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 LaTeX, 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.

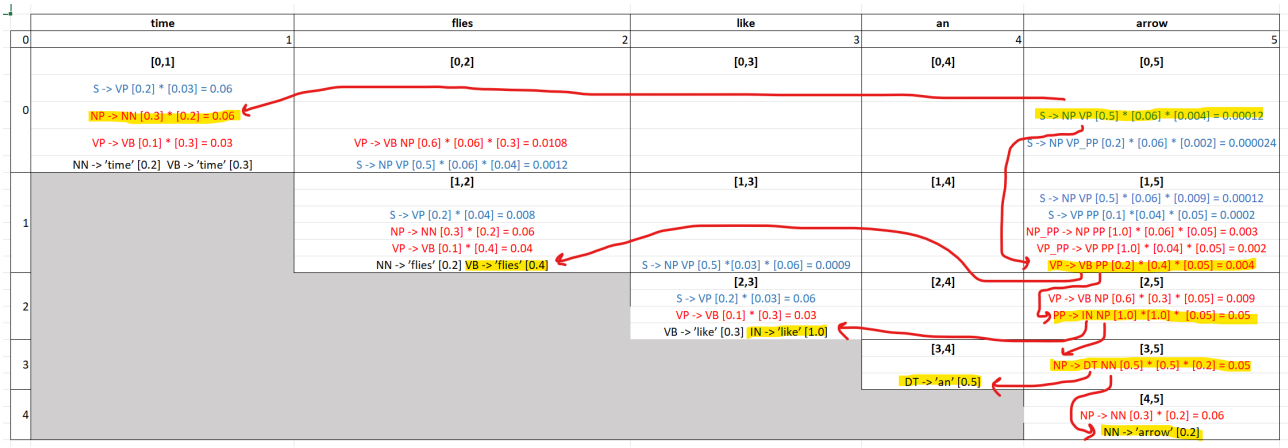


Figure 1: CKY table for "time flies like an arrow"

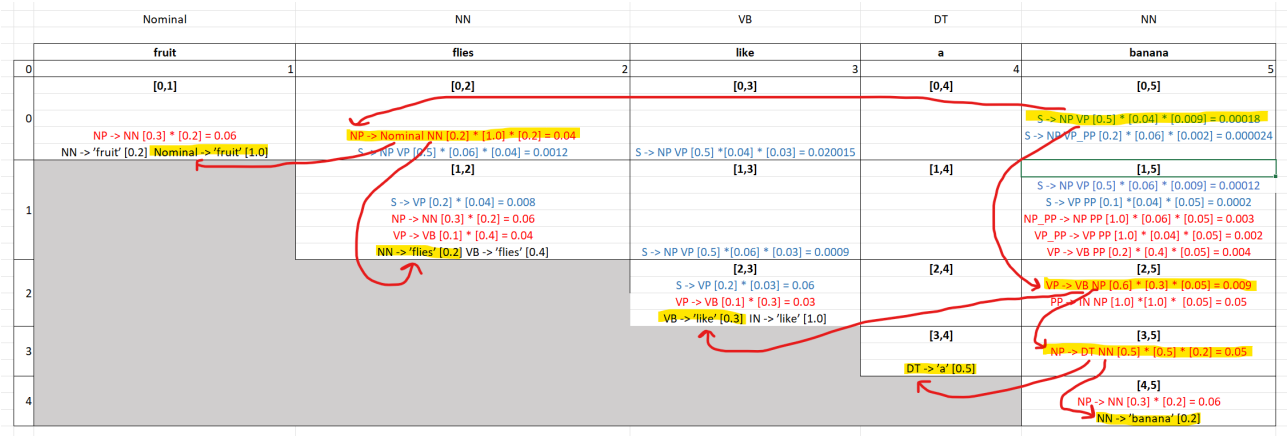
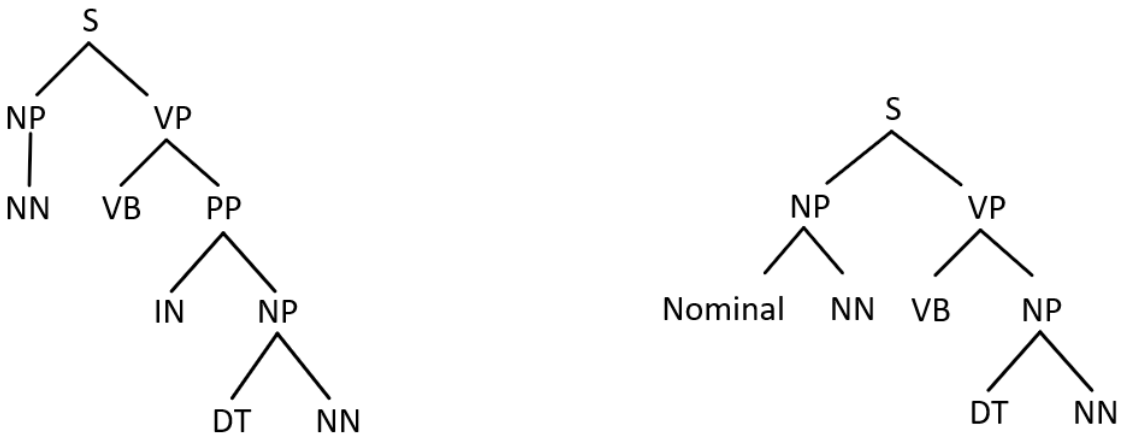


Figure 2: CKY table for "fruit flies like a banana"



(a) "time flies like an arrow"

(b) "fruit flies like a banana"

Figure 3: CKY Trees for both sentences