

自然言語処理論 I

4.文法2(構文解析) その1

1

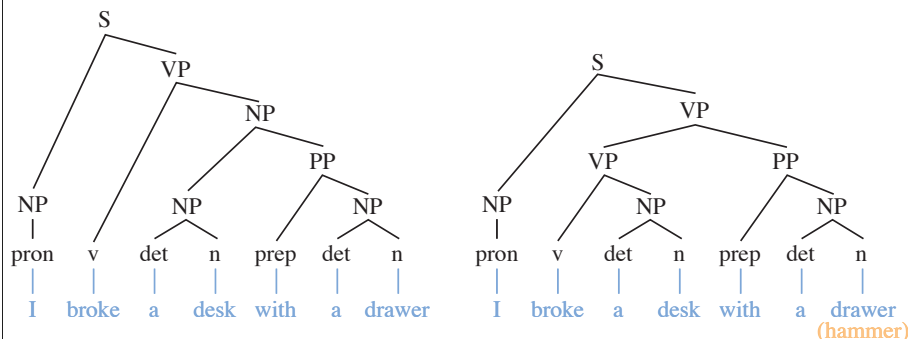
構文解析

- syntactic analysis, parsing
- 文の構文的な構造を決定すること
- 句構造文法が使われることが多い
- 文法による構文木は一般に複数ある
 - 構文木の違い=解釈の違い
- 構文解析の目的
 - 句構造文法の規則を使って、文を生成できる構文木を全て見つけだすこと
 - ◆ 文法が入力文を生成できるかどうかを調べるだけではない

2

構文解析とは

- 構文木の違い=解釈の違い



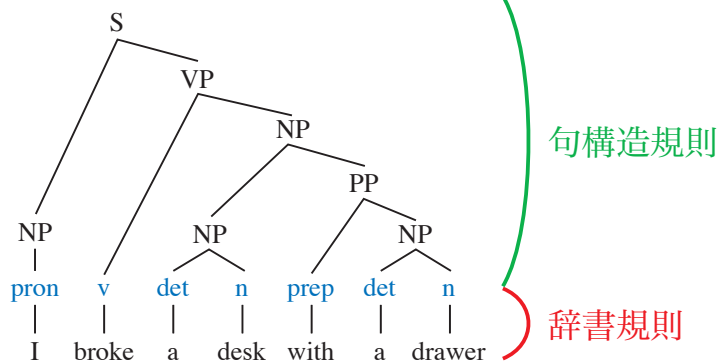
3

前終端記号と辞書規則

- 前終端記号(pre-terminal symbol)
 - 単語を生成する前のシンボル
 - 品詞に相当する
- 2種類の文法規則
 - 句構造規則
非終端記号と前終端記号からなる規則
 - 辞書規則
前終端記号→単語 という規則

4

(句構造規則)		(辞書規則)	
S→NP VP	VP→v	pron→I	
NP→pron	VP→v NP	det→a	
NP→det n	VP→VP PP	v→broke	
NP→NP PP	PP→prep NP	prep→with	
		n→desk, drawer	



5

構文解析の入力

- 単語列を入力とする
 - 膨大な数の辞書規則が必要
- 品詞列を入力とする
 - 単語と品詞の対応付けは形態素解析で行う
 - 前終端記号を終端記号として扱う
 - 句構造規則のみ使う
- 実際には品詞列を入力とする場合が多い

6

構文解析アルゴリズム

- 大きく分けて2種類ある
 - トップダウンアルゴリズム
 - ◆ 開始記号(S)から品詞列へ
 - ボトムアップアルゴリズム
 - ◆ 品詞列から開始記号(S)へ
- 代表的な構文解析アルゴリズム
 - CKY法, チャート法, アーリー法
再帰遷移ネットワーク, 一般化LR法

7

CKY法

- Cocke-Kasami-Younger法
- 文法はチョムスキー標準形に限る
- CKY表
 - 構文解析の途中経過を保持するためのテーブル

8

CKY表

	1	2	3	4	5	6	7	8	
1									
the	2								
man	3								
broke	4								
a	5								
desk	6								
with	7								
a	8								
drawer	9								

- 行列の対角成分に入力文の単語を置く
- a_{ij} は、入力文の*i*番目から*j*番目の単語を支配する非終端記号を置く

CKY法のアルゴリズム

- for $i:=1$ to n do
 $a_{ii} = \{A_a \mid A \rightarrow w_i\}$ (辞書規則の適用)
- for $d:=1$ to $n - 1$ do
 for $i:=1$ to $n - d$ do
 $j=i+d$
 for $k=i$ to $j - 1$ do
 a_{ij} に $\{A_a(B_b, C_c) \mid A \rightarrow B C, B_b \in a_{ik}, C_c \in a_{k+1, j}\}$ を追加
- a_{1n} に開始記号Sがあれば解析成功, それ以外は失敗
 (nは入力単語数、a,b,cは識別番号)

CKY法のアルゴリズム

- 表を埋める順序は以下の通り
 - 変数dが1つの対角線を表す

1									
the	2								
man	3								
broke	4								
a	5								
desk	6								
with	7								
a	8								
drawer	9								

CKY法のアルゴリズム

- マス目 a_{ij} を埋める時、以下の順序で2つの要素の組み合わせをチェック

	1	2	3	4	5	6	7	8	
1									
the	2	①	②	③	④	a_{ij}			
man	3					①			
broke	4					②			
a	5					③			
desk	6					④			
with	7								
a	8								
drawer	9								

- 数字は変数kに対応
- a_{26} の場合
 $(a_{22}, a_{36}), (a_{23}, a_{46}), (a_{24}, a_{56}), (a_{25}, a_{66})$ が規則の右辺になるかどうかをチェック

CKY法(解析例/構文木の復元)

● 解析例

■ 文法

$S \rightarrow NP VP$ $VP \rightarrow v NP$ $det \rightarrow a | the$
 $NP \rightarrow det n$ $VP \rightarrow VP PP$ $v \rightarrow broke$
 $NP \rightarrow NP PP$ $PP \rightarrow prep NP$ $prep \rightarrow with$
 $n \rightarrow man | desk | drawer$

■ 入力文

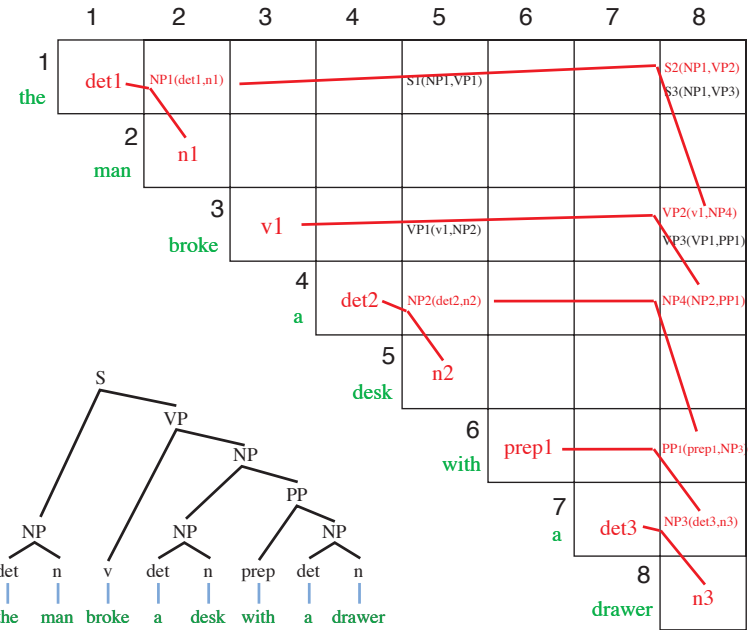
◆ the man broke a desk with a drawer

■ 解析過程 → 黒板、別紙資料(後日配布)

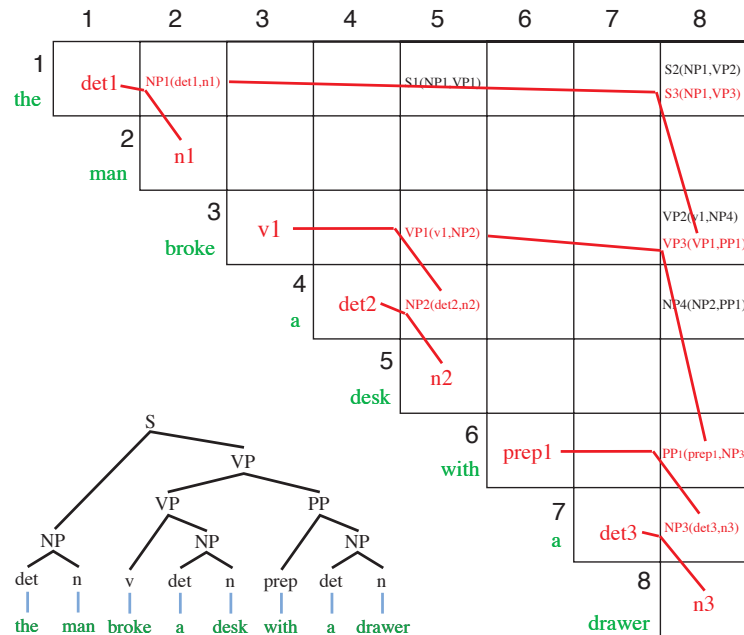
● 構文木の復元

■ 完成したCKY表の履歴をたどる

13



14



15

CKY法の特徴

- 計算量 $O(n^3)$
 - n は入力文の長さ(単語数)
- ボトムアップアルゴリズム
- アルゴリズムが単純で理解しやすい
- チョムスキー標準形の文法でしか解析できない

16

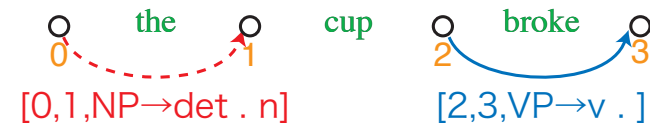
チャート法 (chart parsing)

- 2通りの方式
 - トップダウンチャート法
 - ボトムアップチャート法
- 構文解析の途中経過をチャートに保存

17

チャート法

- 用語
 - 節点(node)
 - ◆ 単語と単語の間に存在する仮想的な点
 - 弧(arc)
 - ◆ 節点間を結び、文の部分的な構造を表す
 - ◆ $[i, j, A \rightarrow \alpha . \beta]$
 - ◆ i は弧の始点、 j は弧の終点
 - ◆ ドットは解析が終了している位置を表す



18

● 用語

- 不活性弧(inactive arc)
 - ◆ ドットが右辺の最後にある弧
- 活性弧(active arc)
 - ◆ 不活性弧以外の弧
- チャート(chart)
 - ◆ ノード、弧の集合
- アジェンダ(agenda)
 - ◆ チャートに追加すべき弧のリスト(スタック)

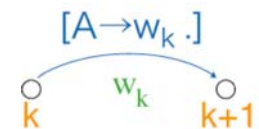
● 記号

- n : 入力単語の長さ
- X, Y, Z : 非終端記号 S : 開始記号
- α, β, γ : 終端記号または非終端記号の列 (空列を含む)

19

● アルゴリズム

- 辞書規則の適用
 - ◆ 入力文の各単語 w_k について、
不活性弧 $[k, k+1, A \rightarrow w_k .]$ を
アジェンダに追加



- 活性弧 $[0, 0, S \rightarrow . \alpha]$ をアジェンダの先頭に追加
 - ◆ S は開始記号

20

- アジェンダが空になるまで以下の操作を繰り返す

1. (弧の選択)

アジェンダから弧を1個選びチャートに追加(以下arc)

2. (弧の結合)

- ◆ arcが活性弧 $[i, j, X \rightarrow \alpha.Y\beta]$ のとき
arcの右にある不活性弧 $[j, k, Y \rightarrow \gamma.]$ を探し、結合する
- ◆ arcが不活性弧 $[i, j, Y \rightarrow \gamma.]$ のとき
arcの左にある活性弧 $[k, i, X \rightarrow \alpha.Y\beta]$ を探し、結合する
- ◆ 結合してできた新しい弧 $[i, k, X \rightarrow \alpha Y(n).\beta]$ を
アジェンダに追加 (n は不活性弧のID(履歴))

3. (新しい弧の提案)

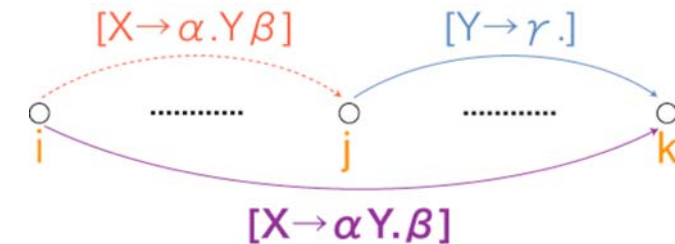
- ◆ arcが活性弧 $[i, j, X \rightarrow \alpha.Y\beta]$ のとき
Yを左辺とする規則 $Y \rightarrow \gamma$ (辞書規則を除く)があれば、
新しい活性弧 $[j, j, Y \rightarrow \gamma.]$ を作ってアジェンダに追加

※ 同じ弧が既にチャートまたはアジェンダにあるときは追加しない 21

弧の結合

- 活性弧+不活性弧で新しい弧を作る

- $[i, j, X \rightarrow \alpha.Y\beta] + [j, k, Y \rightarrow \gamma.]$
⇒ $[i, k, X \rightarrow \alpha Y.\beta]$



22

トップダウンチャート法

- チャートへの弧の追加

- $n:X[\alpha]$ と弧を省略表記するとよい
 - ◆ n : 弧番号, X : 規則の左辺, α : .の右にある記号列
 - ◆ 次にどんな弧を結合すればいいか、わかりやすい

- $[0,n,S \rightarrow \alpha.]$ という不活性弧ができていれば解析成功、それ以外は失敗

23

トップダウンチャート法

- 解析例

- 文法

$S \rightarrow NP VP$	$VP \rightarrow v$	$det \rightarrow the$
$NP \rightarrow det n$	$VP \rightarrow v NP$	$n \rightarrow cup$
		$v \rightarrow broke \mid cup$

- 解析文

- ◆ the cup broke

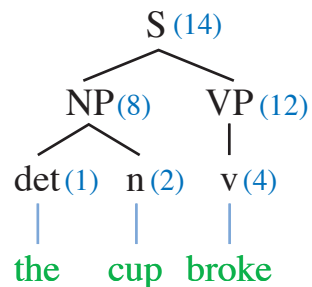
- 解析過程

- ◆ 黒板、別紙資料(後日配布)

24

構文木の復元

- 弧に履歴を残す
 - 弧に識別番号をつける
 - 右辺がどの不活性弧によって構成されるかを記録
- 不活性弧 $[0, n, S \rightarrow \alpha.]$ の履歴をたどれば構文木が復元できる



25

チャート法の特徴

- 計算量は $O(n^3)$
- 任意の文脈自由文法が扱える
- 4種類の方式
 - トップダウンとボトムアップ
 - 縦型探索と横型探索
- 文法の予測能力が使える
 - 無駄な弧を生成しないので効率が良い
 - トップダウンチャート法のみ
- 広く使われている

26

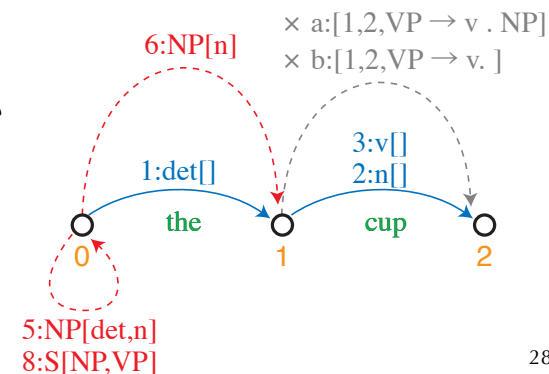
縦型探索と横型探索

- 縦型探索
 - 1つの解の候補の解析を優先的に進める
 - ◆ 文が文法によって生成できるかだけを調べるときに便利
- 横型探索
 - 全ての解の候補の解析を並列に進める
 - ◆ ビームサーチに向いている
- チャート法では両方とも可能
 - アジェンダをスタックにしたときは縦型探索
 - アジェンダをキューにしたときは横型探索

27

文法の予測能力

- 弧 a, b
 - 無駄な弧(cupの品詞がvであるという解釈は誤り)
 - トップダウンチャート法では生成されない
 - ◆ 文法によって
detの後には
vが現われない
ことが予想
されている



28