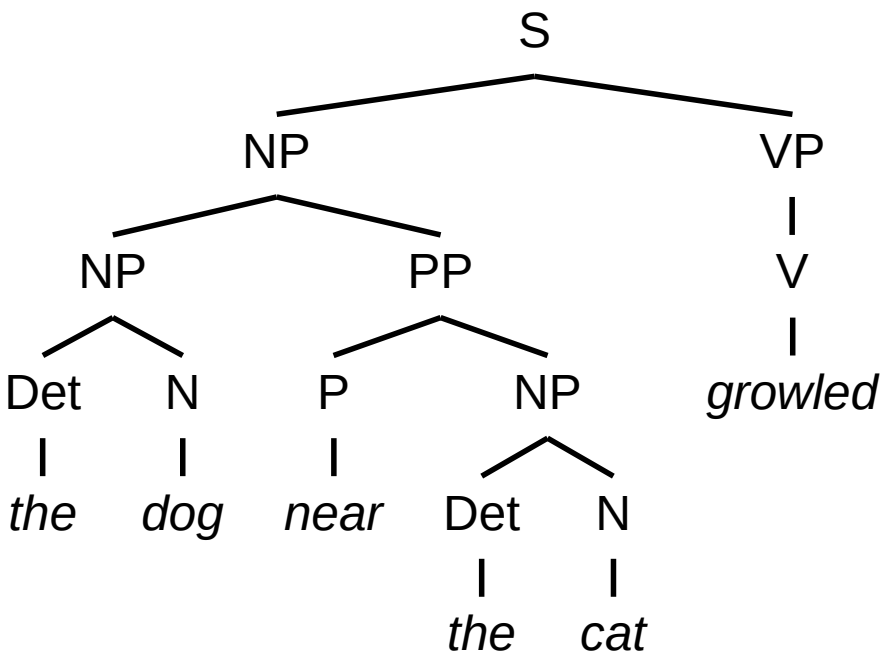


Context-free Grammars: an example II

S $\rightarrow$ NP VP
NP $\rightarrow$ Det N
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
VP $\rightarrow$ V

Det $\rightarrow$ the
N $\rightarrow$ dog
N $\rightarrow$ cat
P $\rightarrow$ near
V $\rightarrow$ growled

Here is a *derivation* and the resulting *derivation tree*:



Probabilistic Context-Free Grammars

A *probabilistic* context-free grammar (PCFG) consists of a tuple (N, V, S, R, P) such that:

- N is a finite set of non-terminal symbols;
- V is a finite set of terminal symbols;
- S is the start symbol;
- R is a finite set of rules of the form $X \rightarrow \alpha$ where $X \in N$ and α is a sequence of symbols drawn from $N \cup V$;
- P is a mapping from R into probabilities, such that for each $X \in N$,

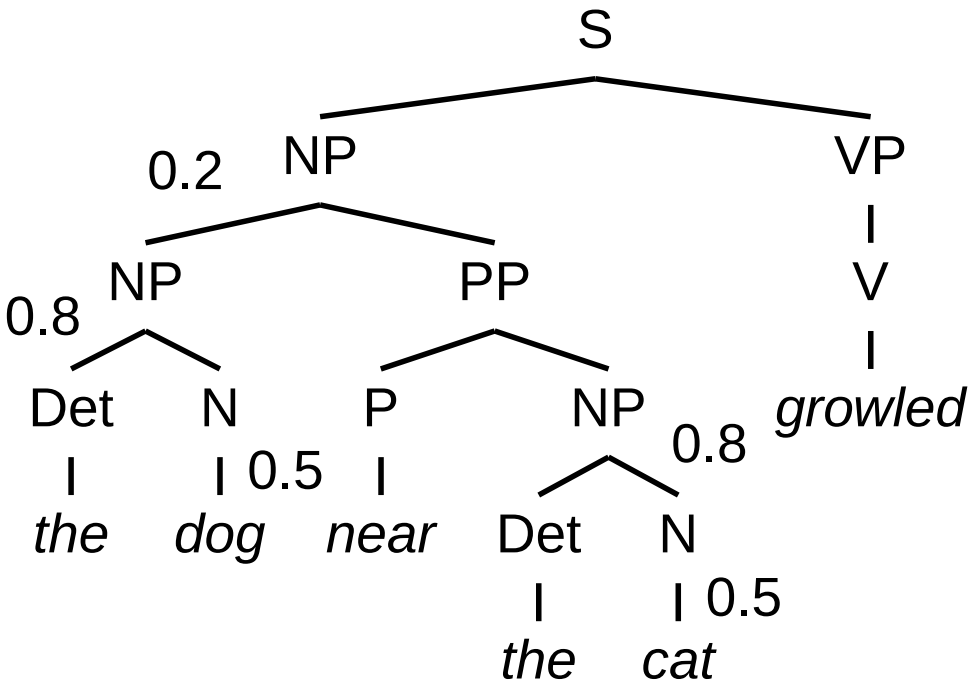
$$\sum_{[X \rightarrow \alpha] \in R} P(X \rightarrow \alpha) = 1$$

PCFG *derivations* and *derivation trees* are just like for CFGs. The probability $P(T)$ of a derivation tree is simply the product of the probabilities of each rule application.

Example PCFG

1	S	$\rightarrow$NP VP
0.8	NP	$\rightarrow$Det N
0.2	NP	$\rightarrow$NP PP
1	PP	$\rightarrow$P NP
1	VP	$\rightarrow$V

1	Det	$\rightarrow$ the
0.5	N	$\rightarrow$ dog
0.5	N	$\rightarrow$ cat
1	P	$\rightarrow$ near
1	V	$\rightarrow$ growled



$$P(T) = 1 \times 0.2 \times 0.8 \times 1 \times 0.5 \times 0.8 \times 1 \times \times 0.8 \times 1 \times 0.5 \times 1 \times 1$$
$$= 0.032$$

PCFG review (2)

- We just learned how to calculate the *probability of a tree*
- The *probability of a string* $w_{1\ldots n}$ is the sum of the probabilities of all trees whose yield **is** $w_{1\ldots n}$
- The *probability of a string prefix* $w_{1\ldots i}$ is the sum of the probabilities of all trees whose yield **begins with** $w_{1\ldots i}$
- If we had the probabilities of two string prefixes $w_{1\ldots i-1}$ and $w_{1\ldots i}$, we could calculate the conditional probability $P(w_i|w_{1\ldots i-1})$ as their ratio:

$$P(w_i|w_{1\ldots i-1}) = \frac{P(w_{1\ldots i})}{P(w_{1\ldots i-1})}$$

Inference over infinite tree sets

Consider the following noun-phrase grammar:

$\frac{2}{3}$	NP $\rightarrow$ Det N	1	Det $\rightarrow$ the
$\frac{1}{3}$	NP $\rightarrow$ NP PP	$\frac{2}{3}$	N $\rightarrow$ dog
1	PP $\rightarrow$ P NP	$\frac{1}{3}$	N $\rightarrow$ cat
		1	P $\rightarrow$ near

Inference over infinite tree sets

Consider the following noun-phrase grammar:

			1	Det $\rightarrow$ the
$\frac{2}{3}$	NP $\rightarrow$ Det N		$\frac{2}{3}$	N $\rightarrow$ dog
$\frac{1}{3}$	NP $\rightarrow$ NP PP		$\frac{1}{3}$	N $\rightarrow$ cat
1	PP $\rightarrow$ P NP		1	P $\rightarrow$ near

Question: given a sentence starting with

the. . .

what is the probability that the next word is *dog*?

Inference over infinite tree sets

Consider the following noun-phrase grammar:

			1	Det $\rightarrow$ the
$\frac{2}{3}$	NP $\rightarrow$ Det N		$\frac{2}{3}$	N $\rightarrow$ dog
$\frac{1}{3}$	NP $\rightarrow$ NP PP		$\frac{1}{3}$	N $\rightarrow$ cat
1	PP $\rightarrow$ P NP		1	P $\rightarrow$ near

Question: given a sentence starting with

the. . .

what is the probability that the next word is *dog*?
Intuitively, the answers to this question should be

$$P(\text{dog}|\text{the}) = \frac{2}{3}$$

Inference over infinite tree sets

Consider the following noun-phrase grammar:

			1	Det $\rightarrow$ the
$\frac{2}{3}$	NP $\rightarrow$ Det N		$\frac{2}{3}$	N $\rightarrow$ dog
$\frac{1}{3}$	NP $\rightarrow$ NP PP		$\frac{1}{3}$	N $\rightarrow$ cat
1	PP $\rightarrow$ P NP		1	P $\rightarrow$ near

Question: given a sentence starting with

the. . .

what is the probability that the next word is *dog*?
Intuitively, the answers to this question should be

$$P(\text{dog}|\text{the}) = \frac{2}{3}$$

because the second word HAS to be either *dog* or *cat*.

Inference over infinite tree sets (2)

$$\begin{array}{lll} \frac{2}{3} & \text{NP} \rightarrow \text{Det N} \\ \frac{1}{3} & \text{NP} \rightarrow \text{NP PP} \\ 1 & \text{PP} \rightarrow \text{P NP} \end{array} \qquad \begin{array}{lll} 1 & \text{Det} \rightarrow \text{the} \\ \frac{2}{3} & \text{N} \rightarrow \text{dog} \\ \frac{1}{3} & \text{N} \rightarrow \text{cat} \\ 1 & \text{P} \rightarrow \text{near} \end{array}$$

- We “should” just enumerate the trees that cover *the dog . . .* ,

Inference over infinite tree sets (2)

$\frac{2}{3}$ NP $\rightarrow$ Det N
$\frac{1}{3}$ NP $\rightarrow$ NP PP
1 PP $\rightarrow$ P NP

1 Det $\rightarrow$ the
$\frac{2}{3}$ N $\rightarrow$ dog
$\frac{1}{3}$ N $\rightarrow$ cat
1 P $\rightarrow$ near

- We “should” just enumerate the trees that cover *the dog . . .* , and divide their total probability by that of *the . . .*

Inference over infinite tree sets (2)

$$\begin{array}{lll} \frac{2}{3} & \text{NP} \to \text{Det N} \\ \frac{1}{3} & \text{NP} \to \text{NP PP} \\ 1 & \text{PP} \to \text{P NP} \end{array} \qquad \begin{array}{lll} 1 & \text{Det} \to \text{the} \\ \frac{2}{3} & \text{N} \to \text{dog} \\ \frac{1}{3} & \text{N} \to \text{cat} \\ 1 & \text{P} \to \text{near} \end{array}$$

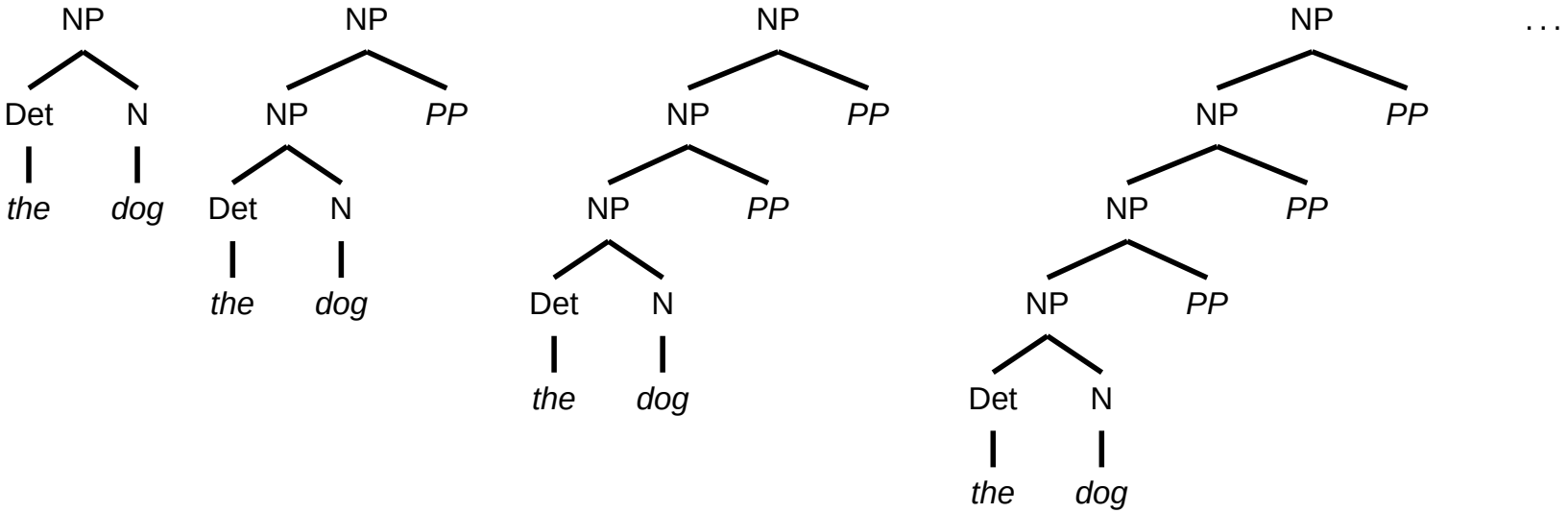
- We “should” just enumerate the trees that cover *the dog . . .* , and divide their total probability by that of *the . . .*
- . . . but there are infinitely many trees.

Inference over infinite tree sets (2)

$\frac{2}{3}$ NP $\rightarrow$ Det N
$\frac{1}{3}$ NP $\rightarrow$ NP PP
1 PP $\rightarrow$ P NP

1 Det $\rightarrow$ the
$\frac{2}{3}$ N $\rightarrow$ dog
$\frac{1}{3}$ N $\rightarrow$ cat
1 P $\rightarrow$ near

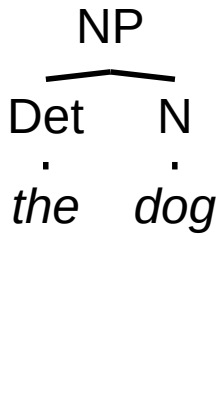
- We "should" just enumerate the trees that cover *the dog . . .* , and divide their total probability by that of *the . . .*
- . . . but there are infinitely many trees.



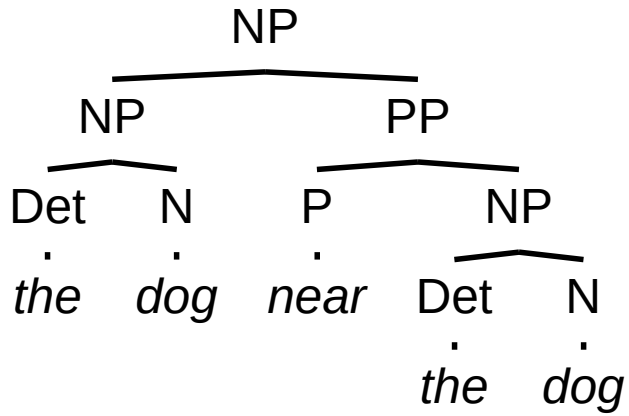
$\frac{2}{3}$	NP $\rightarrow$ Det N		1	Det $\rightarrow$ the
$\frac{1}{3}$	NP $\rightarrow$ NP PP		$\frac{2}{3}$	N $\rightarrow$ dog
1	PP $\rightarrow$ P NP		$\frac{1}{3}$	N $\rightarrow$ cat
			1	P $\rightarrow$ near

Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree.

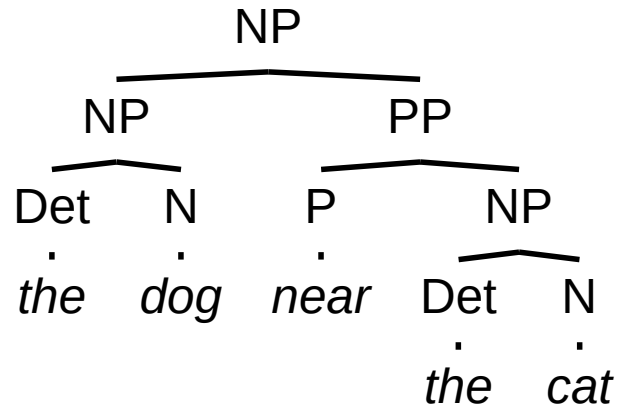
It has a corresponding marginal probability in the PCFG.



$\frac{4}{9}$



$\frac{8}{81}$

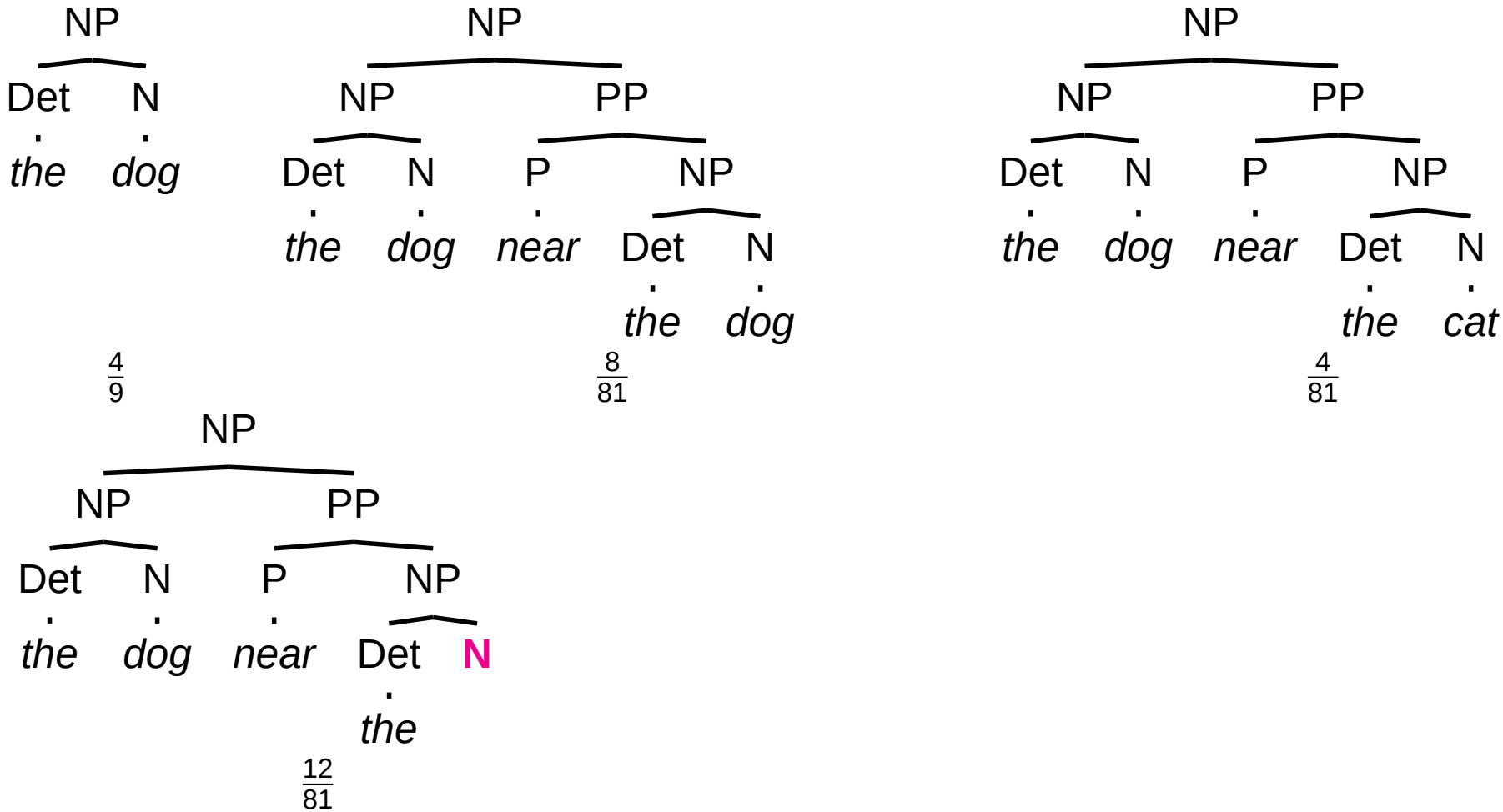


$\frac{4}{81}$

$\frac{2}{3}$	NP → Det N		1	Det → the	
$\frac{1}{3}$	NP → NP PP		$\frac{2}{3}$	N → dog	
1	PP → P NP		$\frac{1}{3}$	N → cat	
			1	P → near	

Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree.

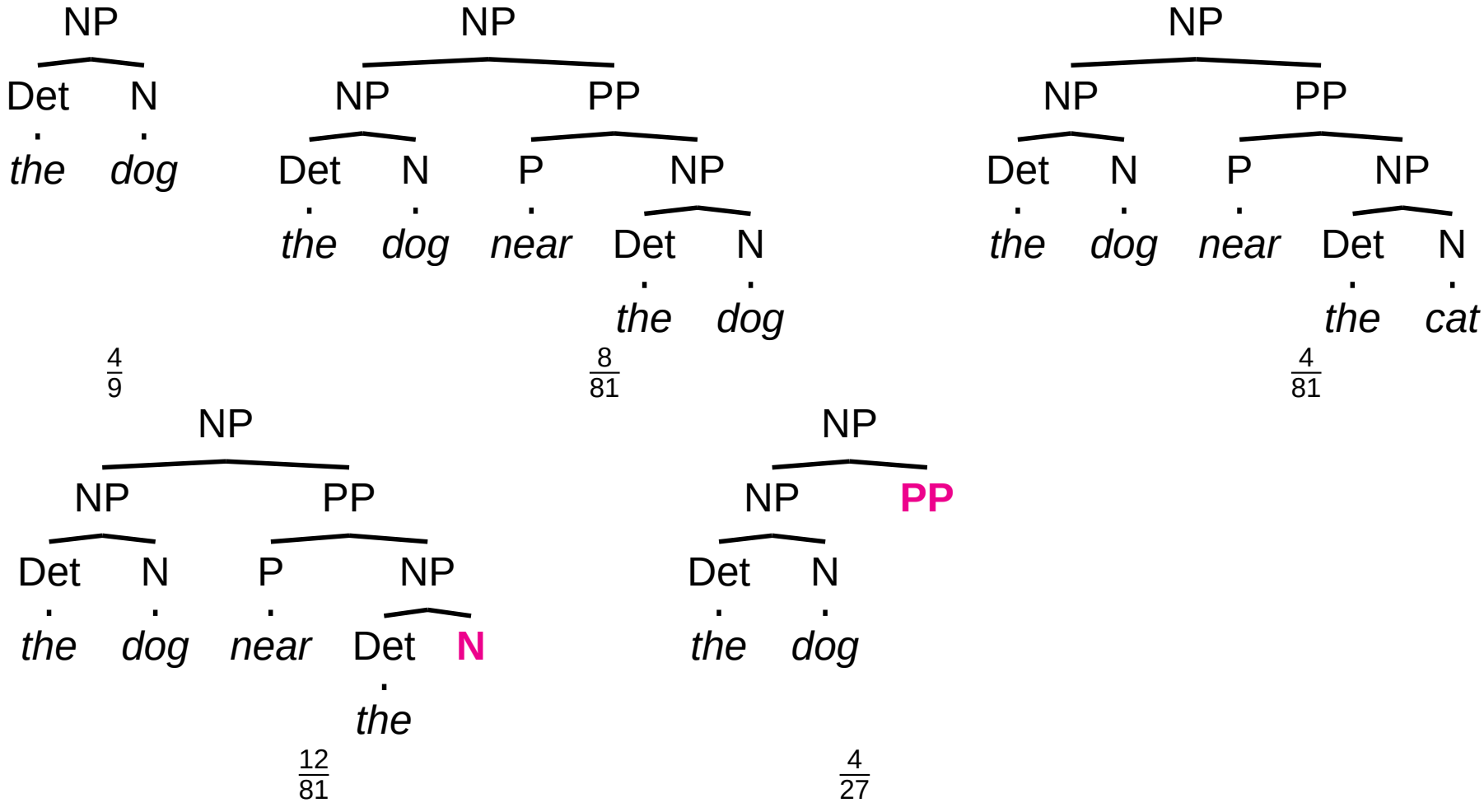
It has a corresponding marginal probability in the PCFG.



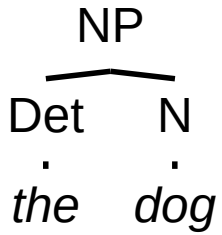
$\frac{2}{3}$	NP $\rightarrow$ Det N		1	Det $\rightarrow$ the
$\frac{1}{3}$	NP $\rightarrow$ NP PP		$\frac{2}{3}$	N $\rightarrow$ dog
1	PP $\rightarrow$ P NP		$\frac{1}{3}$	N $\rightarrow$ cat
			1	P $\rightarrow$ near

Shortcut 1: you can think of a *partial* tree as marginalizing over all completions of the partial tree.

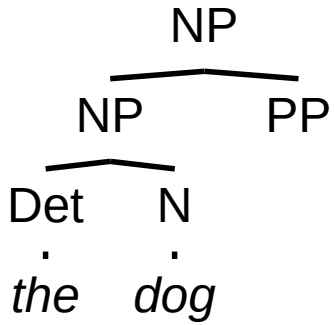
It has a corresponding marginal probability in the PCFG.



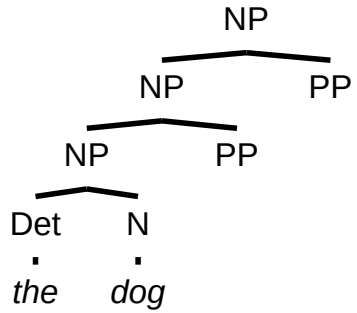
Problem 2: there are still an infinite number of incomplete trees covering a partial input.



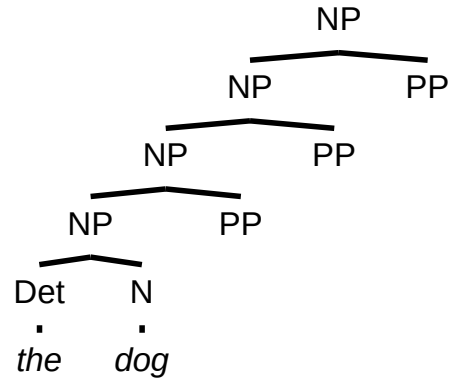
$\frac{4}{9}$



$\frac{4}{27}$

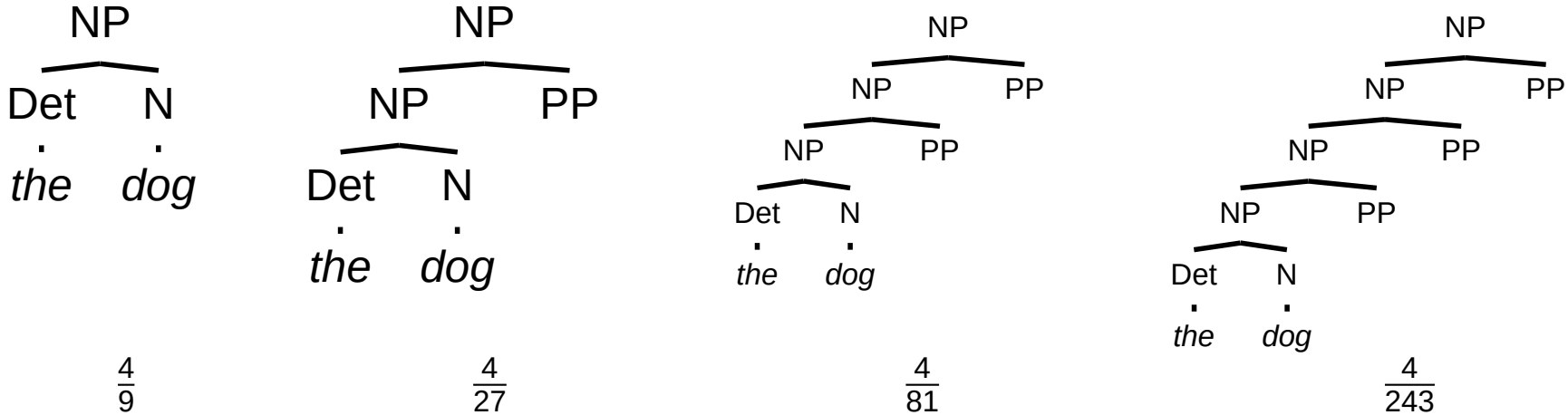


$\frac{4}{81}$



$\frac{4}{243}$

Problem 2: there are still an infinite number of incomplete trees covering a partial input.

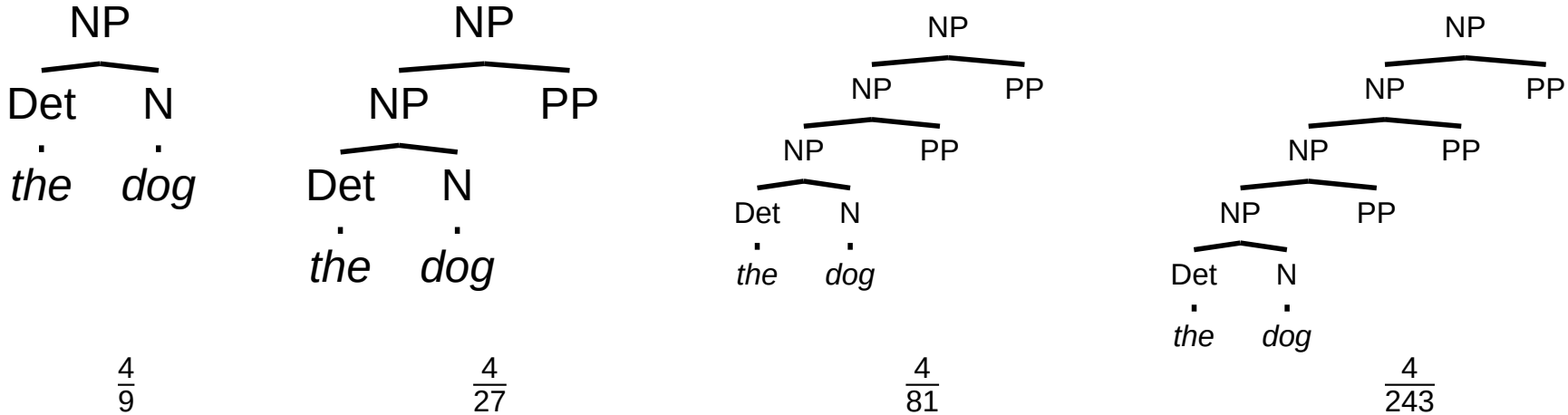


$\frac{4}{9}$ $\frac{4}{27}$ $\frac{4}{81}$ $\frac{4}{243}$

BUT! These tree probabilities form a geometric series:

$$P(the\ dog\ldots) = \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots$$

Problem 2: there are still an infinite number of incomplete trees covering a partial input.

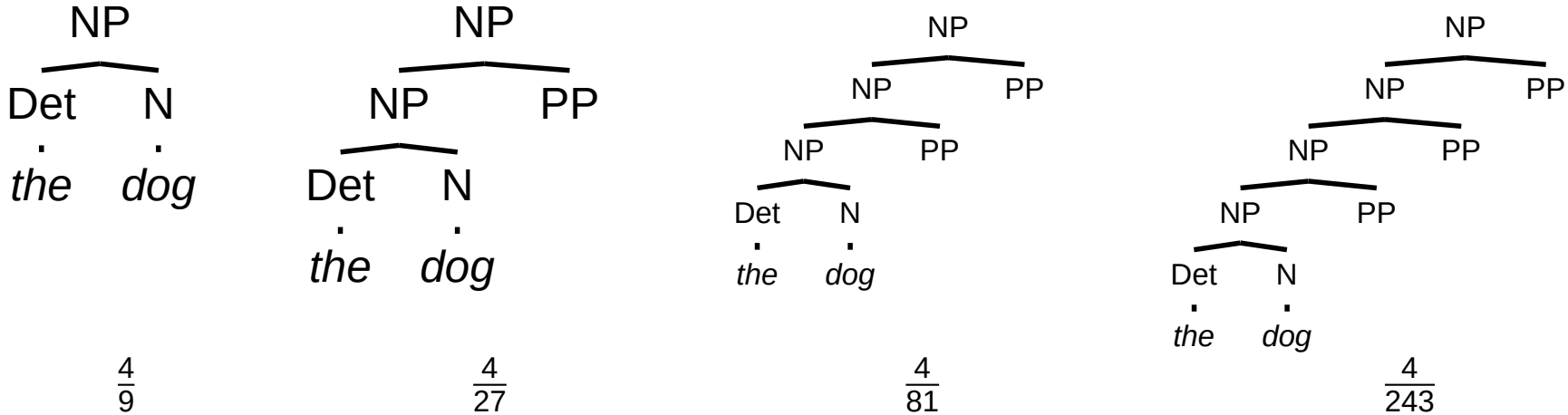


$\frac{4}{9}$ $\frac{4}{27}$ $\frac{4}{81}$ $\frac{4}{243}$

BUT! These tree probabilities form a geometric series:

$$
\begin{aligned}
P(the\ dog\ldots) &= \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots \\
&= \frac{4}{9}\sum_{i=0}^{\infty}\left(\frac{1}{3}\right)^{i}
\end{aligned}
$$

Problem 2: there are still an infinite number of incomplete trees covering a partial input.

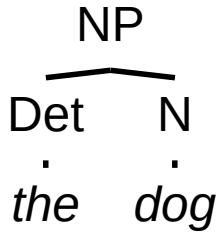


$\frac{4}{9}$ $\frac{4}{27}$ $\frac{4}{81}$ $\frac{4}{243}$

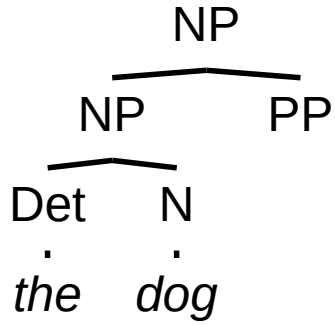
BUT! These tree probabilities form a geometric series:

$$
\begin{aligned}
P(the\ dog\ldots) &= \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots \\
&= \frac{4}{9}\sum_{i=0}^{\infty}\left(\frac{1}{3}\right)^{i} \\
&= \frac{2}{3}
\end{aligned}
$$

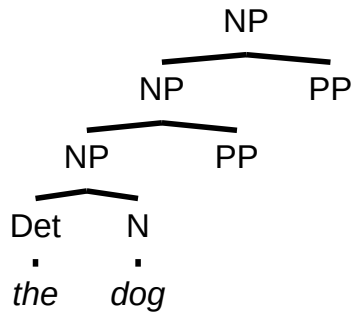
Problem 2: there are still an infinite number of incomplete trees covering a partial input.



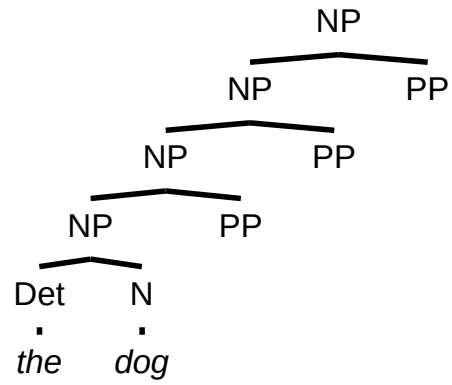
$\frac{4}{9}$



$\frac{4}{27}$



$\frac{4}{81}$



$\frac{4}{243}$

BUT! These tree probabilities form a geometric series:

$$
\begin{aligned}
P(the\ dog\ldots) &= \frac{4}{9} + \frac{4}{27} + \frac{4}{81} + \frac{4}{243} + \cdots \\
&= \frac{4}{9}\sum_{i=0}^{\infty}\left(\frac{1}{3}\right)^{i} \\
&= \frac{2}{3}
\end{aligned}
$$

. . . which matches the original rule probability

$\frac{2}{3}\ N \rightarrow dog$

Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

$A \rightarrow B\ \alpha$ $B \rightarrow A\ \beta$

(Stolcke, 1995)

Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

$$A \rightarrow B\ \alpha \qquad\qquad B \rightarrow A\ \beta$$

We can formulate a stochastic *left-corner matrix* of transitions between categories:

$$P_L = \begin{array}{c|cccc} & A & B & \dots & K \\ \hline A & 0.3 & 0.7 & \cdots & 0 \\ B & 0.1 & 0.1 & \cdots & 0.2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K & 0.2 & 0.1 & \cdots & 0.2 \end{array}$$

(Stolcke, 1995)

Generalizing the geometric series induced by rule recursion

In general, these infinite tree sets arise due to *left recursion* in a probabilistic grammar

$$A \rightarrow B\ \alpha \qquad\qquad B \rightarrow A\ \beta$$

We can formulate a stochastic *left-corner matrix* of transitions between categories:

$$P_L = \begin{array}{c|cccc} & A & B & \dots & K \\ \hline A & 0.3 & 0.7 & \cdots & 0 \\ B & 0.1 & 0.1 & \cdots & 0.2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K & 0.2 & 0.1 & \cdots & 0.2 \end{array}$$

and solve for its closure $R_L = (I - P_L)^{-1}$.

(Stolcke, 1995)

Generalizing the geometric series

1	ROOT	$\rightarrow$	NP	1	Det	$\rightarrow$	the
$\frac{2}{3}$	NP	$\rightarrow$	Det N	$\frac{2}{3}$	N	$\rightarrow$	dog
$\frac{1}{3}$	NP	$\rightarrow$	NP PP	$\frac{1}{3}$	N	$\rightarrow$	cat
1	PP	$\rightarrow$	P NP	1	P	$\rightarrow$	near

- The closure of our left-corner matrix is

$$R_L = \begin{array}{c} \\ \text{ROOT} \\ \text{NP} \\ \text{PP} \\ \text{Det} \\ \text{N} \\ \text{P} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{ROOT} & \text{NP} & \text{PP} & \text{Det} & \text{N} & \text{P} \end{array} \\ \begin{pmatrix} 1 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Generalizing the geometric series

1	ROOT	$\rightarrow$	NP	1	Det	$\rightarrow$	the
$\frac{2}{3}$	NP	$\rightarrow$	Det N	$\frac{2}{3}$	N	$\rightarrow$	dog
$\frac{1}{3}$	NP	$\rightarrow$	NP PP	$\frac{1}{3}$	N	$\rightarrow$	cat
1	PP	$\rightarrow$	P NP	1	P	$\rightarrow$	near

- The closure of our left-corner matrix is

$$R_L = \begin{array}{c} \\ \text{ROOT} \\ \text{NP} \\ \text{PP} \\ \text{Det} \\ \text{N} \\ \text{P} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{ROOT} & \text{NP} & \text{PP} & \text{Det} & \text{N} & \text{P} \end{array} \\ \begin{pmatrix} 1 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

- Refer to an entry (X, Y) in this matrix as $R(X \overset{*}{\Rightarrow}_L Y)$

Generalizing the geometric series

1	ROOT	$\rightarrow$	NP	1	Det	$\rightarrow$	the
$\frac{2}{3}$	NP	$\rightarrow$	Det N	$\frac{2}{3}$	N	$\rightarrow$	dog
$\frac{1}{3}$	NP	$\rightarrow$	NP PP	$\frac{1}{3}$	N	$\rightarrow$	cat
1	PP	$\rightarrow$	P NP	1	P	$\rightarrow$	near

- The closure of our left-corner matrix is

$$R_L = \begin{array}{c} \\ \text{ROOT} \\ \text{NP} \\ \text{PP} \\ \text{Det} \\ \text{N} \\ \text{P} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{ROOT} & \text{NP} & \text{PP} & \text{Det} & \text{N} & \text{P} \end{array} \\ \begin{pmatrix} 1 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

- Refer to an entry (X, Y) in this matrix as $R(X \stackrel{*}{\Rightarrow}_L Y)$
- Note that the $\frac{3}{2}$ "bonus" accrued for left-recursion of NPs appears in the (ROOT,NP) and (NP,NP) cells of the matrix

Generalizing the geometric series

1	ROOT	$\rightarrow$	NP	1	Det	$\rightarrow$	the
$\frac{2}{3}$	NP	$\rightarrow$	Det N	$\frac{2}{3}$	N	$\rightarrow$	dog
$\frac{1}{3}$	NP	$\rightarrow$	NP PP	$\frac{1}{3}$	N	$\rightarrow$	cat
1	PP	$\rightarrow$	P NP	1	P	$\rightarrow$	near

- The closure of our left-corner matrix is

$$R_L = \begin{array}{c} \\ \text{ROOT} \\ \text{NP} \\ \text{PP} \\ \text{Det} \\ \text{N} \\ \text{P} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{ROOT} & \text{NP} & \text{PP} & \text{Det} & \text{N} & \text{P} \end{array} \\ \begin{pmatrix} 1 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

- Refer to an entry (X, Y) in this matrix as $R(X \stackrel{*}{\Rightarrow}_L Y)$
- Note that the $\frac{3}{2}$ "bonus" accrued for left-recursion of NPs appears in the (ROOT,NP) and (NP,NP) cells of the matrix
- We need to do the same with unary chains, constructing a unary-closure matrix R_U.

Efficient incremental parsing: the probabilistic Earley algorithm

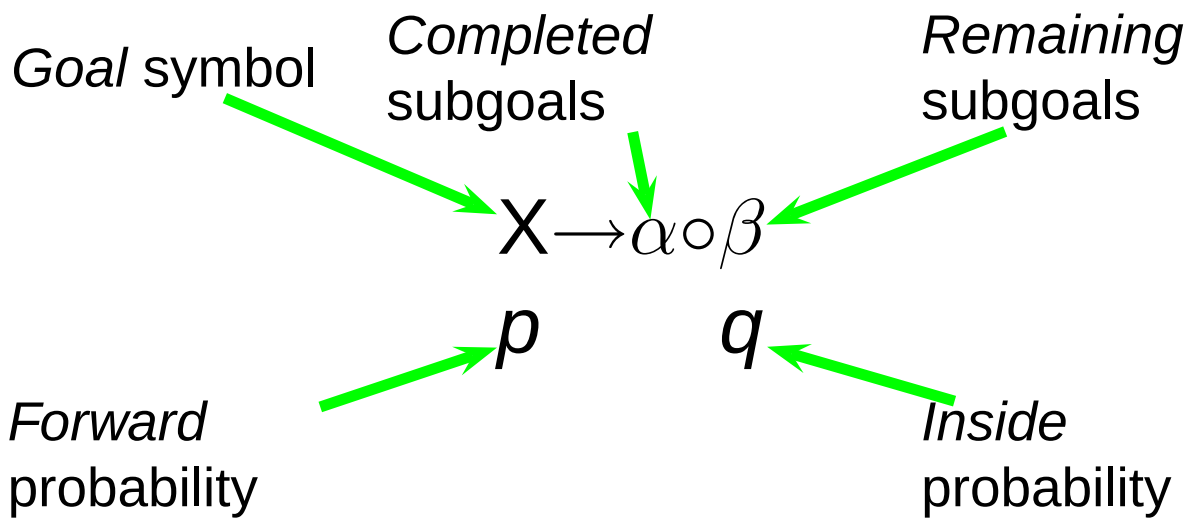
We can use the Earley algorithm (Earley, 1970) in a probabilistic incarnation (Stolcke, 1995) to deal with these infinite tree sets.

The (slightly oversimplified) probabilistic Earley algorithm has two fundamental types of operations:

- **Prediction**: if Y is a possible goal, and Y can lead to Z through a left corner, choose a rule $Z \rightarrow \alpha$ and set up α as a new sequence of possible goals.
- **Completion**: if Y is a possible goal, Y can lead to Z through unary rewrites, and we encounter a completed Z, absorb it and move on to the next sub-goal in the sequence.

Efficient incremental parsing: the probabilistic Earley algorithm

- Parsing consists of constructing a *chart* of *states* (items)
- A state has the following structure:



- The *forward* probability is the total probability of getting from the root at the start of the sentence through to this state
- The *inside* probability is the "bottom-up" probability of the state

Efficient incremental parsing: the probabilistic Earley algorithm

Inference rules for probabilistic Earley:

- **Prediction**:

$$\frac{\begin{array}{cc} X \rightarrow \beta \circ Y\,\gamma & \\ p & q \end{array} \qquad a : R(Y \overset{*}{\Rightarrow}_L Z) \qquad b : Z \rightarrow \alpha}{\begin{array}{cc} Z \rightarrow \circ\alpha & \\ abp & b \end{array}}$$

Efficient incremental parsing: the probabilistic Earley algorithm

Inference rules for probabilistic Earley:

- **Prediction**:

$$\frac{\begin{array}{c} X \rightarrow \beta \circ Y \gamma \\ p \qquad\quad q \end{array} \qquad a : R(Y \stackrel{*}{\Rightarrow}_L Z) \qquad b : Z \rightarrow \alpha}{\begin{array}{c} Z \rightarrow \circ \alpha \\ abp \qquad b \end{array}}$$

- **Completion**:

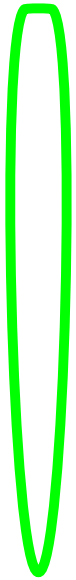
$$\frac{\begin{array}{c} X \rightarrow \beta \circ Y \gamma \\ p \qquad\quad q \end{array} \qquad a : R(Y \stackrel{*}{\Rightarrow}_U Z) \qquad \begin{array}{c} Z \rightarrow \alpha \circ \\ b \qquad\quad c \end{array}}{\begin{array}{c} X \rightarrow \beta Y \circ \gamma \\ acp \qquad acq \end{array}}$$

Efficient incremental parsing: probabilistic Earley

the dog near the

Efficient incremental parsing: probabilistic Earley

ROOT$\rightarrow\circ$NP
1 1



the dog near the

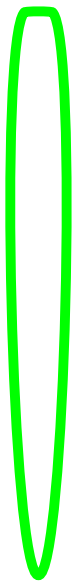
Efficient incremental parsing: probabilistic Earley

Det$\rightarrow\circ$the
1 1

NP$\rightarrow\circ$Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP$\rightarrow\circ$NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

ROOT$\rightarrow\circ$NP
1 1



the dog near the

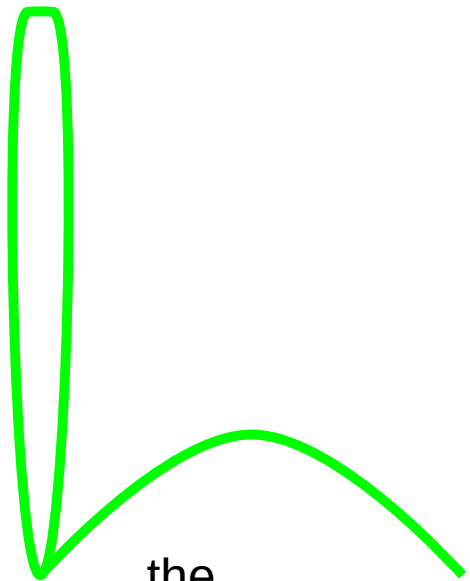
Efficient incremental parsing: probabilistic Earley

Det$\rightarrow\circ$the
1 1

NP$\rightarrow\circ$Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP$\rightarrow\circ$NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

ROOT$\rightarrow\circ$NP
1 1



the dog near the

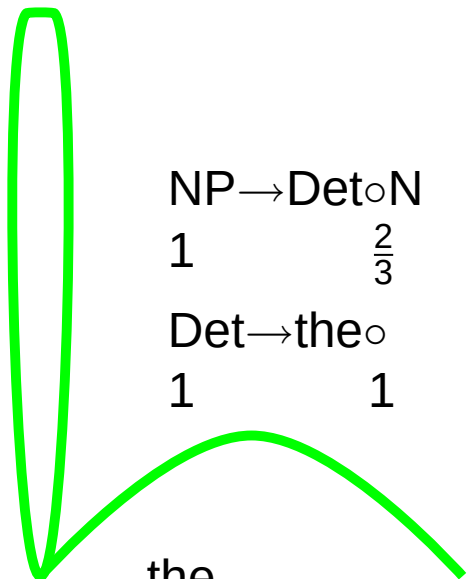
Efficient incremental parsing: probabilistic Earley

Det$\rightarrow\circ$the
1 1

NP$\rightarrow\circ$Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP$\rightarrow\circ$NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

ROOT$\rightarrow\circ$NP
1 1



NP$\rightarrow$Det$\circ$N
1 $\frac{2}{3}$

Det$\rightarrow$the$\circ$
1 1

the dog near the

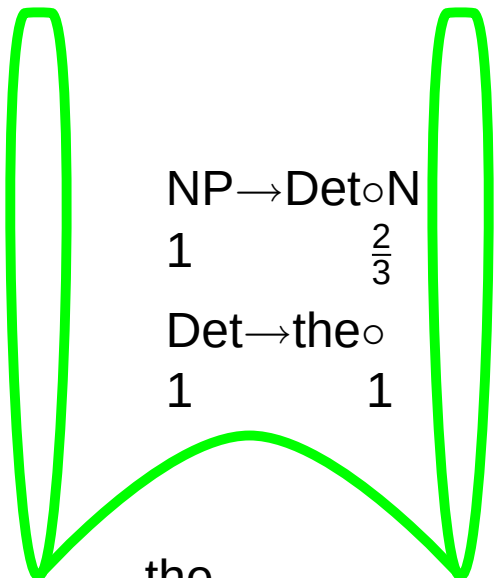
Efficient incremental parsing: probabilistic Earley

Det$\rightarrow\circ$the
1 1

NP$\rightarrow\circ$Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP$\rightarrow\circ$NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

ROOT$\rightarrow\circ$NP
1 1



the dog near the

Efficient incremental parsing: probabilistic Earley

Det$\rightarrow\circ$the
1 1

NP$\rightarrow\circ$Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP$\rightarrow\circ$NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

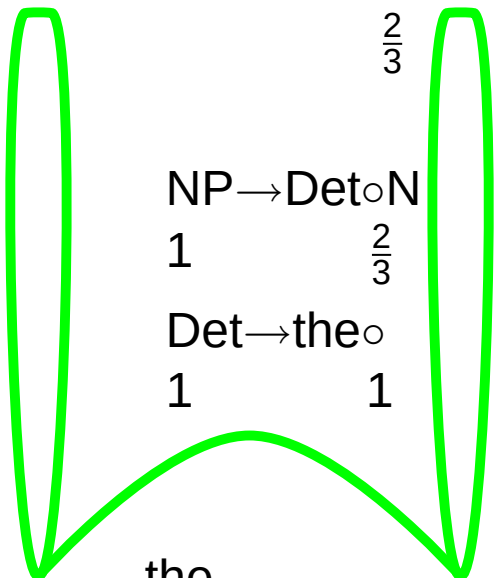
ROOT$\rightarrow\circ$NP
1 1

N$\rightarrow\circ$cat
$\frac{1}{3}$ $\frac{1}{3}$

N$\rightarrow\circ$dog
$\frac{2}{3}$ $\frac{2}{3}$

NP$\rightarrow$Det$\circ$N
1 $\frac{2}{3}$

Det$\rightarrow$the$\circ$
1 1



the dog near the

Efficient incremental parsing: probabilistic Earley

Det→○the
1 1

NP→○Det N
$\frac{2}{3} \times \frac{3}{2}$ $\frac{2}{3}$

NP→○NP PP
$\frac{1}{3} \times \frac{3}{2}$ $\frac{1}{3}$

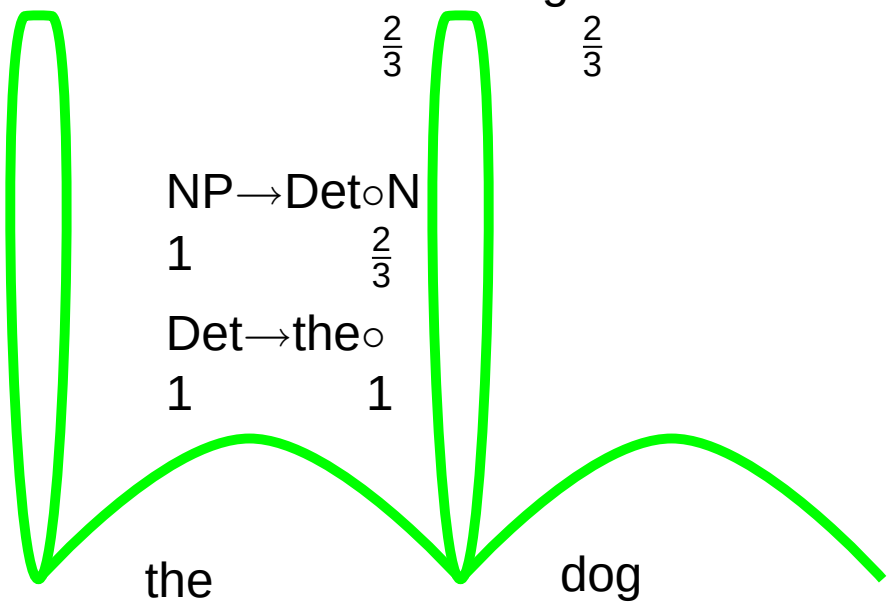
ROOT→○NP
1 1

N→○cat
$\frac{1}{3}$ $\frac{1}{3}$

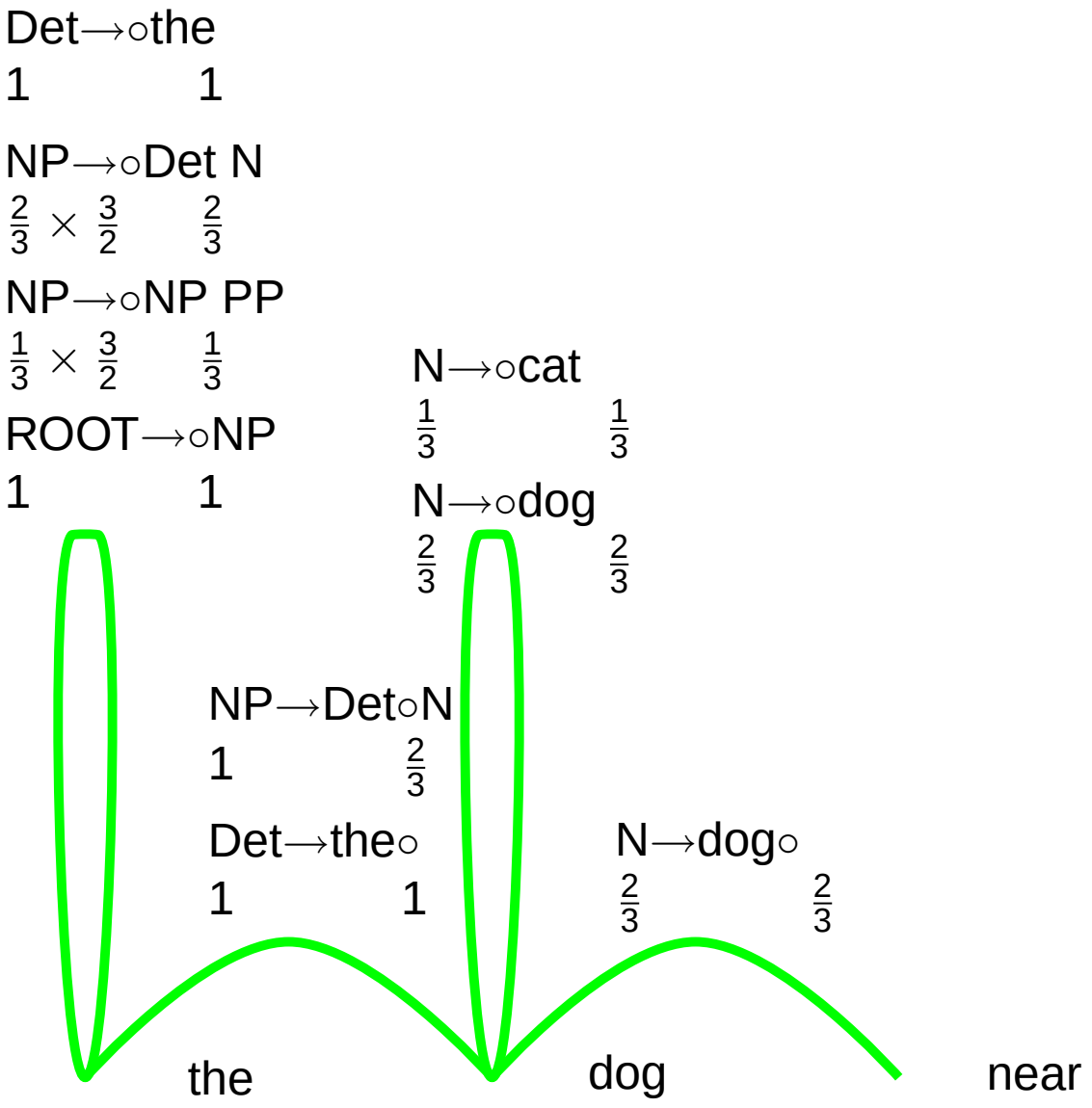
N→○dog
$\frac{2}{3}$ $\frac{2}{3}$

NP→Det○N
1 $\frac{2}{3}$

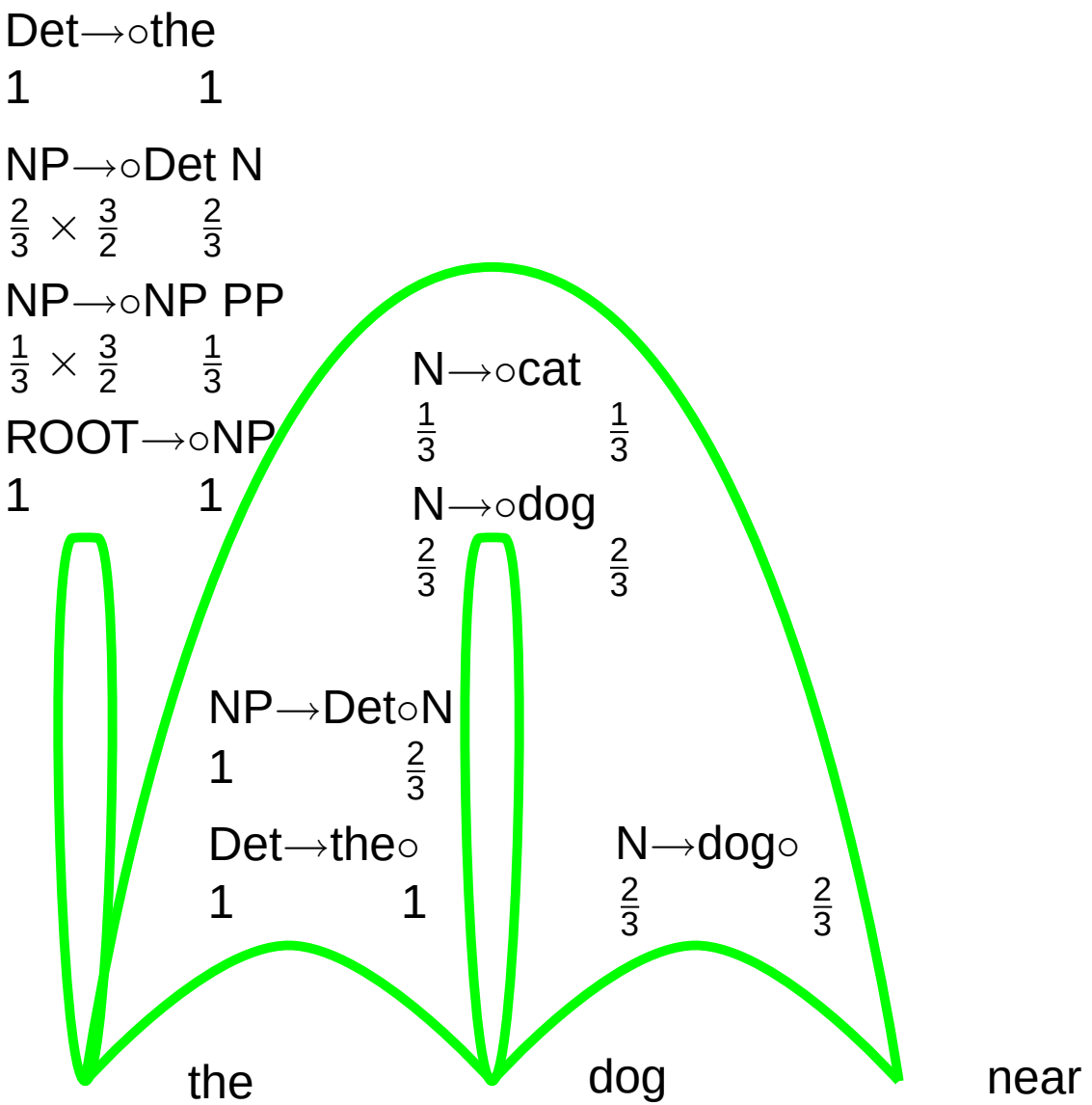
Det→the○
1 1



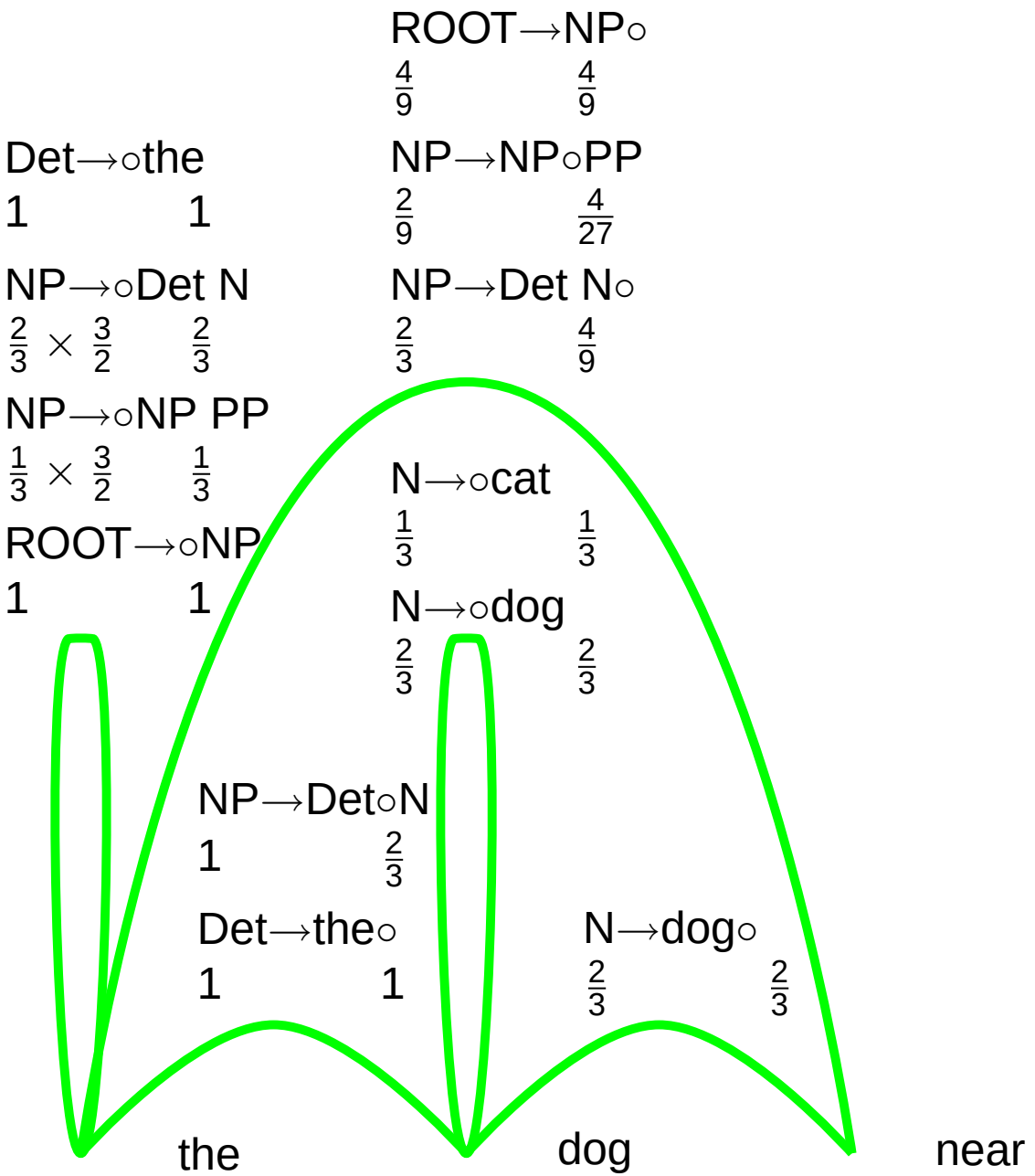
Efficient incremental parsing: probabilistic Earley



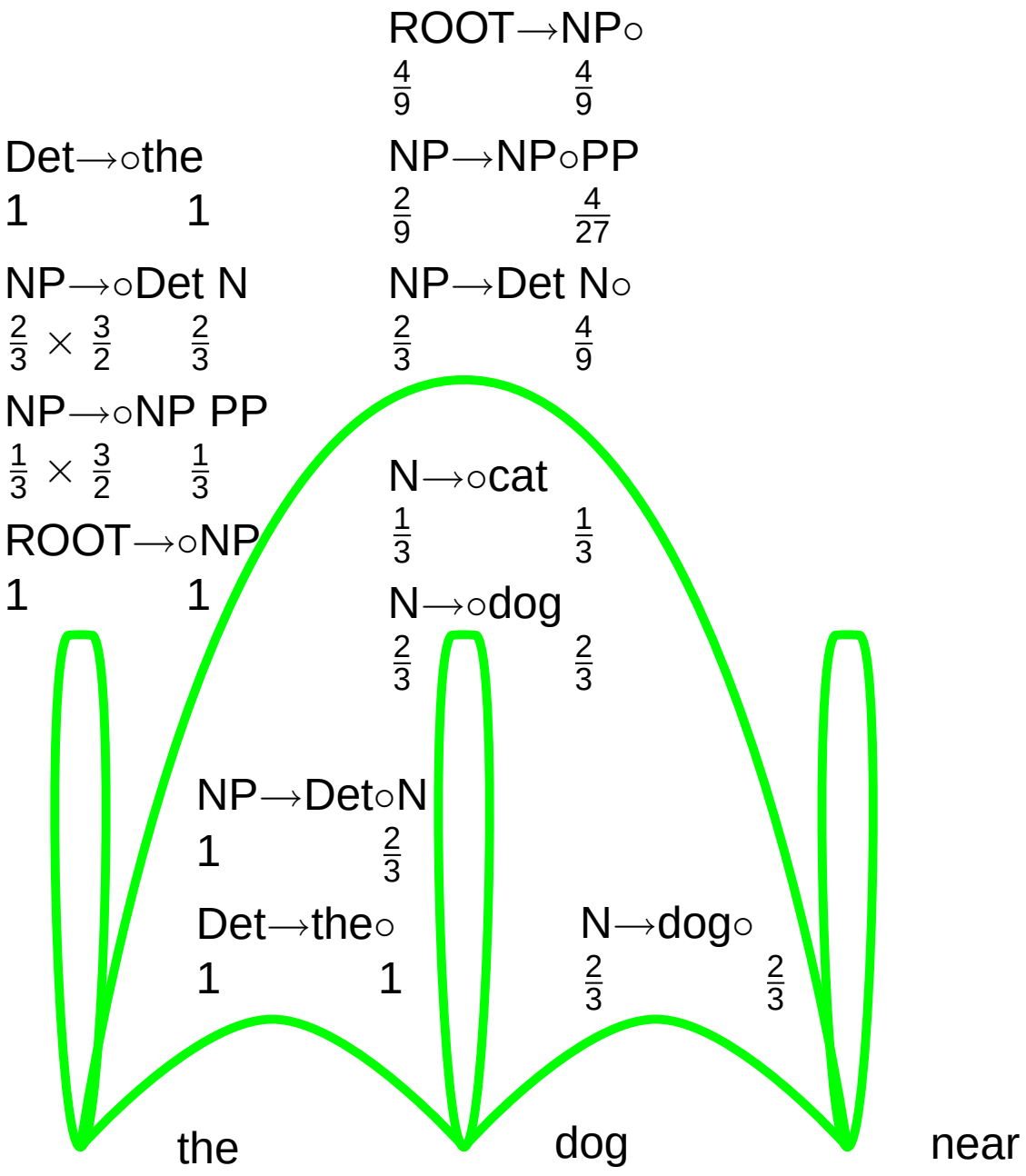
Efficient incremental parsing: probabilistic Earley



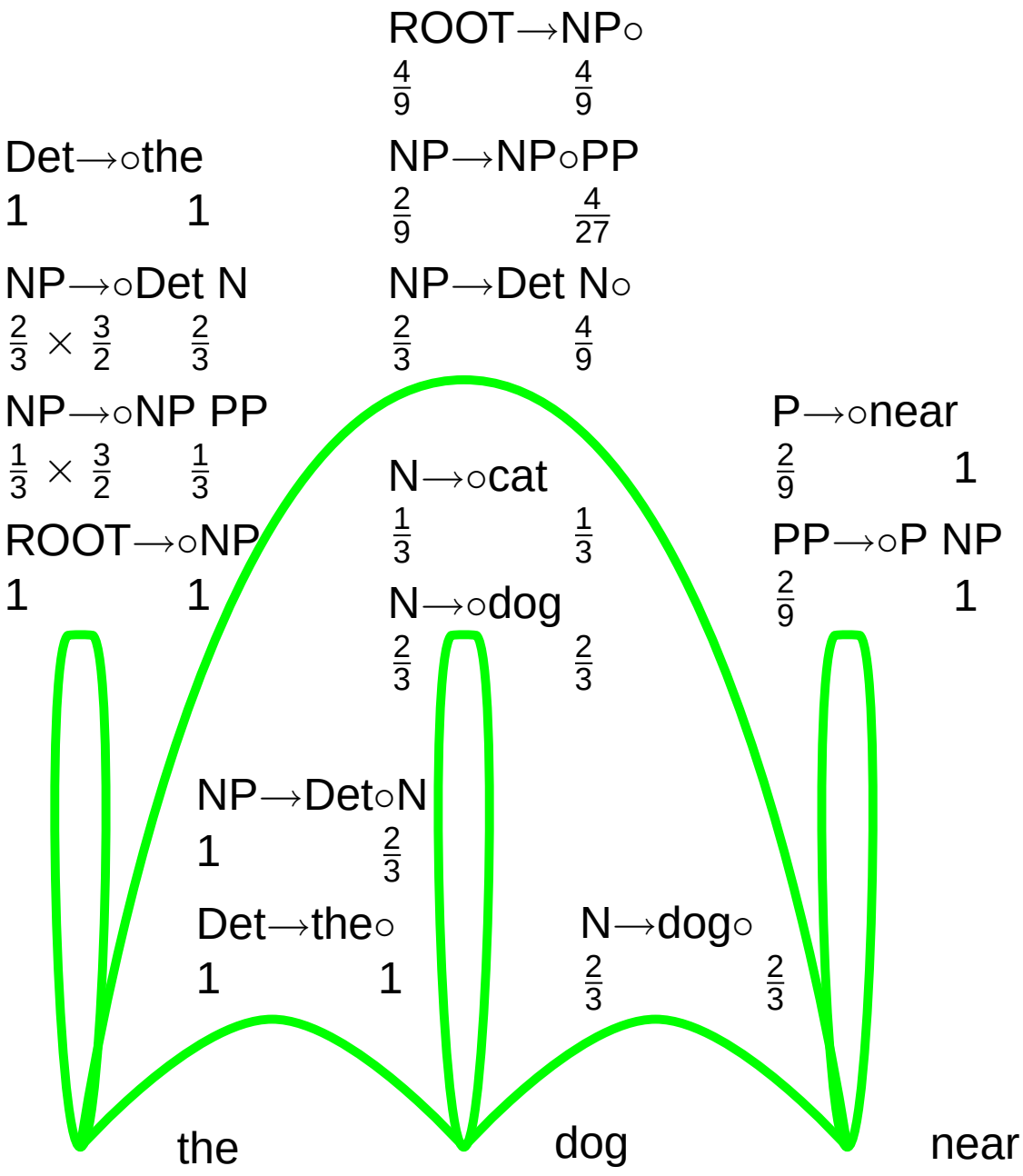
Efficient incremental parsing: probabilistic Earley



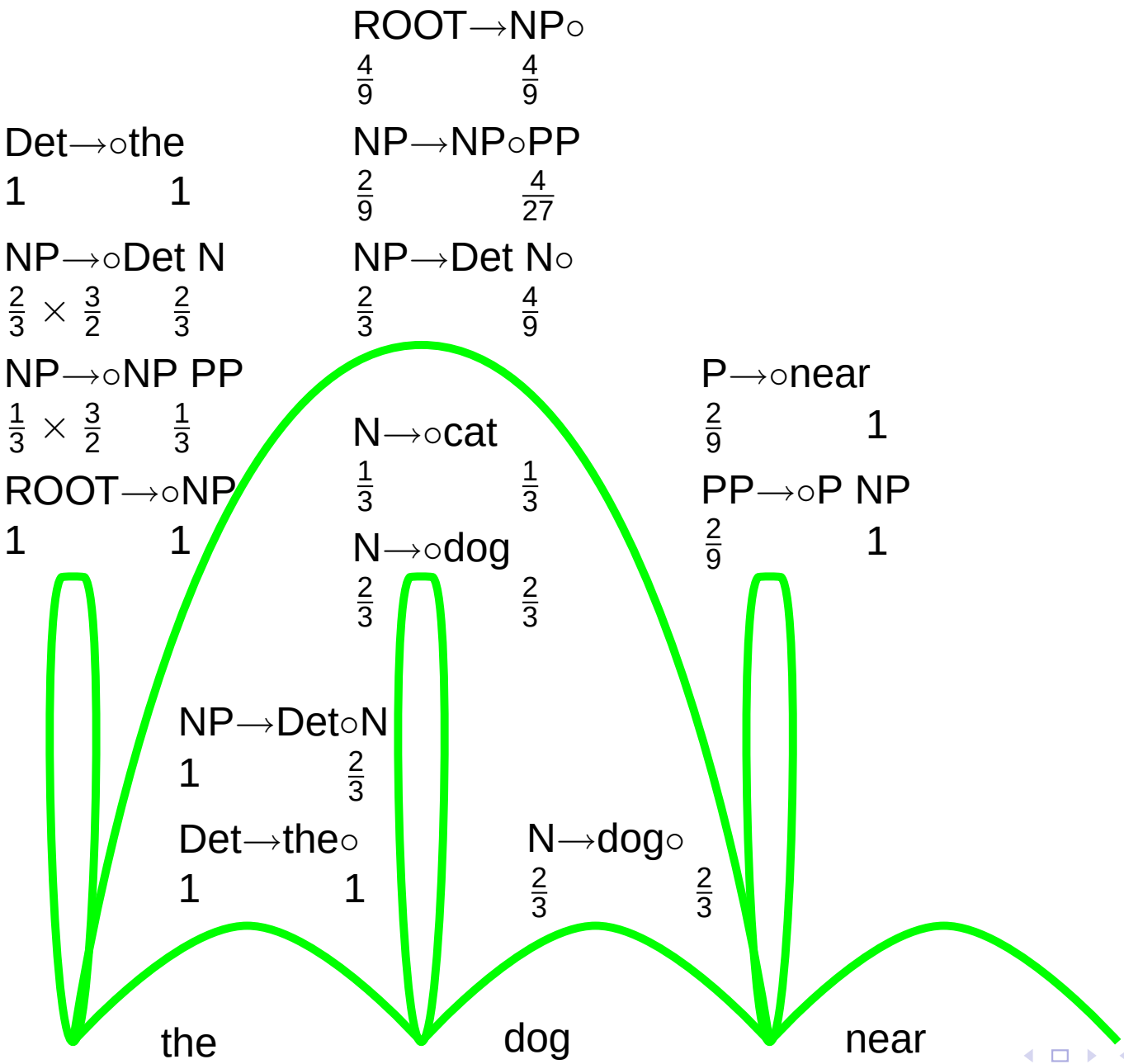
Efficient incremental parsing: probabilistic Earley



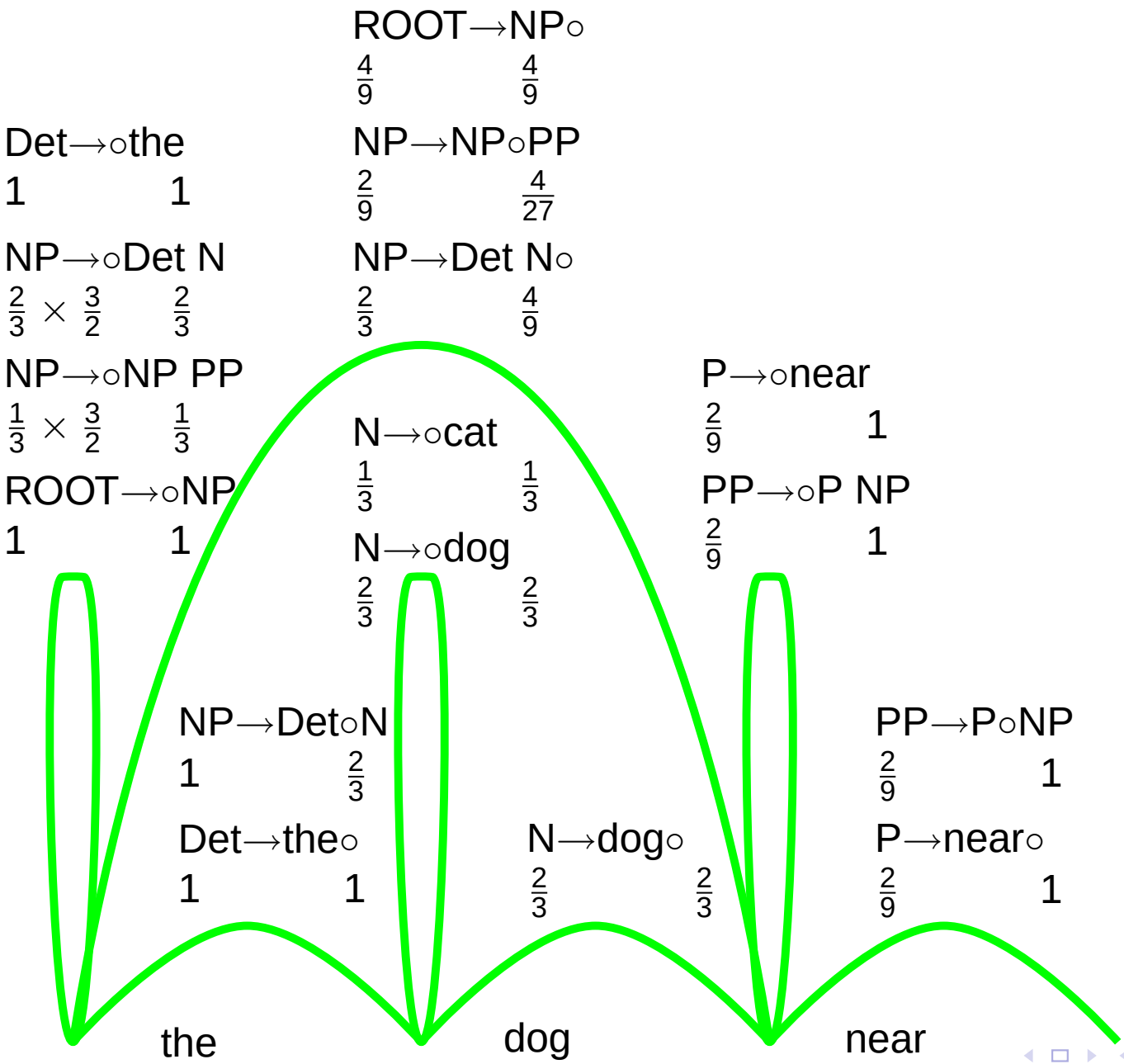
Efficient incremental parsing: probabilistic Earley



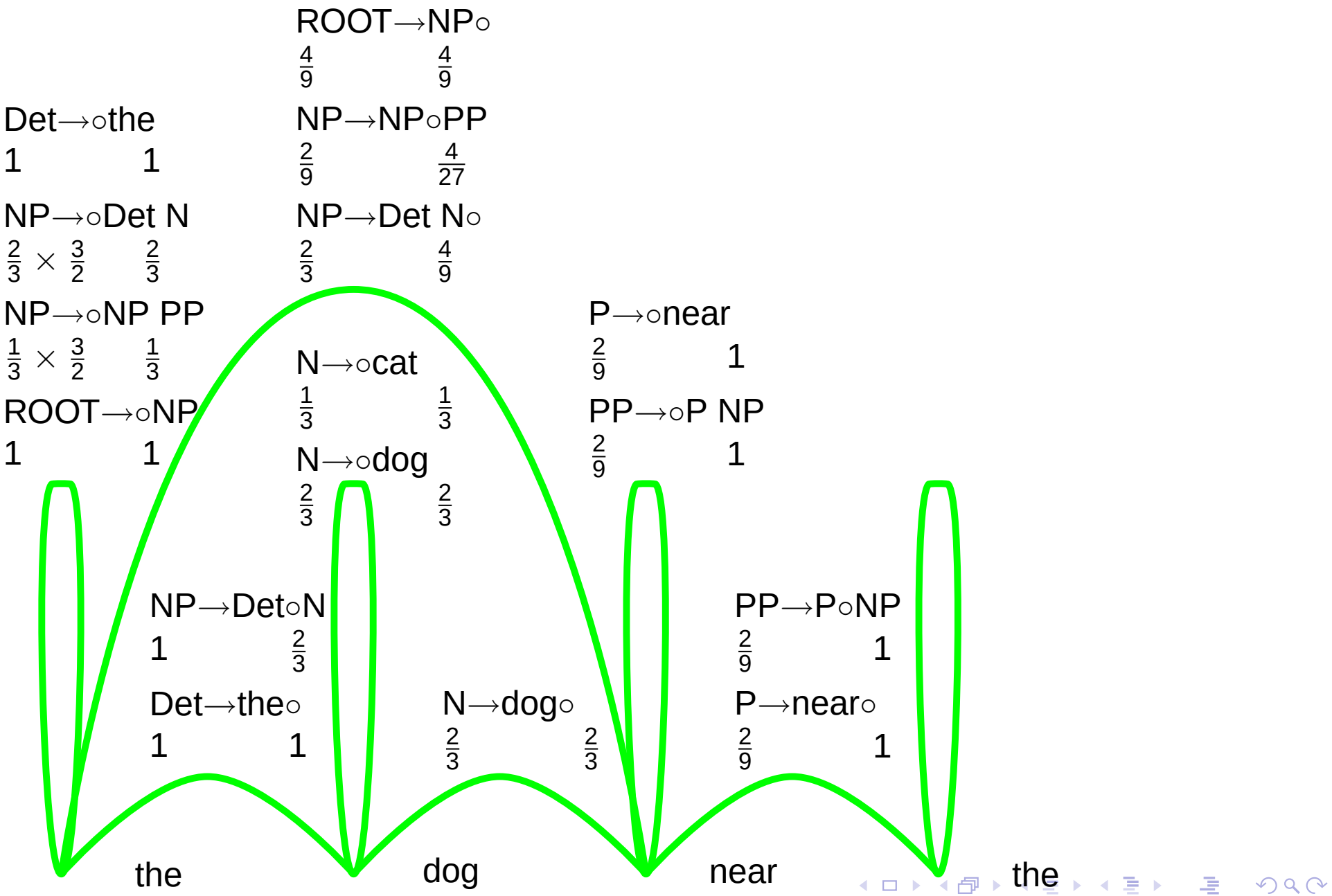
Efficient incremental parsing: probabilistic Earley



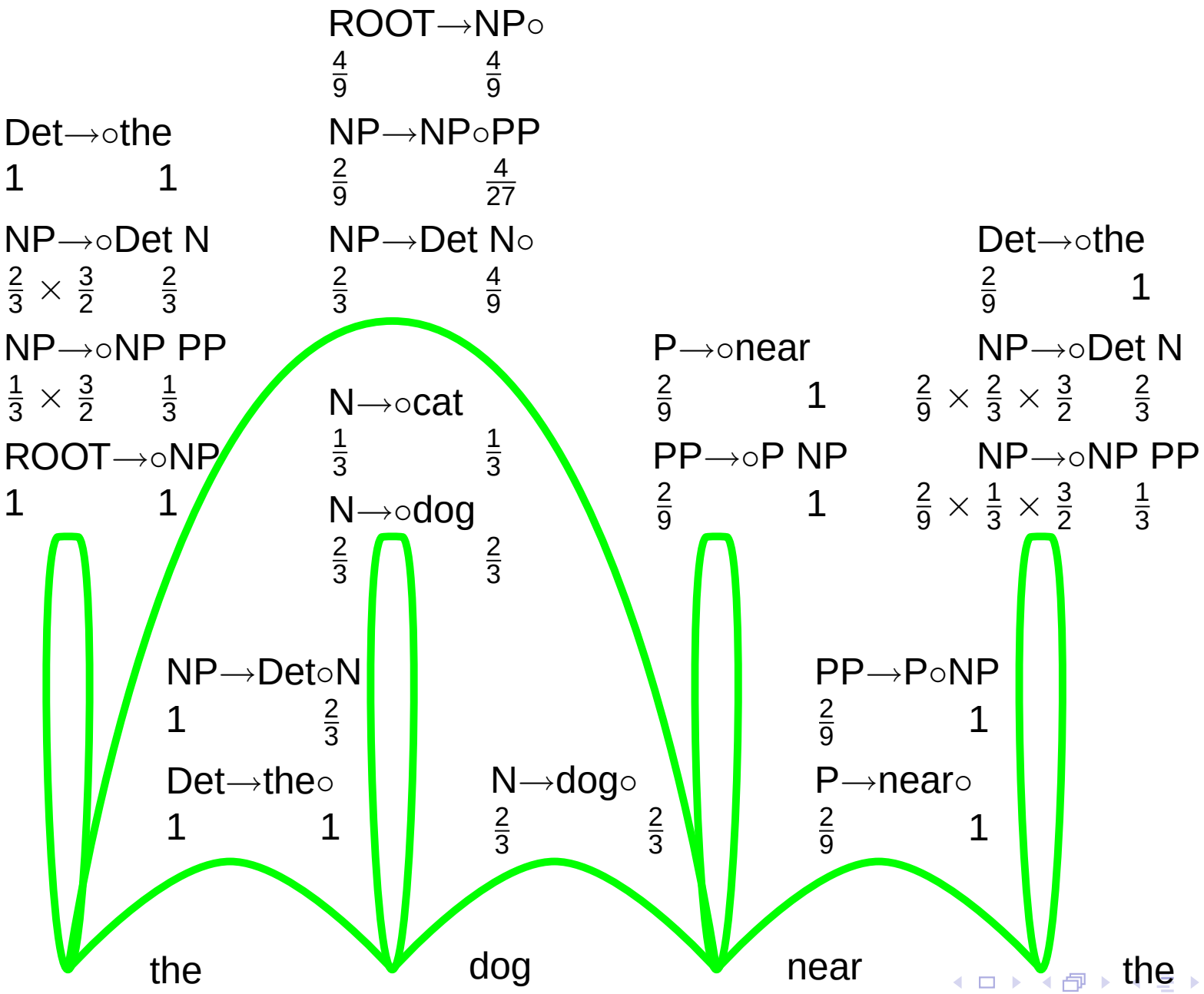
Efficient incremental parsing: probabilistic Earley



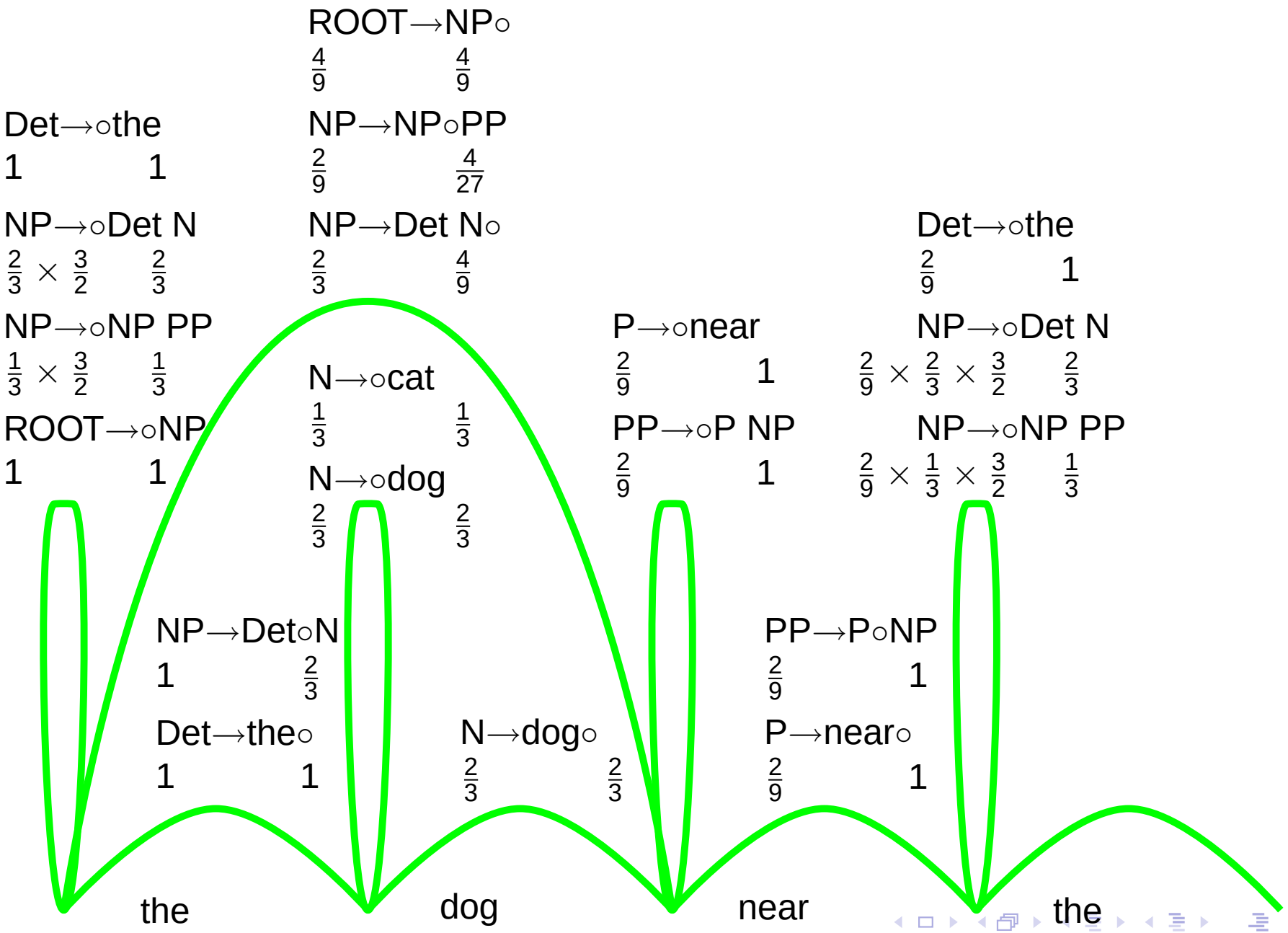
Efficient incremental parsing: probabilistic Earley



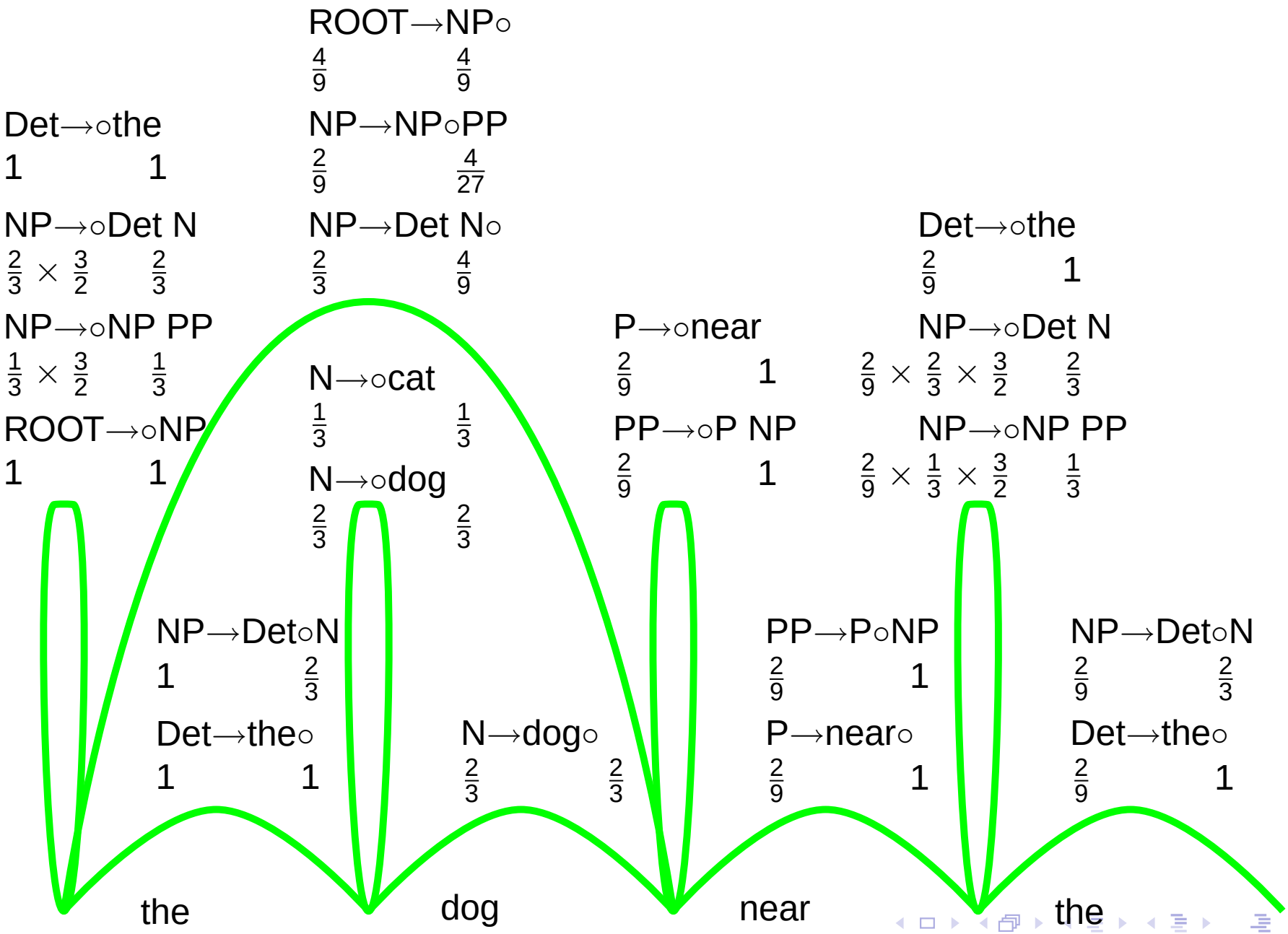
Efficient incremental parsing: probabilistic Earley



Efficient incremental parsing: probabilistic Earley



Efficient incremental parsing: probabilistic Earley



Prefix probabilities from probabilistic Earley

- If you have just processed word w_i, then the prefix probability of $w_{1\ldots i}$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$

Prefix probabilities from probabilistic Earley

- If you have just processed word w_i, then the prefix probability of $w_{1\ldots i}$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$
- In our example, we see:

$$
\begin{aligned}
P(\text{the}) &= 1 \\
P(\text{the dog}) &= \tfrac{2}{3} \\
P(\text{the dog near}) &= \tfrac{2}{9} \\
P(\text{the dog near the}) &= \tfrac{2}{9}
\end{aligned}
$$

Prefix probabilities from probabilistic Earley

- If you have just processed word w_i, then the prefix probability of $w_{1\ldots i}$ can be obtained by summing all forward probabilities of items that have the form $X \rightarrow \alpha \circ w_i \beta$
- In our example, we see:

$$
\begin{aligned}
P(\text{the}) &= 1 \\
P(\text{the dog}) &= \tfrac{2}{3} \\
P(\text{the dog near}) &= \tfrac{2}{9} \\
P(\text{the dog near the}) &= \tfrac{2}{9}
\end{aligned}
$$

- Taking the ratios of these prefix probabilities can give us conditional word probabilities

Probabilistic Earley as an “eager” algorithm

- From the *inside probabilities* of the states on the chart, the posterior distribution on (incremental) trees can be directly calculated
- This posterior distribution is *precisely* the correct result of the application of Bayes’ rule
- Hence, probabilistic Earley is also performing rational disambiguation
- Hale (2001) called this the “eager” property of an incremental parsing algorithm.

Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events
- Infinitely many trees can be constructed in polynomial time ( ) and space ( )
- The *prefix probability* of the string is calculated in the process
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated

Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events
- Infinitely many trees can be constructed in polynomial time ($O(n^3)$) and space ( )
- The *prefix probability* of the string is calculated in the process
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated

Probabilistic Earley algorithm: key ideas

- We want to use probabilistic grammars for both disambiguation and calculating probability distributions over upcoming events
- Infinitely many trees can be constructed in polynomial time ($O(n^3)$) and space ($O(n^2)$)
- The *prefix probability* of the string is calculated in the process
- By taking the log-ratio of two prefix probabilities, the surprisal of a word in its context can be calculated

Other introductions

- You can read about the (non-probabilistic) Earley algorithm in (Jurafsky and Martin, 2000, Chapter 13)
- Prefix probabilities can also be calculated with an extension of the CKY algorithm due to Jelinek and Lafferty (1991)

Also. . .

Applications of the idea of surprisal to comprehension and production

References I

Altmann, G. T. and Kamide, Y. (1999). Incremental interpretation at verbs: restricting the domain of subsequent reference. *Cognition*, 73(3):247–264.

Attneave, F. (1959). *Applications of Information Theory to Psychology: A summary of basic concepts, methods and results*. Holt, Rinehart and Winston.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Ehrlich, S. F. and Rayner, K. (1981). Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, 20:641–655.

Grodner, D. and Gibson, E. (2005). Some consequences of the serial nature of linguistic input. *Cognitive Science*, 29(2):261–290.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 159–166.

References II

Itti, L. and Baldi, P. (2005). Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems*.

Jelinek, F. and Lafferty, J. D. (1991). Computation of the probability of initial substring generation by stochastic context free grammars. *Computational Linguistics*, 17(3):315–323.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

Kaiser, E. and Trueswell, J. C. (2004). The role of discourse context in the processing of a flexible word-order language. *Cognition*, 94:113–147.

Levy, R. (2005). *Probabilistic Models of Word Order and Syntactic Discontinuity*. PhD thesis, Stanford University.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

References III

Mandelbrot, B. (1953). An informational theory of the statistical structure of language. In Jackson, W., editor, *Communication Theory: papers read at a symposium on applications of communication theory*. New York: Academic Press.

Norris, D. (2006). The Bayesian Reader: Explaining word recognition as an optimal Bayesian decision process. *Psychological Review*, 113(2):327–357.

Smith, N. (2006). Surprisal-based sentence processing as optimal behavior. M.S., UC San Diego.

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K., and Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.