

# CKY (11/12)

## CS 585, Fall 2015

Introduction to Natural Language Processing  
<http://people.cs.umass.edu/~brenocon/inlp2015/>

**Brendan O'Connor**

College of Information and Computer Sciences  
University of Massachusetts Amherst

# CKY

## Grammar

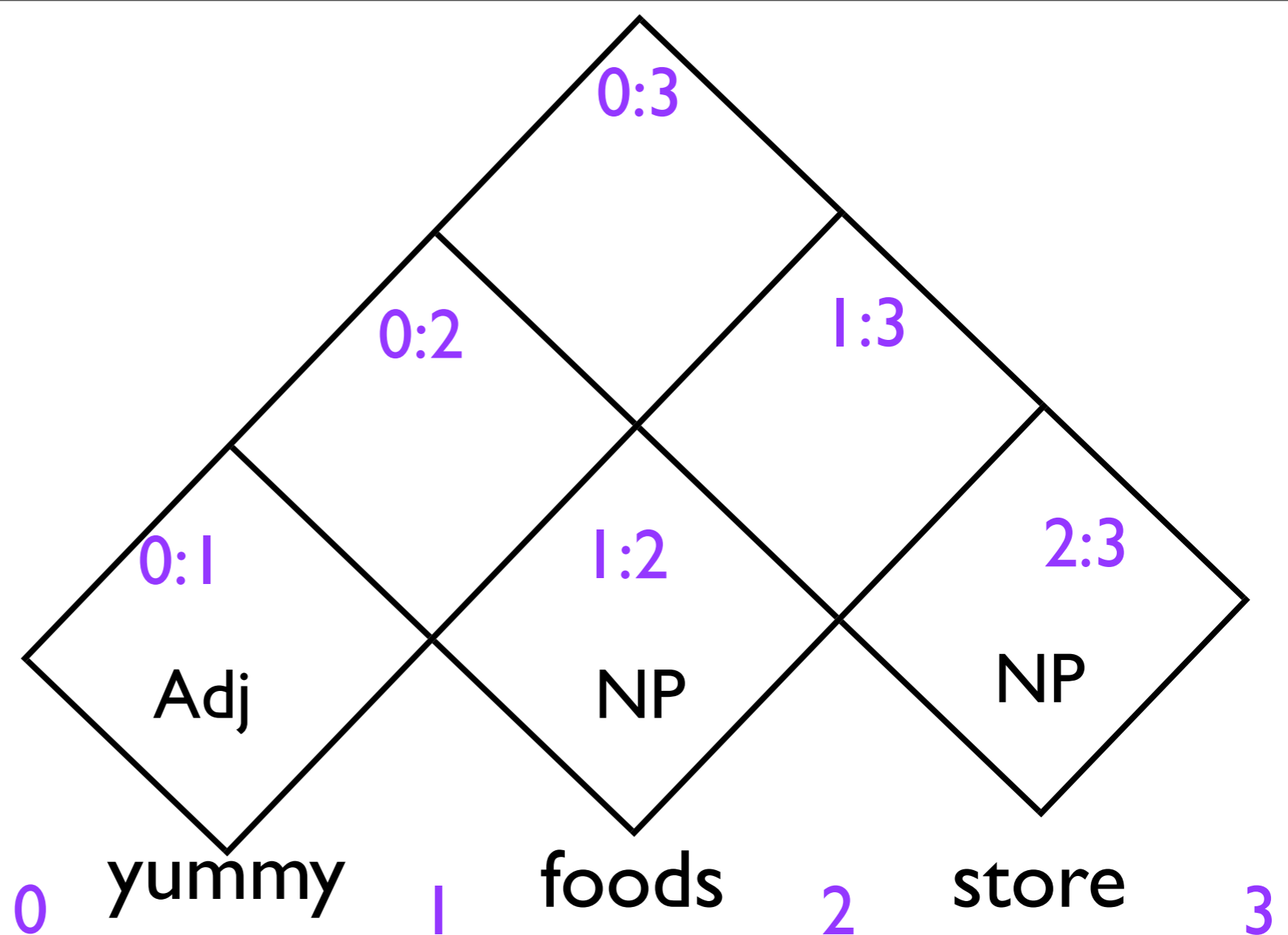
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

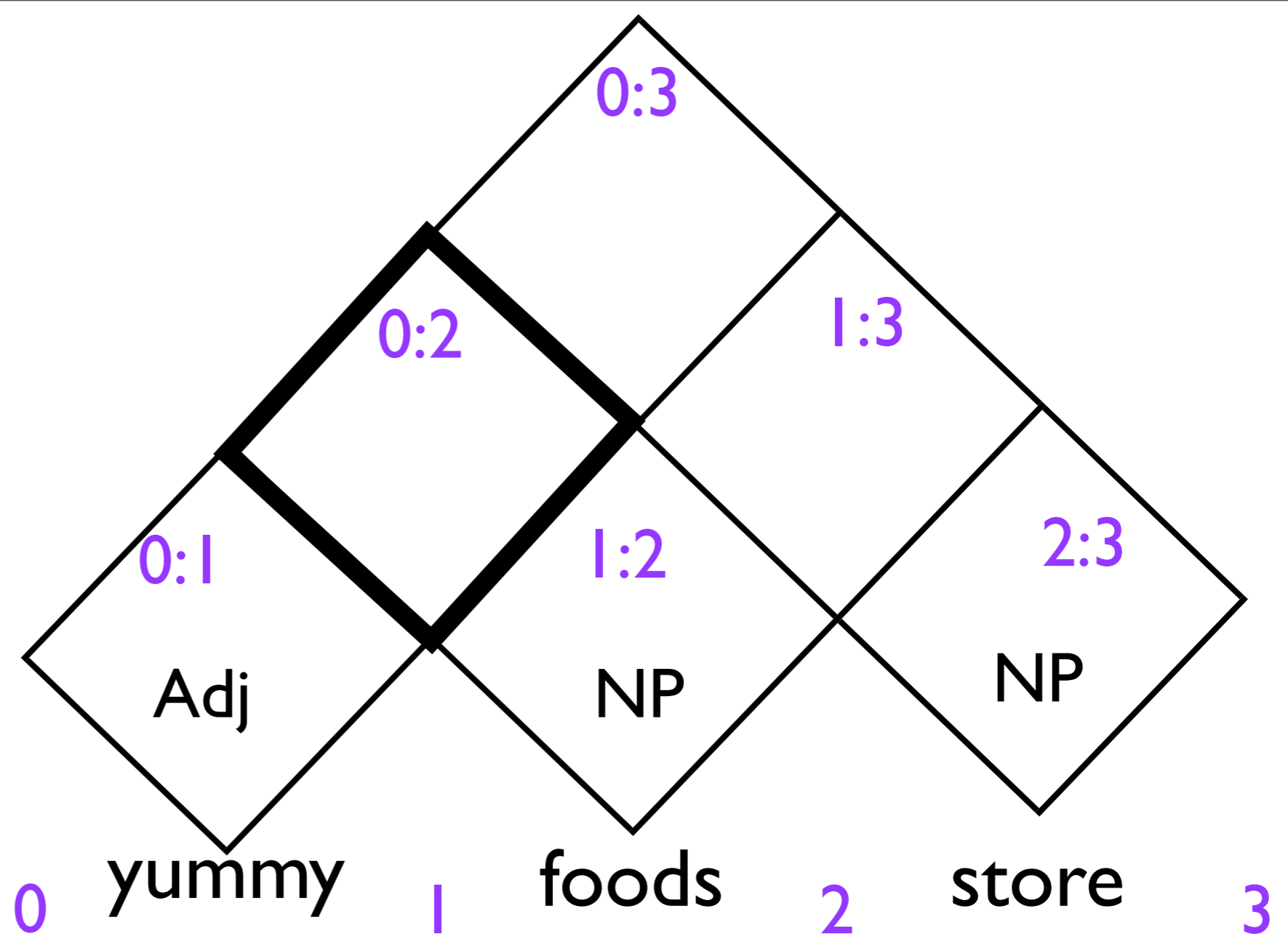
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

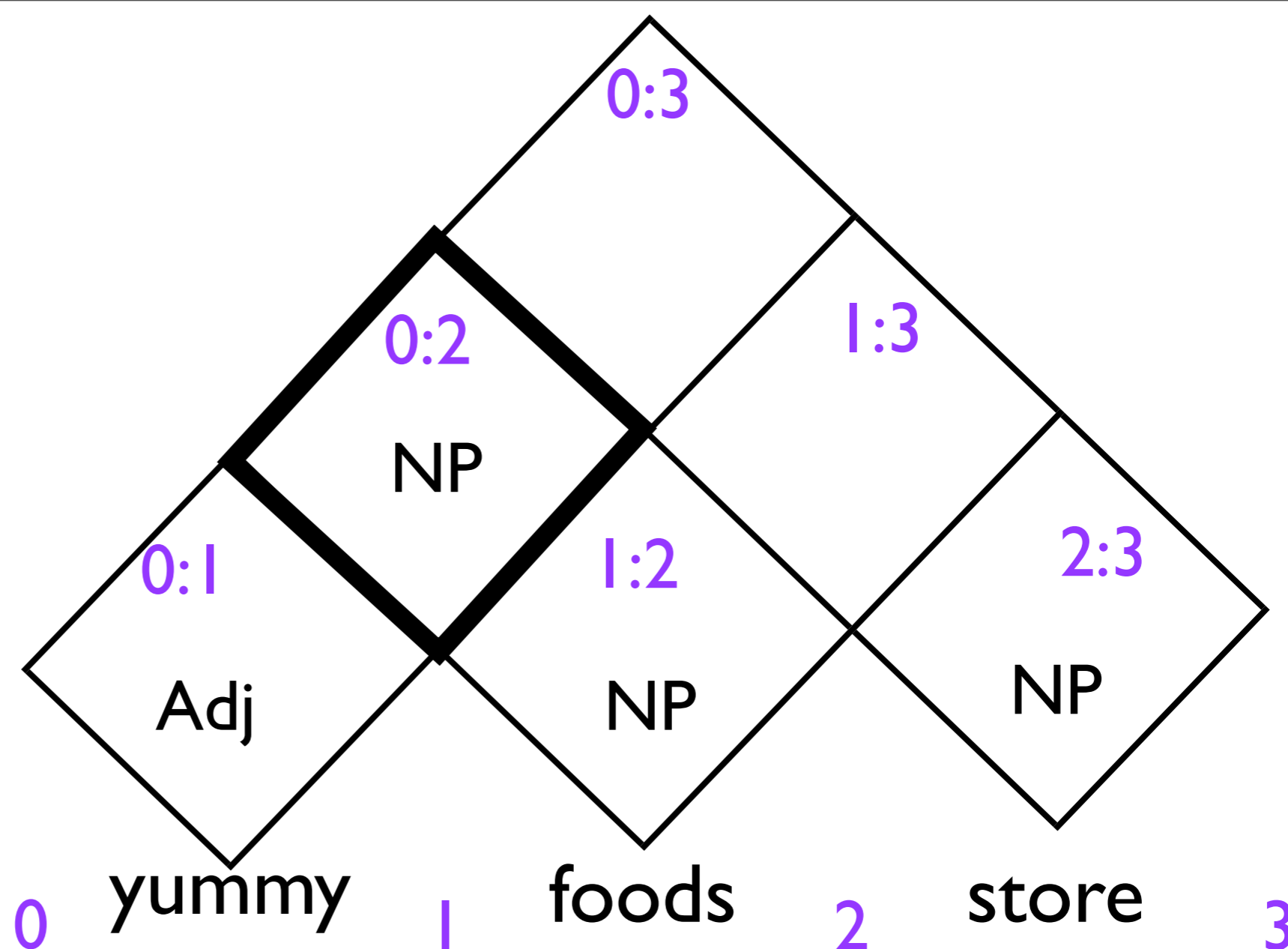
Adj -> yummy

NP -> foods

NP -> store

NP -> NP NP

NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

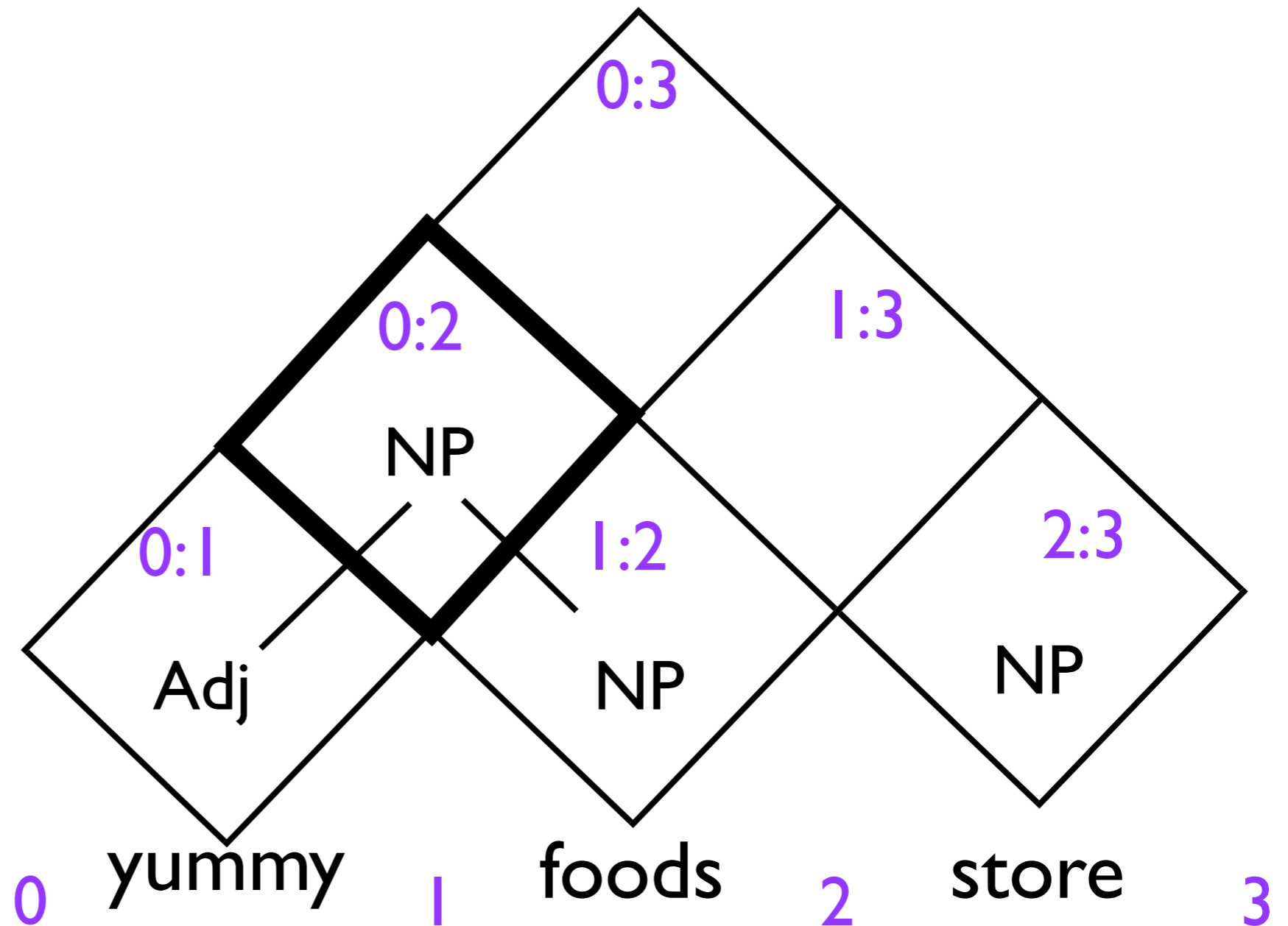
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

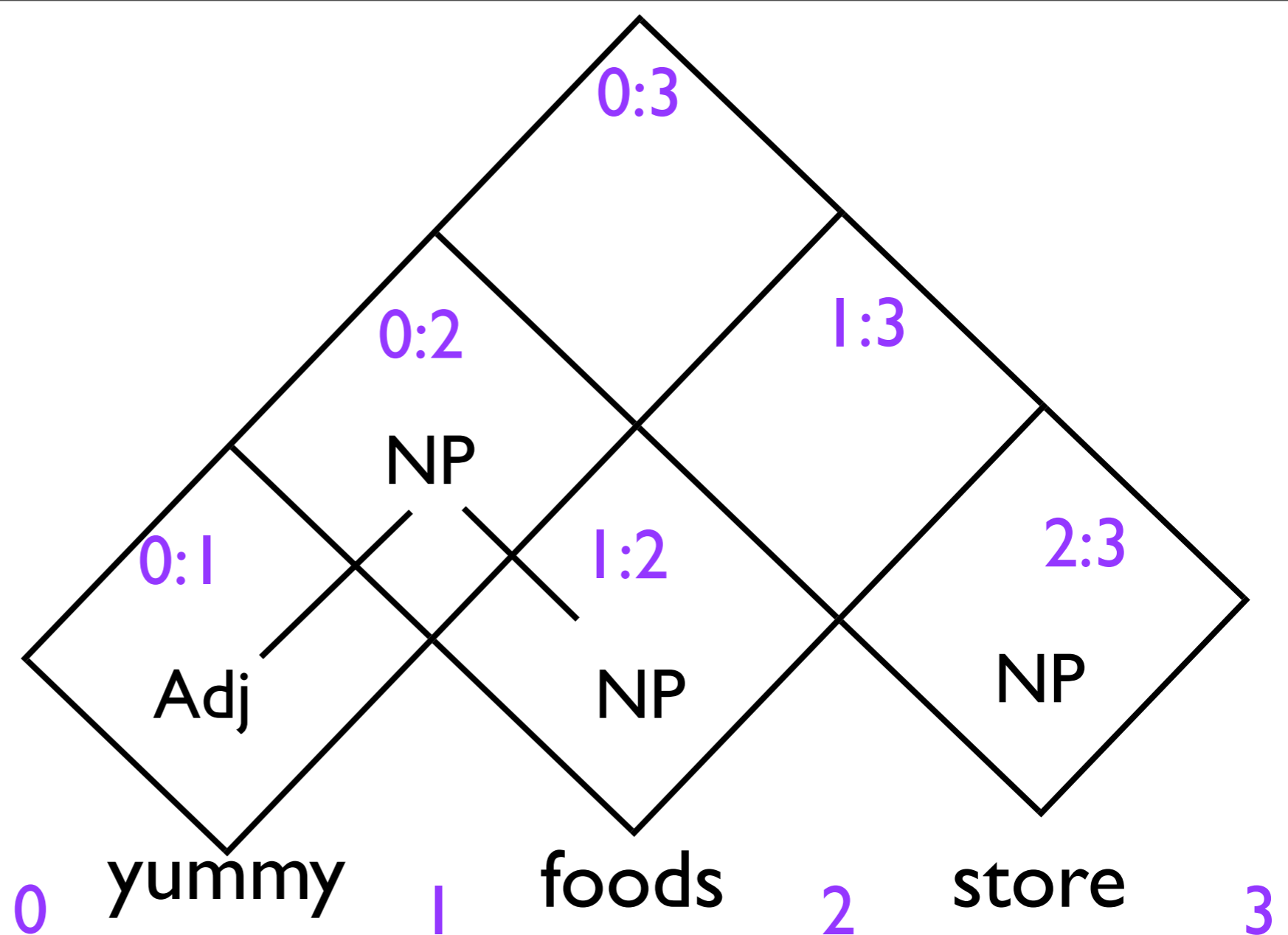
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

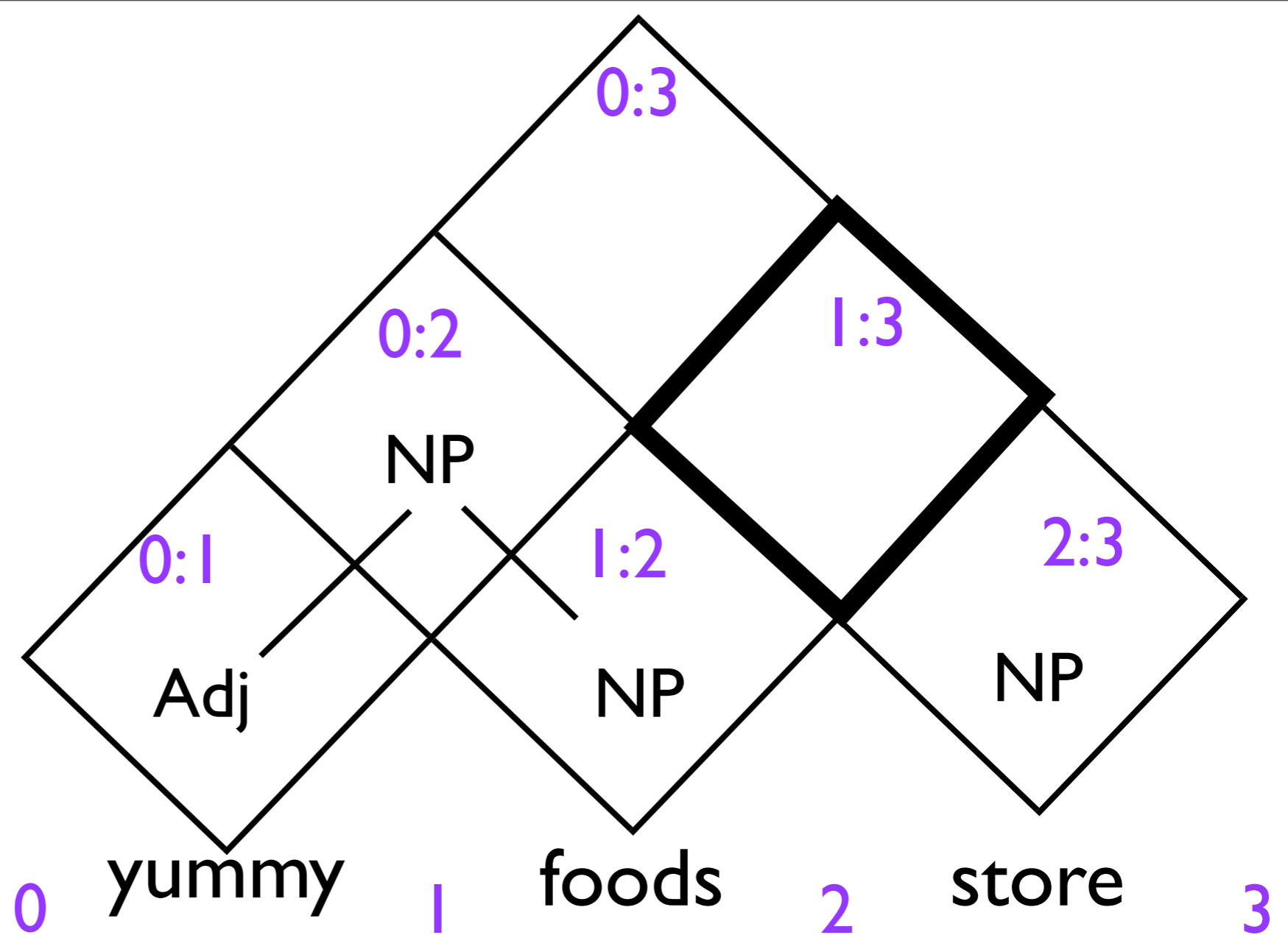
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

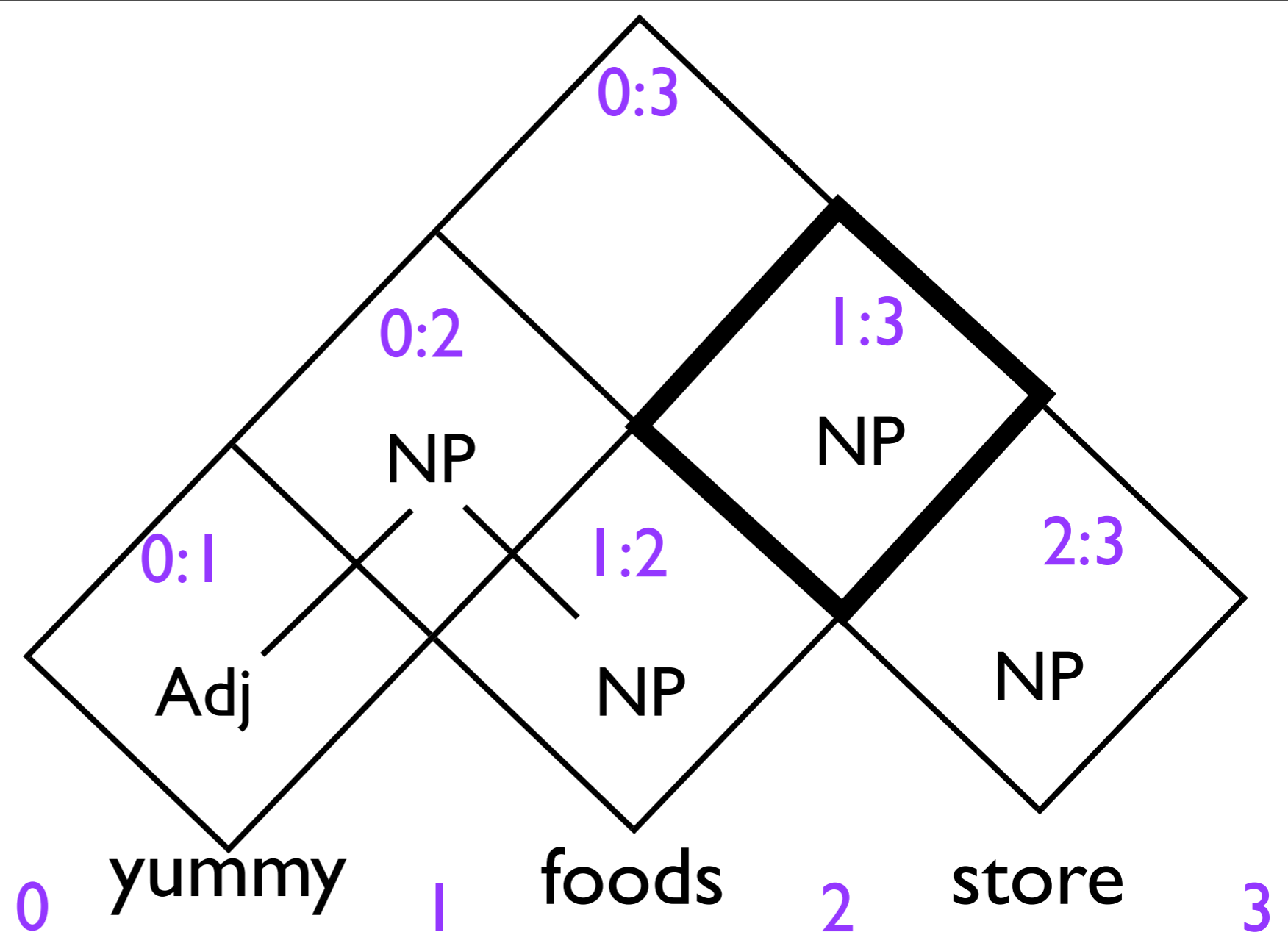
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

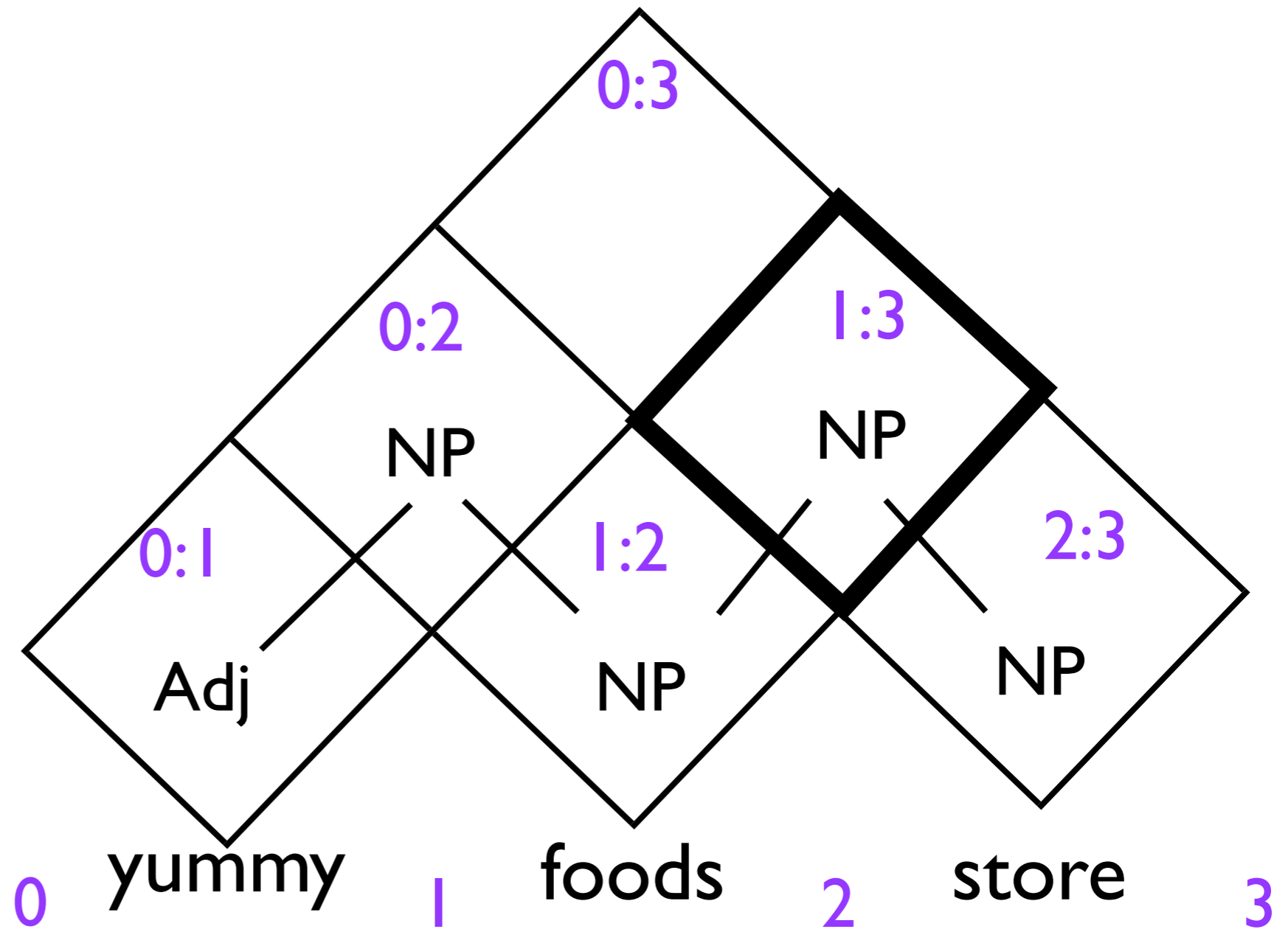
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.



# CKY

Grammar  
 Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)  
 For possible splitpoint  $k=(i+1)..(j-1)$ :  
 For every B in  $[i,k]$  and C in  $[k,j]$ ,  
 If exists rule  $A \rightarrow B C$ ,  
add A to cell  $[i,j]$  (Recognizer)  
 ... or ...  
add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

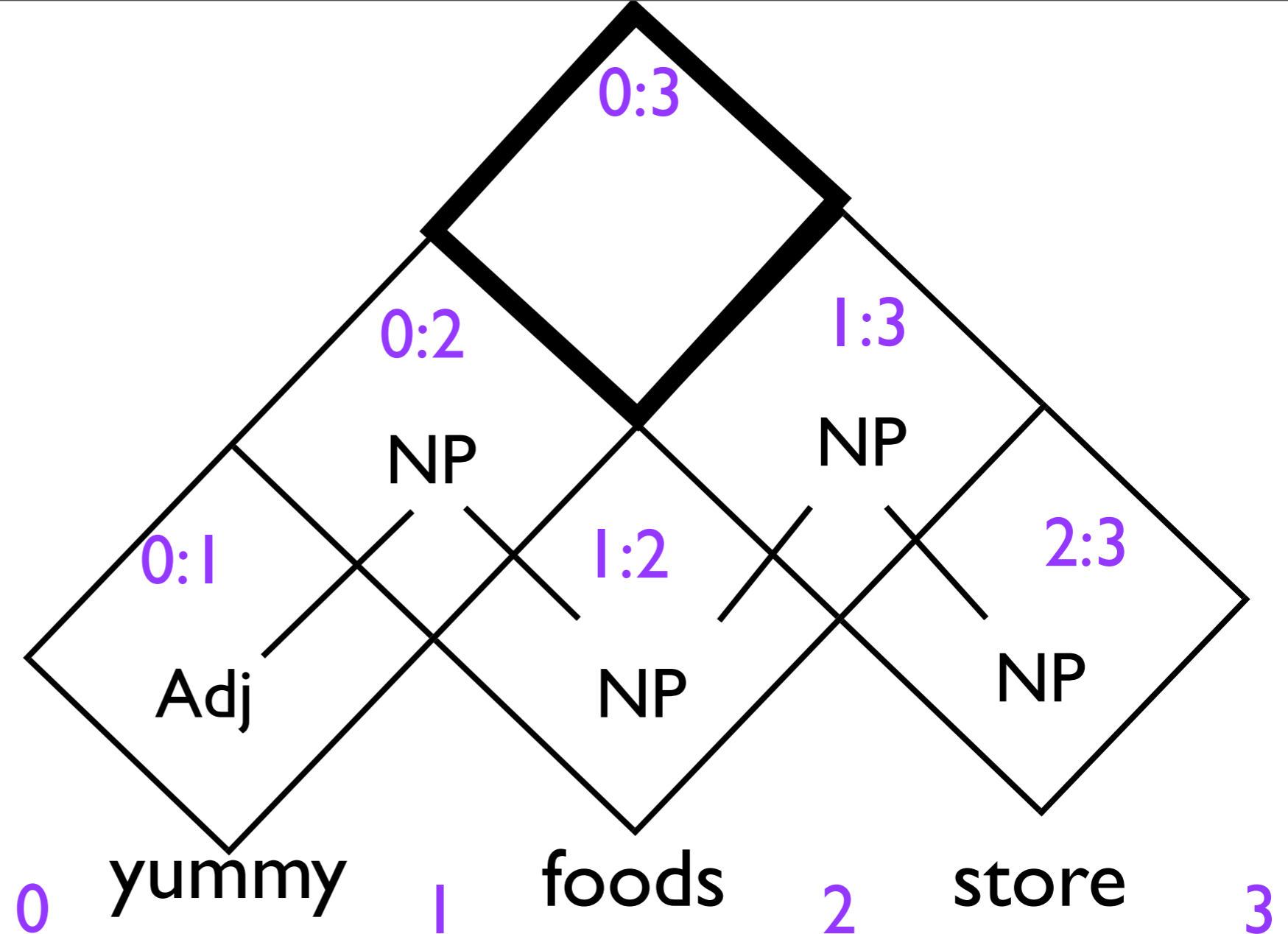
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

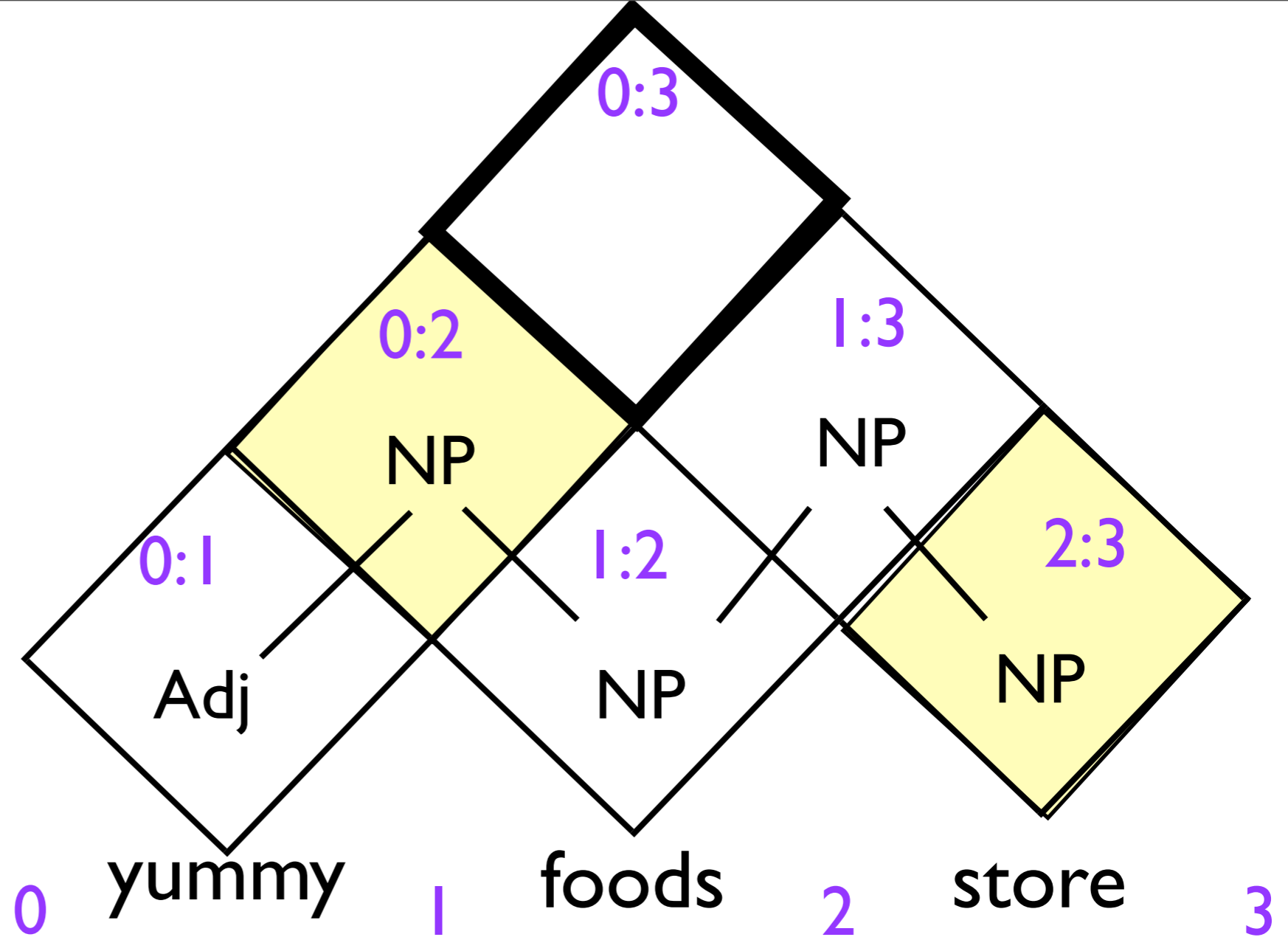
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

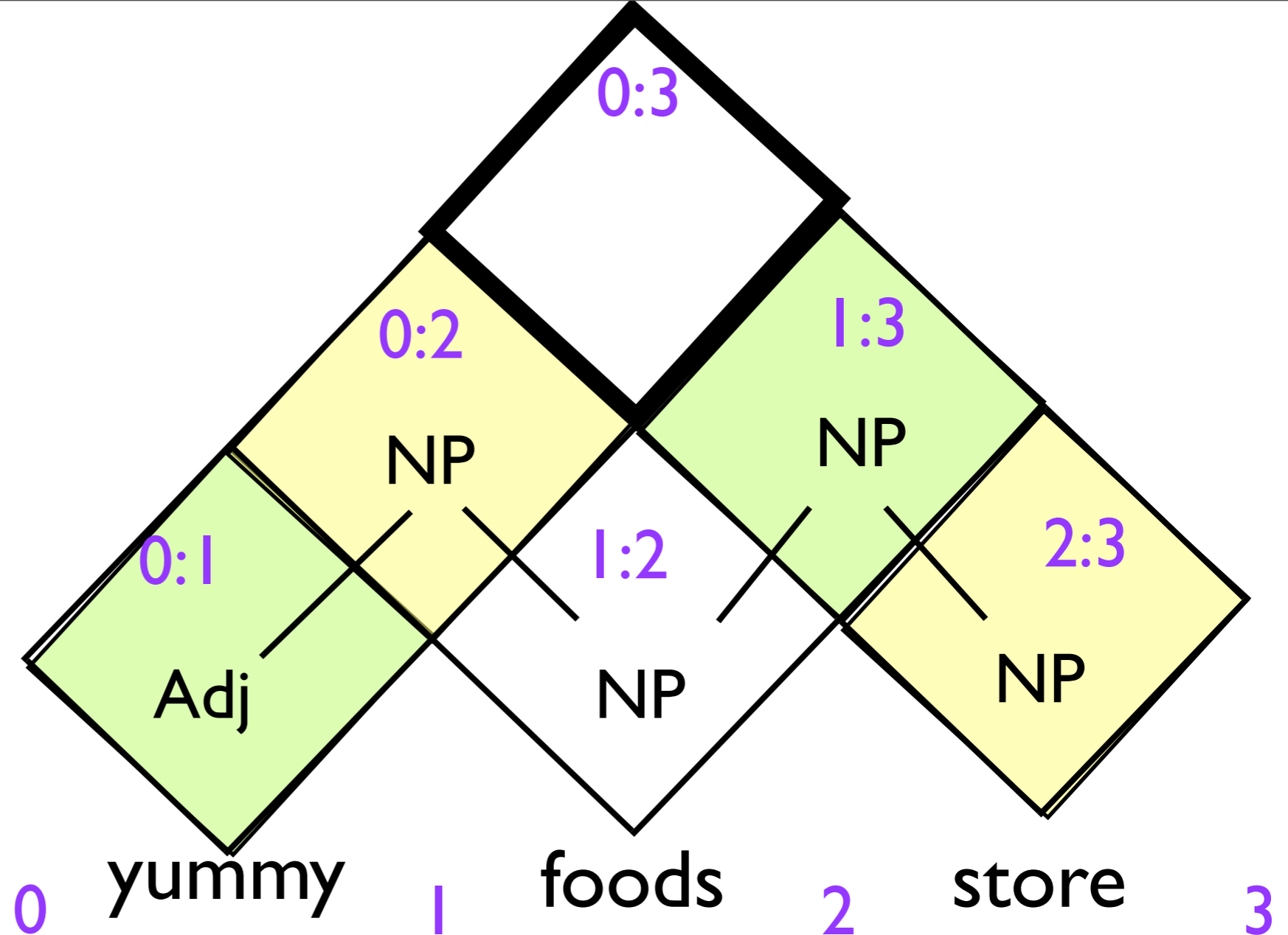
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

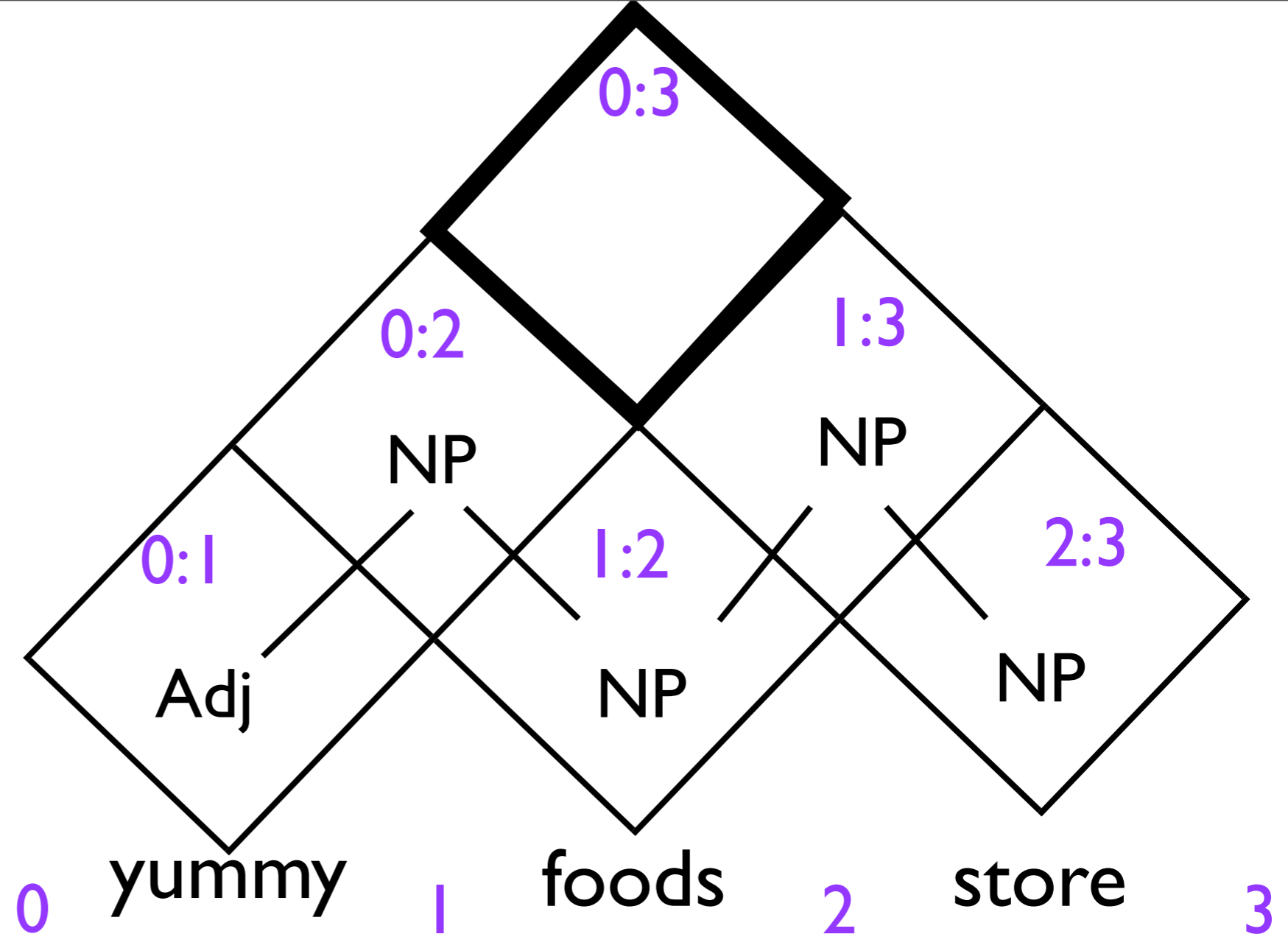
Adj  $\rightarrow$  yummy

NP  $\rightarrow$  foods

NP  $\rightarrow$  store

NP  $\rightarrow$  NP NP

NP  $\rightarrow$  Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

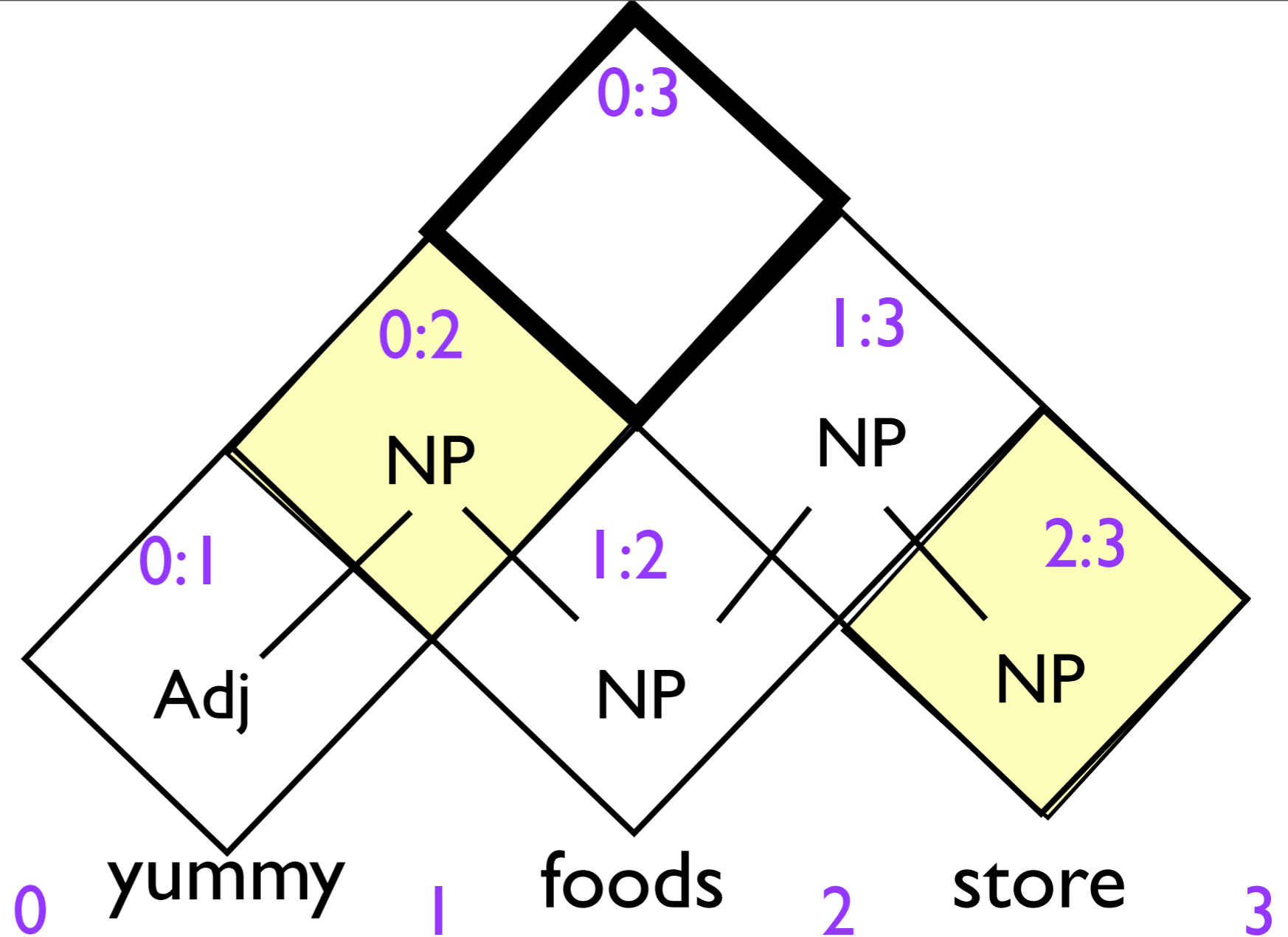
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

## Grammar

Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

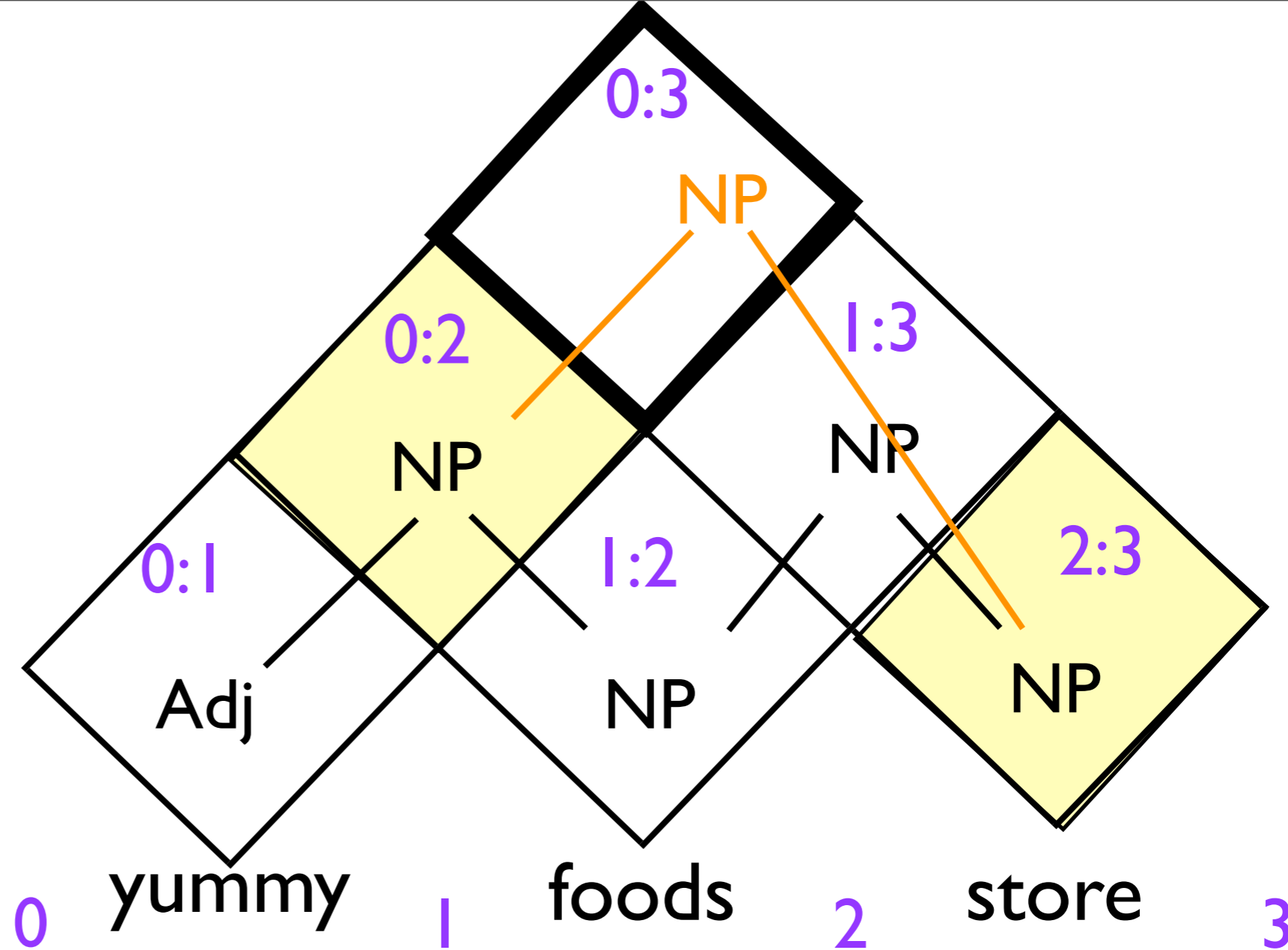
add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

Grammar  
 Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)  
 For possible splitpoint  $k=(i+1)..(j-1)$ :  
 For every B in  $[i,k]$  and C in  $[k,j]$ ,  
 If exists rule  $A \rightarrow B C$ ,  
add A to cell  $[i,j]$  (Recognizer)  
 ... or ...  
add (A,B,C, k) to cell  $[i,j]$  (Parser)

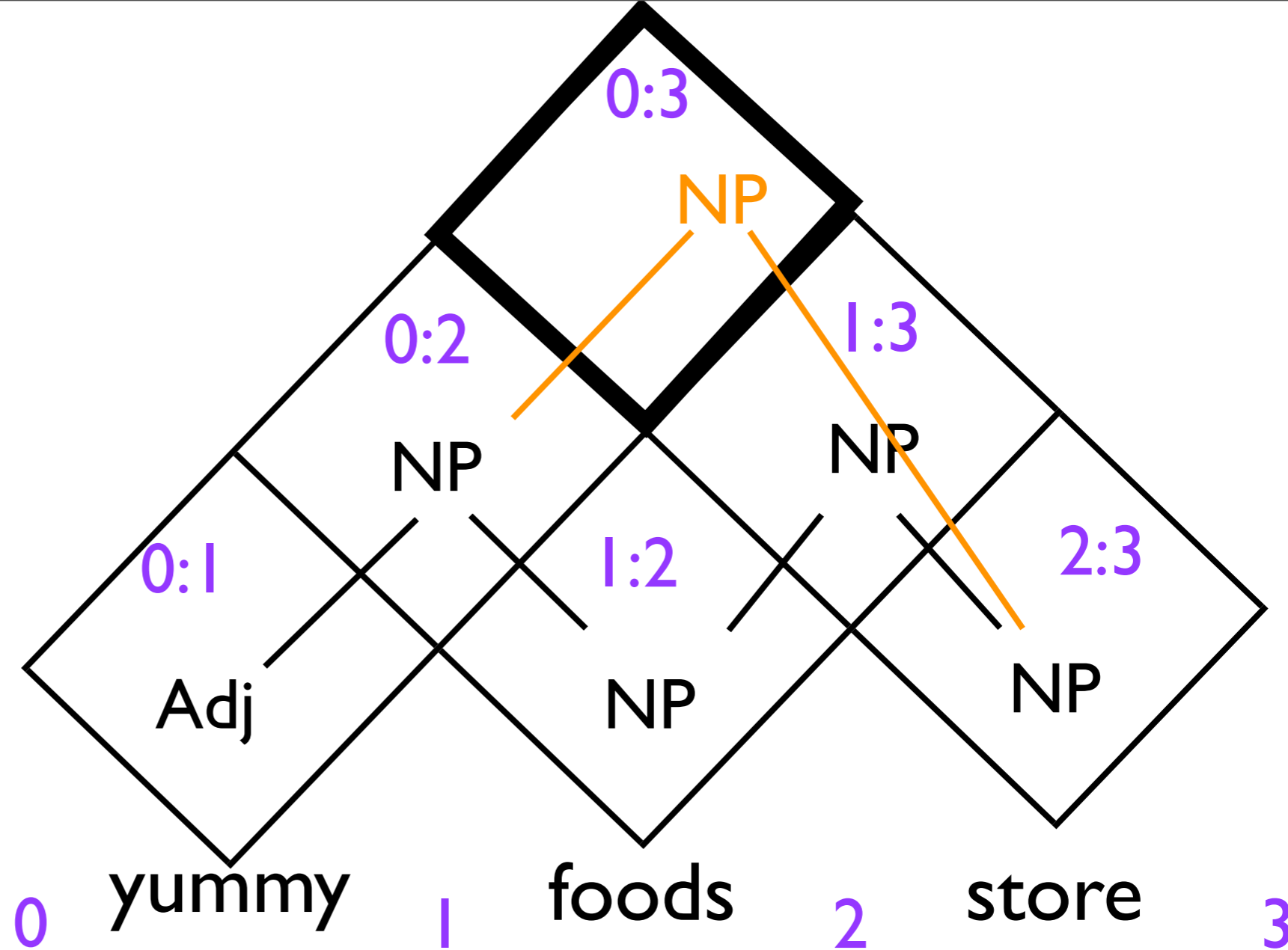
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

Grammar

Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)  
 For possible splitpoint  $k=(i+1)..(j-1)$ :  
 For every B in  $[i,k]$  and C in  $[k,j]$ ,  
 If exists rule  $A \rightarrow B C$ ,  
add A to cell  $[i,j]$  (Recognizer)  
 ... or ...  
add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.



# CKY

## Grammar

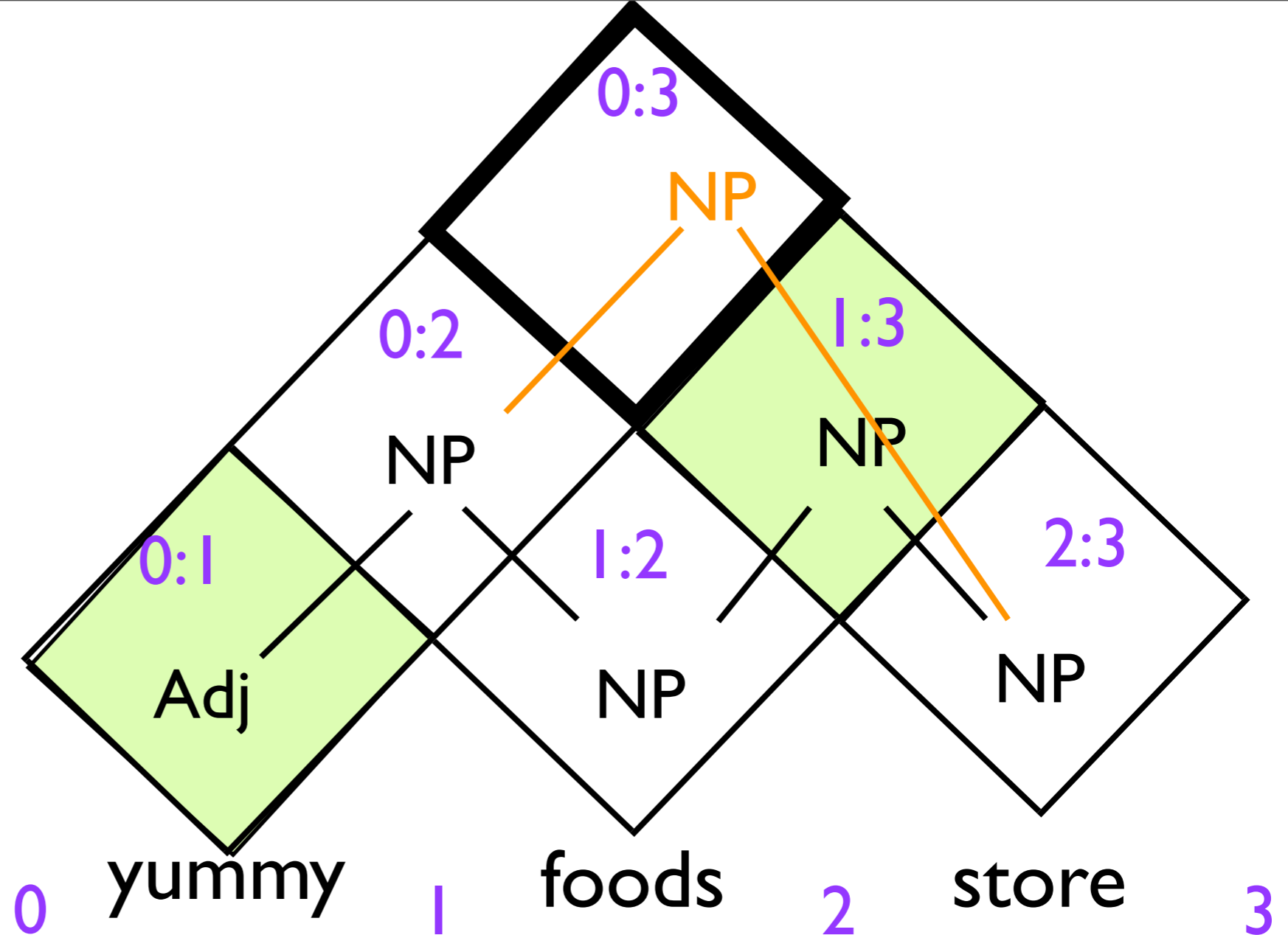
Adj -> yummy

NP -> foods

NP -> store

NP -> NP NP

NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every B in  $[i,k]$  and C in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add A to cell  $[i,j]$  (Recognizer)

... or ...

add (A,B,C, k) to cell  $[i,j]$  (Parser)

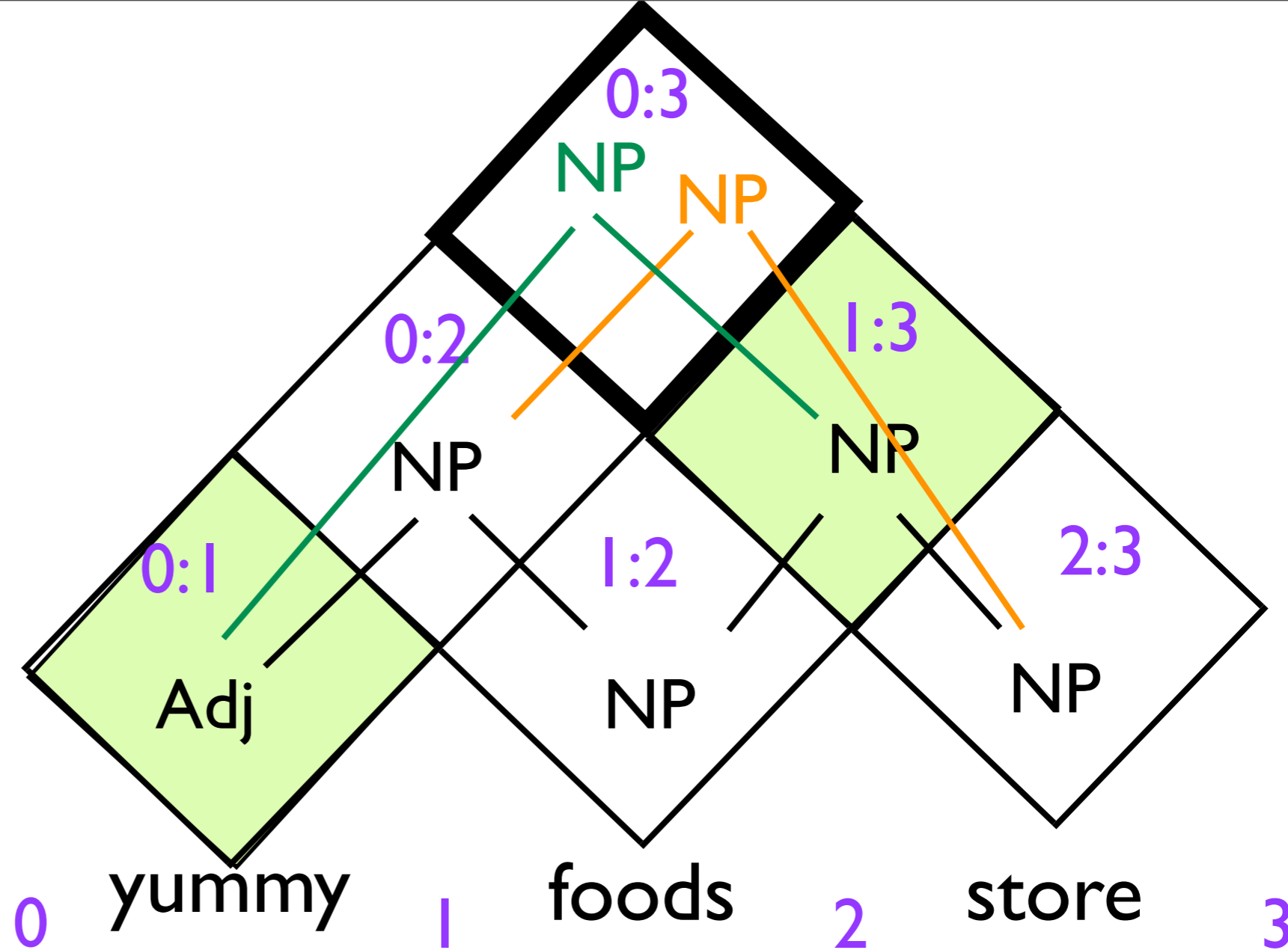
Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

# CKY

Grammar

Adj -> yummy  
 NP -> foods  
 NP -> store  
 NP -> NP NP  
 NP -> Adj NP



For cell  $[i,j]$  (loop through them bottom-up)  
 For possible splitpoint  $k=(i+1)..(j-1)$ :  
 For every B in  $[i,k]$  and C in  $[k,j]$ ,  
 If exists rule  $A \rightarrow B C$ ,  
add A to cell  $[i,j]$  (Recognizer)  
 ... or ...  
add (A,B,C, k) to cell  $[i,j]$  (Parser)

Recognizer: per span, record list of possible nonterminals

Parser: per span, record possible ways the nonterminal was constructed.

For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

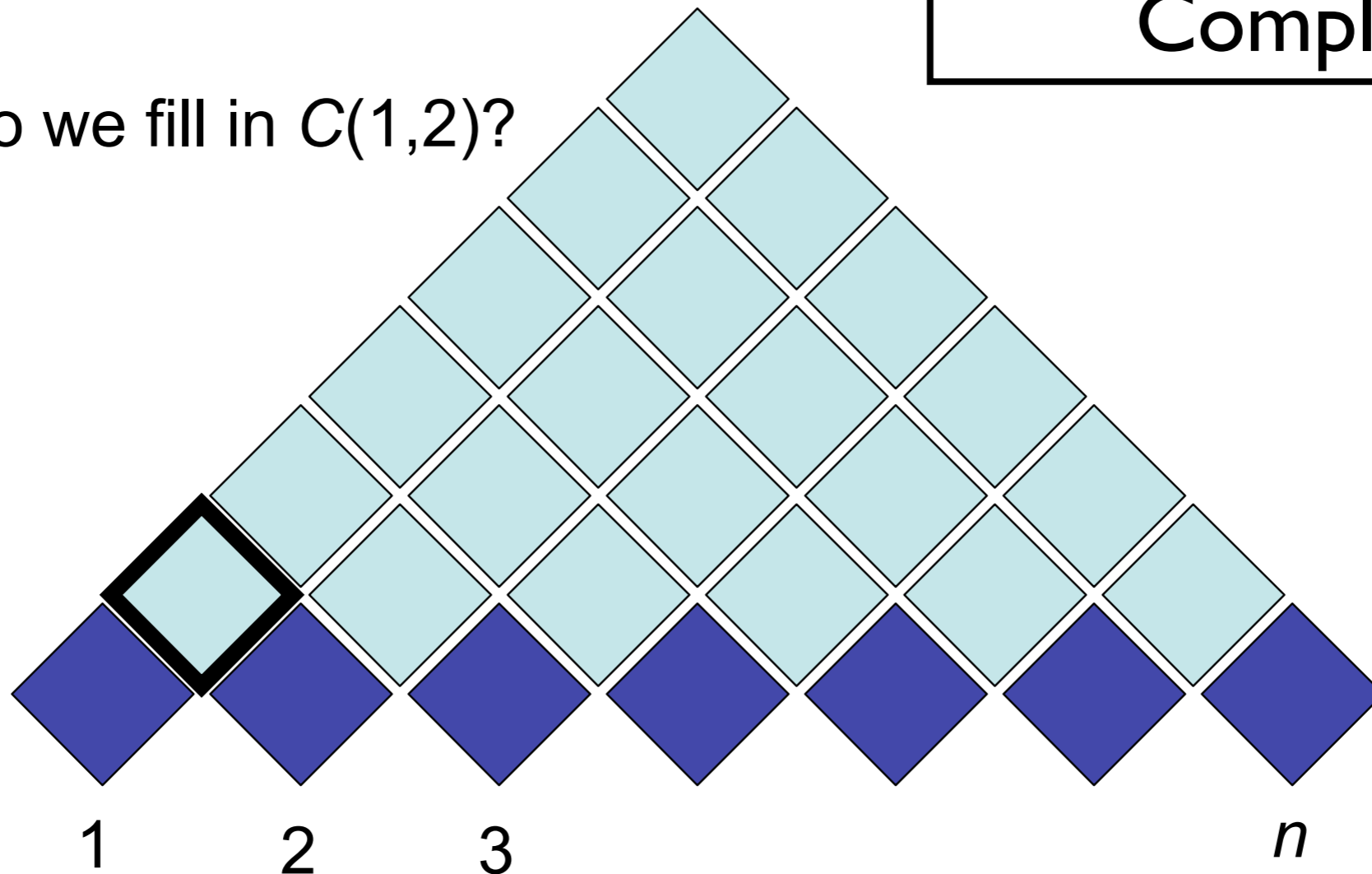
For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,2)$ ?



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

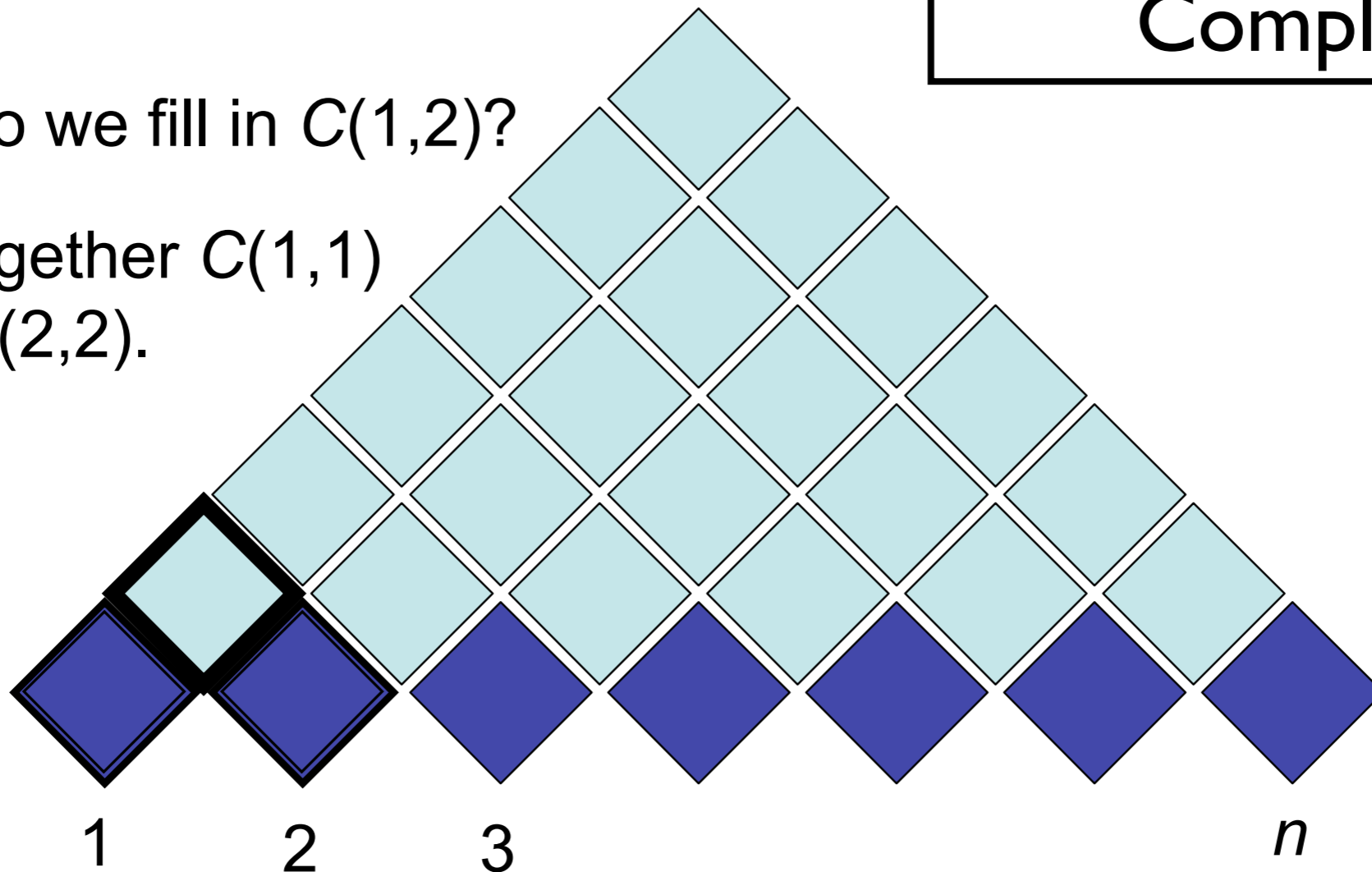
If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,2)$ ?

Put together  $C(1,1)$   
and  $C(2,2)$ .



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

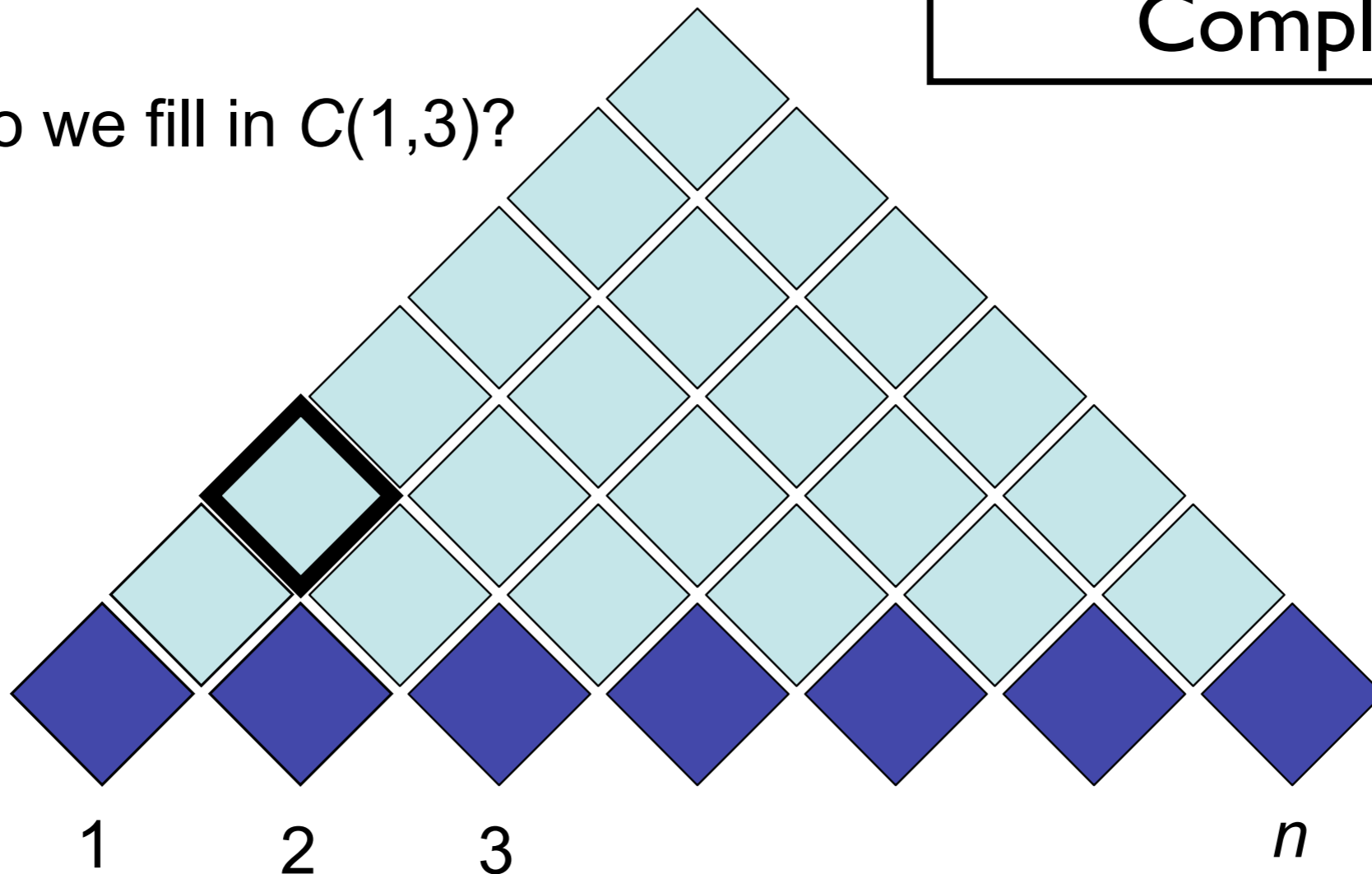
For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,3)$ ?



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

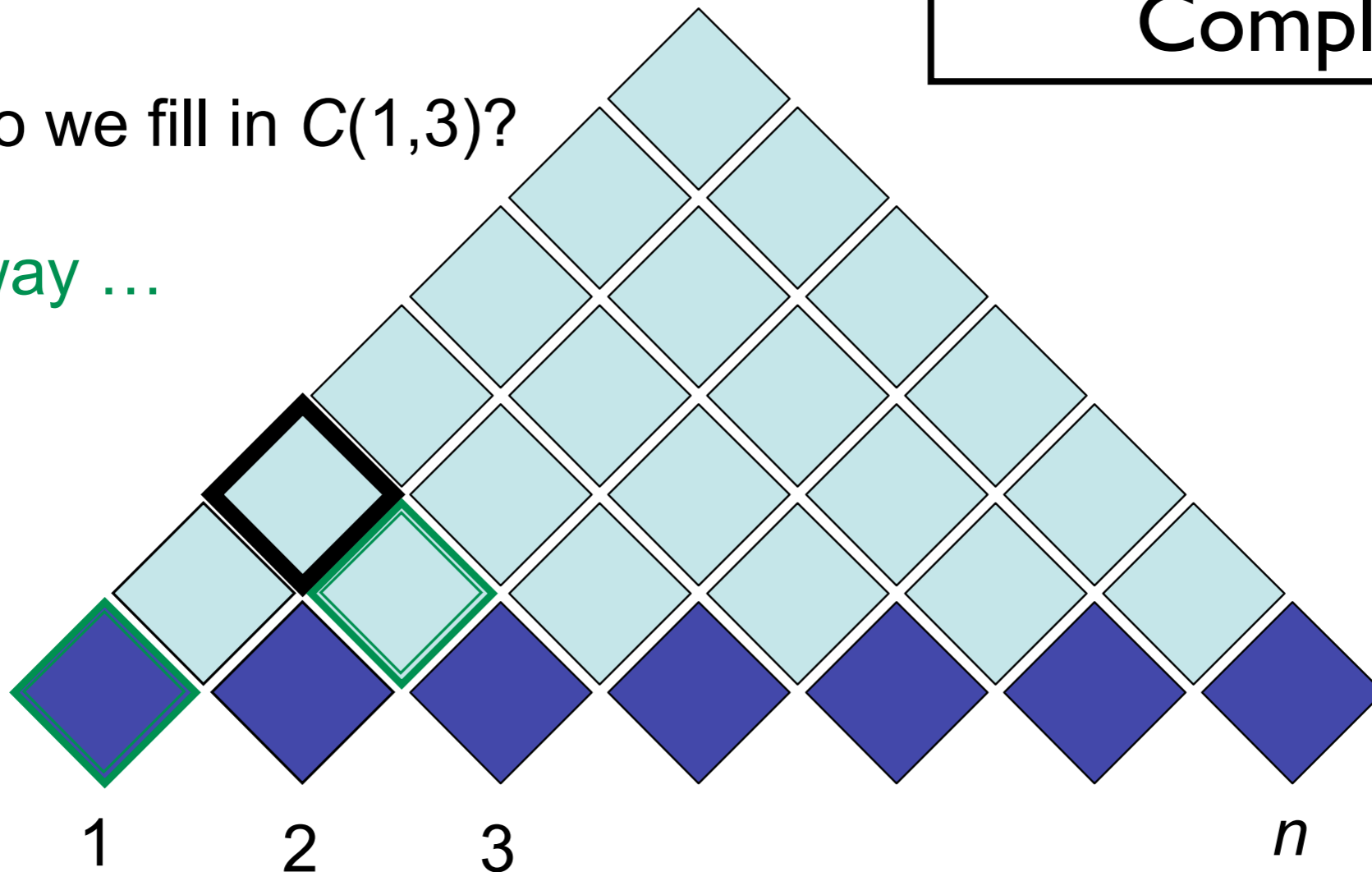
If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,3)$ ?

One way ...



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

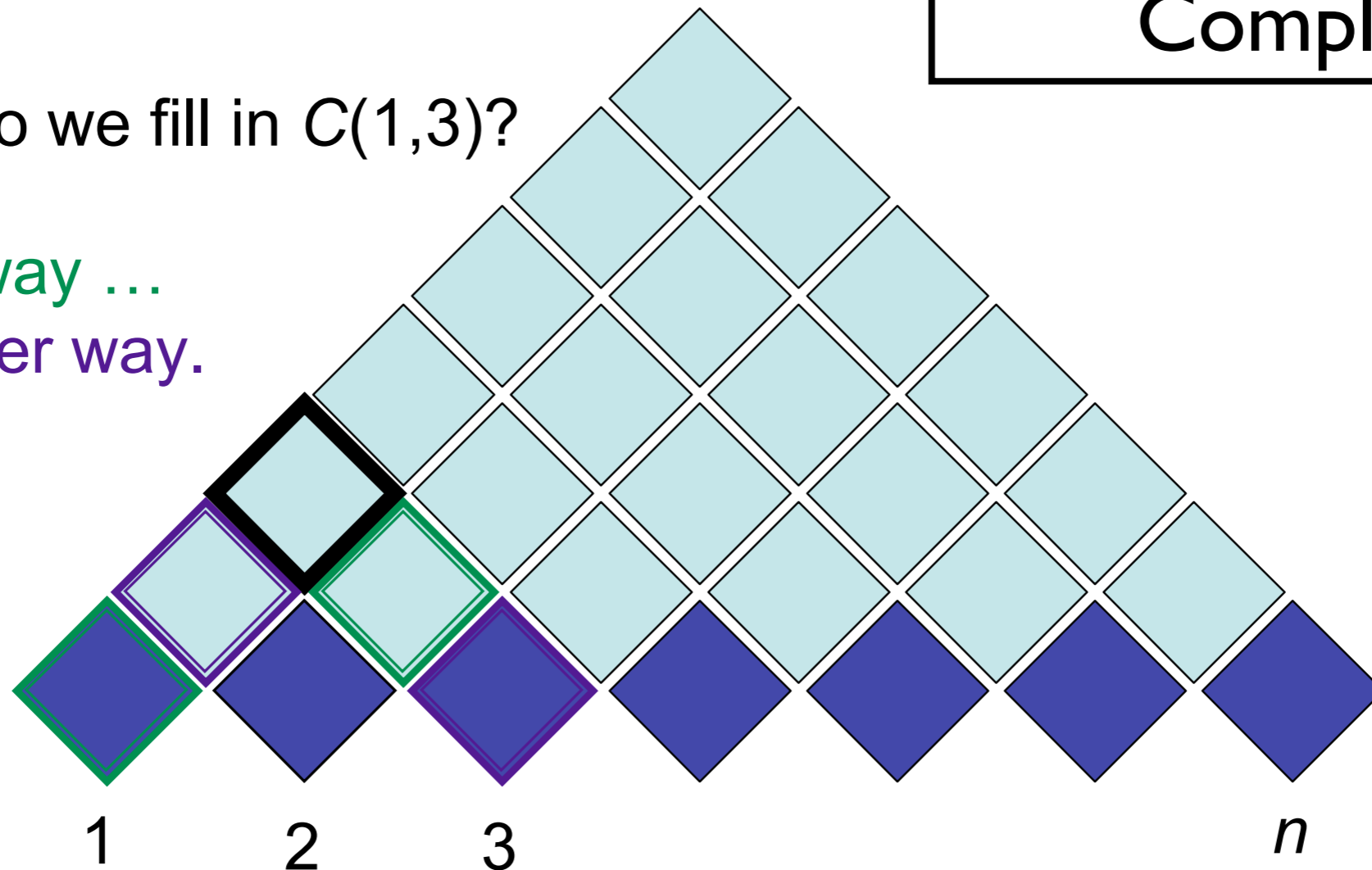
add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,3)$ ?

One way ...

Another way.



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

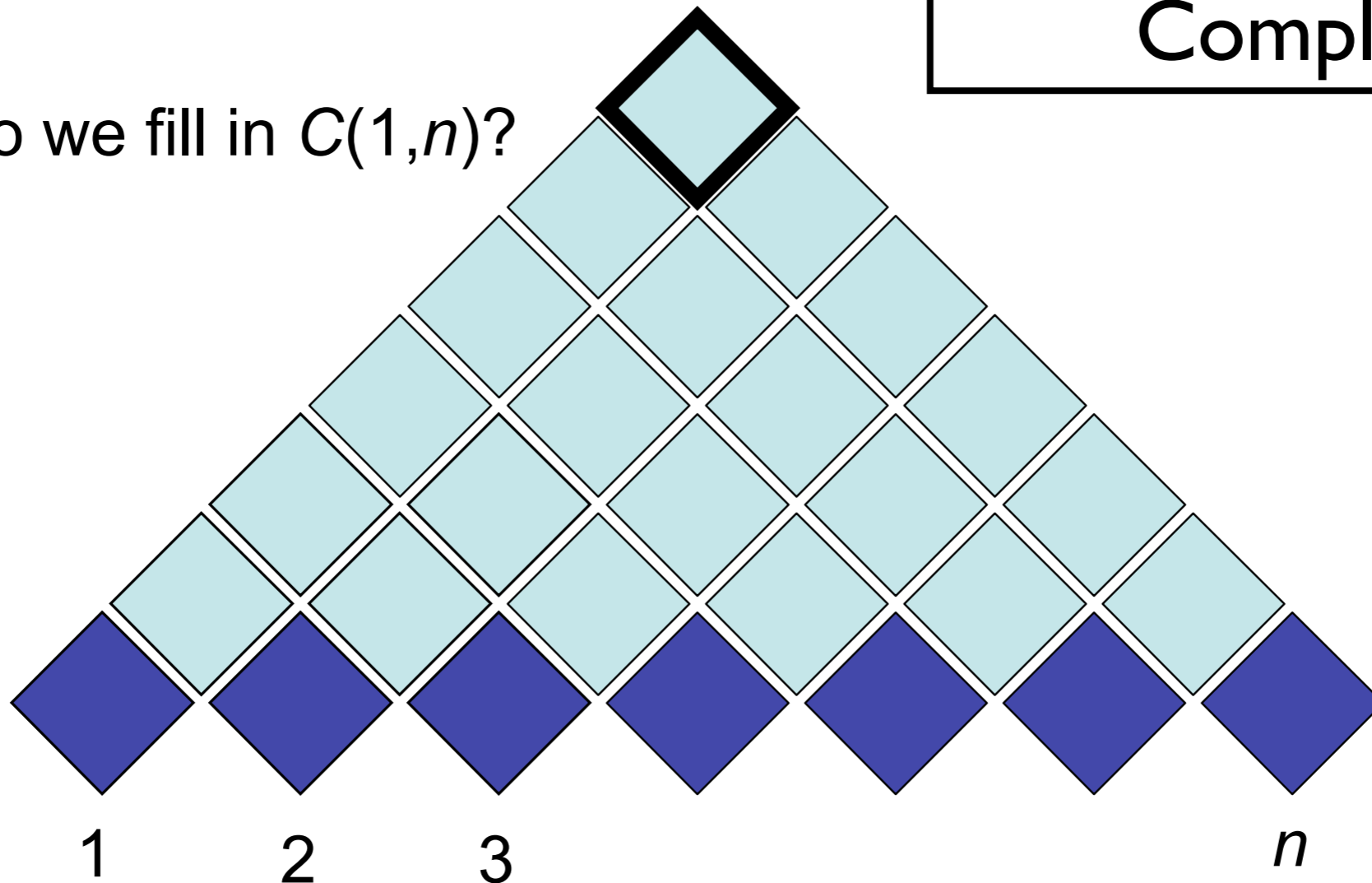
For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,n)$ ?





For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

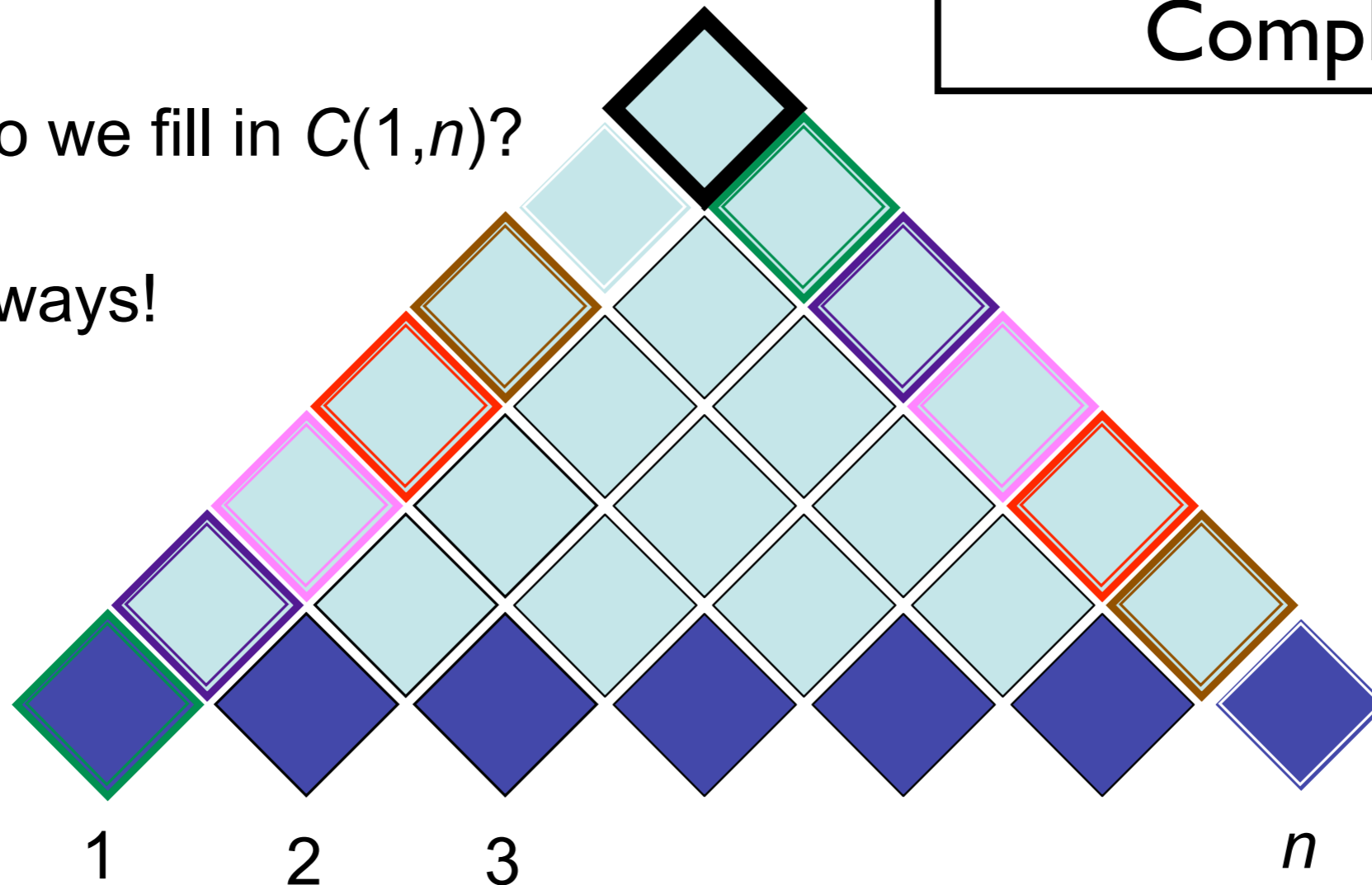
If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,n)$ ?

$n - 1$  ways!



For cell  $[i,j]$

For possible splitpoint  $k=(i+1)..(j-1)$ :

For every  $B$  in  $[i,k]$  and  $C$  in  $[k,j]$ ,

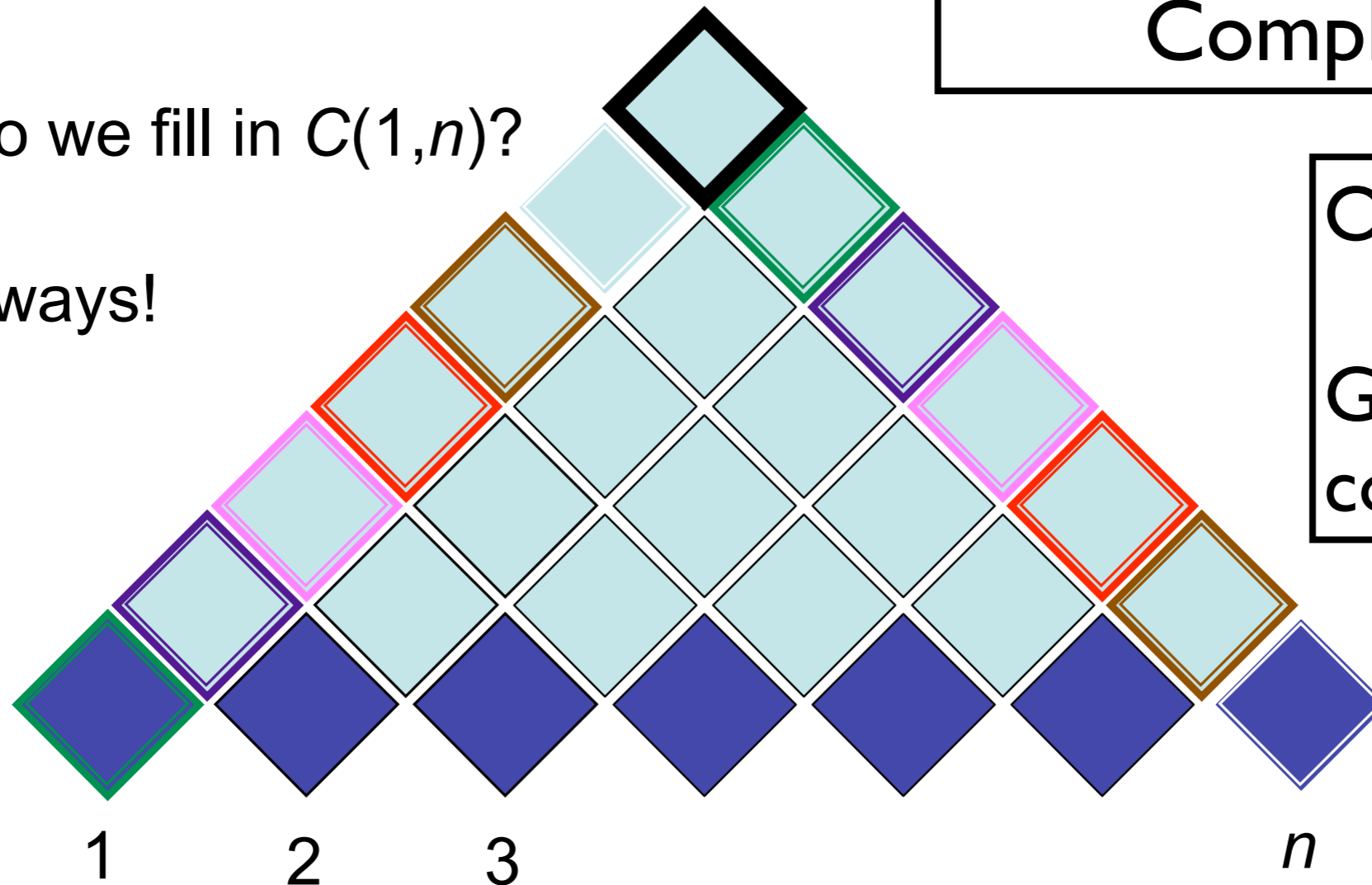
If exists rule  $A \rightarrow B C$ ,

add  $A$  to cell  $[i,j]$

Computational  
Complexity ?

How do we fill in  $C(1,n)$ ?

$n - 1$  ways!



$O(G n^3)$

$G =$  grammar  
constant