



アルゴリズムとデータ構造III

5回目: 11月12日

構文解析 CKY法, 動的計画法

授業資料 <http://ir.cs.yamanashi.ac.jp/~ysuzuki/algorithm3/index.html>

授業の予定(中間試験まで)

1	10/01	スタック(後置記法で書かれた式の計算)
2	10/15	文脈自由文法, 構文解析, CYK法
3	10/22	構文解析 CYK法
4	10/29	構文解析 CYK法
5	11/12	構文解析 CYK法, 動的計画法
6	11/	構文解析(チャート法), グラフ(ダイクストラ法, DPマッチング)
7	11/	グラフ(DPマッチング, A*アルゴリズム)
8	11/19	グラフ(A*アルゴリズム), 前半のまとめ
9	11/26	中間試験

10/08, 11/05の代わりにの補講日は後日相談

授業の予定(中間試験以降)

10	12/03	全文検索アルゴリズム (simple search, KMP)
11	12/10	全文検索アルゴリズム (BM, Aho-Corasick)
12	12/17	全文検索アルゴリズム (Aho-Corasick), データ圧縮
13	01/07	暗号 (黄金虫, 踊る人形) 符号化 (モールス信号, Zipfの法則, ハフマン符号) テキスト圧縮
14	01/14	テキスト圧縮 (zip), 音声圧縮 (ADPCM, MP3, CELP), 画像圧縮 (JPEG)
15	01/21	期末試験



本日のメニュー

- 構文解析
 - CYK法(続き)
- 動的計画法
 - (ダイクストラ法)

練習問題1

- CYK法を使って“*I eat pizza with Nana*”の構文解析結果を作成しなさい。

	(1)	S	→	N	V
	(2)	S	→	S	PP
	(3)	S	→	V	N
	(4)	V	→	V	N
	(5)	PP	→	P	N
	(6)	N	→	N	PP
<hr/>					
	(7)	N	→	I	
	(8)	N	→	Nana	
	(9)	N	→	pizza	
	(10)	V	→	eat	
	(11)	P	→	with	

A → BC

A → a (辞書規則)

CYK法で構文解析

I eat pizza with Nana.

	1. I	2. eat	3. pizza	4. with	5. Nana
1. I	N → I				
2. eat		V → eat			
3. pizza			N → pizza		
4. with				P → with	
5. Nana					N → Nana

(1)	S	→	N	V	
(2)	S	→	S	PP	• N → I
(3)	S	→	V	N	• N → Nana
(4)	V	→	V	N	• N → pizza
(5)	PP	→	P	N	• V → eat
(6)	N	→	N	PP	• P → with

CYK法で構文解析

I eat pizza with Nana.

	1. I	2. eat	3. pizza	4. with	5. Nana
1. I	N → I	S → N V			
2. eat		V → eat	S → V N V → V N		
3. pizza			N → pizza		
4. with				P → with	PP → P N
5. Nana					N → Nana

(1)	S	→	N V		
(2)	S	→	S PP	• N → I	
(3)	S	→	V N	• N → Nana	
(4)	V	→	V N	• N → pizza	
(5)	PP	→	P N	• V → eat	
(6)	N	→	N PP	• P → with	

CYK法で構文解析

I eat pizza with Nana.

	1. I	2. eat	3. pizza	4. with	5. Nana
1. I	N → I	S → N V	S → N V		
2. eat		V → eat	S → V N V → V N		
3. pizza			N → pizza		N → N PP
4. with				P → with	PP → P N
5. Nana	(1) S → N V (2) S → S PP (3) S → V N (4) V → V N (5) PP → P N (6) N → N PP		• N → I • N → Nana • N → pizza • V → eat • P → with		N → Nana

CYK法で構文解析

I eat pizza with Nana.

	1. I	2. eat	3. pizza	4. with	5. Nana
1. I	N → I	S → N V	S → N V		
2. eat		V → eat	S → V N V → V N		S → S PP S → V N V → V N
3. pizza			N → pizza		N → N PP
4. with				P → with	PP → P N
5. Nana	(1) S → N V (2) S → S PP (3) S → V N (4) V → V N (5) PP → P N (6) N → N PP		• N → I • N → Nana • N → pizza • V → eat • P → with		N → Nana

CYK法で構文解析

I eat pizza with Nana.

1. I 2. eat 3. pizza 4. with 5. Nana

1. I	N → I	S → N V	S → N V		S → N V S → S PP
2. eat		V → eat	S → V N V → V N		S → S PP S → V N V → V N
3. pizza			N → pizza		N → N PP
4. with				P → with	PP → P N
5. Nana					N → Nana

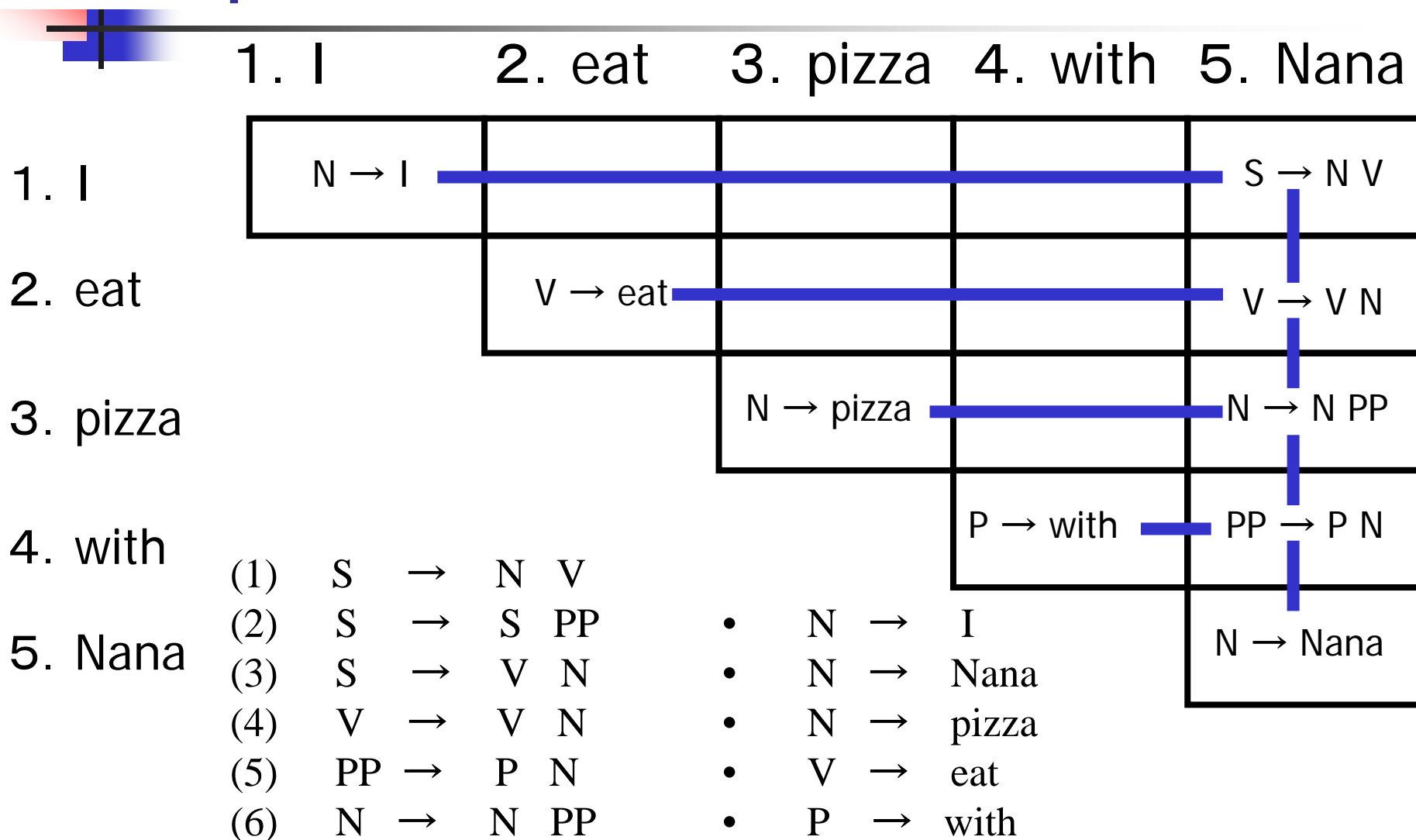
- (1) S → N V
- (2) S → S PP
- (3) S → V N
- (4) V → V N
- (5) PP → P N
- (6) N → N PP

- N → I
- N → Nana
- N → pizza
- V → eat
- P → with

CYK法で構文解析

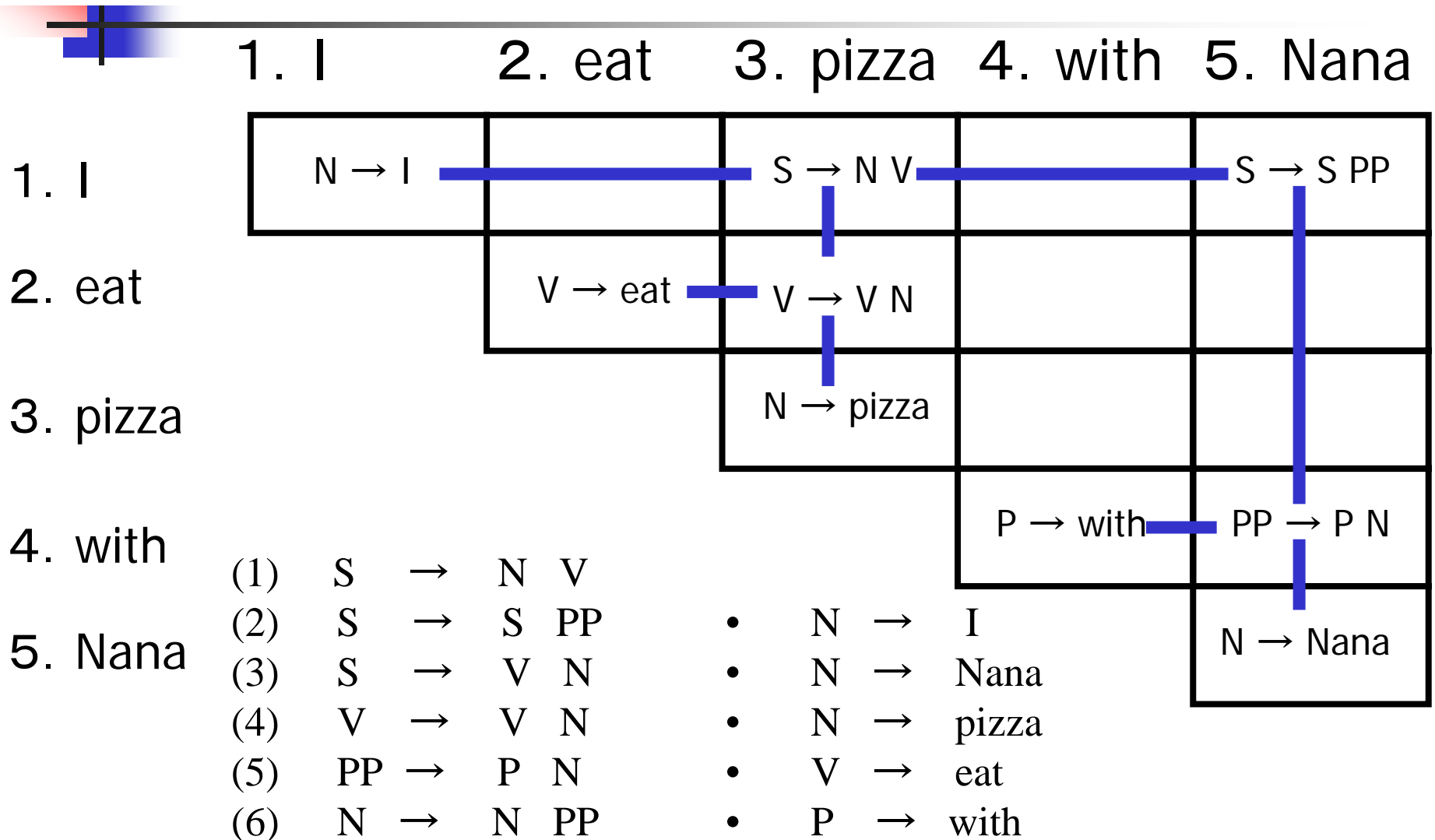
$S \rightarrow N V$ の構文木

I eat pizza with Nana.



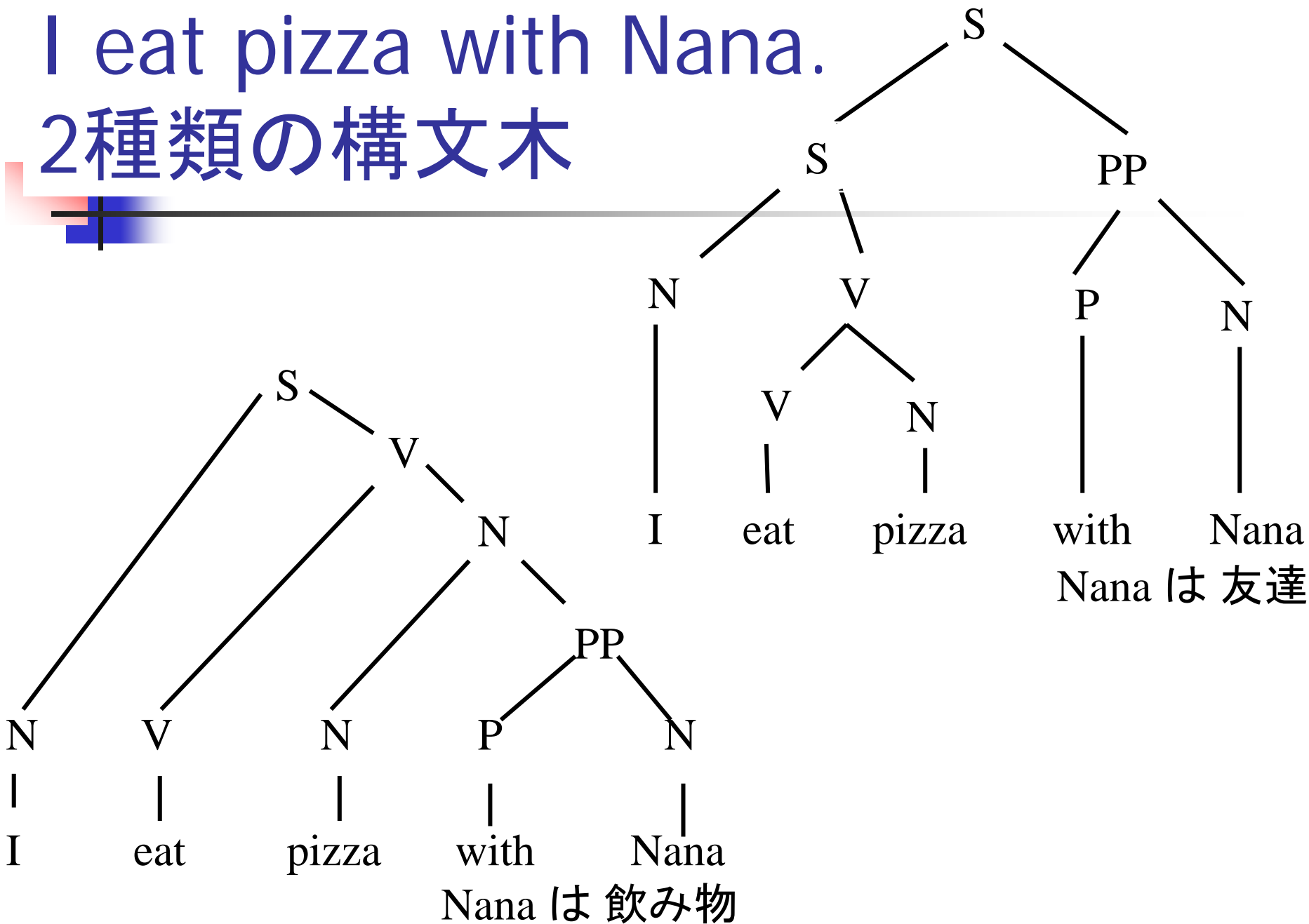
CYK法で構文解析 $S \rightarrow S PP$ の構文木

I eat pizza with Nana.



I eat pizza with Nana.

2種類の構文木



CYK法のアルゴリズム

adv→急いで B	vp→adv v A			
	v→走る C	np→v n		
		n→一郎	pp→n p	
			p→を	
				v→見た

- BとCの結果をAのマスを埋めるために利用
 - A→BC
- 部分問題の解をより大きな問題を解くために利用
- 同じ問題を2度解かなくても済むように解を格納

CYKアルゴリズム $T_{1,2}$ の計算

1. $A \rightarrow a$ の生成規則を用いて, 主対角線上の要素を計算

for $i=1$ to N

$$T_{i,j} = \{A \mid A \rightarrow w_i\}$$

2. $A \rightarrow BC$ の生成規則を用いて, 2番目以降の対角線上の要素を計算

for $n=1$ to $N-1$

for $i=1$ to $N-n$

$$T_{i,i+n} = \bigcup_{j=1}^n \{A \mid A \rightarrow BC, B \in T_{i,i+j-1}, C \in T_{i+j,i+n}\}$$

3. $S \in T_{1,N}$ であれば, $w_1 \cdots w_N$ は開始記号 S から導出可能

1,1	1,2	1,3	1,4	1,5
	2,2	2,3	2,4	2,5
		3,3	3,4	3,5
			4,4	4,5
				5,5

$T_{1,2}$ の計算

$i=1, n=1, j=1 (1 \leq j \leq n)$

$$T_{1,2} = A \rightarrow BC, B = T_{1,1+1-1} = T_{1,1}, C = T_{1+1,1+1} = T_{2,2}$$

CYKアルゴリズム $T_{2,4}$ の計算

2. $A \rightarrow BC$ の生成規則を用いて,2番目以降の対角線上の要素を計算

for $n=1$ to $N-1$

for $i=1$ to $N-n$

$$T_{i,i+n} = \bigcup_{j=1}^n \{A \mid A \rightarrow BC, B \in T_{i,i+j-1}, C \in T_{i+j,i+n}\}$$

1,1	1,2	1,3	1,4	1,5
	2,2	2,3	2,4	2,5
		3,3	3,4	3,5
			4,4	4,5
				5,5

$T_{2,4}$ の計算

$i=2, n=2$ ($4=2+n$), $j=1,2$ ($1 \leq j \leq n$)

$T_{2,4} = A \rightarrow BC, B = T_{2,2+j-1}, C = T_{2+j,2+2}$

$j=1$ の時 $T_{2,4} = A \rightarrow BC, B = T_{2,2}, C = T_{3,4}$

$j=2$ の時 $T_{2,4} = A \rightarrow BC, B = T_{2,3}, C = T_{4,4}$

構文解析アルゴリズム



■ ボトムアップアルゴリズム

■ 戦略

- 単語列から出発
- Sを導出 → 解析終了

■ 代表的なアルゴリズム

- CYK法
- LR法

■ トップダウンアルゴリズム

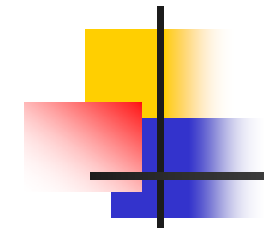
■ 戦略

- Sから出発
- 目的の単語列を導出 → 解析終了

■ 代表的なアルゴリズム

- アーリー法(Earley parser)
- LL法

動的計画法(Dynamic Programming)



- 部分問題の解をより大きな問題を解くために利用
- 同じ問題を2度解かなくても済むように解を格納

- アルゴリズムの例
 - CYK法(構文解析)
 - ダイクストラ法(最短経路問題)
 - DPマッチング(パターンマッチング DNAの解析にも利用)
 - DPを使った解法(ナップサック問題)
 - ビタビアルゴリズム(音声認識など)