COMP90042
Web search and text analysis

Workshop Week 10

xudong.han@unimelb.edu.au
https://github.com/HanXudong/COMP90042_Workshops

regular grammar

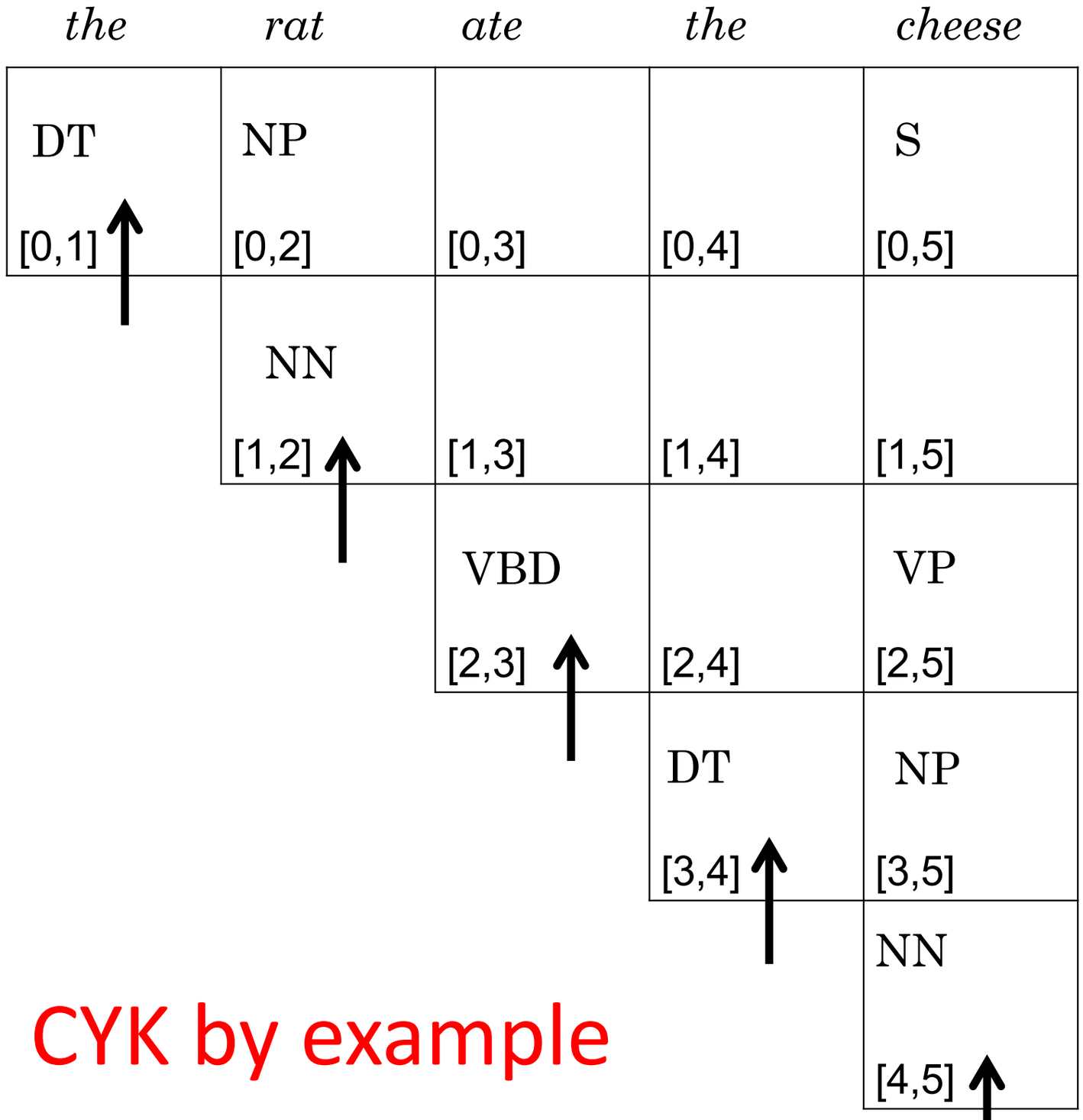
- For example: a simple regular grammar
- Rules: S -> A, A -> aA, A-> ϵ
- S is the start symbol
- It will generate words such as a, aa, aaa, aaa
- The set of words generated by this regular grammar is a regular language
- This regular language can also be expressed in regular expression a*

Q1 What is chart parsing? Why is it important?

Parsing CFGs

- Parsing: given string, identify possible structures
- Brute force search is intractable for non-trivial grammars
  - Good solutions use dynamic programming
- Two general strategies
  - Bottom-up
    - Start with words, work up towards S
    - CYK parsing

Q1 What is chart parsing? Why is it important?



S → NP VP
NP → DT NN
VP → VBD NP
DT → *the*
NN → *rat*
NN → *cheese*
VBD → *ate*

CYK by example

Q2. What changes need to be made to grammar?

```
 S   ->  NP   VP
VP   ->   V   NP      |     V      NP      PP
PP   ->   P   NP
 V   ->  saw   |   walked
NP   ->  Joh   |     Bob    |     Det      N     |   Det  N  PP
Det  ->   a    |      an    |      the     |    my
 N   ->  ma    |     cat    |  telescope   |   park
 P   ->  on    |      by    |      with
```

- **Productions** (rules)

$$W \rightarrow X\ Y\ Z$$

  - Exactly one non-terminal on left-hand side (LHS)
  - An ordered list of symbols on right-hand side (RHS)
    — can be **Terminals** or **Non-terminals**

Q2. What changes need to be made to grammar?

S	->	NP	VP					
VP	->	V	NP	|	V	NP	PP	
PP	->	P	NP					
V	->	saw	|	walked				
NP	->	Joh	|	Bob	|	Det	N	| Det N PP
Det	->	a	|	an	|	the	|	my
N	->	ma	|	cat	|	telescope	|	park
P	->	on	|	by	|	with		

CYK parsing:

Convert grammar to Chomsky Normal Form (CNF)

- Change grammar so all rules of form
  $A \rightarrow B\ C$ or $A \rightarrow a$

Q2-a-(i) “a man saw John”

0 1 2 3 4

a	man	saw	John
[0,1]	[0,2]	[0,3]	[0,4]
	[1,2]	[1,3]	[1,4]
		[2,3]	[2,4]
			[3,4]

S -> NP VP

VP -> V NP | V X

PP -> P NP

V -> saw | walked

NP -> John | Bob | Det N | Det Y

Det -> a | an | the | my

N -> man | cat | telescope | park

P -> on | by | with

X -> NP PP

Y -> N PP

Q2-a-(i) “a man saw John”

0 1 2 3 4

a	man	saw	John
[0,1]	[0,2]	[0,3]	[0,4]
	[1,2]	[1,3]	[1,4]
		[2,3]	[2,4]
			[3,4]

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

Q2-a-(iii) “park by the cat with my telescope”

park	*by*	*the*	*cat*	*with*	*my*	*telescope*
[0,1] N	[0,2] -	[0,3] -	[0,4] Y	[0,5] -	[0,6] -	[0,7] Y
	[1,2] P	[1,3] -	[1,4] PP	[1,5] -	[1,6] -	[1,7] PP
		[2,3] Det	[2,4] NP	[2,5] -	[2,6] -	[2,7] NP, X
			[3,4] N	[3,5] -	[3,6] -	[3,7] Y
				[4,5] P	[4,6] -	[4,7] PP
					[5,6] Det	[5,7] NP
						[6,7] N

Q2-a-(ii) “an park by Bob walked an park with Bob”

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP, X	[0,5] -	[0,6] -	[0,7] S	[0,8] -	[0,9] S, S
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] -	[1,8] -	[1,9] -
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] -	[2,8] -	[2,9] -
			[3,4] NP	[3,5] -	[3,6] -	[3,7] S	[3,8] -	[3,9] S, S
				[4,5] V	[4,6] -	[4,7] VP	[4,8] -	[4,9] VP, VP
					[5,6] Det	[5,7] NP	[5,8] -	[5,9] NP, X
						[6,7] N	[6,8] -	[6,9] Y
							[7,8] P	[7,9] PP
								[8,9] NP

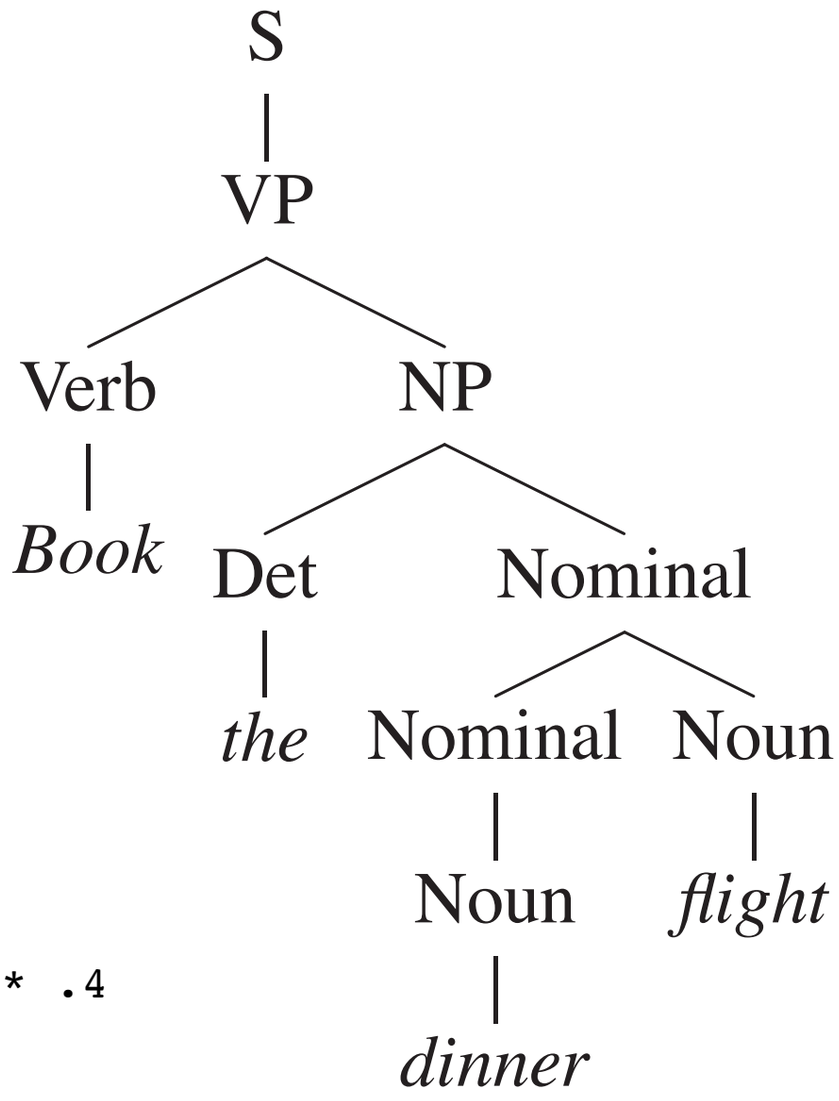
Q3 and Q4

- What differentiates probabilistic parsing from chart parsing? Why is this important? How does this affect the algorithms used for parsing?

- What is a probabilistic grammar and what problem does it attempt to solve?

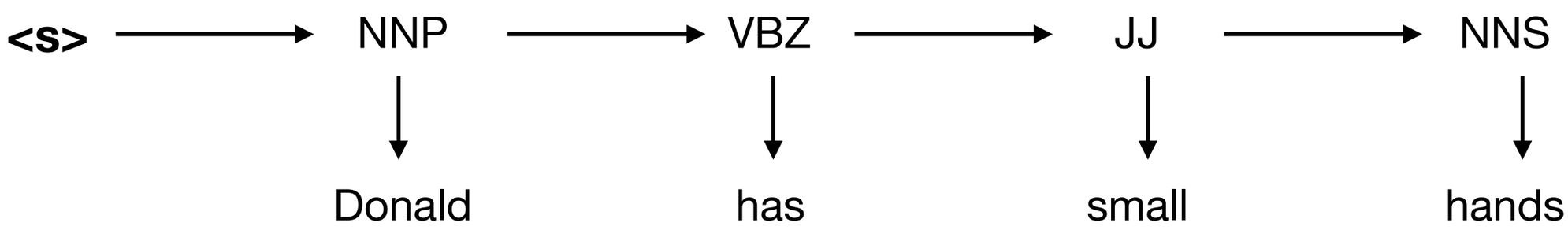
Q3 and Q4

- Given a tree, we can compute its probability
  - Decomposes into probability of each production
- E.g., for tree on right,
  - P(tree) =
    P(S → VP) ×
    P(VP → Verb NP) ×
    P(Verb → *Book*) ×
    P(NP → Det Nominal) ×
    P(Det → *the*) ×
    P(Nominal → Nominal Noun) ×
    P(Nominal → Noun) ×
    P(Noun → *dinner*) ×
    P(Noun → *flight*) = 2.16×10^{-6}




```
I.e., .05 * .2 * .3 * .2 * .6 * .2 * .75 * .1 * .4
```

Q5



Transition	Emission
<s> -> NNP NNP’	NNP -> Donald
NNP’ -> VBZ VBZ’	VBZ -> has
VBZ’ -> JJ JJ’	JJ -> small
JJ’ -> NNS	NNS -> hands