

# CS542 (Fall 2021) Written Assignment 3

## Statistical Parser: CKY Algorithm\*

Eric Martin  
Due November 7, 2021

### Assignment

You are given a small grammar on the next page. Your task is to use the probabilistic CKY algorithm to fill in the *table* and *back* arrays for the two sentences “time flies like an arrow” and “fruit flies like a banana”. Please also draw the final tree for each sentence. Note that because the grammar is not in Chomsky normal form (it contains unary rules), you will need to modify the algorithm in Figure C.3 of the Jurafsky and Martin book to handle the unary rules.

```
function PROBABILISTIC-CKY(words,grammar) returns most probable parse
                                     and its probability
for j ← from 1 to LENGTH(words) do
  for all { A | A → words[j] ∈ grammar }
    table[j − 1, j, A] ← P(A → words[j])
  for i ← from j − 2 downto 0 do
    for k ← i + 1 to j − 1 do
      for all { A | A → BC ∈ grammar,
                and table[i, k, B] > 0 and table[k, j, C] > 0 }
        if (table[i, j, A] < P(A → BC) × table[i, k, B] × table[k, j, C]) then
          table[i, j, A] ← P(A → BC) × table[i, k, B] × table[k, j, C]
          back[i, j, A] ← {k, B, C}
  return BUILD-TREE(back[1, LENGTH(words), S], table[1, LENGTH(words), S])
```

**Figure C.3** The probabilistic CKY algorithm for finding the maximum probability parse of a string of *num\_words* words given a PCFG grammar with *num\_rules* rules in Chomsky normal form. *back* is an array of backpointers used to recover the best parse. The *build\_tree* function is left as an exercise to the reader.

---

\*This assignment is adapted from Hal Daume's.

## Grammar

S	-> NP VP	[0.5]
S	-> VP	[0.2]
S	-> NP VP_PP	[0.2]
S	-> VP PP	[0.1]
VP_PP	-> VP PP	[1.0]
NP	-> DT NN	[0.5]
NP	-> Nominal NN	[0.2]
NP	-> NN	[0.3]
VP	-> VB NP	[0.6]
VP	-> VB PP	[0.2]
VP	-> VB NP_PP	[0.1]
VP	-> VB	[0.1]
NP_PP	-> NP PP	[1.0]
PP	-> IN NP	[1.0]
DT	-> 'a'	[0.5]
DT	-> 'an'	[0.5]
NN	-> 'time'	[0.2]
NN	-> 'flies'	[0.2]
NN	-> 'arrow'	[0.2]
NN	-> 'fruit'	[0.2]
NN	-> 'banana'	[0.2]
VB	-> 'time'	[0.3]
VB	-> 'flies'	[0.4]
VB	-> 'like'	[0.3]
IN	-> 'like'	[1.0]
Nominal	-> 'fruit'	[1.0]

## Submission Instructions

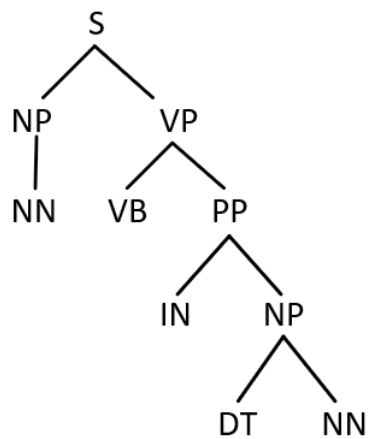
Please submit your solutions (in PDF format) to the submission box on Canvas. If you are familiar with L<sup>A</sup>T<sub>E</sub>X, you may use the table and tree templates below (see HW3.tex, which is downloadable from Canvas). Alternatively, you may draw your tables and trees by hand and scan them into a PDF.

	time	flies	like	an	arrow
0	[0,1]	[0,2]	[0,3]	[0,4]	[0,5]
0	$S \rightarrow VP [0.2] * [0.03] = 0.06$ $NP \rightarrow NN [0.3] * [0.2] = 0.06$ $VP \rightarrow VB [0.1] * [0.3] = 0.03$ $NN \rightarrow 'time' [0.2] \quad VB \rightarrow 'time' [0.3]$	$VP \rightarrow VB NP [0.6] * [0.06] * [0.3] = 0.0108$ $S \rightarrow NP VP [0.5] * [0.06] * [0.04] = 0.0012$			$S \rightarrow NP VP [0.5] * [0.06] * [0.004] = 0.00012$ $S \rightarrow NP VP_PP [0.2] * [0.06] * [0.002] = 0.000024$
1		[1,2]	[1,3]	[1,4]	[1,5]
1		$S \rightarrow VP [0.2] * [0.04] = 0.008$ $NP \rightarrow NN [0.3] * [0.2] = 0.06$ $VP \rightarrow VB [0.1] * [0.4] = 0.04$ $NN \rightarrow 'flies' [0.2] \quad VB \rightarrow 'flies' [0.4]$	$S \rightarrow NP VP [0.5] * [0.03] * [0.06] = 0.0009$		$S \rightarrow NP VP [0.5] * [0.06] * [0.009] = 0.00012$ $S \rightarrow VP PP [0.1] * [0.04] * [0.05] = 0.0002$ $NP_PP \rightarrow NP PP [1.0] * [0.06] * [0.05] = 0.003$ $VP_PP \rightarrow VP PP [1.0] * [0.04] * [0.05] = 0.002$ $VP \rightarrow VB PP [0.2] * [0.4] * [0.05] = 0.004$
2			[2,3]	[2,4]	[2,5]
2			$S \rightarrow VP [0.2] * [0.03] = 0.06$ $VP \rightarrow VB [0.1] * [0.3] = 0.03$ $VB \rightarrow 'like' [0.3] \quad IN \rightarrow 'like' [1.0]$		$VP \rightarrow VB NP [0.6] * [0.3] * [0.05] = 0.009$ $PP \rightarrow IN NP [1.0] * [1.0] * [0.05] = 0.05$
3				[3,4]	[3,5]
3				$DT \rightarrow 'an' [0.5]$	$NP \rightarrow DT NN [0.5] * [0.5] * [0.2] = 0.05$
4					[4,5]
4					$NP \rightarrow NN [0.3] * [0.2] = 0.06$ $NN \rightarrow 'arrow' [0.2]$

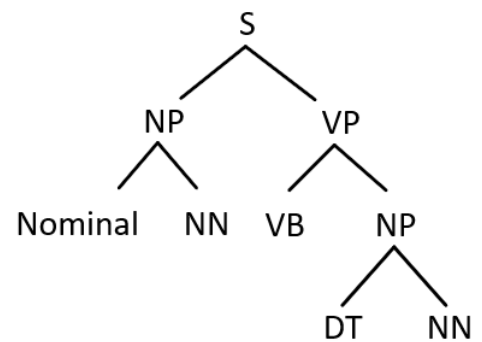
Figure 1: CKY table for "time flies like an arrow"

	Nominal	NN	VB	DT	NN
	fruit	flies	like	a	banana
0	[0,1]	[0,2]	[0,3]	[0,4]	[0,5]
0	NP -> NN [0.3] * [0.2] = 0.06 NN -> 'fruit' [0.2] Nominal -> 'fruit' [1.0]	NP -> Nominal NN [0.2] * [1.0] * [0.2] = 0.04 S -> NP VP [0.5] * [0.06] * [0.04] = 0.0012	S -> NP VP [0.5] * [0.04] * [0.03] = 0.020015		S -> NP VP [0.5] * [0.04] * [0.009] = 0.00018 S -> NP VP_PP [0.2] * [0.06] * [0.002] = 0.000024
1		[1,2] S -> VP [0.2] * [0.04] = 0.008 NP -> NN [0.3] * [0.2] = 0.06 VP -> VB [0.1] * [0.4] = 0.04 NN -> 'flies' [0.2] VB -> 'flies' [0.4]	[1,3] S -> NP VP [0.5] * [0.06] * [0.03] = 0.0009	[1,4]	[1,5] S -> NP VP [0.5] * [0.06] * [0.009] = 0.00012 S -> VP PP [0.1] * [0.04] * [0.05] = 0.0002 NP_PP -> NP PP [1.0] * [0.06] * [0.05] = 0.003 VP_PP -> VP PP [1.0] * [0.04] * [0.05] = 0.002 VP -> VB PP [0.2] * [0.4] * [0.05] = 0.004
2			[2,3] S -> VP [0.2] * [0.03] = 0.06 VP -> VB [0.1] * [0.3] = 0.03 VB -> 'like' [0.3] IN -> 'like' [1.0]	[2,4]	[2,5] VP -> VB NP [0.6] * [0.3] * [0.05] = 0.009 PP -> IN NP [1.0] * [1.0] * [0.05] = 0.05
3				[3,4] DT -> 'a' [0.5]	[3,5] NP -> DT NN [0.5] * [0.5] * [0.2] = 0.05
4					[4,5] NP -> NN [0.3] * [0.2] = 0.06 NN -> 'banana' [0.2]

Figure 2: CKY table for "fruit flies like a banana"



(a) "time flies like an arrow"



(b) "fruit flies like a banana"

Figure 3: CKY Trees for both sentences