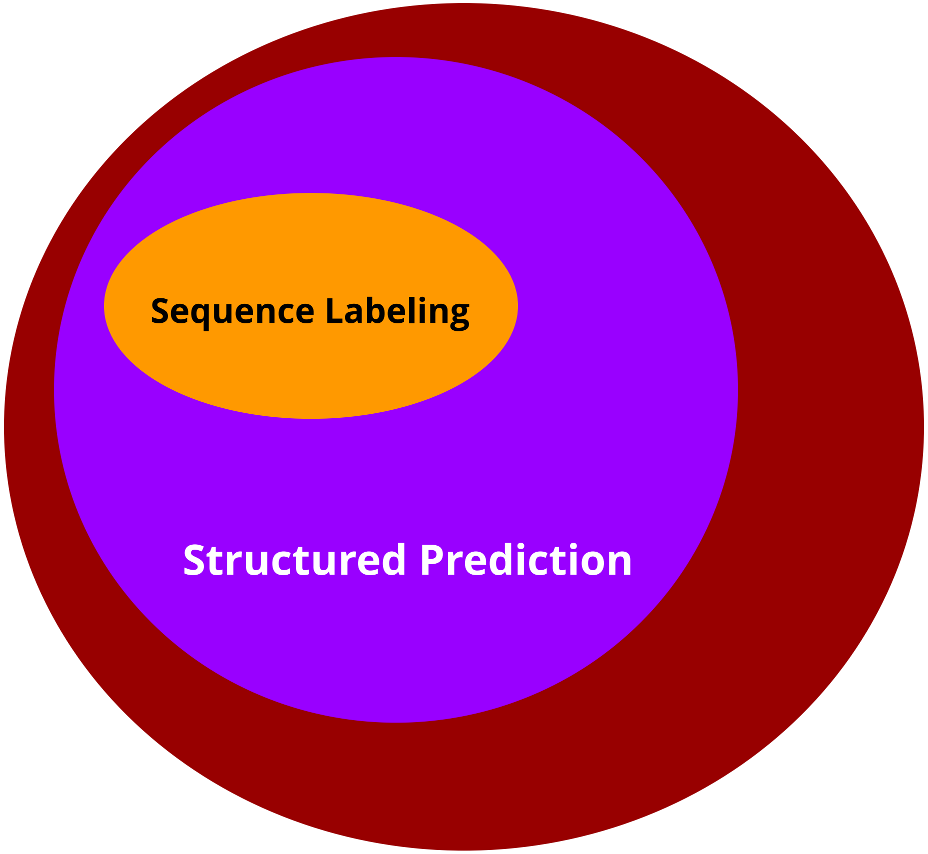
50.040 Natural Language Processing

Lu, Wei

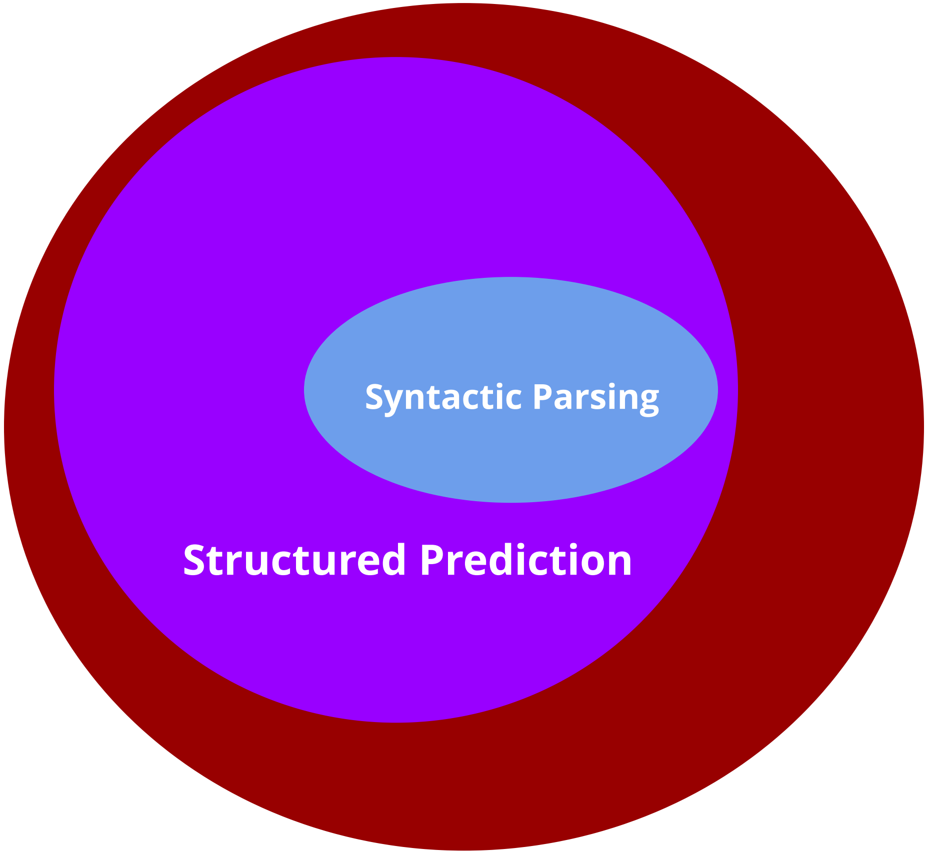
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Tasks in NLP



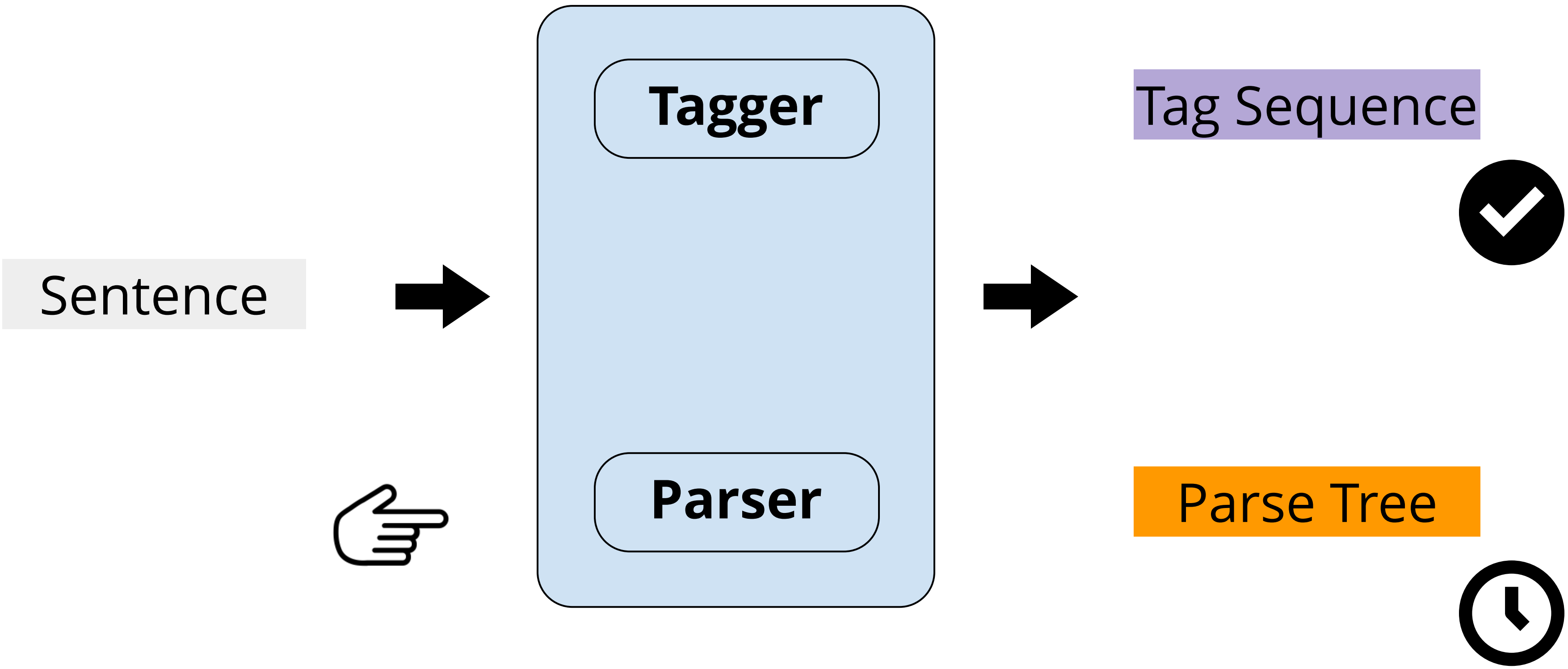
Supervised

Tasks in NLP



Supervised

Structured Prediction



Structured Space $\mathcal{S}_i$ Structured Prediction Model M Structured Space $\mathcal{S}_o$

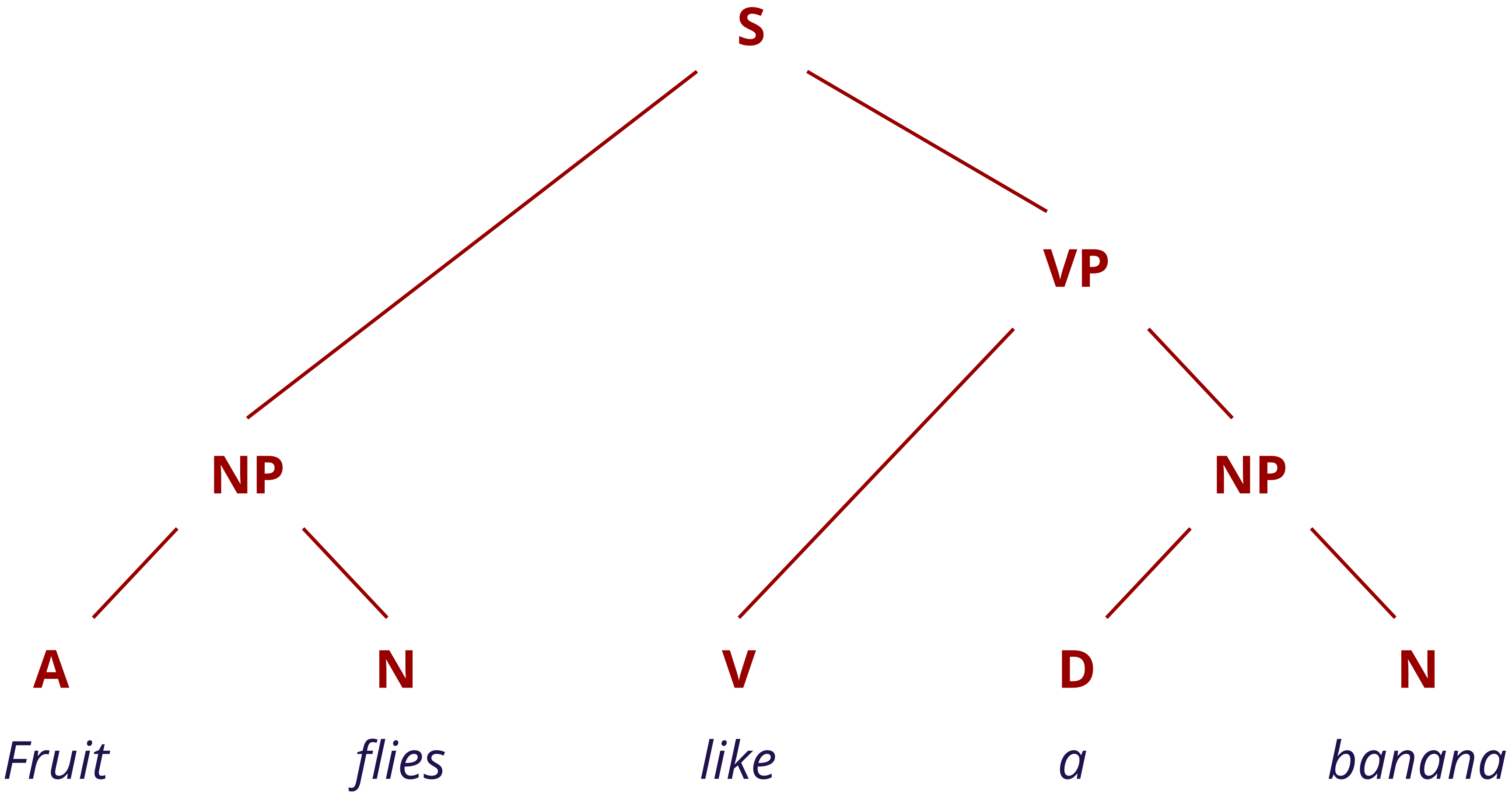
$$\mathsf{M} : \mathcal{S}_i \rightarrow \mathcal{S}_o$$

Syntactic Parsing

The task of mapping a sentence into its syntactic structure (usually in the form of a tree or a graph).

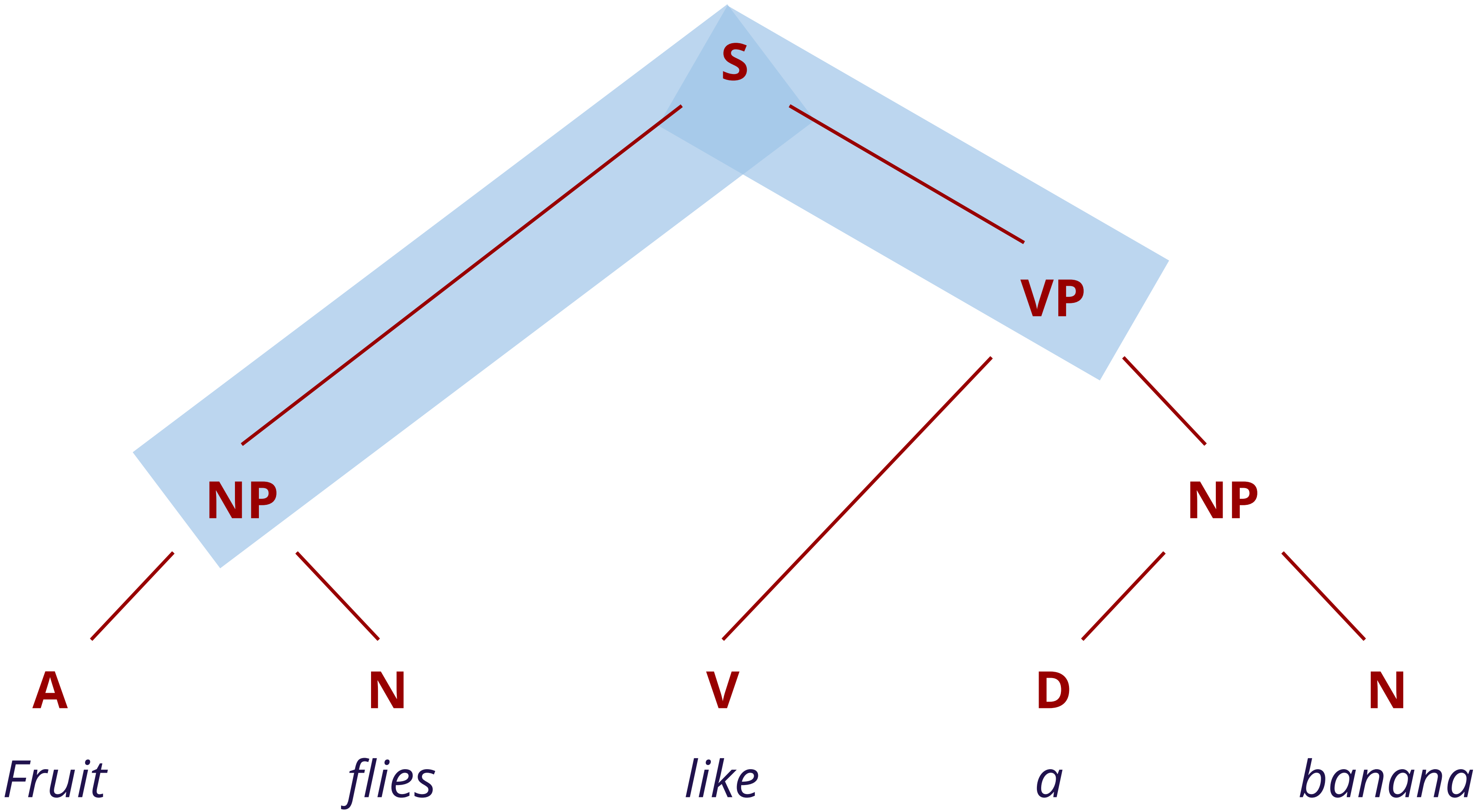
In this course

Constituency Parsing



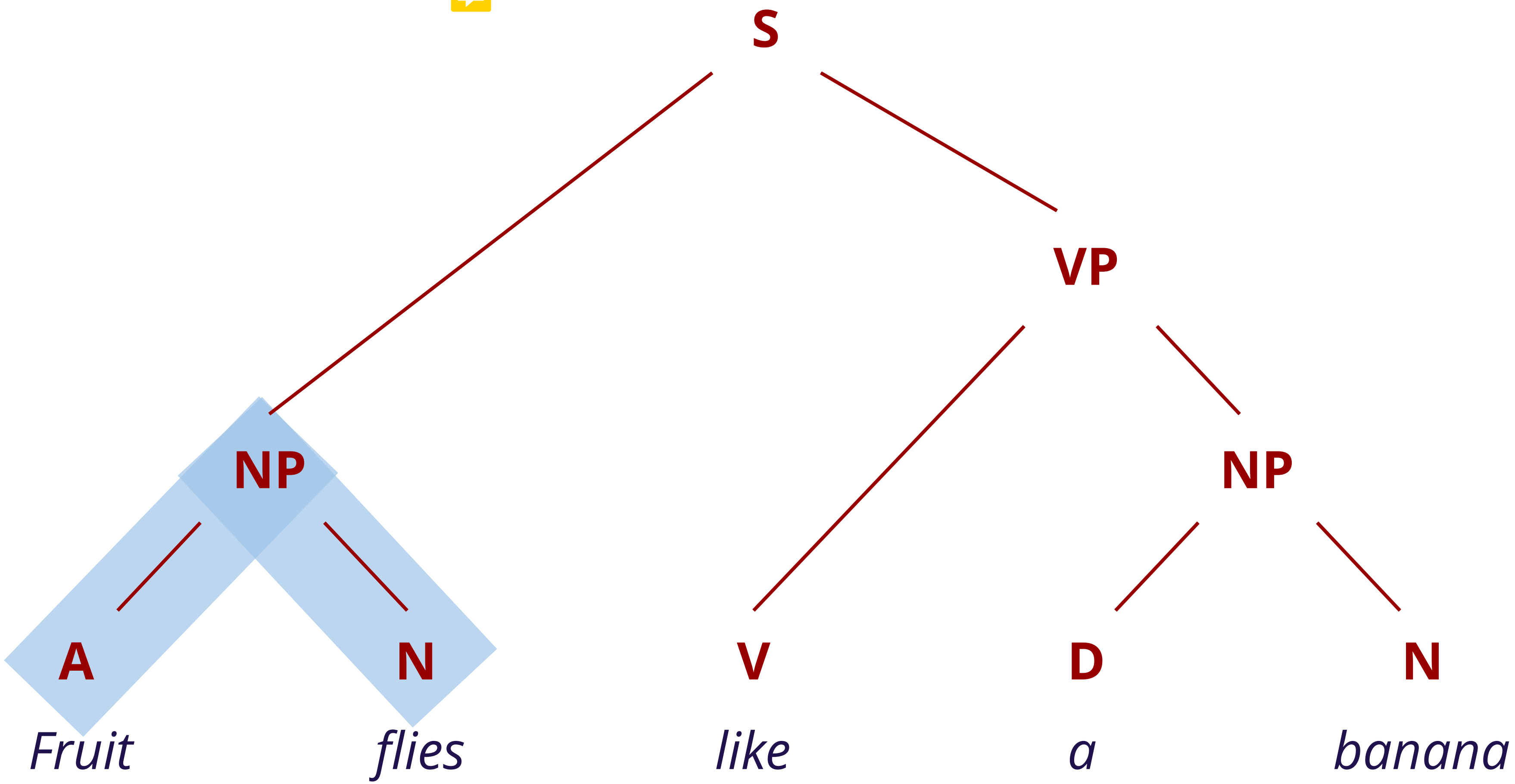
Grammar

$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP}$



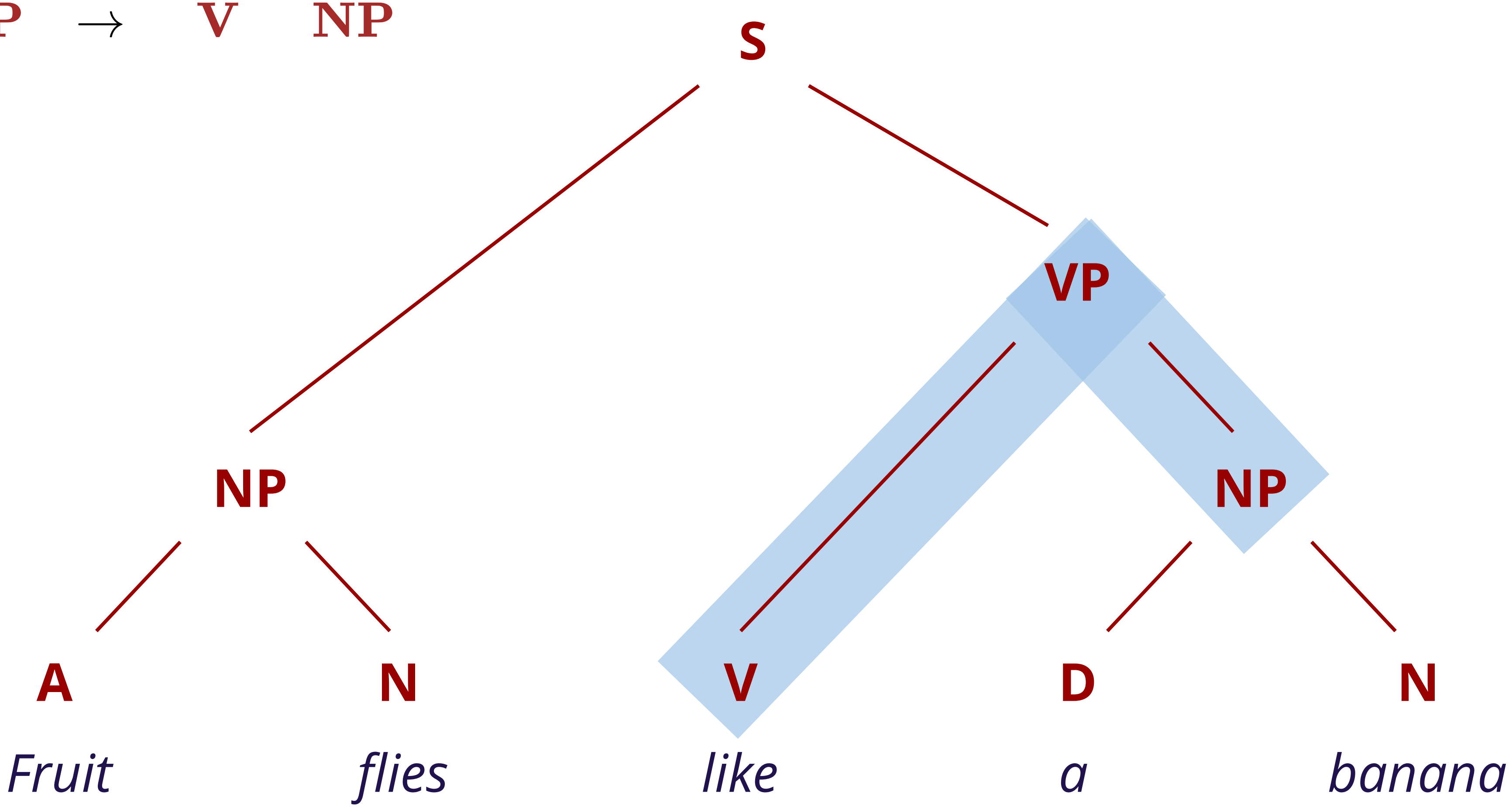
Grammar

$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP}$$
$$\mathbf{NP} \rightarrow \mathbf{A} \quad \mathbf{N}$$



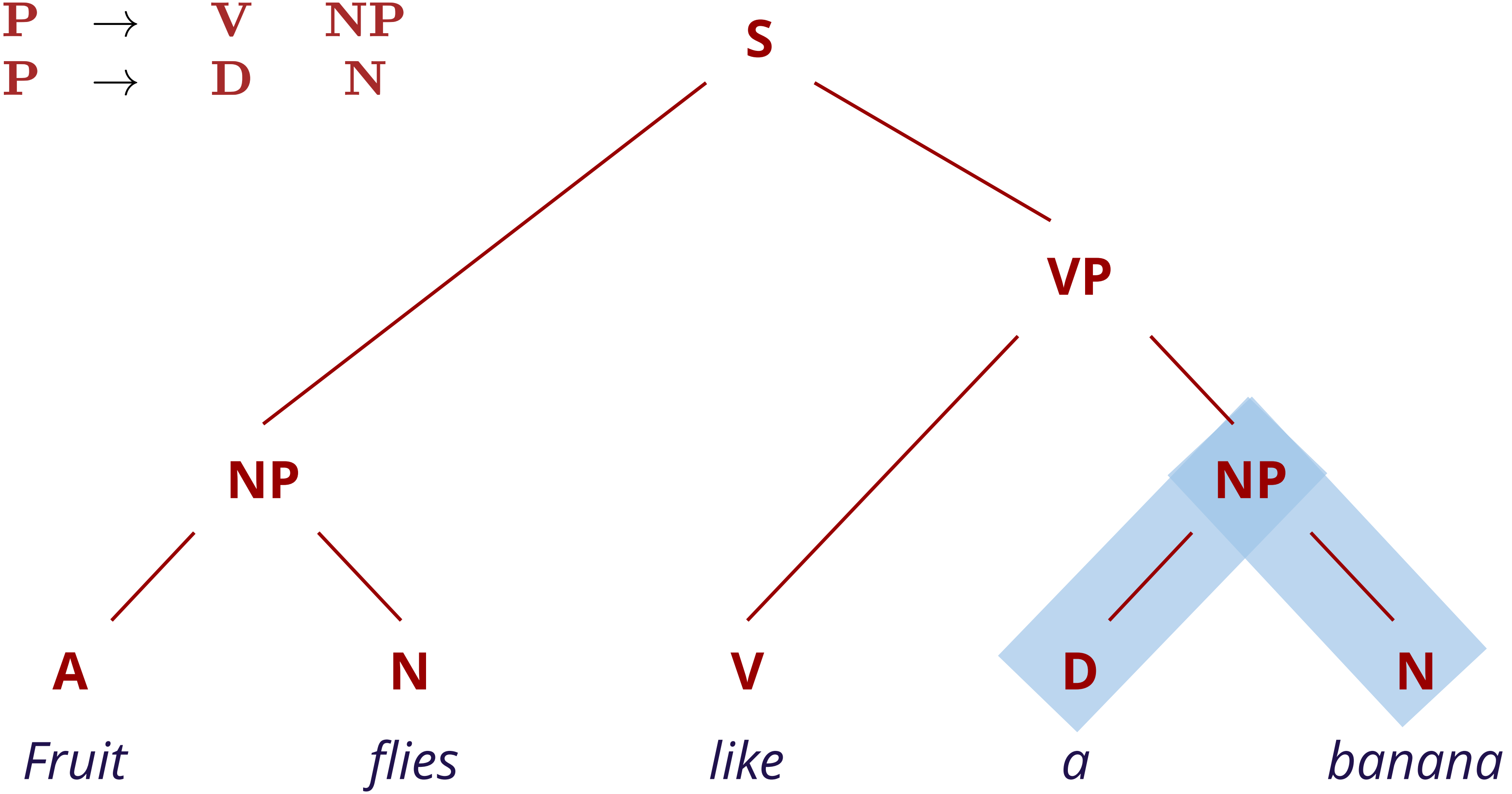
Grammar

$$\begin{array}{cccc} \mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} \\ \mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} \\ \mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} \end{array}$$



Grammar

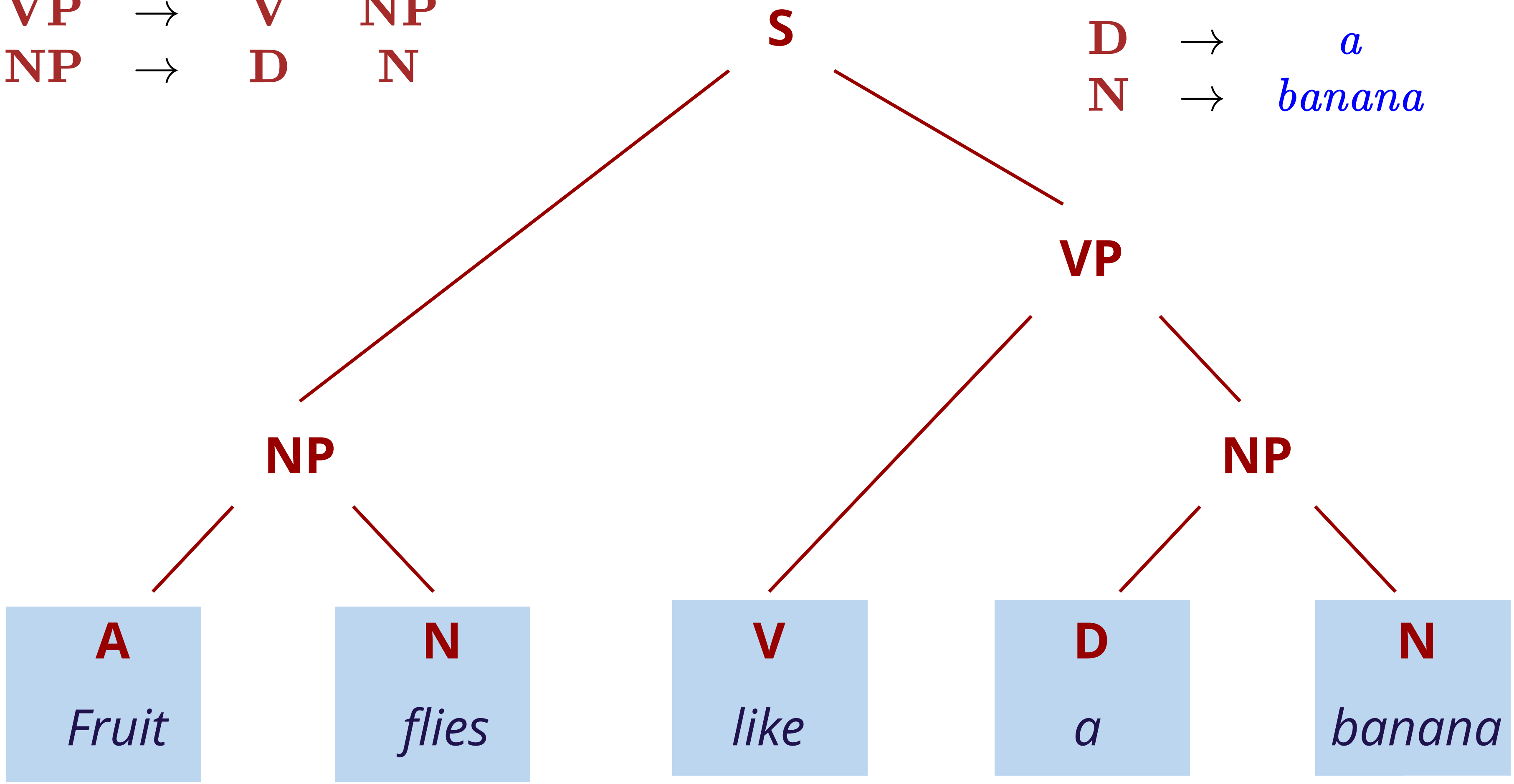
$$
\begin{array}{llll}
\mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} \\
\mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} \\
\mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} \\
\mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N}
\end{array}
$$



Grammar

S	$\rightarrow$	NP	VP
NP	$\rightarrow$	A	N
VP	$\rightarrow$	V	NP
NP	$\rightarrow$	D	N

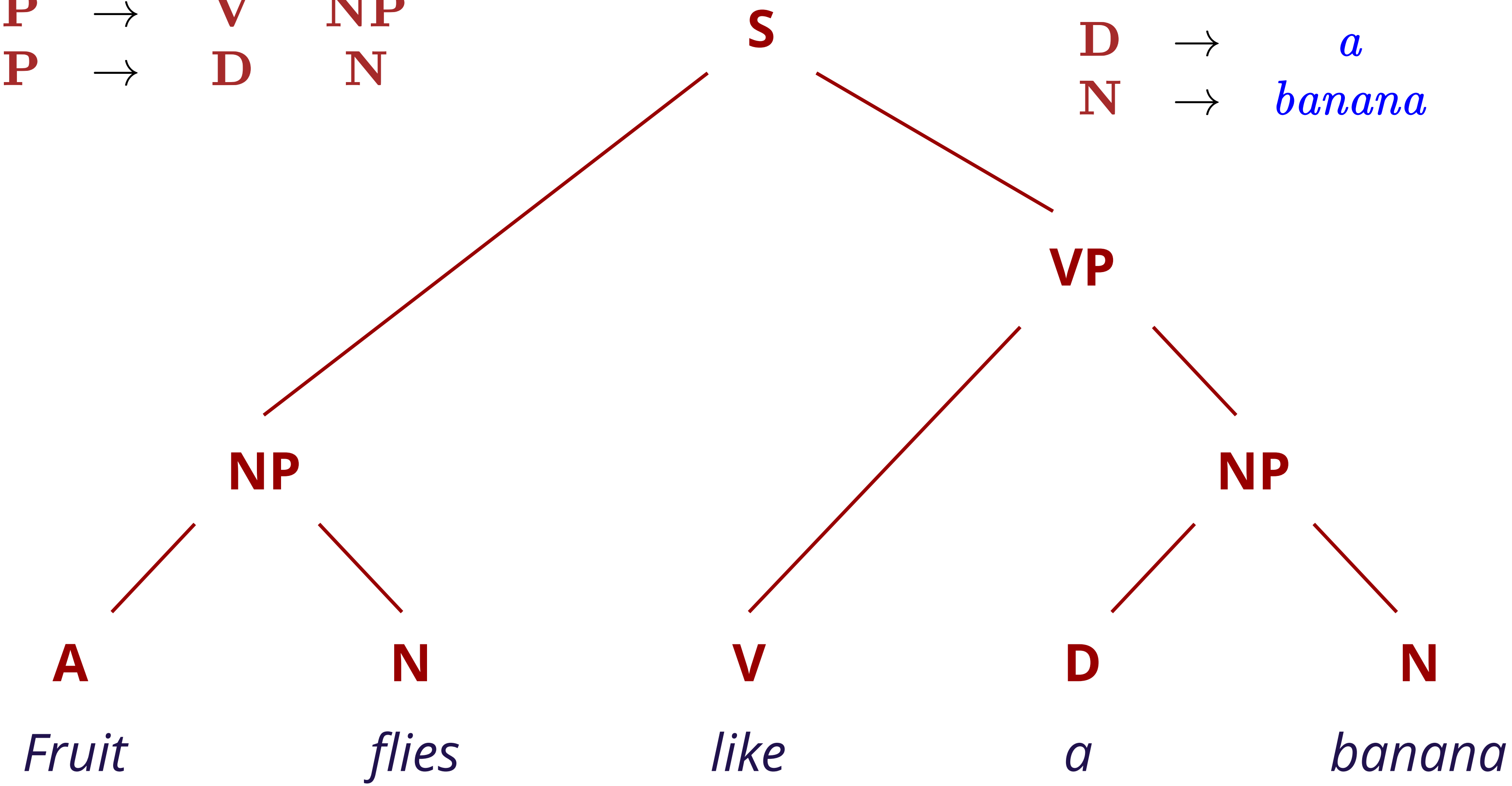
A	$\rightarrow$	*Fruit*
N	$\rightarrow$	*flies*
V	$\rightarrow$	*like*
D	$\rightarrow$	*a*
N	$\rightarrow$	*banana*



Context Free Grammar

$$
\begin{array}{lcll}
\mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} \\
\mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} \\
\mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} \\
\mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N}
\end{array}
$$

$$
\begin{array}{lcl}
\mathbf{A} & \rightarrow & \mathit{Fruit} \\
\mathbf{N} & \rightarrow & \mathit{flies} \\
\mathbf{V} & \rightarrow & \mathit{like} \\
\mathbf{D} & \rightarrow & \mathit{a} \\
\mathbf{N} & \rightarrow & \mathit{banana}
\end{array}
$$



Context Free Grammar

$$
\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{NP}\ \mathbf{VP} \\
\mathbf{NP} &\rightarrow \mathbf{A}\ \mathbf{N} \\
\mathbf{VP} &\rightarrow \mathbf{V}\ \mathbf{NP} \\
\mathbf{NP} &\rightarrow \mathbf{D}\ \mathbf{N}
\end{aligned}
\qquad
\begin{aligned}
\mathbf{A} &\rightarrow \textit{Fruit} \\
\mathbf{N} &\rightarrow \textit{flies} \\
\mathbf{V} &\rightarrow \textit{like} \\
\mathbf{D} &\rightarrow \textit{a} \\
\mathbf{N} &\rightarrow \textit{banana}
\end{aligned}
$$

Non-terminal Symbols

Context Free Grammar

$$\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{NP}\ \mathbf{VP} \\
\mathbf{NP} &\rightarrow \mathbf{A}\ \mathbf{N} \\
\mathbf{VP} &\rightarrow \mathbf{V}\ \mathbf{NP} \\
\mathbf{NP} &\rightarrow \mathbf{D}\ \mathbf{N}
\end{aligned}$$

$$\begin{aligned}
\mathbf{A} &\rightarrow \boxed{\textit{Fruit}} \\
\mathbf{N} &\rightarrow \textit{flies} \\
\mathbf{V} &\rightarrow \textit{like} \\
\mathbf{D} &\rightarrow \textit{a} \\
\mathbf{N} &\rightarrow \textit{banana}
\end{aligned}$$

Terminal Symbols

Context Free Grammar

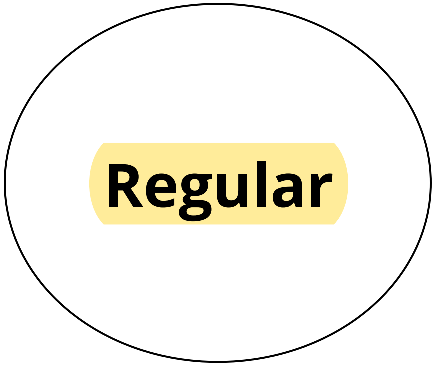
Grammar with rules of the form

$$\mathbf{A} \rightarrow \mathbf{exp}$$

A non-terminal symbol

A list of non-terminal and terminal symbols

Languages

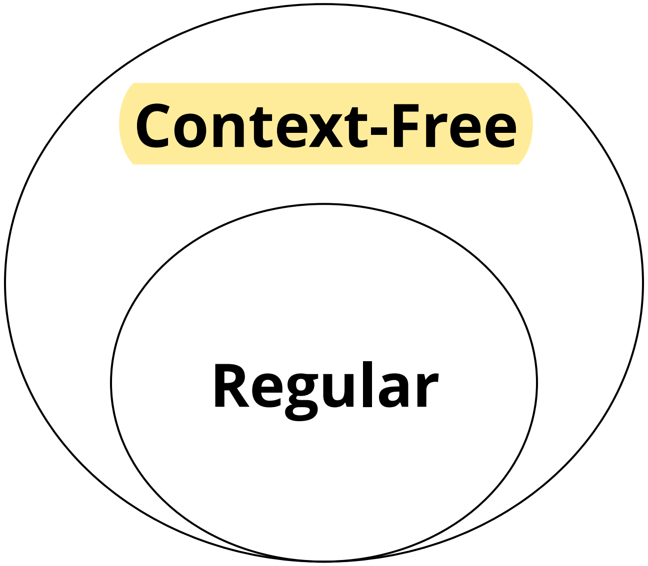


A terminal symbol

$$\mathbf{A} \rightarrow \alpha$$

$$\mathbf{A} \rightarrow \alpha \; \mathbf{B}$$

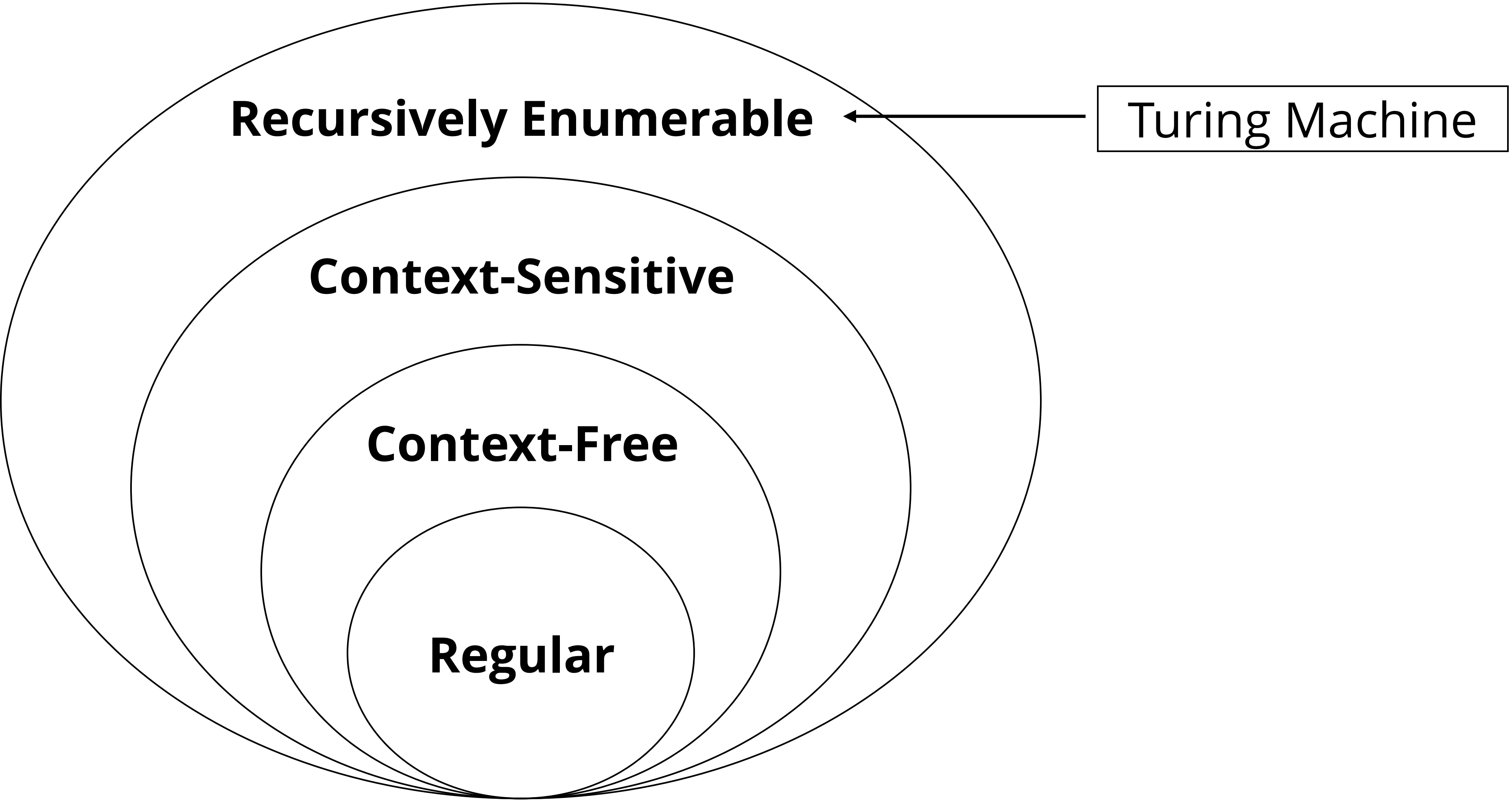
Languages



A list of non-terminal and terminal symbols

$$\mathbf{A} \rightarrow \mathbf{exp}$$

Languages



Recursively Enumerable
Context-Sensitive
Context-Free
Regular
Turing Machine

Chomsky Normal Form

A CFG is said to be in Chomsky normal form (CNF) if all its production rules are of either of the following forms:

$$\mathbf{A} \rightarrow \mathbf{B} \quad \mathbf{C}$$
$$\mathbf{A} \rightarrow \alpha$$

Some textbooks also consider epsilon rules

$$\begin{array}{lcll} \mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} \\ \mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} \\ \mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} \\ \mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N} \end{array}$$

$$\begin{array}{lcl} \mathbf{A} & \rightarrow & \mathit{Fruit} \\ \mathbf{N} & \rightarrow & \mathit{flies} \\ \mathbf{V} & \rightarrow & \mathit{like} \\ \mathbf{D} & \rightarrow & \mathit{a} \\ \mathbf{N} & \rightarrow & \mathit{banana} \end{array}$$

Chomsky Normal Form

A CFG is said to be in Chomsky normal form (CNF) if all its production rules are of either of the following forms:

$$\mathbf{A} \rightarrow \mathbf{B} \quad \mathbf{C}$$
$$\mathbf{A} \rightarrow \alpha$$

Parsing with a CFG in CNF will be convenient.

$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP}$$
$$\mathbf{NP} \rightarrow \mathbf{A} \quad \mathbf{N}$$
$$\mathbf{VP} \rightarrow \mathbf{V} \quad \mathbf{NP}$$
$$\mathbf{NP} \rightarrow \mathbf{D} \quad \mathbf{N}$$

$$\mathbf{A} \rightarrow \mathit{Fruit}$$
$$\mathbf{N} \rightarrow \mathit{flies}$$
$$\mathbf{V} \rightarrow \mathit{like}$$
$$\mathbf{D} \rightarrow \mathit{a}$$
$$\mathbf{N} \rightarrow \mathit{banana}$$

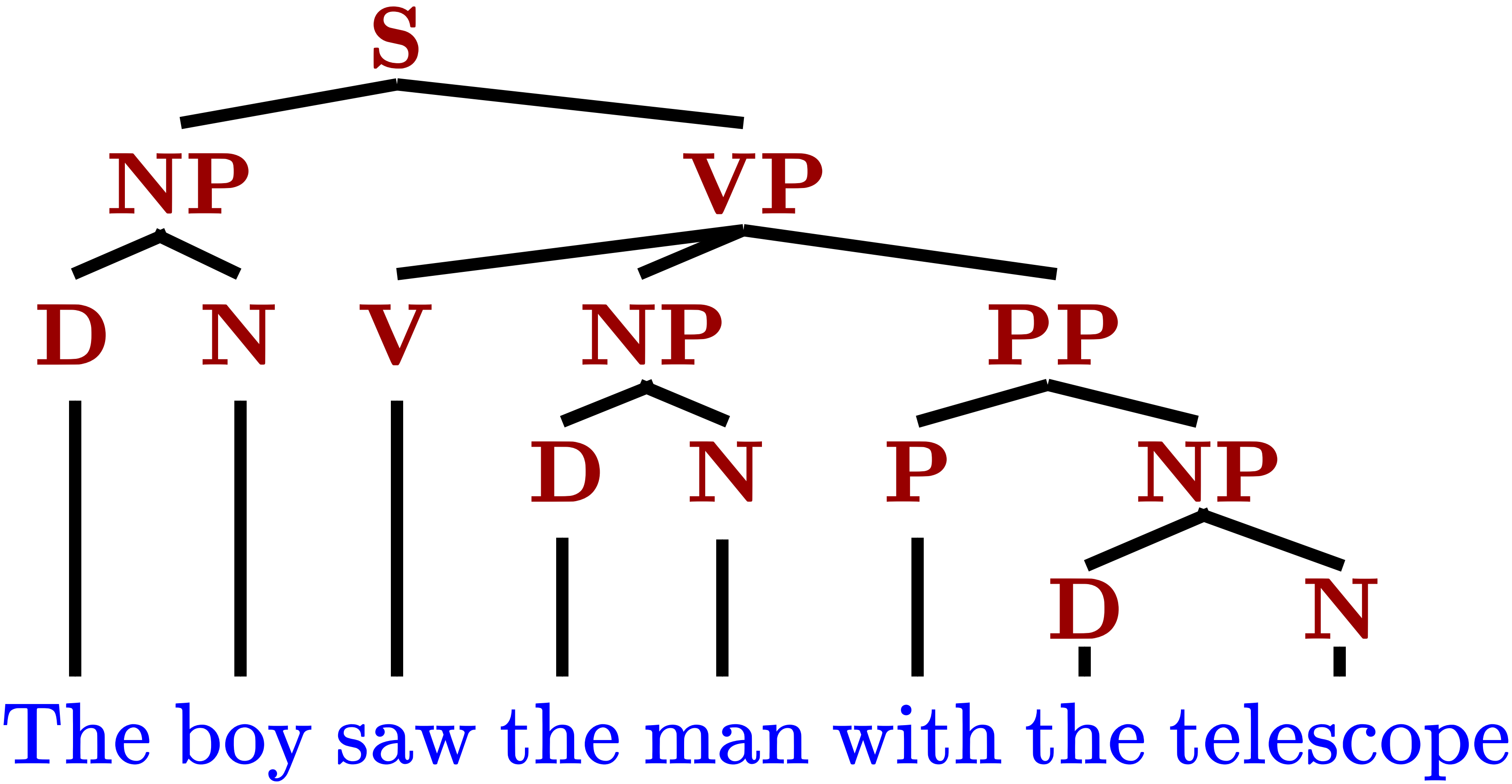
Parsing with CFG

The boy saw the man with the telescope

Using the telescope, the boy saw the man

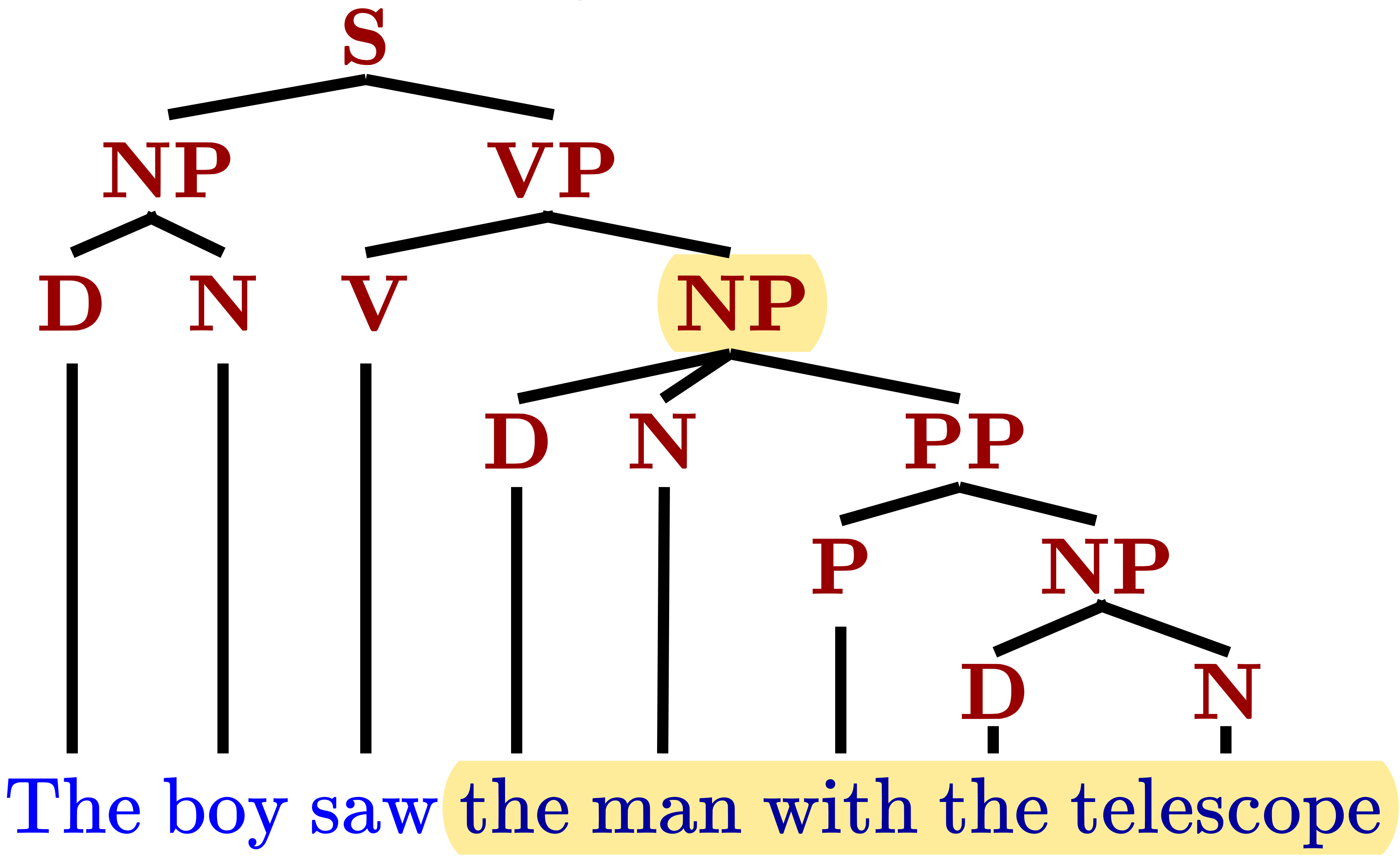
The boy saw the man. The man had a telescope

Parsing with CFG



Using the telescope, the boy saw the man

Parsing with CFG



The boy saw the man. The man had a telescope

Question

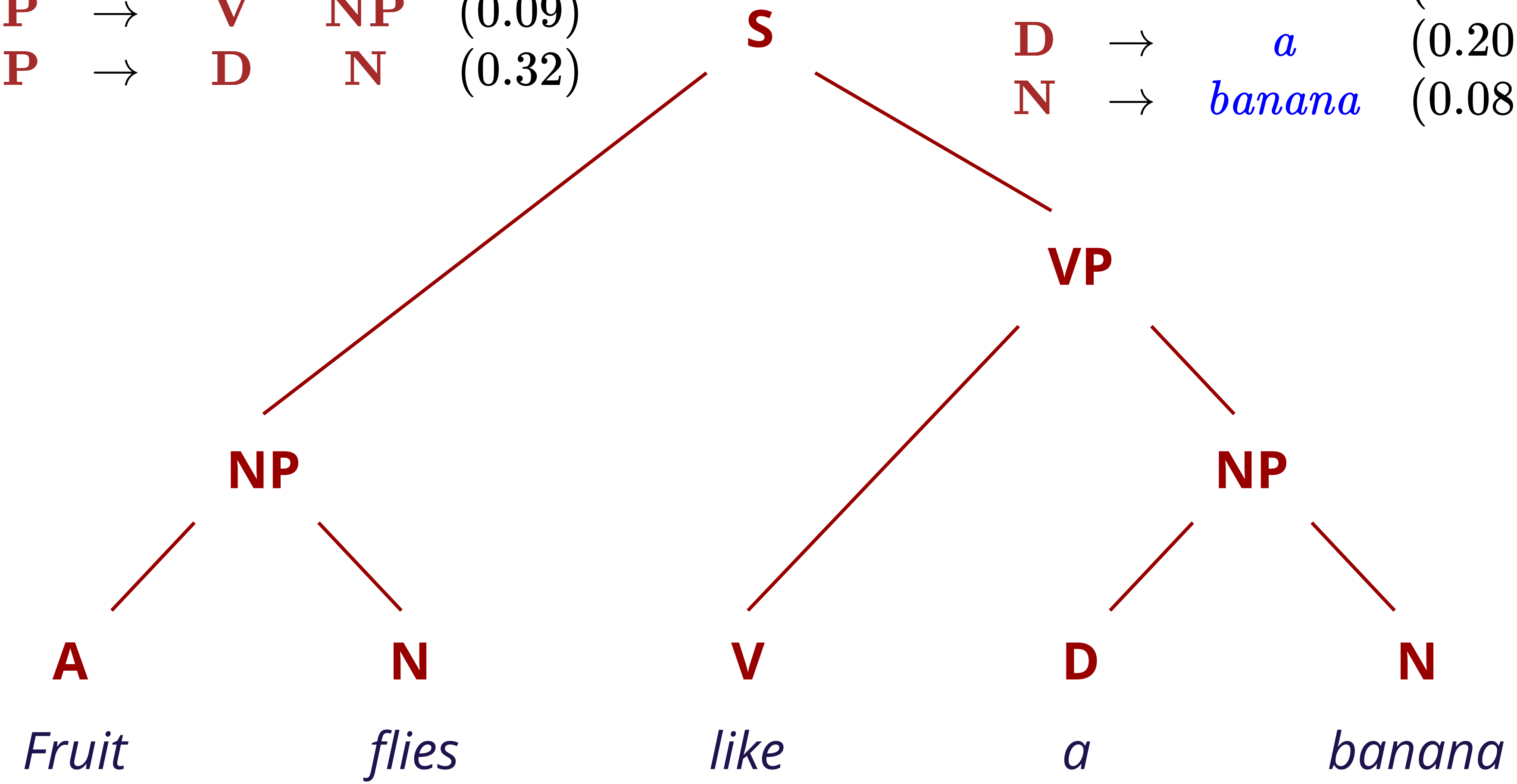
How to know which parse is better?

We need to assign scores to such trees.

Probabilistic CFG

$$
\begin{array}{llllr}
\mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} & (0.12) \\
\mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} & (0.23) \\
\mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} & (0.09) \\
\mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N} & (0.32)
\end{array}
$$

$$
\begin{array}{lllr}
\mathbf{A} & \rightarrow & \mathit{Fruit} & (0.221) \\
\mathbf{N} & \rightarrow & \mathit{flies} & (0.206) \\
\mathbf{V} & \rightarrow & \mathit{like} & (0.108) \\
\mathbf{D} & \rightarrow & \mathit{a} & (0.209) \\
\mathbf{N} & \rightarrow & \mathit{banana} & (0.089)
\end{array}
$$

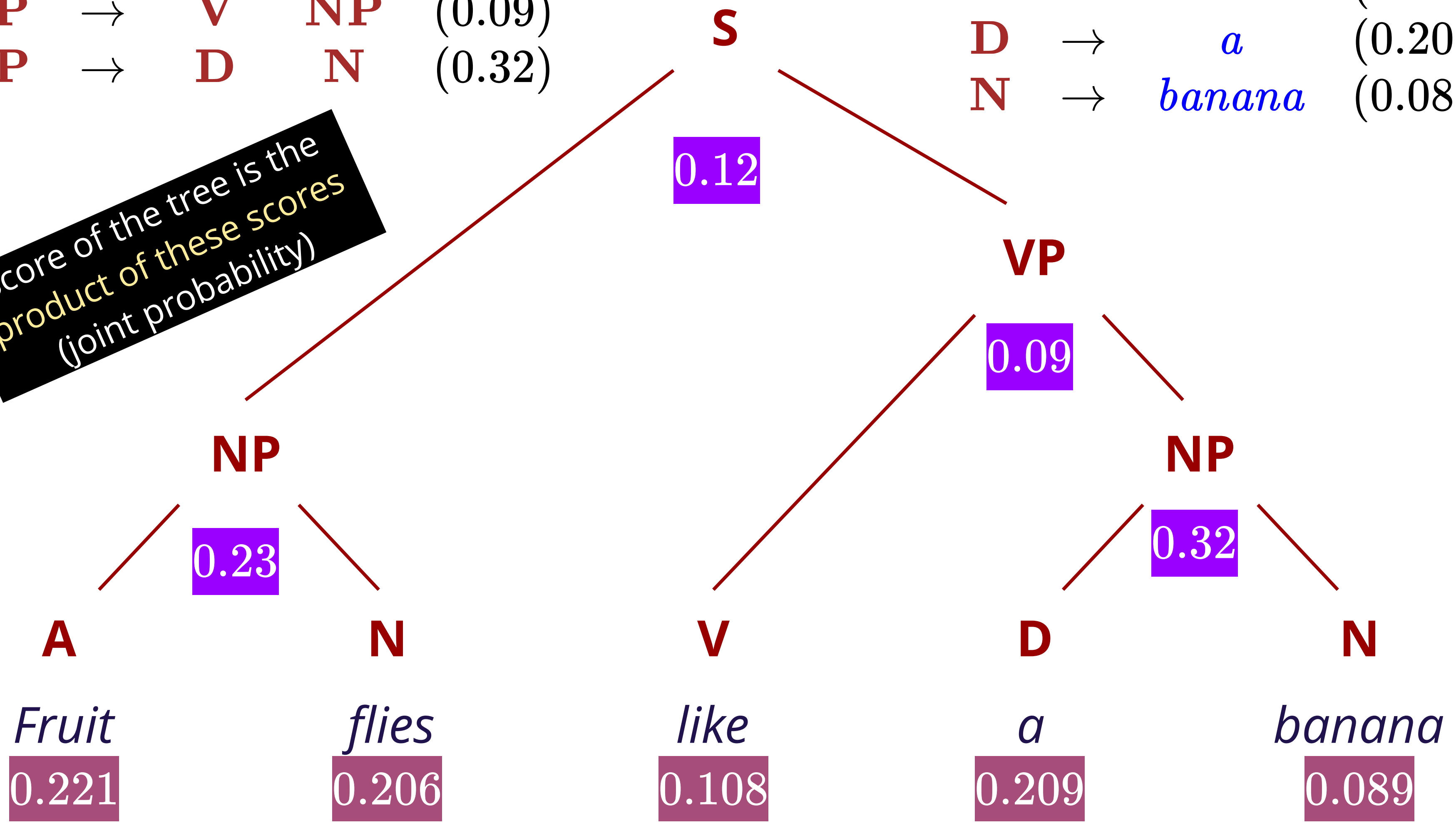


Probabilistic CFG

$$
\begin{array}{llll}
\mathbf{S} & \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} & (0.12) \\
\mathbf{NP} & \rightarrow \quad \mathbf{A} \quad \mathbf{N} & (0.23) \\
\mathbf{VP} & \rightarrow \quad \mathbf{V} \quad \mathbf{NP} & (0.09) \\
\mathbf{NP} & \rightarrow \quad \mathbf{D} \quad \mathbf{N} & (0.32)
\end{array}
$$

$$
\begin{array}{lll}
\mathbf{A} & \rightarrow \quad \textit{Fruit} & (0.221) \\
\mathbf{N} & \rightarrow \quad \textit{flies} & (0.206) \\
\mathbf{V} & \rightarrow \quad \textit{like} & (0.108) \\
\mathbf{D} & \rightarrow \quad \textit{a} & (0.209) \\
\mathbf{N} & \rightarrow \quad \textit{banana} & (0.089)
\end{array}
$$

Score of the tree is the product of these scores (joint probability)



PCFG Learning

Where do we get the scores for the rules?

Given	Find
A collection of sentences annotated with their parse trees.	The probability for each rule that appears in these parse trees.

How do we learn such probabilities?

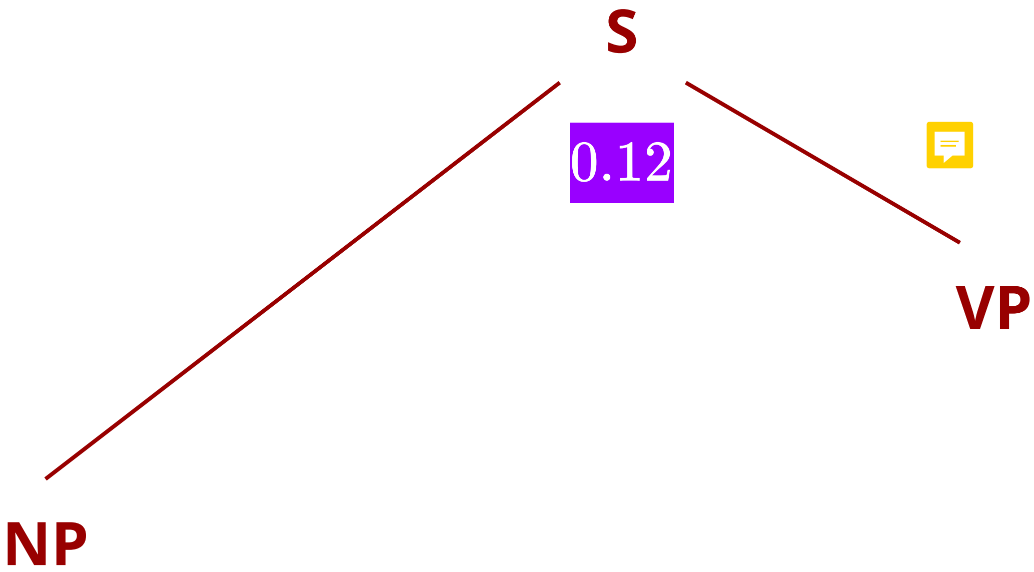
How is each tree generated?

Generative Model

S

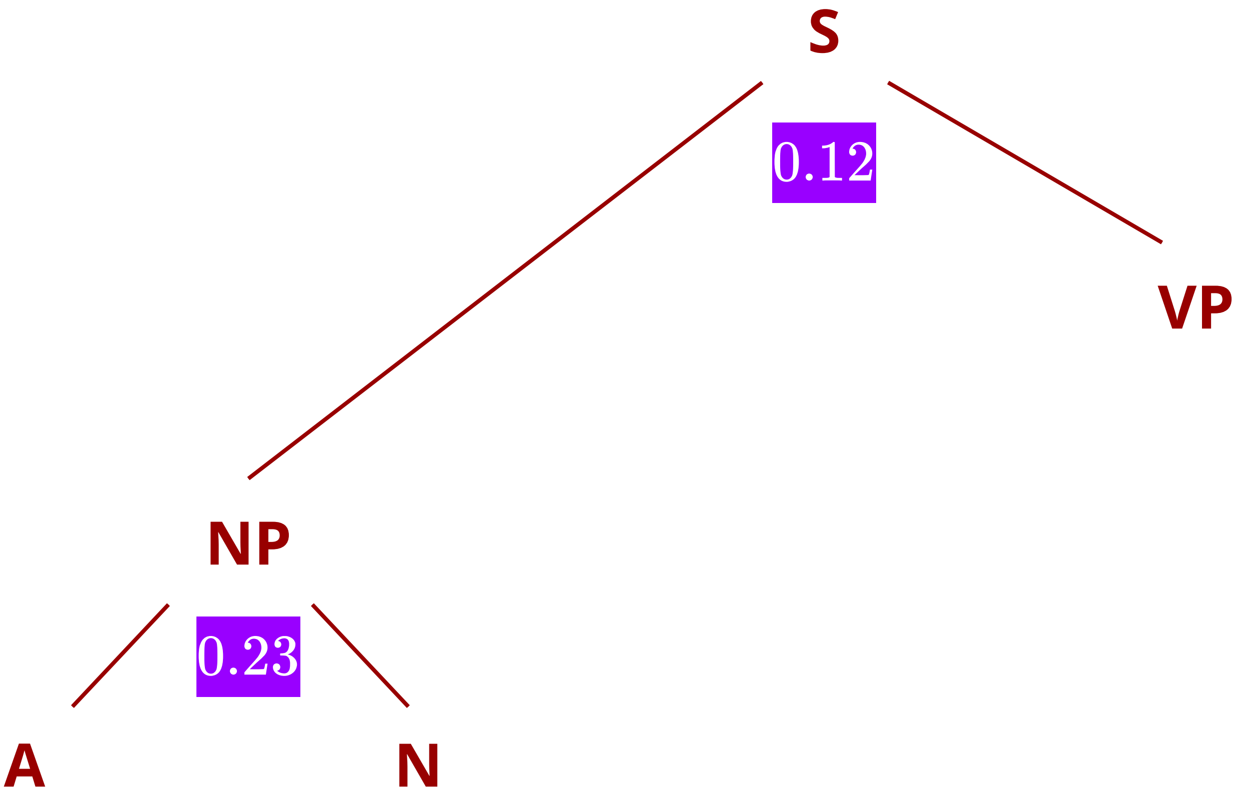
$$p(tree) = \ldots$$

Generative Model



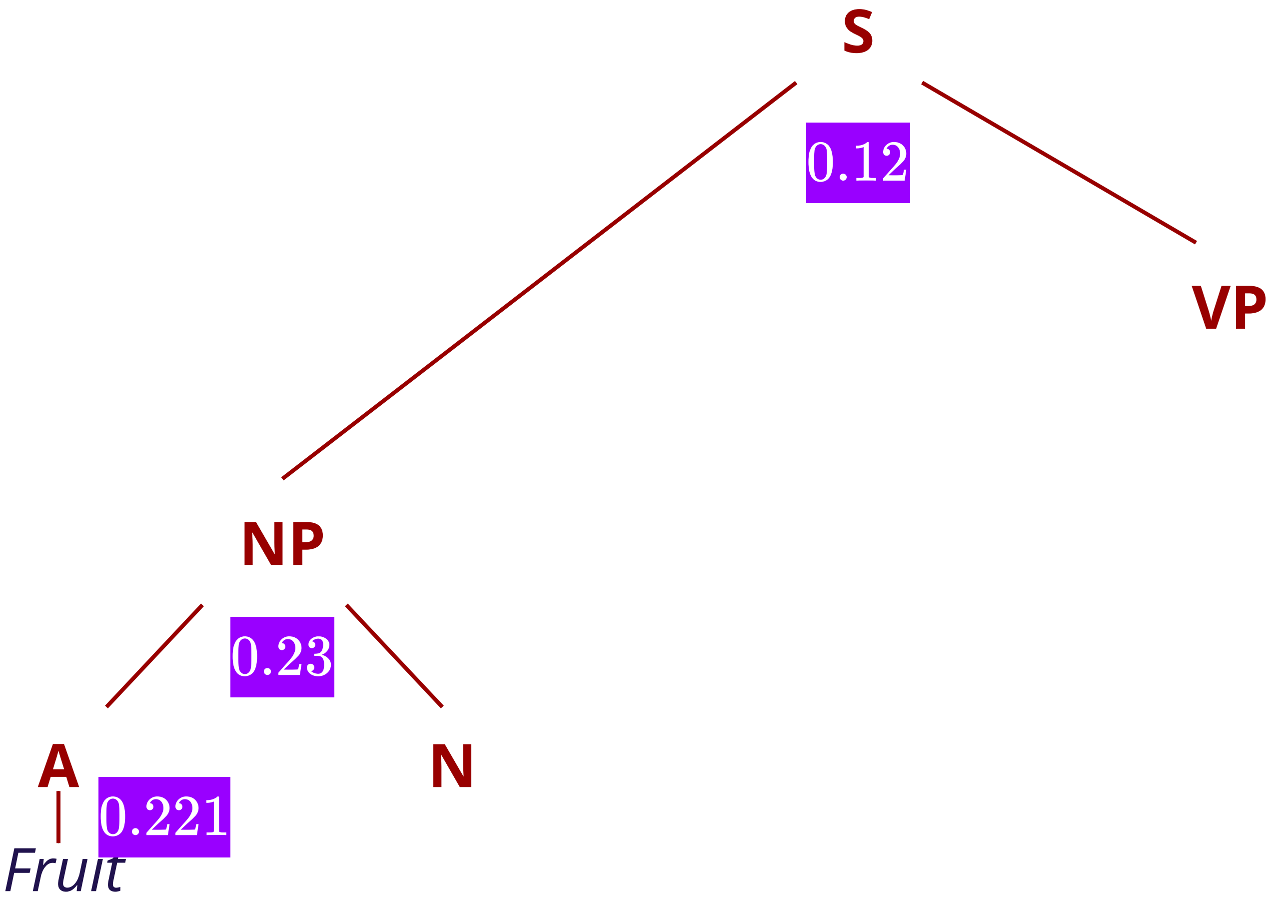
$$p(tree) = 0.12 \times \ldots$$

Generative Model



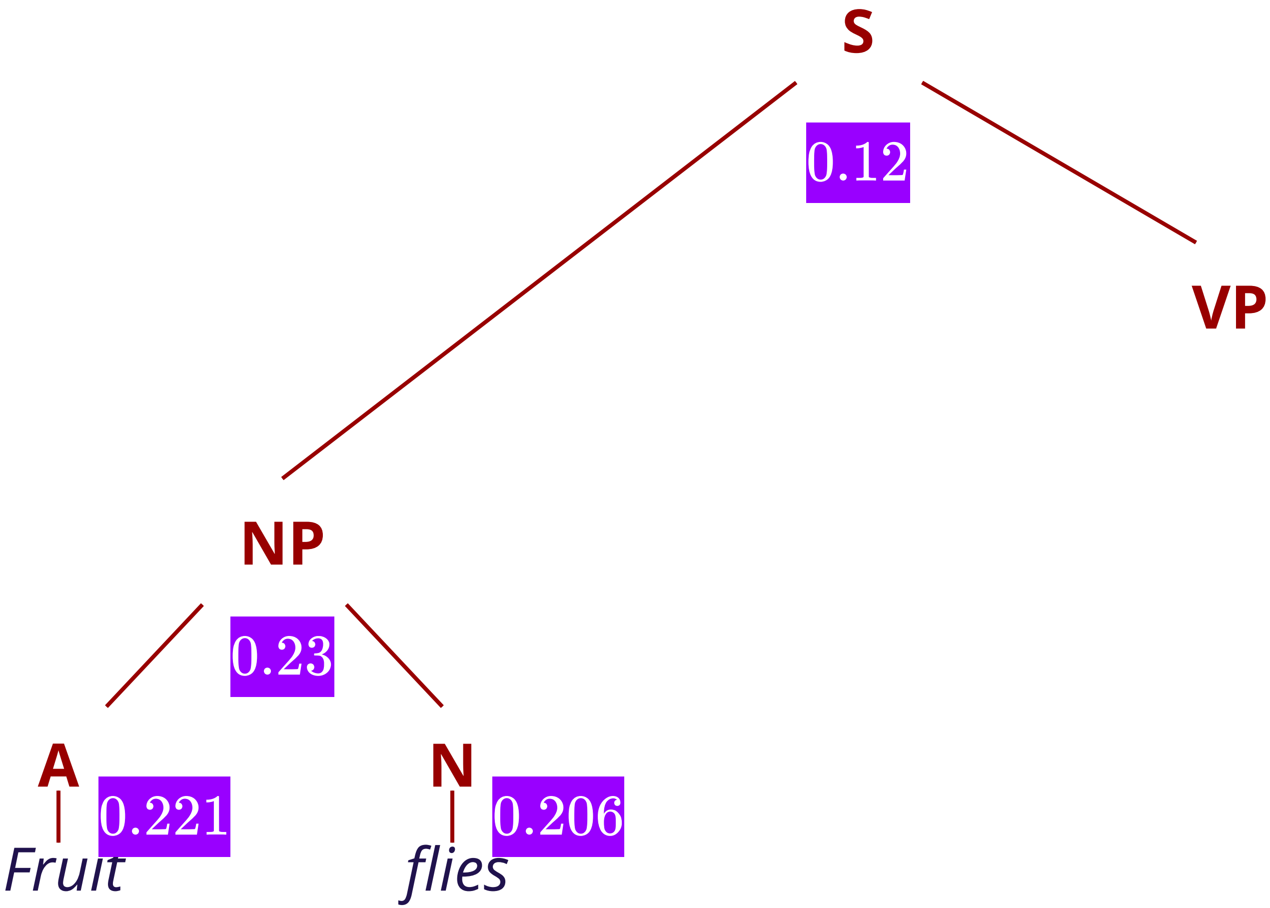
$$p(tree) = 0.12 \times 0.23 \times \ldots$$

Generative Model



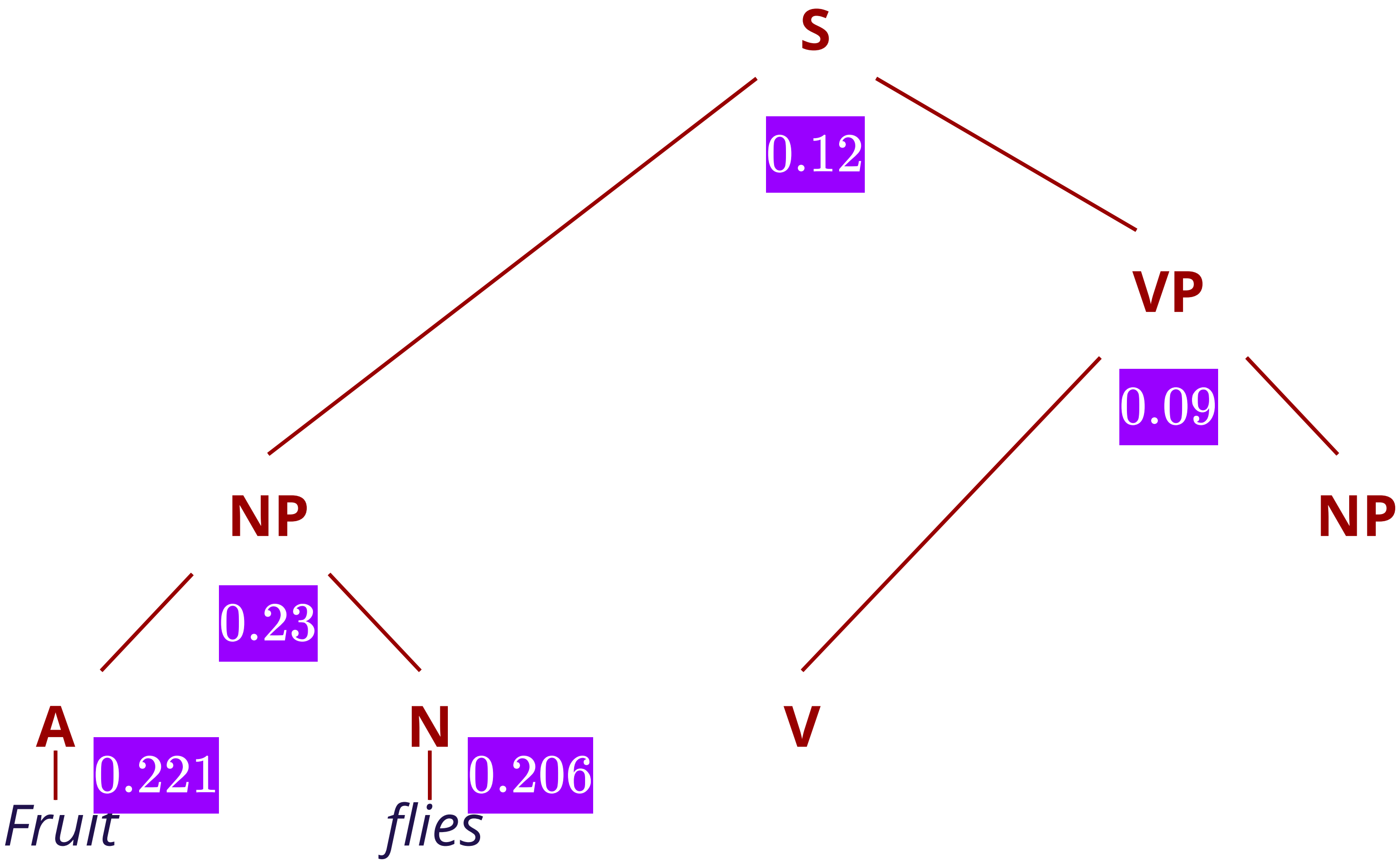
$$p(tree) = 0.12 \times 0.23 \times 0.221 \times \ldots$$

Generative Model



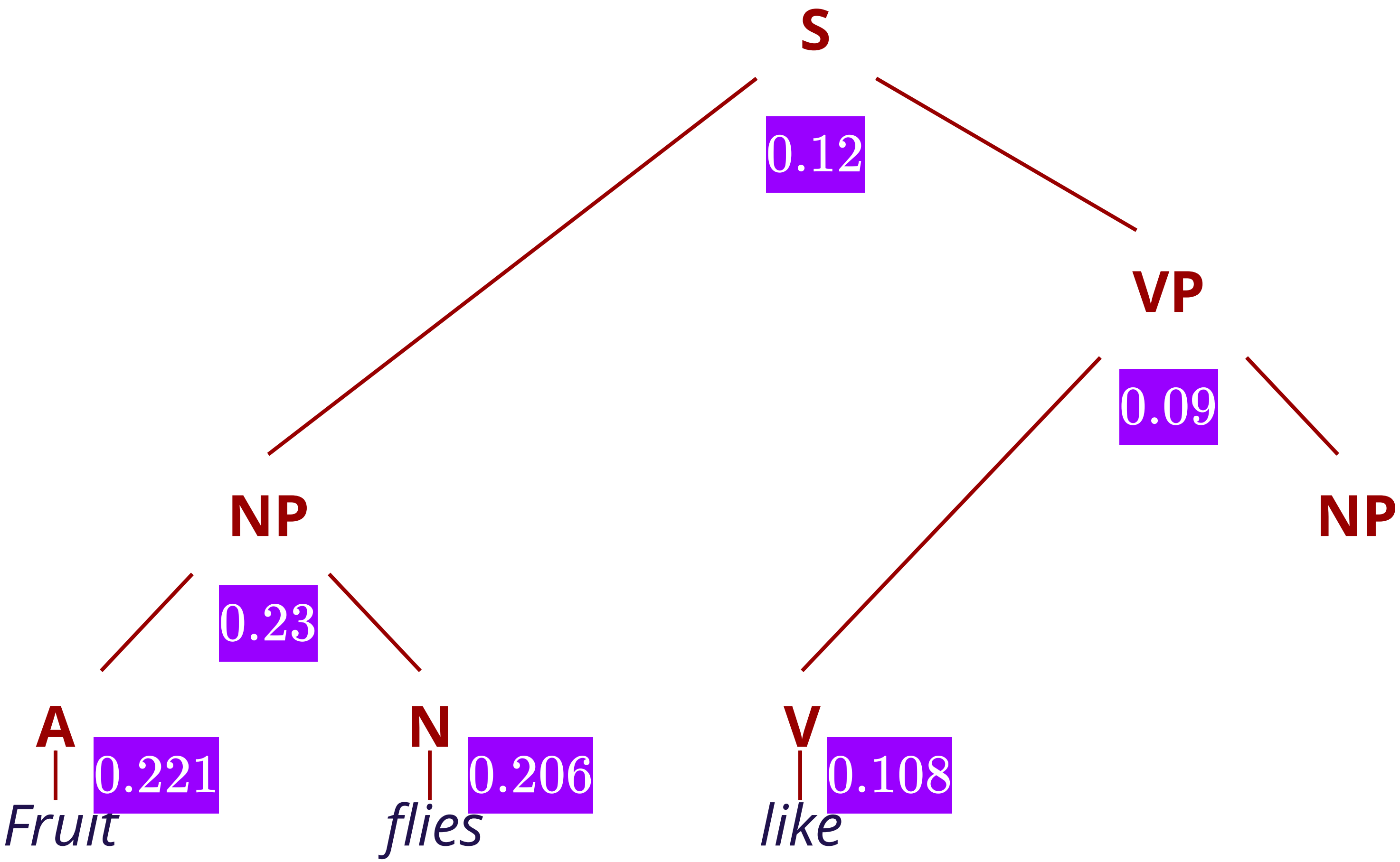
$$p(tree) = 0.12 \times 0.23 \times 0.221 \times 0.206 \times \ldots$$

Generative Model



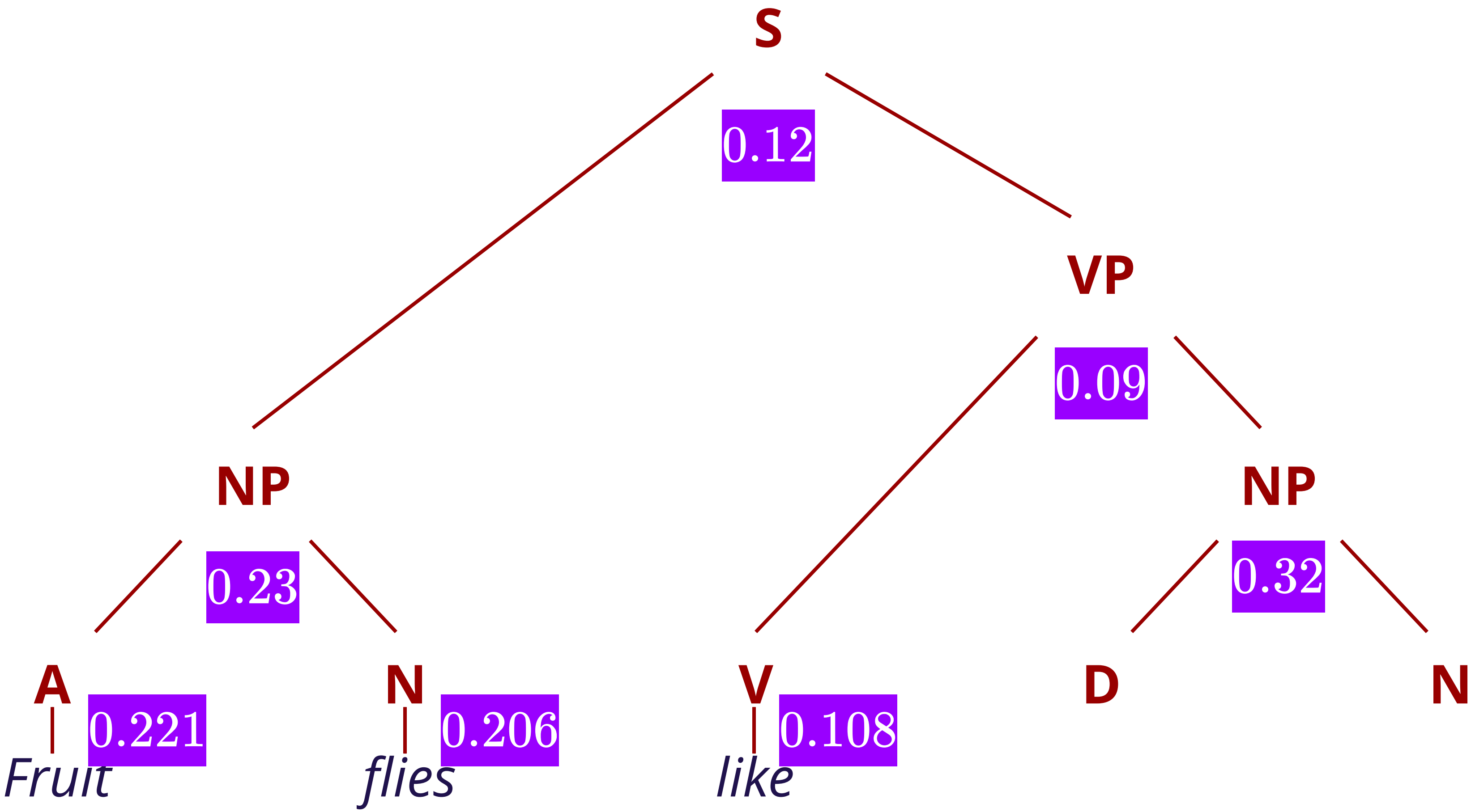
$$p(tree) = 0.12 \times 0.23 \times 0.221 \times 0.206 \times 0.09 \times \ldots$$

Generative Model



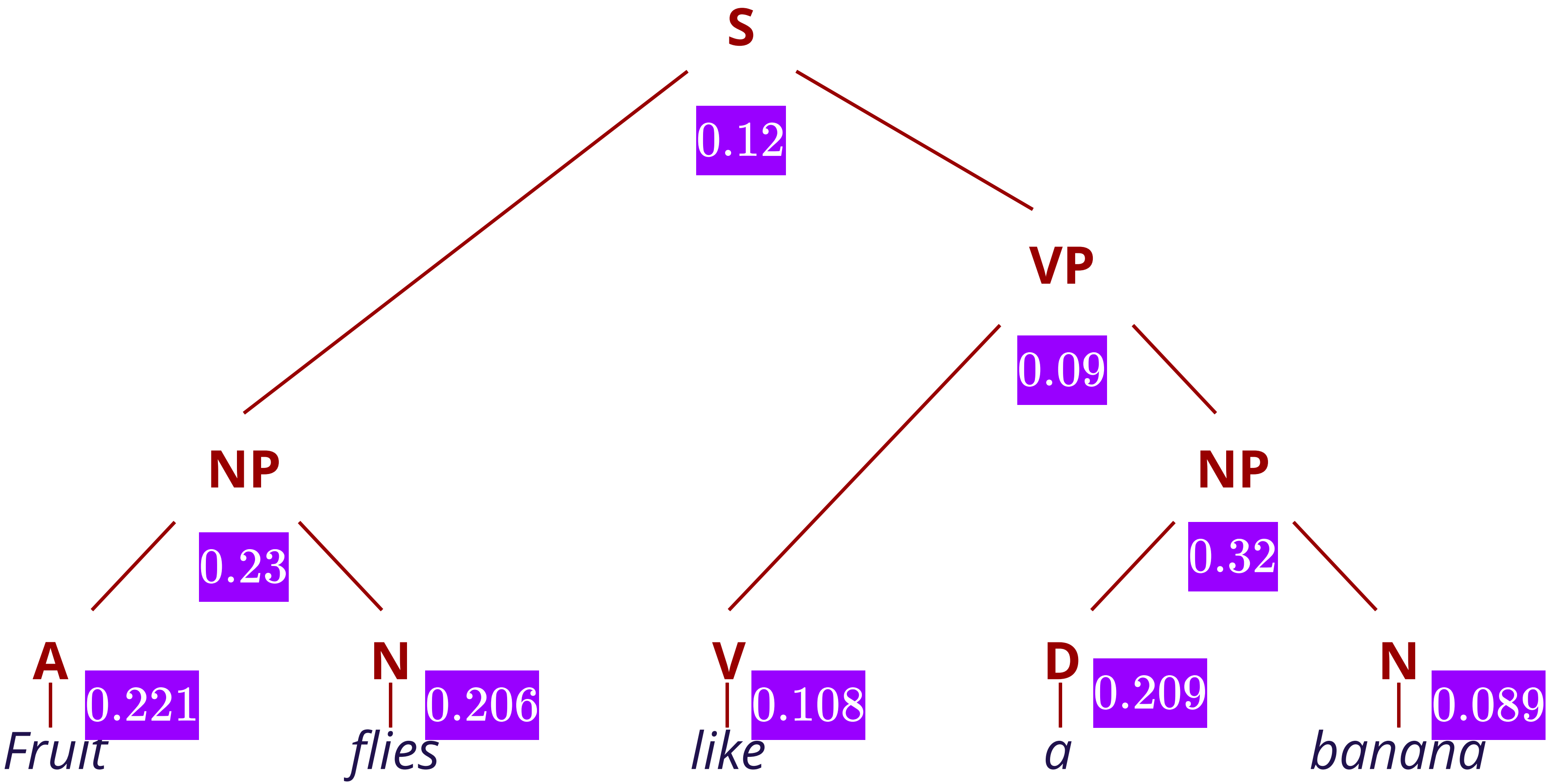
$$p(tree) = 0.12 \times 0.23 \times 0.221 \times 0.206 \times 0.09 \times 0.108 \times \ldots$$

Generative Model



$$p(tree) = 0.12 \times 0.23 \times 0.221 \times 0.206 \times 0.09 \times 0.108 \times 0.32 \times \ldots$$

Generative Model

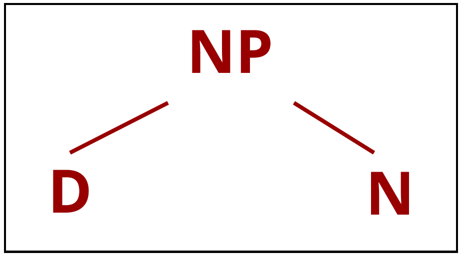


$$p(tree) = 0.12 \times 0.23 \times 0.221 \times 0.206 \times 0.09 \times 0.108 \times 0.32 \times 0.209 \times 0.089$$

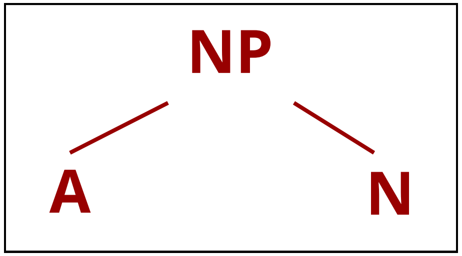
Question

How to learn the probability for each PCFG rule based on the generative process?

PCFG Learning



This rule appears 54 times in the training set



This rule appears 37 times in the training set

What is the probability for the rule $\mathbf{NP} \rightarrow \mathbf{A\ N}$?

$$\frac{37}{37+54}$$

PCFG Learning

Maximum Likelihood Estimator!!

The number of times we see the rule $\mathbf{A} \to \mathbf{exp}$ in the training set.

$$p(\mathbf{A} \to \mathbf{exp}) = \frac{\text{count}(\mathbf{A} \to \mathbf{exp})}{\text{count}(\mathbf{A})}$$

The number of times we see the non-terminal $\mathbf{A}$ in the training set.

PCFG Decoding

Given	Find
A probabilistic CFG and an input sentence.	The most probable parse tree for the given sentence.

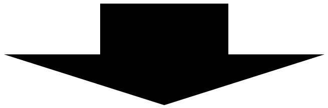
Hmm... seems we need a procedure similar to Viterbi, but works for PCFG?

Cocke–Kasami–Younger (CKY) Algorithm

$$
\begin{array}{llllr}
\mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} & (0.12) \\
\mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} & (0.23) \\
\mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} & (0.09) \\
\mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N} & (0.32)
\end{array}
$$

$$
\begin{array}{lllr}
\mathbf{A} & \rightarrow & \mathit{Fruit} & (0.221) \\
\mathbf{N} & \rightarrow & \mathit{flies} & (0.206) \\
\mathbf{V} & \rightarrow & \mathit{like} & (0.108) \\
\mathbf{D} & \rightarrow & \mathit{a} & (0.209) \\
\mathbf{N} & \rightarrow & \mathit{banana} & (0.089)
\end{array}
$$

Fruit *flies* *like* *a* *banana*

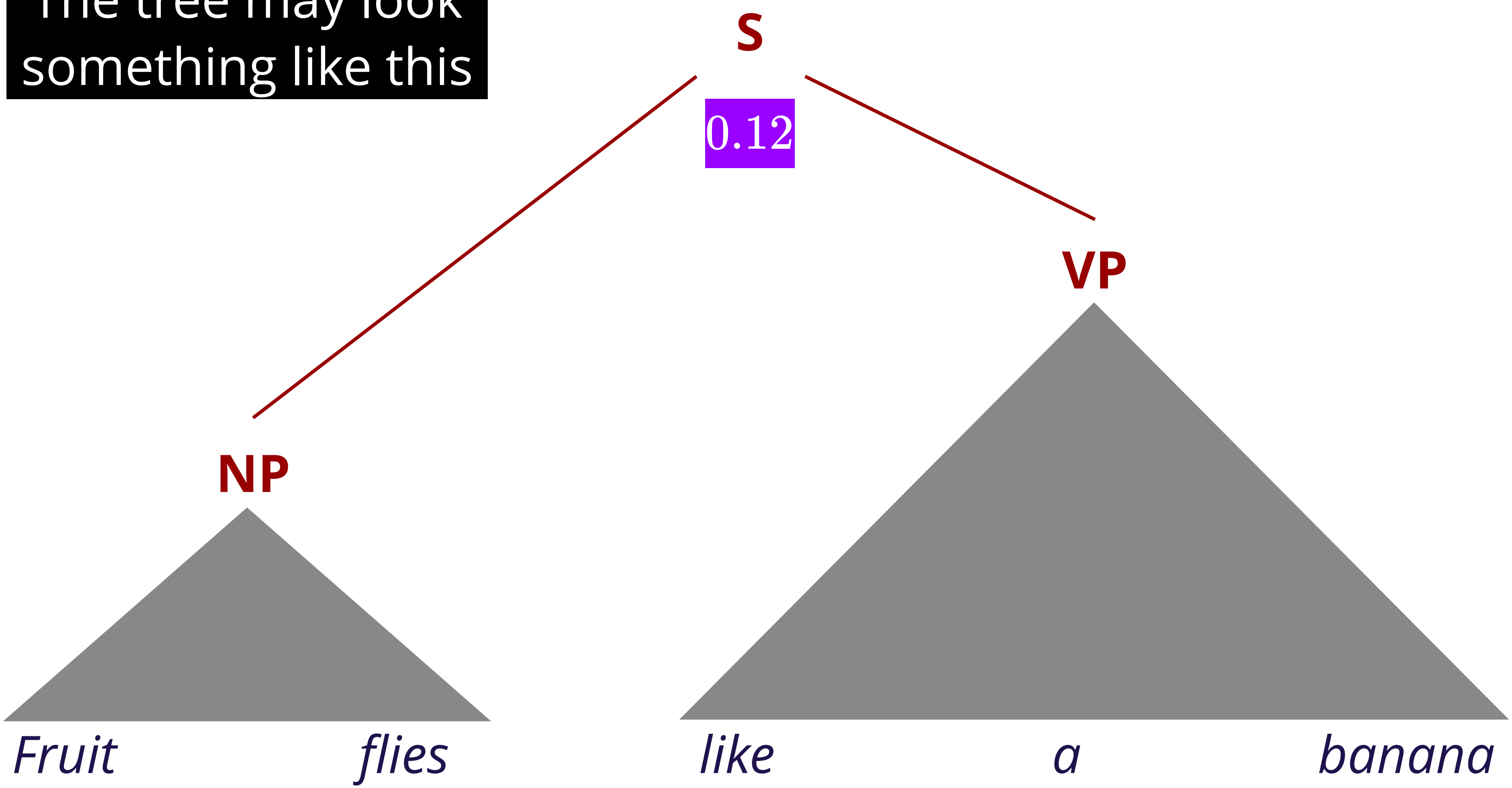


Parse Tree

CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

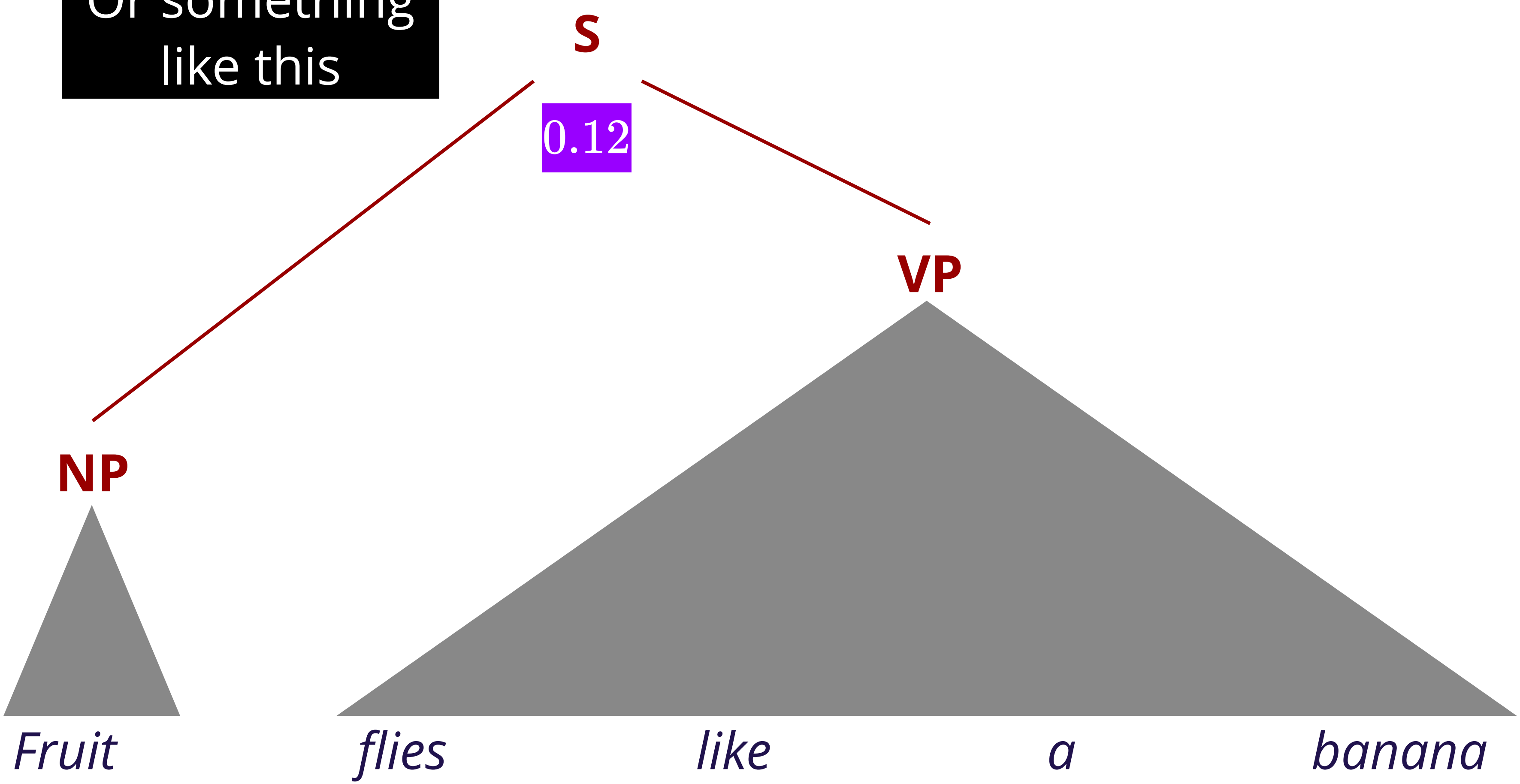
The tree may look something like this



CKY Algorithm

$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

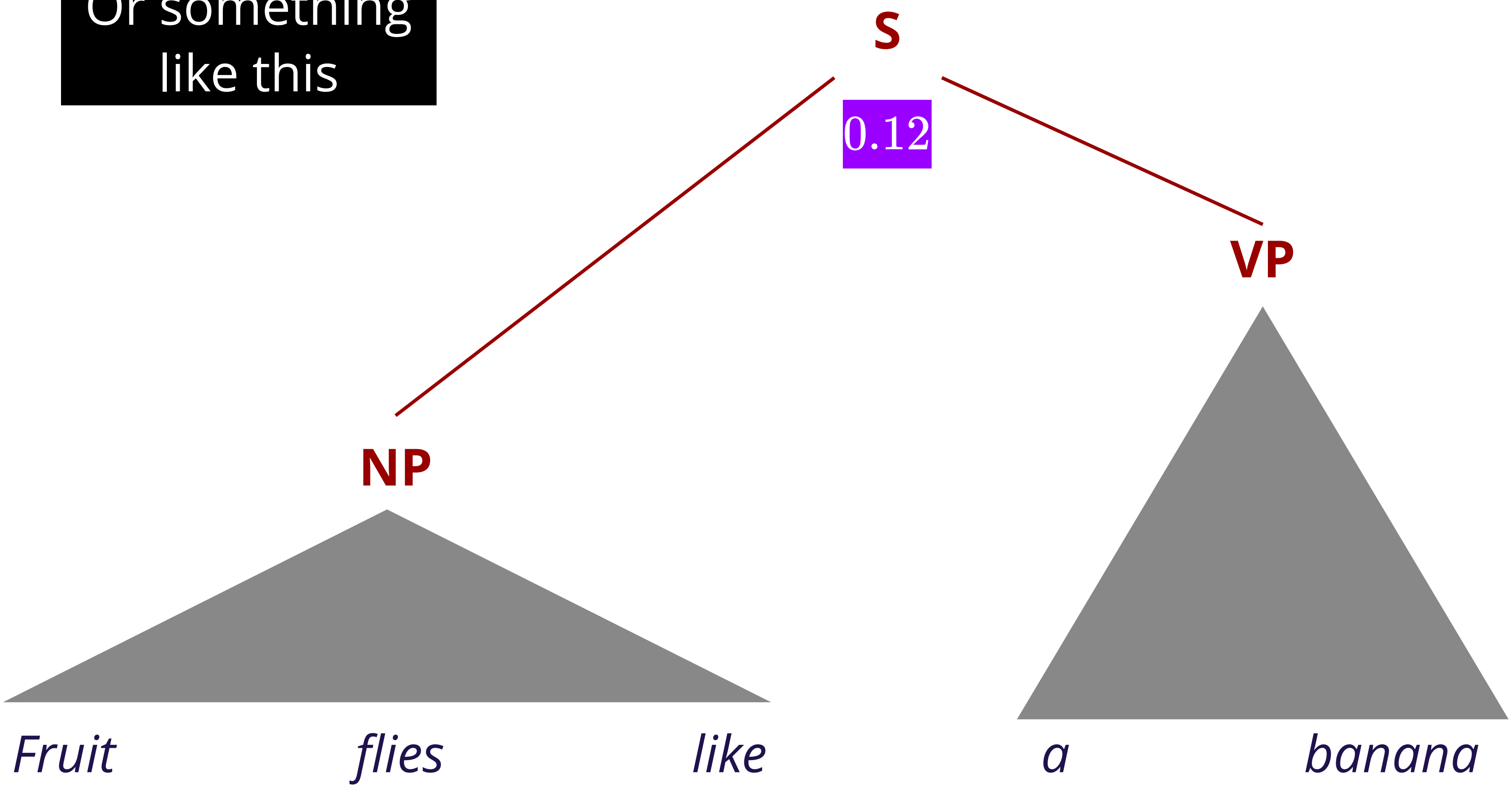
Or something like this



CKY Algorithm

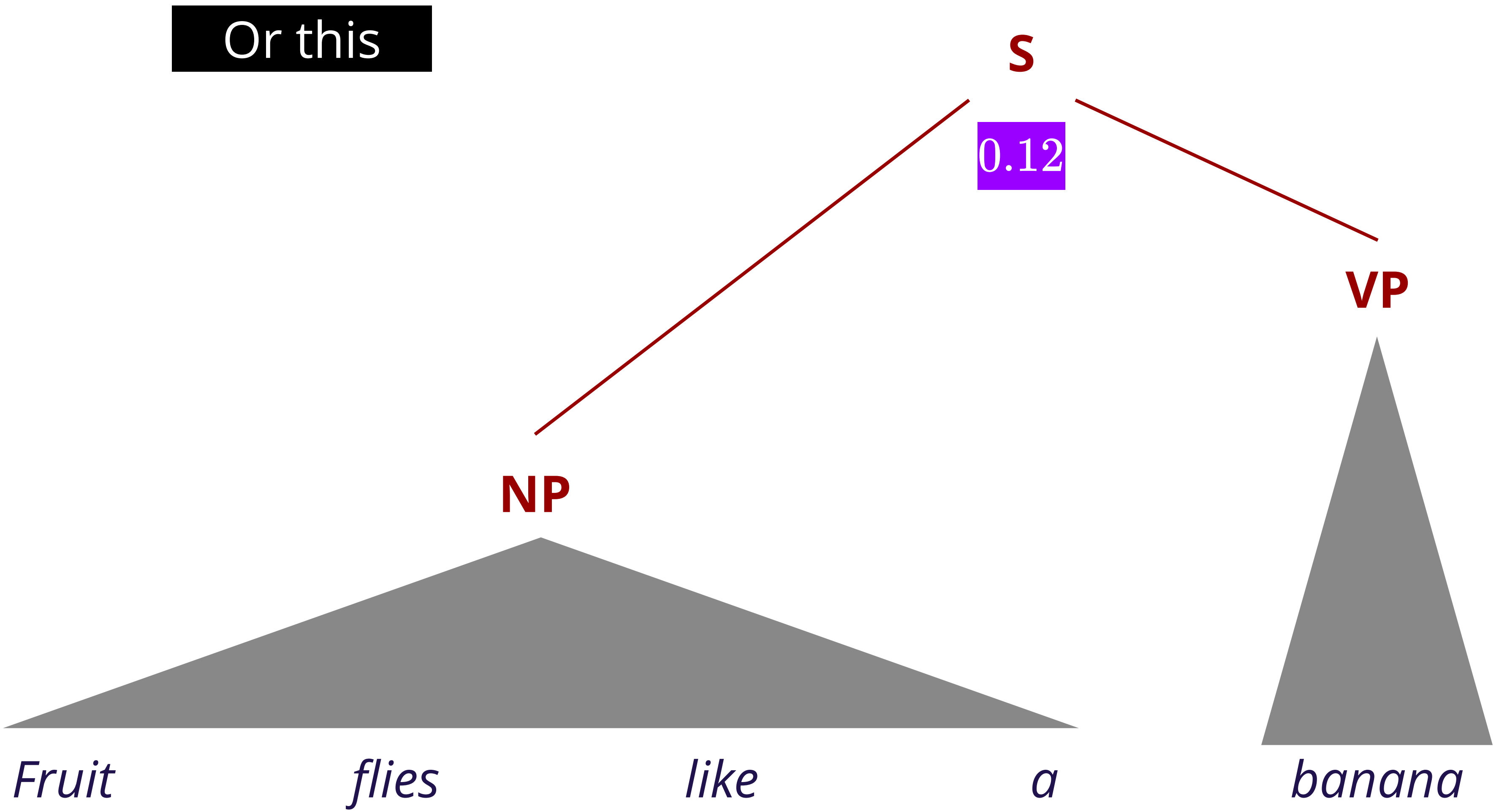
$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

Or something like this



CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$



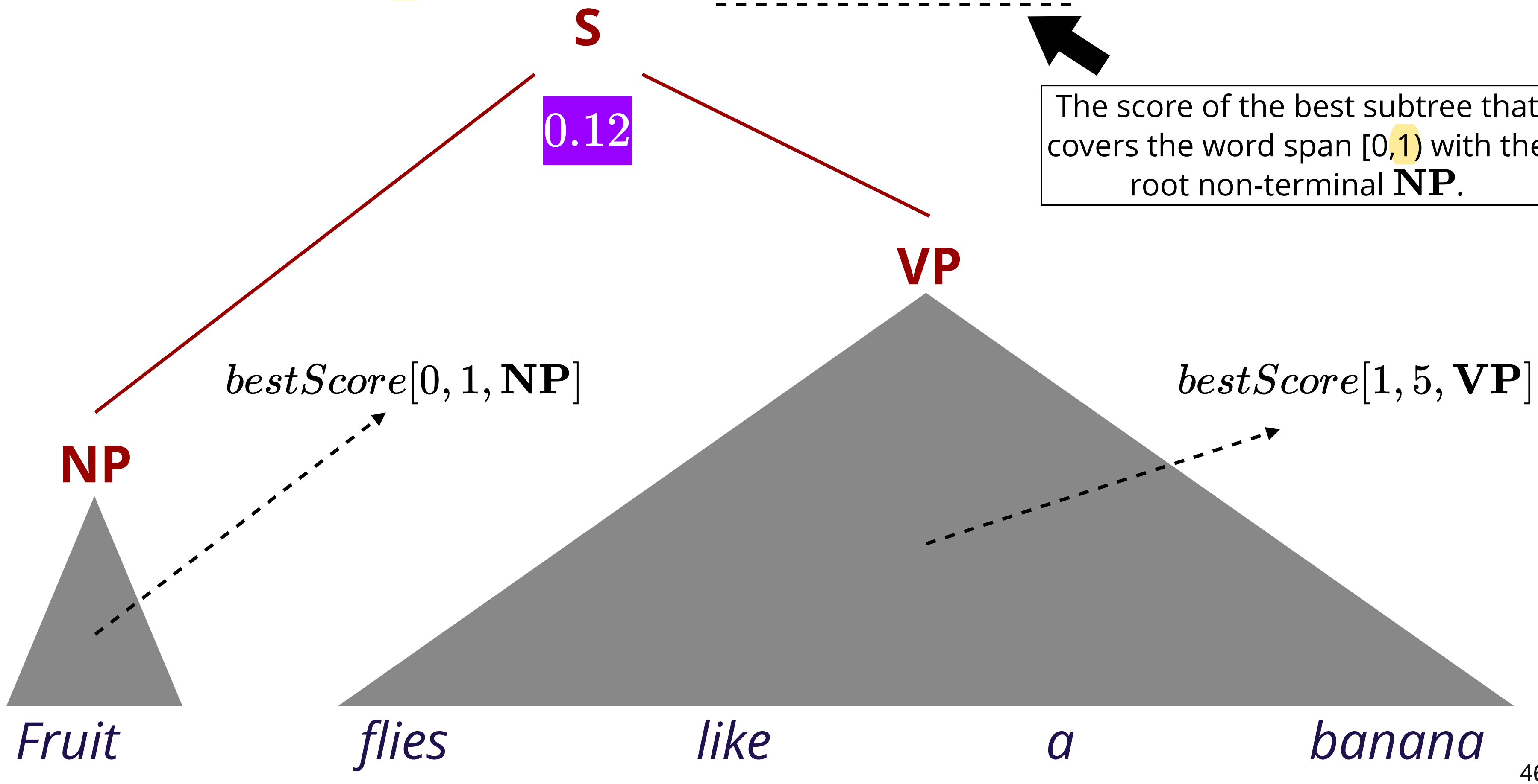
CKY Algorithm

$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

Partition position: 1

$$0.12 \times bestScore[0, 1, \mathbf{NP}] \times bestScore[1, 5, \mathbf{VP}]$$

The score of the best subtree that covers the word span [0,1) with the root non-terminal $\mathbf{NP}$.

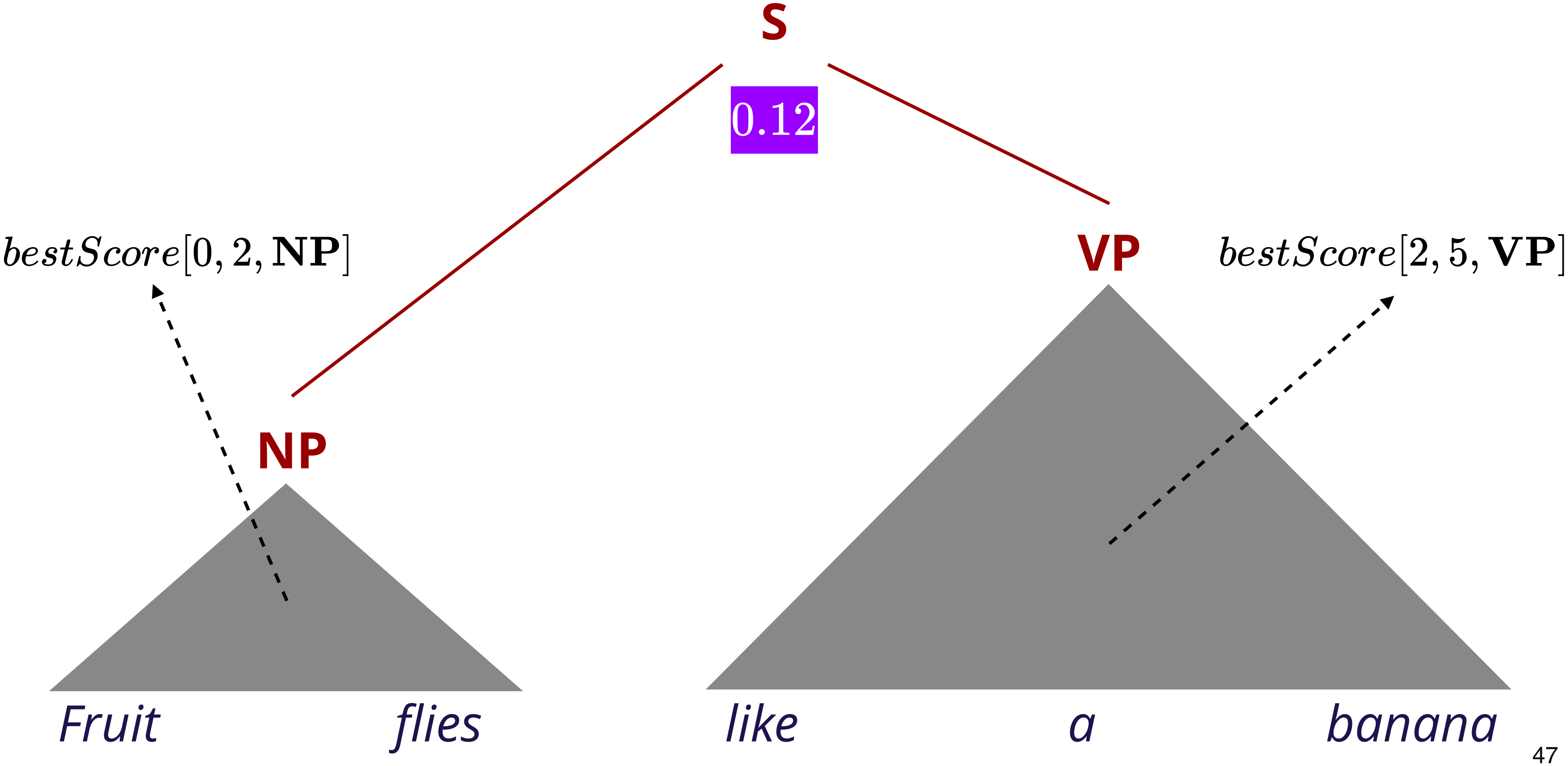


CKY Algorithm

$$\mathbf{S} \rightarrow \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

Partition position: 2

$$0.12 \times bestScore[0, 2, \mathbf{NP}] \times bestScore[2, 5, \mathbf{VP}]$$

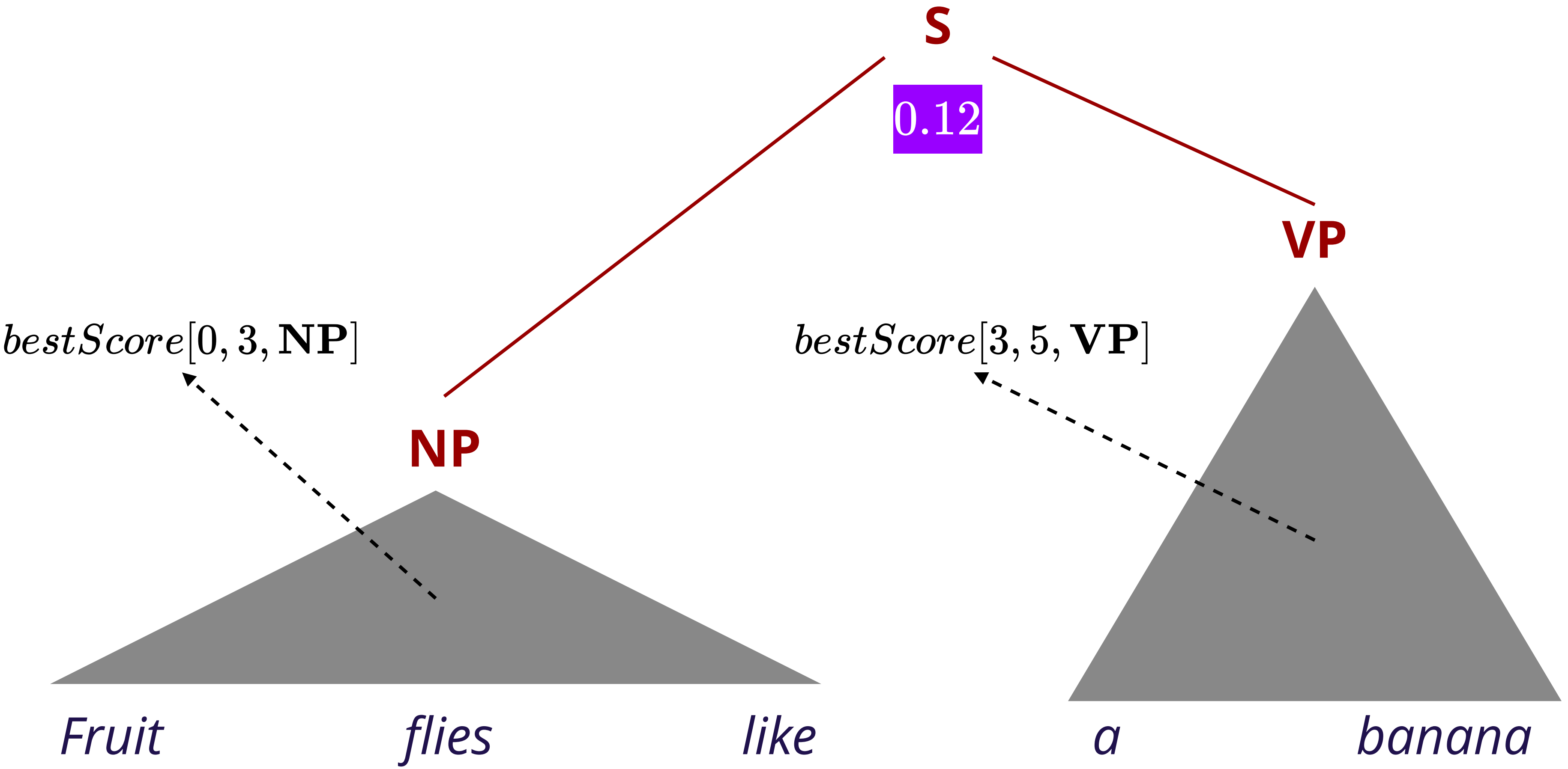


CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

Partition position: 3

$$0.12 \times bestScore[0, 3, \mathbf{NP}] \times bestScore[3, 5, \mathbf{VP}]$$

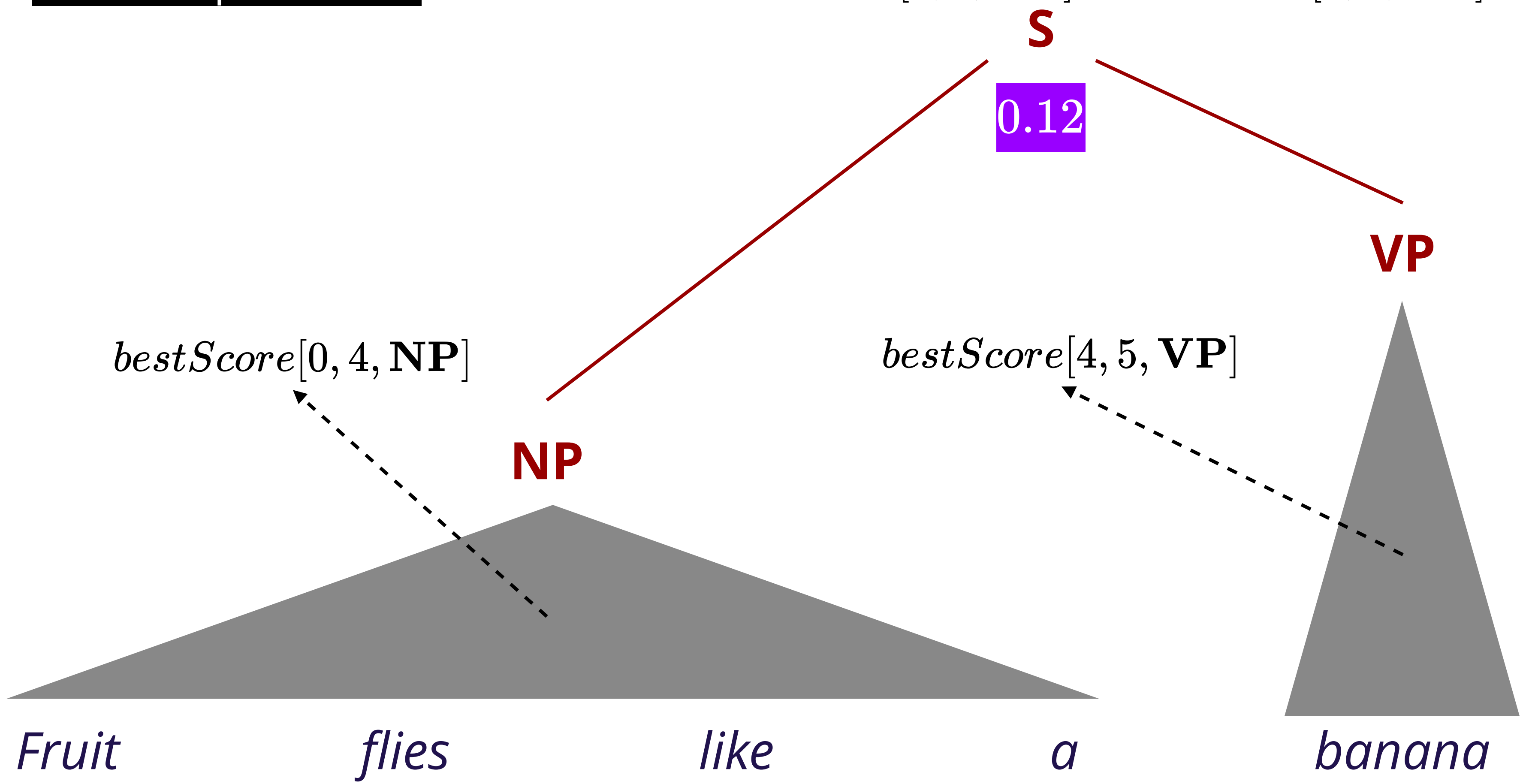


CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

Partition position: 4

$$0.12 \times bestScore[0,4,\mathbf{NP}] \times bestScore[4,5,\mathbf{VP}]$$

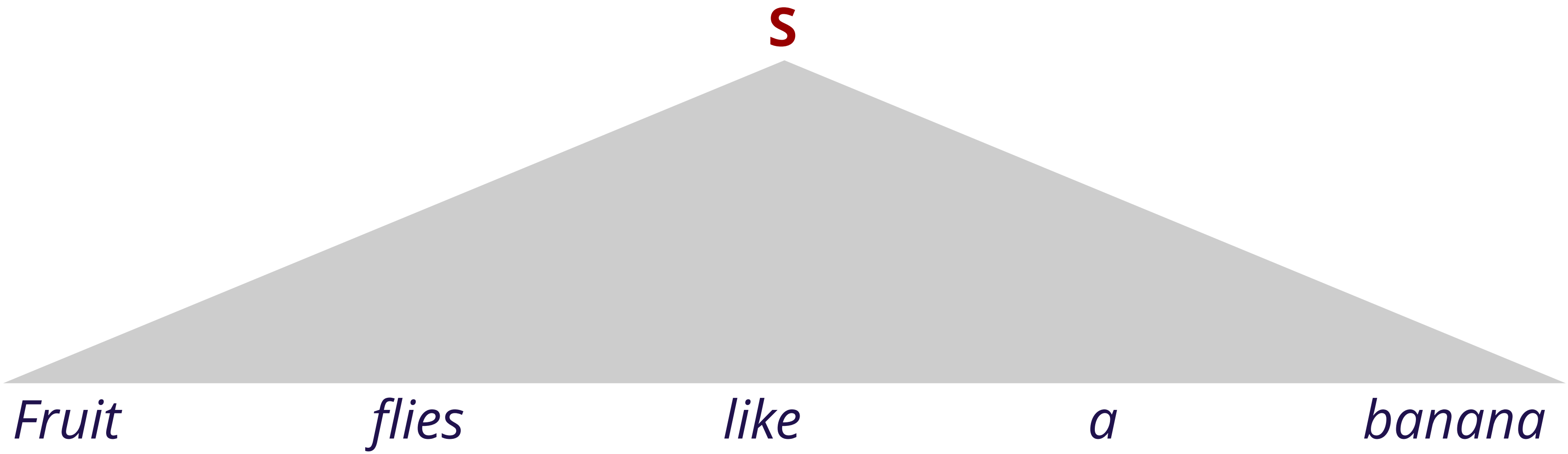


CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

$\boxed{\text{We shall pick the split point that gives the highest score.}}$

$$bestScore[0,5,\mathbf{S}] = \max_{1 \leq k \leq 4}(0.12 \times bestScore[0,k,\mathbf{NP}] \times bestScore[k,5,\mathbf{VP}])$$



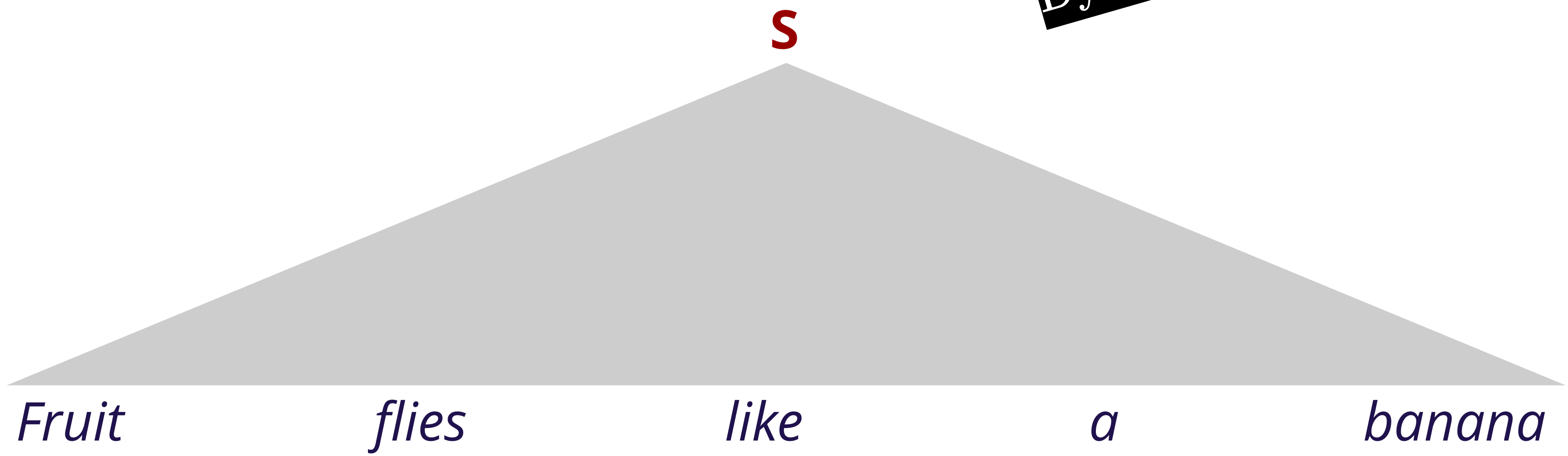
CKY Algorithm

$$\mathbf{S} \quad \rightarrow \quad \mathbf{NP} \quad \mathbf{VP} \quad (0.12)$$

$$bestScore[0,5,\mathbf{S}] = \max_{1 \le k \le 4}(0.12 \times bestScore[0,k,\mathbf{NP}] \times bestScore[k,5,\mathbf{VP}])$$

Q1. How to find such *bestScore* values?

Dynamic programming!



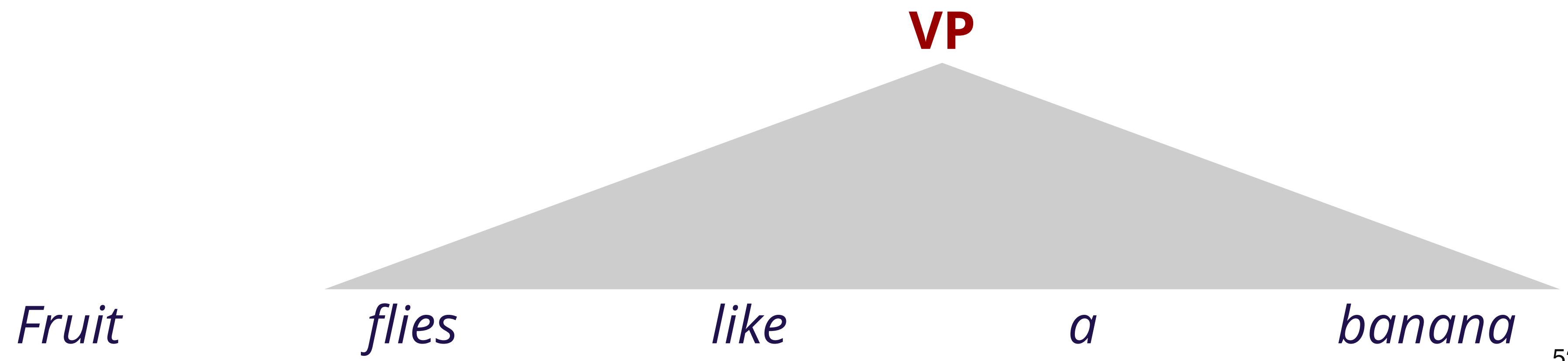
CKY Algorithm

$$
\begin{array}{llll}
\mathbf{VP} & \rightarrow & \mathbf{V} \quad \mathbf{NP} & (0.29) \\
\mathbf{VP} & \rightarrow & \mathbf{V} \quad \mathbf{VP} & (0.12) \\
\mathbf{VP} & \rightarrow & \mathbf{V} \quad \mathbf{PP} & (0.08)
\end{array}
$$

$$
\begin{aligned}
bestScore[1,5,\mathbf{VP}] = \max\{ & \max_{1 \le k \le 4}(0.29 \times bestScore[1,k,\mathbf{V}] \times bestScore[k,5,\mathbf{NP}]), \\
& \max_{1 \le k \le 4}(0.12 \times bestScore[1,k,\mathbf{V}] \times bestScore[k,5,\mathbf{VP}]), \\
& \max_{1 \le k \le 4}(0.08 \times bestScore[1,k,\mathbf{V}] \times bestScore[k,5,\mathbf{PP}])\}
\end{aligned}
$$

You need two 'for' loops!!

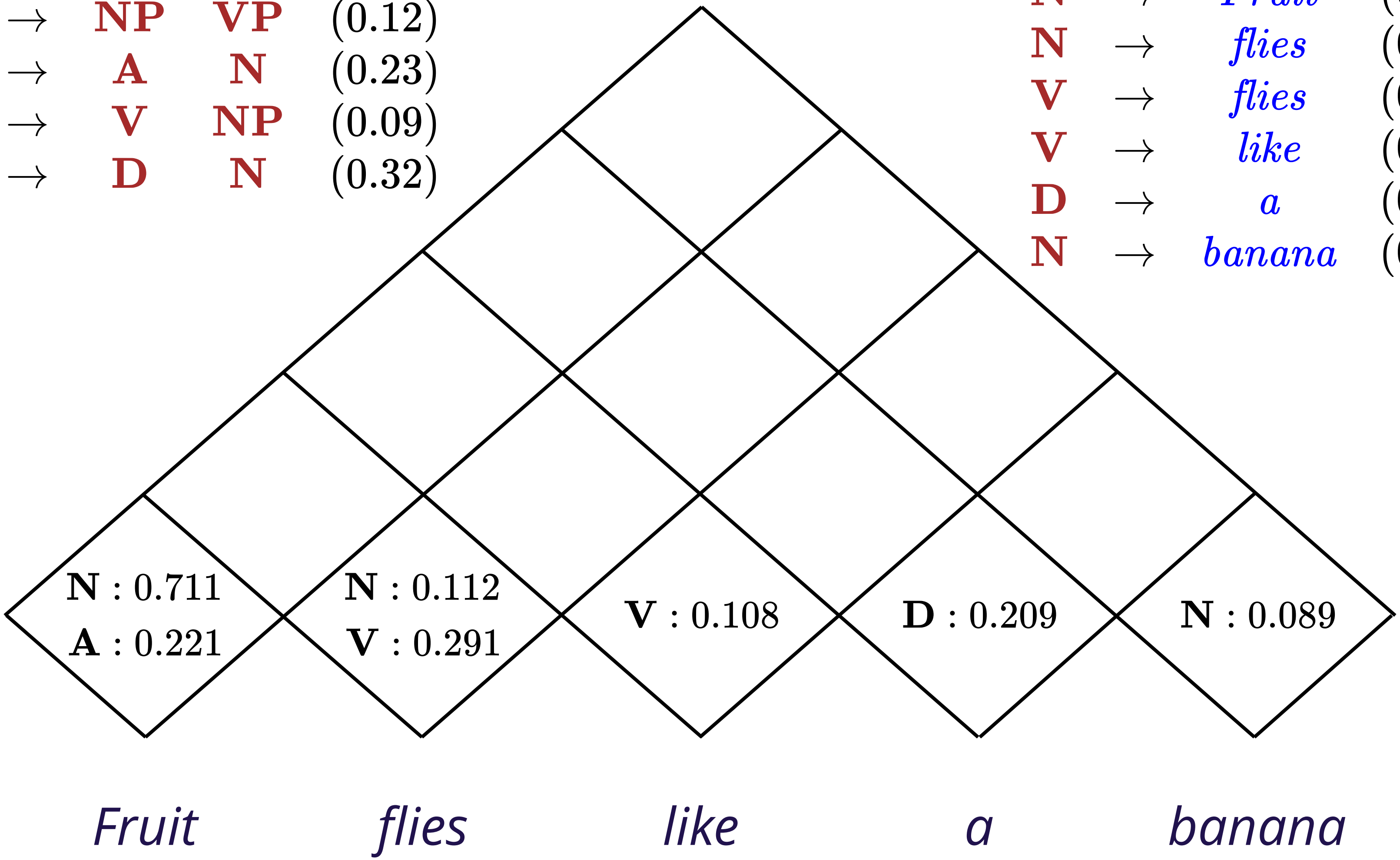
Q2. What if we have multiple rules that start with the same non-terminal?



CKY Algorithm

$$\begin{array}{llll} \mathbf{S} \rightarrow \mathbf{NP}\ \mathbf{VP} & (0.12) \\ \mathbf{NP} \rightarrow \mathbf{A}\ \mathbf{N} & (0.23) \\ \mathbf{VP} \rightarrow \mathbf{V}\ \mathbf{NP} & (0.09) \\ \mathbf{NP} \rightarrow \mathbf{D}\ \mathbf{N} & (0.32) \end{array}$$

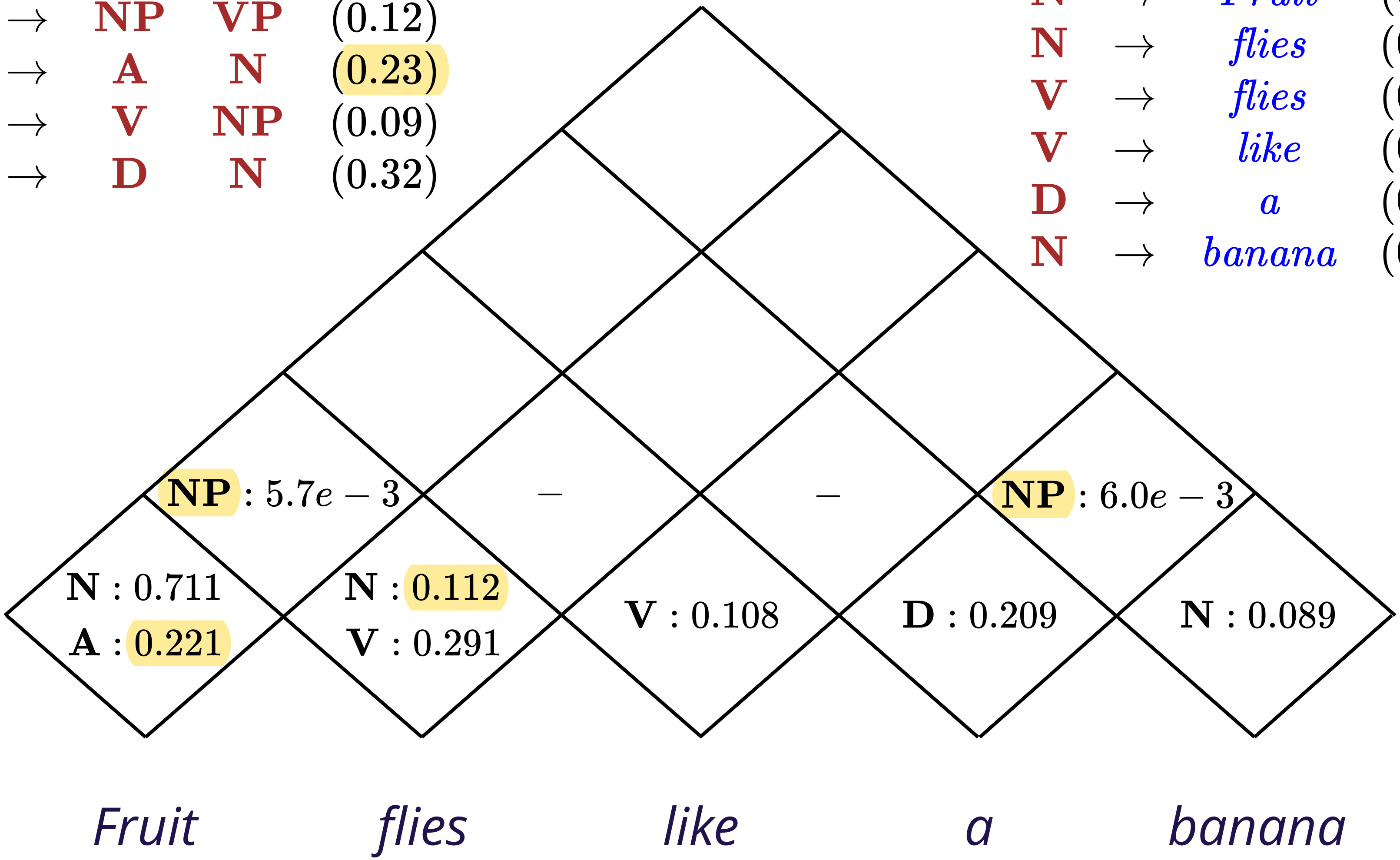
$$\begin{array}{lll} \mathbf{A} \rightarrow \textit{Fruit} & (0.221) \\ \mathbf{N} \rightarrow \textit{Fruit} & (0.711) \\ \mathbf{N} \rightarrow \textit{flies} & (0.112) \\ \mathbf{V} \rightarrow \textit{flies} & (0.291) \\ \mathbf{V} \rightarrow \textit{like} & (0.108) \\ \mathbf{D} \rightarrow \textit{a} & (0.209) \\ \mathbf{N} \rightarrow \textit{banana} & (0.089) \end{array}$$



CKY Algorithm

S	$\rightarrow$	**NP**	**VP**	(0.12)
NP	$\rightarrow$	**A**	**N**	(0.23)
VP	$\rightarrow$	**V**	**NP**	(0.09)
NP	$\rightarrow$	**D**	**N**	(0.32)

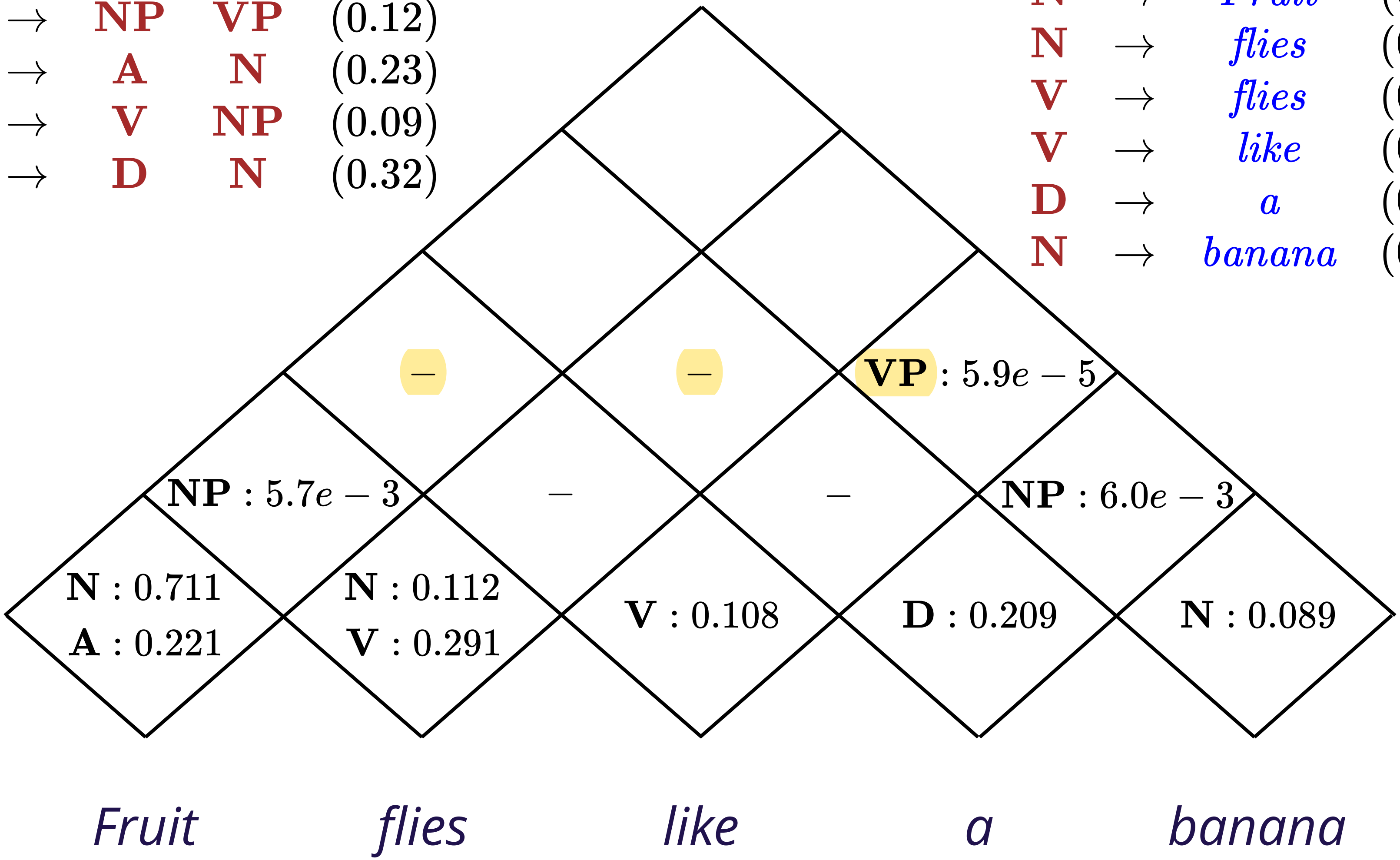
A	$\rightarrow$	*Fruit*	(0.221)
N	$\rightarrow$	*Fruit*	(0.711)
N	$\rightarrow$	*flies*	(0.112)
V	$\rightarrow$	*flies*	(0.291)
V	$\rightarrow$	*like*	(0.108)
D	$\rightarrow$	*a*	(0.209)
N	$\rightarrow$	*banana*	(0.089)



CKY Algorithm

$$
\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{NP}\ \mathbf{VP} && (0.12)\\
\mathbf{NP} &\rightarrow \mathbf{A}\ \mathbf{N} && (0.23)\\
\mathbf{VP} &\rightarrow \mathbf{V}\ \mathbf{NP} && (0.09)\\
\mathbf{NP} &\rightarrow \mathbf{D}\ \mathbf{N} && (0.32)
\end{aligned}
$$

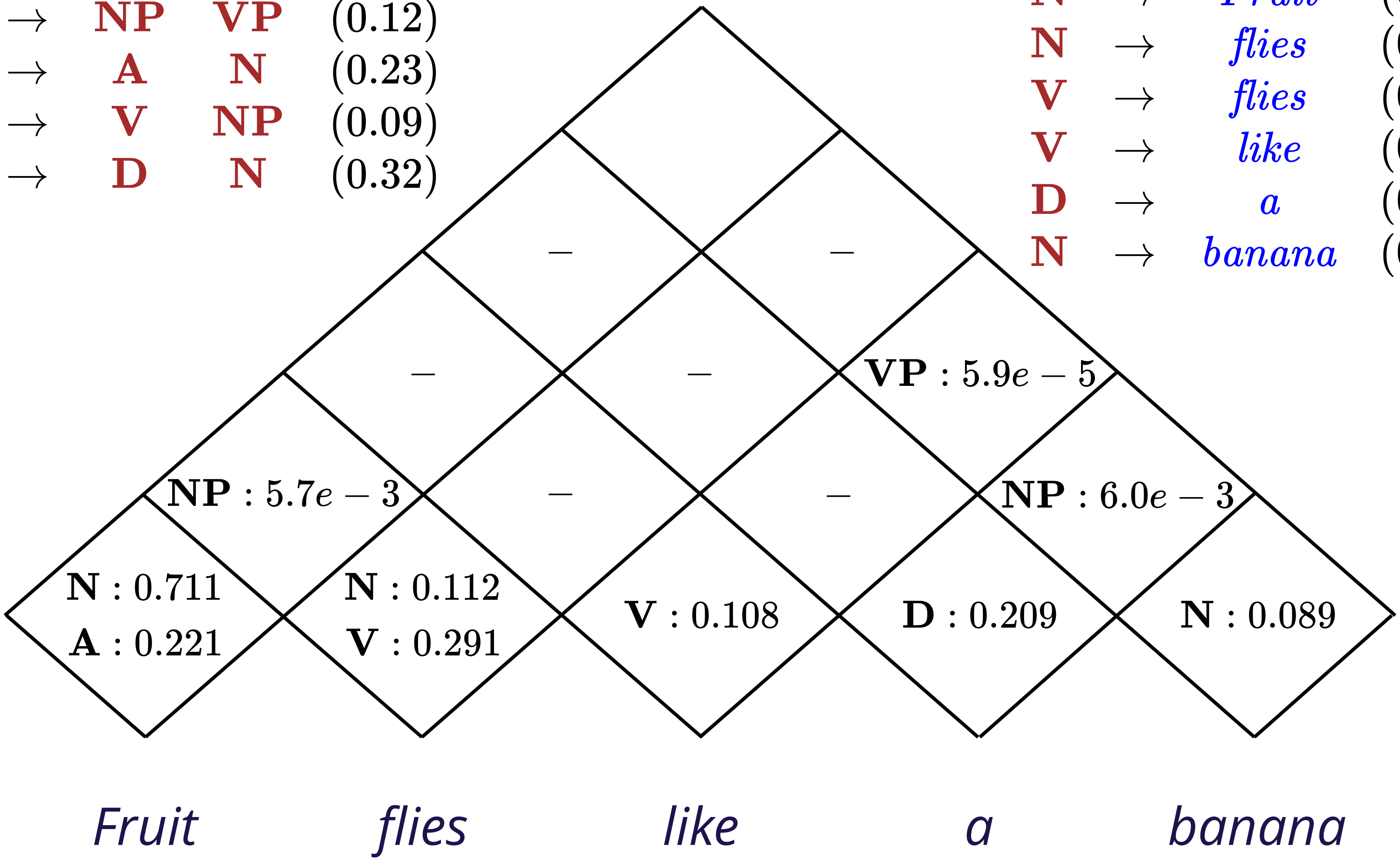
$$
\begin{aligned}
\mathbf{A} &\rightarrow \mathit{Fruit} && (0.221)\\
\mathbf{N} &\rightarrow \mathit{Fruit} && (0.711)\\
\mathbf{N} &\rightarrow \mathit{flies} && (0.112)\\
\mathbf{V} &\rightarrow \mathit{flies} && (0.291)\\
\mathbf{V} &\rightarrow \mathit{like} && (0.108)\\
\mathbf{D} &\rightarrow \mathit{a} && (0.209)\\
\mathbf{N} &\rightarrow \mathit{banana} && (0.089)
\end{aligned}
$$



CKY Algorithm

S	$\rightarrow$	**NP**	**VP**	(0.12)
NP	$\rightarrow$	**A**	**N**	(0.23)
VP	$\rightarrow$	**V**	**NP**	(0.09)
NP	$\rightarrow$	**D**	**N**	(0.32)

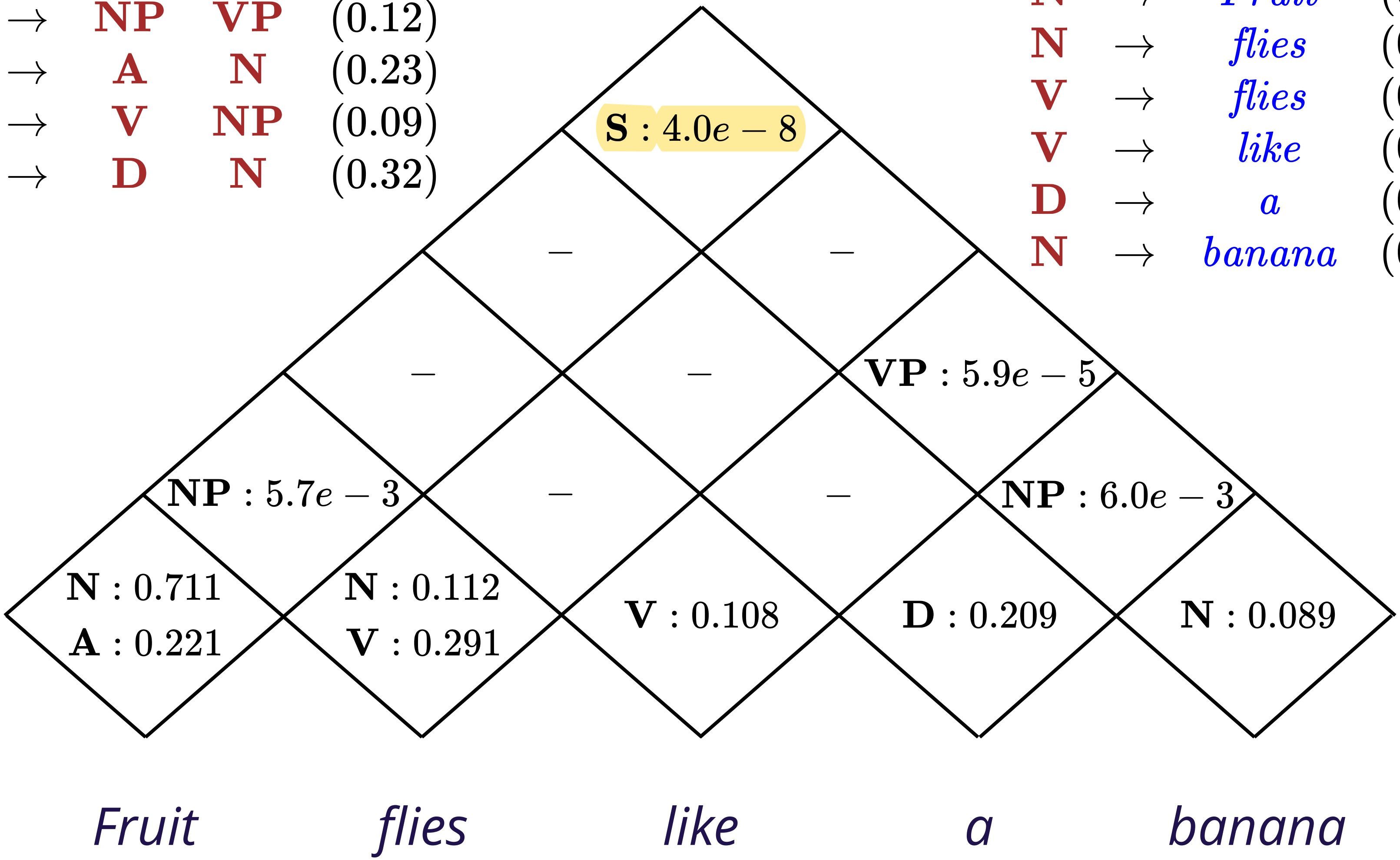
A	$\rightarrow$	*Fruit*	(0.221)
N	$\rightarrow$	*Fruit*	(0.711)
N	$\rightarrow$	*flies*	(0.112)
V	$\rightarrow$	*flies*	(0.291)
V	$\rightarrow$	*like*	(0.108)
D	$\rightarrow$	*a*	(0.209)
N	$\rightarrow$	*banana*	(0.089)



CKY Algorithm

$$
\begin{array}{lllll}
\mathbf{S} & \rightarrow & \mathbf{NP} & \mathbf{VP} & (0.12) \\
\mathbf{NP} & \rightarrow & \mathbf{A} & \mathbf{N} & (0.23) \\
\mathbf{VP} & \rightarrow & \mathbf{V} & \mathbf{NP} & (0.09) \\
\mathbf{NP} & \rightarrow & \mathbf{D} & \mathbf{N} & (0.32)
\end{array}
$$

$$
\begin{array}{llll}
\mathbf{A} & \rightarrow & \mathit{Fruit} & (0.221) \\
\mathbf{N} & \rightarrow & \mathit{Fruit} & (0.711) \\
\mathbf{N} & \rightarrow & \mathit{flies} & (0.112) \\
\mathbf{V} & \rightarrow & \mathit{flies} & (0.291) \\
\mathbf{V} & \rightarrow & \mathit{like} & (0.108) \\
\mathbf{D} & \rightarrow & \mathit{a} & (0.209) \\
\mathbf{N} & \rightarrow & \mathit{banana} & (0.089)
\end{array}
$$



Question

What is the time complexity of CKY?

Time complexity: $O(n^3|R|)$

n: length of parsed string

|R|: size of grammar rules

We have n^2 entries in the table. But for each entry, we have to consider all the possible split points.

Each rule is in CNF, you have to consider the possibility of each rule to be applied to every span. Within each span, you have to find the split point. Essentially, for each entry, you have to consider all possible rules at worst case.

Probabilistic CFG

Summary

Learning

$$p(\mathbf{A} \rightarrow \mathbf{exp}) = \frac{\text{count}(\mathbf{A} \rightarrow \mathbf{exp})}{\text{count}(\mathbf{A})}$$

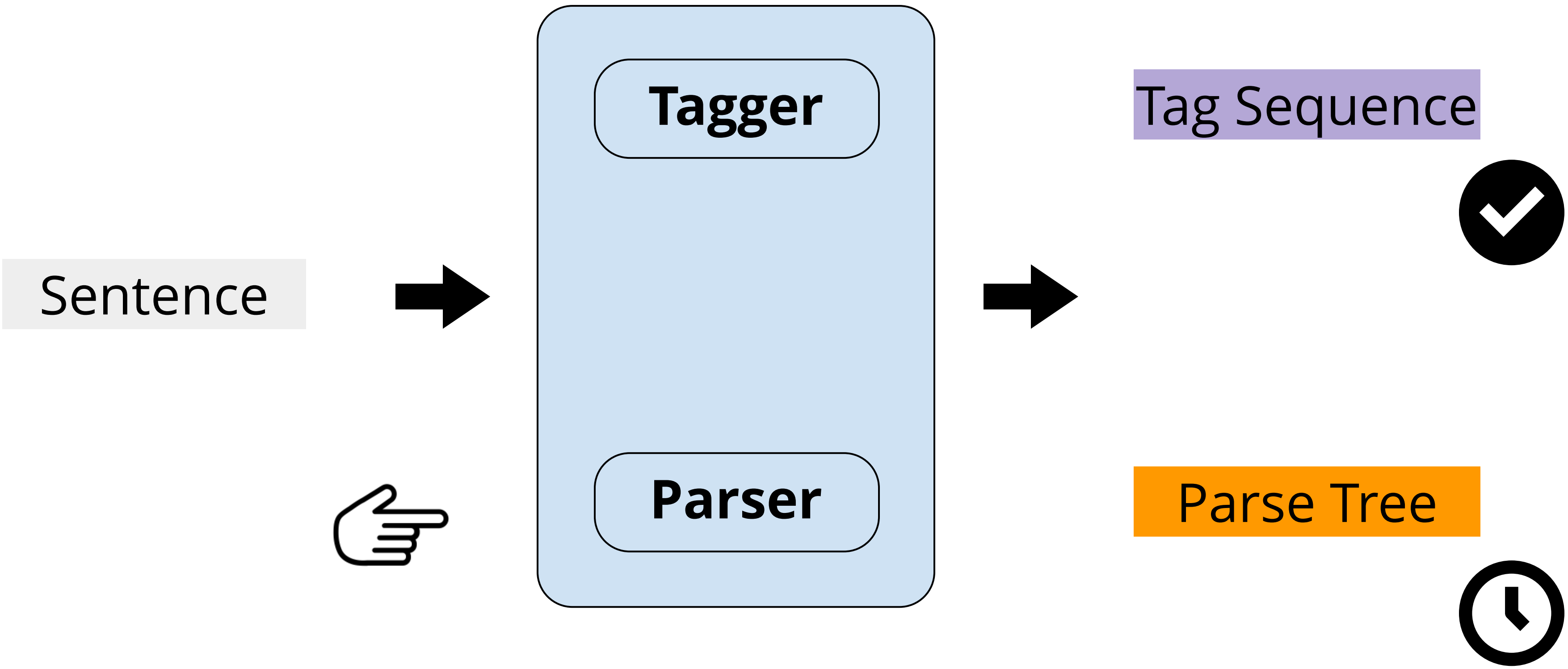
Decoding

CKY Algorithm
(Dynamic Programming)

There are limitations with the probabilistic CFG as it is a generative approach...

We will learn some improved models next time!!

Structured Prediction



Structured Space $\mathcal{S}_i$ Structured Prediction Model M Structured Space $\mathcal{S}_o$

$$\mathsf{M} : \mathcal{S}_i \rightarrow \mathcal{S}_o$$