

# 自然言語処理のための文法形式 (Grammar Formalisms for Natural Language Processing)

松本裕治

2004年5月28日(金)

1

## 文法理論 (自然言語処理のための文法システム)

- 拡張遷移網文法 (Augmented Transition Networks)[Woods 70]
- カテゴリ文法 (Categorial Grammar) (例えば [Lambek 58][Oehrle 88][Wood 93])
- 格文法 (Case Grammar)[Fillmore 68]
- モンタギュ文法 (Montague Grammar)[Montague 73]
- 概念依存 (Conceptual Dependency, CD)[Schank 72]
- 拡張文脈自由文法 (Extended CFG)
  - 属性文法 (Attribute Grammar)
  - LINGOL[Pratt 75]
  - 拡張 LINGOL(Extended LINGOL)[田中 他 77]
- Combinatory Categorial Grammar(CCG)[Steedman 96]

2

- 木接合文法 (Tree Adjoining Grammar, TAG)[Joshi 75,98]
  - Lexicalized Tree Adjoining Grammar (L-TAG)[Schabes 90,92]
  - Tree Insertion Grammar (TIG)[Schabes 95]
- 単一化文法 (Unification-based Grammar)
  - LFG(Lexical Functional Grammar)[Kaplan & Bresnan 82]
  - GPSG(Generalized Phrase Structure Grammar)[Gazdar 82][GKPS 85]
  - Head Grammar[Pollard]
  - HPSG(Head-driven Phrase Structure Grammar)[Pollard & Sag 87, 94] [Sag 03]
  - 論理文法 (Logic Grammars)
    - \* Metamorphosis Grammars(MGs)[Colmerauer 78]
    - \* Functional Unification Grammar(FUG)[Kay 80]
    - \* Definite Clause Grammars(DCGs)[Pereira & Warren 80]

3

- \* Extraposition Grammars(XGs)[Pereira 81]
- \* Gapping Grammars(GGs)[Dahl 84], Discontinuous Grammars(DGs)[Dahl 89]
- \* PATR-II[Shieber 84]

4

## 形式文法の定義

$$G = (V_N, V_T, P, S)$$

$V_N$  非終端記号の有限集合

$V_T$  終端記号の有限集合  $V_N \cap V_T = \phi$

$P$  生成規則 (文法規則) の有限集合

$S$  開始記号 (文範疇)  $S \in V_N$

## 形式文法の生成規則

$$\alpha \rightarrow \beta \quad (\alpha \in (V_N \cup V_T)^+, \beta \in (V_N \cup V_T)^*)$$

## 正規文法

$$A \rightarrow B, A \rightarrow aB \quad (A, B \in V_N, a \in V_T)$$

5

## 文脈自由文法

$$A \rightarrow \beta \quad (A \in V_N, \beta \in (V_N \cup V_T)^*)$$

## 文脈依存文法

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2 \quad (\text{または, } |\alpha| \leq |\beta| \text{ のとき})$$

6

## RTN: Recursive Transition Networks

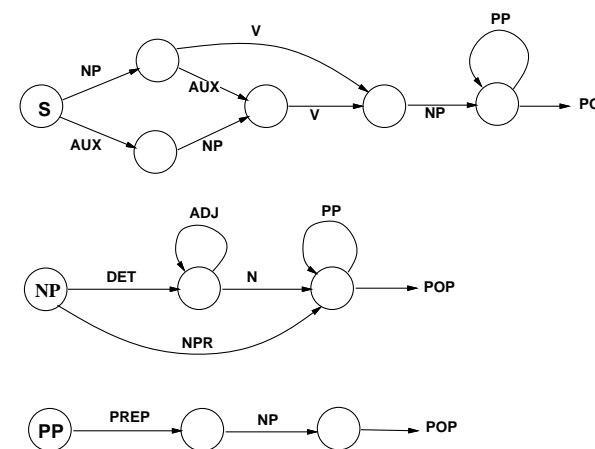
- 状態遷移オートマトン (有限オートマトン) の集合
- 状態遷移の (枝の) ラベルが一つの状態遷移オートマトンの名前
- 文脈自由文法と等価

### 3種類の Transition Networks

- Transition Networks = Finite state Automata
- Recursive Transition Networks = Context Free Grammars
- Augmented Transition Networks = Type-0 Grammars

7

## RTN(Recursive Transition Networks) の例

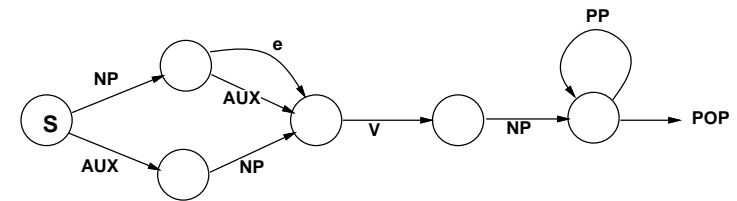


8

## ATN(Augmented Transition Networks)

- RTN の各枝にレジスタを導入し、部分解析木を格納したり、条件のチェックや手続きの実行を記述できるようにしたもの
- 大域的なレジスタとして Hold レジスタをもつ (関係詞節などの痕跡の処理に用いることができる)
- 解析はトップダウン後戻りにより行なわれる
- DCG(Definite Clause Grammar) に変換可能

9



$S \rightarrow ( NP, ( AUX ; [ ] ) ; AUX, NP ), V, NP, PP^* .$

$PP^* \rightarrow [ ] .$

$PP^* \rightarrow PP, PP^* .$

10

## カテゴリ文法 (Categorial Grammar)

- 単語が属する文法範疇 (カテゴリ) が複雑な関数をなすと考え、関数適用を繰り返すことによって文を構成するという立場
- 文法の要素としてあるのは単語だけであり、文法規則と呼ばれるものは明示的には存在しない

### カテゴリの定義

1. 原始カテゴリは、 $N, S$  の2つであり、それぞれ名前 (名詞句)、文に対応する。
2. 複合カテゴリは、他の2つのカテゴリから合成して得られるカテゴリであり、数学的には、一方を定義域、他方を値域とする関数となる。記法としては、 $X$  と  $Y$  がカテゴリなら、 $X/Y, X \setminus Y$  は、複合カテゴリ

11

## カテゴリ文法の演算規則

演算規則名	演算	記号
関数適用 (Application):	$X/Y Y \rightarrow X$	(>A)
	$Y X \setminus Y \rightarrow X$	(<A)
結合 (Associativity):	$(X \setminus Y)/Z \leftrightarrow (X/Z) \setminus Y$	
合成 (Composition):	$X/Y Y/Z \rightarrow X/Z$	(>C)
	$Y \setminus Z X \setminus Y \rightarrow X \setminus Z$	(<C)
繰り上げ (Raising):	$X \rightarrow Y/(Y \setminus X)$	
	$X \rightarrow Y \setminus (Y/X)$	
分配 (Division):	$X/Y \rightarrow (X/Z)/(Y/Z)$	
	$X \setminus Y \rightarrow (X \setminus Z) \setminus (Y \setminus Z)$	

12

## カテゴリの関数適用の記述例

$$(1) \frac{X \quad Y \backslash X}{Y} < A$$

$$(2) \frac{Y / X \quad X}{Y} > A$$

13

## カテゴリ文法の演算規則の適用例

### 1. 関数適用:

John, Mary:  $N$ , loves:  $(S \backslash N) / N$

$$\begin{array}{c} \text{John} \quad \text{loves} \quad \text{Mary} \\ N \quad (S \backslash N) / N \quad N \\ \hline (S \backslash N) > A \\ \hline S < A \end{array}$$

14

### 2. 結合: $(X \backslash Y) / Z \rightarrow (X / Z) \backslash Y$

loves:  $(S \backslash N) / N \rightarrow (S / N) \backslash N$

$$(i) \frac{\begin{array}{c} (\text{John} \quad \text{loves}) \quad \text{Mary} \\ N \quad (S / N) \backslash N \quad N \\ \hline S / N < A \end{array}}{S} > A$$

$$(ii) \frac{\begin{array}{c} (\text{Mary} \quad \text{cooked}) \quad \text{and} \quad (\text{John} \quad \text{ate}) \quad \text{the cookie} \\ N \quad (S / N) \backslash N \quad \text{Conj} \quad N \quad (S / N) \backslash N \quad N \\ \hline (S / N) < A \quad (S / N) < A \\ \hline (S / N) \quad \text{Conj} \\ \hline S < A \end{array}}{S} > A$$

15

### 3. 合成: $(<C) Y \backslash Z \ X \backslash Y \rightarrow X \backslash Z$

繰り上げ:  $X \rightarrow Y \backslash (Y / X)$

Mary, Bill:  $N \rightarrow (S \backslash N) \backslash ((S \backslash N) / N)$  (=  $vp \backslash (vp / N)$  と略記)

$$\begin{array}{c} \text{John} \quad \text{saw} \quad \text{Mary} \quad \text{yesterday} \quad \text{and} \quad \text{Bill} \quad \text{today} \\ N \quad vp / N \quad vp \backslash (vp / N) \quad vp \backslash vp \quad \text{Conj} \quad vp \backslash (vp / N) \quad vp \backslash vp \\ \hline vp \backslash (vp / N) < C \quad vp \backslash (vp / N) < C \\ \hline vp \backslash (vp / N) \quad \text{Conj} \\ \hline vp (= (S \backslash N)) < A \\ \hline S < A \end{array}$$

### (Co-ordination) のための規則

$$X^+ \text{ Conj } X \Rightarrow X$$

16

4. 分配:  $X \setminus Y \rightarrow (X \setminus Z) \setminus (Y \setminus Z)$

$$\begin{array}{c} \text{(i) } \begin{array}{ccc} \text{John} & \text{lives} & \text{here} \\ N & S \setminus N & S \setminus S \end{array} \\ \hline <A \\ S \\ \hline <A \\ S \end{array}$$

(ii) here:  $S \setminus S \rightarrow (S \setminus N) \setminus (S \setminus N)$

$$\begin{array}{c} \text{John} \quad (\text{lives} \quad \text{here}) \\ N \quad S \setminus N \quad (S \setminus N) \setminus (S \setminus N) \\ \hline <A \\ (S \setminus N) \\ \hline <A \\ S \end{array}$$

## 木接合文法 (Tree Adjoining Grammar)

木接合文法は、5 つ組  $(\Sigma, NT, I, A, S)$  によって定義される。

- $\Sigma$  は終端記号の集合、 $NT$  は非終端記号の集合、
- $I$  と  $A$  はそれぞれ句構造木の集合であり、 $I$  の要素を初期木 (initial trees)、 $A$  の要素を補助木 (auxiliary trees) と呼ぶ。
- $S$  は  $NT$  の特別な要素であり、文範疇を表す。
- $I$  と  $A$  の要素を総称して基本木 (elementary trees) という。

初期木 普通の句構造木。(単語以外の) 葉には下矢印をもつ。

補助木 根の範疇と同一名の範疇を葉にも持つような句構造木。

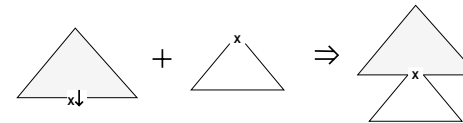
## 木接合文法の基本操作

置換 (substitution): 句構造文法と同様の木の生成操作。下矢印が付けられた節点に対し、その節点と同じラベルを根にもつ初期木によって置き換える操作。

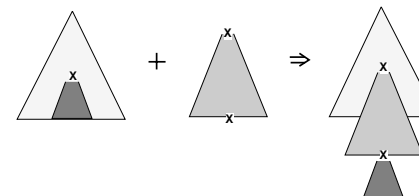
接合 (adjoining): 木の内部節点 (ラベル  $X$  をもつと仮定する) の位置で、木を上下に分割し、その間に  $X$  を根とする補助木を挿入する操作。この操作は、adjunction と呼ばれる。

## 置換操作、接合操作の模式図

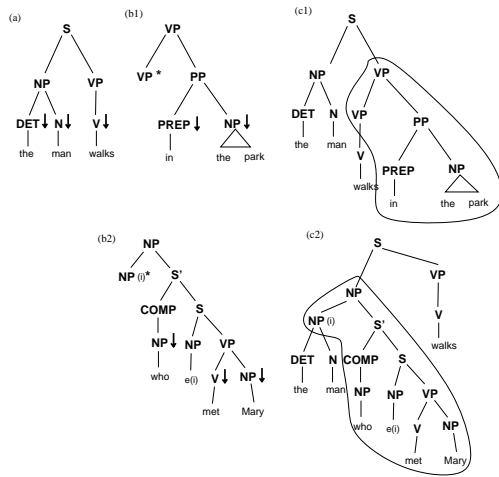
- 置換



- 接合



## 木接合文法の使用例



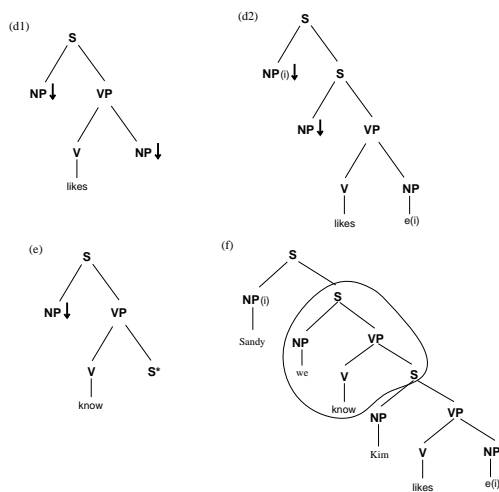
21

## 語彙化木接合文法 (Lexicalized Tree Adjoining Grammar)

- 少なくとも一つの葉が単語によってラベル付けされている木接合文法
- 単語の記述が文法のすべて
- 単語の記述に対して置換と接合操作を経ることにより、言語のあらゆる適格な文が生成される

22

## 語彙化木接合文法の例



23

## 動詞の格フレームに関する議論の展開

- 格フレーム (下位範疇化フレーム) : 用言が補語としてとる名詞句の一覧
- 表層格 : 日本語の「が」格、「を」格、英語の主格、目的格などの表層的な格
- 深層格 : 動作主体、対象など、用言と補語との意味関係に注目した格
- 旧来の見方 : 各用言は個別の格フレームを持ち、表層格と深層格の対応が内部でとられている
- 用言がもつ深層格の一覧は項構造 (argument structure) とも呼ばれる
- 近年の見方 : 項構造が、ある規則性をもって表層格に対応する。そのような規則を linking rule と呼ぶ

24

## 文法理論に関する進展

- 言語の文法を、個別の文法規則の集まりとしてみるのではなく、文法規則といえるものは言語のもつ文構造全般に渡る性質や制約を記述するものであり、個々の単語(語彙項目)がそれぞれの働きを予測できるほどの内部構造をもっている(語彙化文法の考え方)
- 語彙を文法の主要な構成要素と考える語彙化文法の例を以下に紹介する。いずれも単一化文法と呼ばれる文法理論の一種

25

## Unification Based Grammar Formalisms (単一化文法)

- LFG(Lexical Functional Grammar)
- GPSG(Generalized Phrase Structure Grammar)
- HG(Head Grammar)
- HPSG(Head-driven Phrase Structure Grammar)

## Logic Grammars

- FUG(Functional Unification Grammar)
- MG(Metamorphosis Grammars)
- DCG(Definite Clause Grammars)
- XG(Extraposition Grammars)

26

## 単一化文法が共通にもつ性質

1. 宣言的な意味を持っていること
2. 文脈自由文法を構文上の基本構造としていること
3. 文法を書き換え規則としてではなく、許容可能性条件 (node admissibility condition) として解釈していること
4. 再帰的に定義された素性構造、あるいは、非循環グラフ (acyclic graph) をデータ構造として用いていること
5. 構文的な情報を構築するための基本的な演算として単一化 (unification) を用いていること

28

- DG(Discontinuous Grammars)

- PATR-II

LFG, GPSG, …

自然言語の普遍的性質を記述するために言語学上の興味に基づいて提案された文法形式

FUG, DCG, PATR-II, …

言語学的に自然であり、なおかつ、計算可能な文法記述形式

27

## 確定節文法 (Definite Clause Grammars)

- 一階述語論理の部分クラス, Horn 節論理によって自然言語の文法を記述する
- 構文構造は文脈自由文法
- 非終端記号が引数を持ち, 任意の Prolog プログラムを本体部に含み得る
- Prolog に簡単にコンパイルでき, 下降型構文解析プログラムとして動作可能

29

## [例]

```
setence(SFrame, s(NP,VP)) -->
    np(NPFrame, NP),
    { case(NPFrame, subject) },
    vp(VPFrame, VP),
    { make_new_frame(NPFrame, VPFrame, SFrame) }.
```

setence は np と vp の列からなるが, これらは引数に格納された情報を持ち運ぶことができ, それらの値が '{ '}' の中の手続きを満足していなければならない.

30

## 単一化に基づく語彙化文法

### 語彙機能文法 (Lexical Functional Grammar, LFG)

- 文脈自由文法に基づく文法規則 (c-構造規則と呼ばれる) と文法規則に付与された付加情報 (f-構造規則と呼ばれる) よりなる。
- 語彙項目 (単語) が, 構文や意味に関する情報の多くを持つ (語彙化文法の一つ)

c-構造: 文の構文的な構造を表わすための木構造。

f-構造: 文の詳細な情報を表す素性構造 (属性と値の対の集合)。文法的な適格性を決定する。f-構造は, 語彙項目に書かれた f-構造および c-構造に与えられた注釈 (annotation) により計算される。

31

### 主辞駆動句構造文法 (Head-driven Phrase Structure Grammar, HPSG)

- 語彙項目 (単語) が, 構文や意味に関する情報に関するほとんどの情報を持つ (語彙化文法の一つ)
- 文の構造を決めるための個別の文法規則は存在せず, 語や句が結びついてより大きな構造を作る時の一般的な制約 (原理) が決められているだけと仮定する。

32



## 語彙機能文法 (LFG, Lexical Functional Grammar)

### LFG の文法規則の例 (c-構造規則および f-構造注釈)

S → NP VP ← c-構造  
(↑ SUBJ) = ↓ ↑ = ↓ ← f-構造注釈

VP → V NP NP ← c-構造  
↑ = ↓ (↑ OBJ) = ↓ (↑ OBJ2) = ↓ ← f-構造注釈

文 (S) が、名詞句 (NP) と動詞句 (VP) からできるという文法規則、および、動詞句 (VP) が、動詞 (V) と二つの名詞句 (NP) からできるという規則。

f-構造注釈は、左辺の句のもつ情報が右辺の句のもつ情報をどのように組み合わせてできるかを記述している。

33

### 文法規則の具体的な意味

S → NP VP ← c-構造  
(↑ SUBJ) = ↓ ↑ = ↓ ← f-構造注釈

c-構造規則: 文 (S) が、名詞句 (NP) と動詞句 (VP) からできるということ

f-構造注釈: 上矢印が左辺の句がもつ内部表現、下矢印が右辺の個々の句の内部表現を表す

右辺の NP の下の f-構造注釈: S 中の SUBJ という素性の値が NP の値と等しいという意味

右辺の VP の下の f-構造注釈: S のもつ値が VP がもつ値をそのまま引き継ぐという意味

34

## 語彙項目の例

a: DET (↑ SPEC) = A  
(↑ NUM) = SG

girl: N (↑ NUM) = SG  
(↑ PRED) = 'GIRL'

handed: V (↑ TENSE) = PAST  
(↑ PRED) = 'HAND((↑ SUBJ)(↑ OBJ)(↑ OBJ2))'

the: DET (↑ SPEC) = THE

### 文の f-構造の構成

c-構造によって得られた構文解析木に従い、語彙項目の f-構造と使用した c-構造に与えられている f-構造注釈よりなる連立方程式を解くことによって得られる。

35

## 下位範疇化 (subcategorization)

語彙項目に含まれる PRED が下位範疇化情報 (項構造) を記述している

ask: V (↑ TENSE) = PRES  
(↑ PRED) = 'ASK((↑ SUBJ)(↑ OBJ)(↑ COMP))'

go: V (↑ TENSE) = PRES  
(↑ PRED) = 'GO((↑ SUBJ))'

- ask は、主語、目的語、節 (that 節など) をとり、go は、主語のみをとる

36

## その他の動詞の記述 (主要部分のみ)

try: V (↑ PRED) = 'TRY<((↑ SUBJ)(↑ VCOMP))>  
 (↑ VCOMP SUBJ) = (↑ SUBJ)  
 (↑ VCOMP TO) =<sub>c</sub> +

promise: V (↑ PRED) = 'PROMISE<((↑ SUBJ)(↑ OBJ)(↑ VCOMP))>  
 (↑ VCOMP SUBJ) = (↑ SUBJ)  
 (↑ VCOMP TO) =<sub>c</sub> +

persuade: V (↑ PRED) = 'PERSUADE<((↑ SUBJ)(↑ OBJ)(↑ VCOMP))>  
 (↑ VCOMP SUBJ) = (↑ OBJ)  
 (↑ VCOMP TO) =<sub>c</sub> +

37

- try は、主語と不定詞句をとること、および、不定詞句の主語が文の主語と一致することが記述されている。
- promise は、主語と目的語と不定詞句をとること、および、不定詞句の主語が文の主語と一致することが記述されている。
- persuade(説得する) は、主語と目的語と不定詞句をとること、および、不定詞句の主語が文の目的語と一致することが記述されている。

38

## 文法規則と語彙記述の例

### 文法規則の例 (c-構造規則および f-構造注釈)

S → NP VP  
 (↑ SUBJ) = ↓ ↑ = ↓

VP → V NP (VP')  
 ↑ = ↓ (↑ OBJ) = ↓ (↑ VCOMP) = ↓

VP' → (to) VP  
 (↑ TO) = + ↑ = ↓  
 (↑ INF) =<sub>c</sub> +

39

### 語彙項目の例

a: DET (↑ SPEC) = A  
 (↑ NUM) = SG

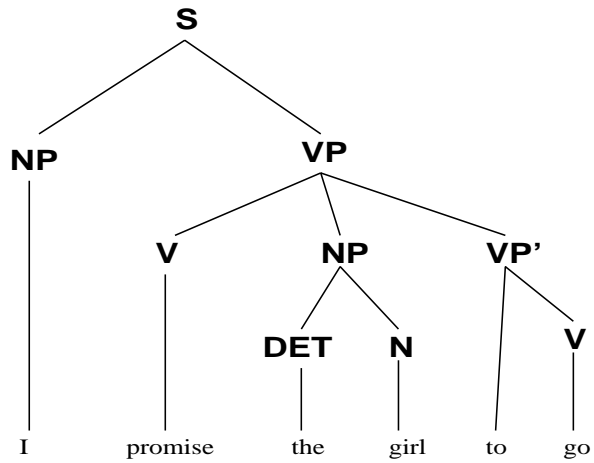
the: DET (↑ SPEC) = THE

girl: N (↑ NUM) = SG  
 (↑ PRED) = 'GIRL'

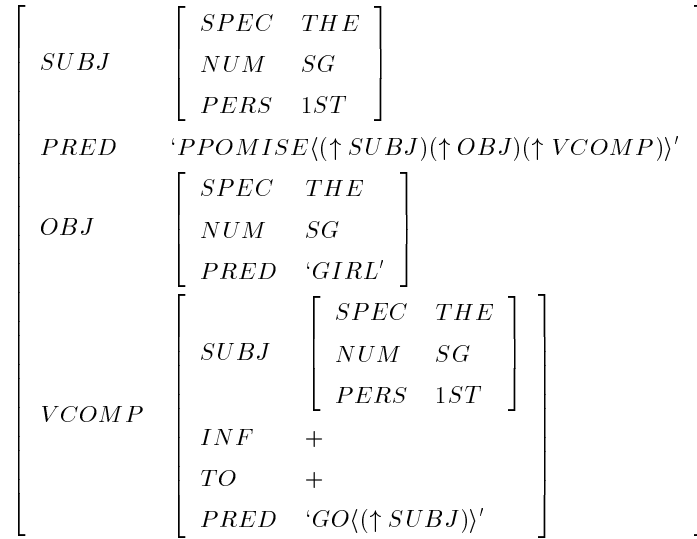
I: NP (↑ SPEC) = THE  
 (↑ NUM) = SG  
 (↑ PERS) = 1ST

40

**LFG の c-構造と f-構造の例 (promise の例)**

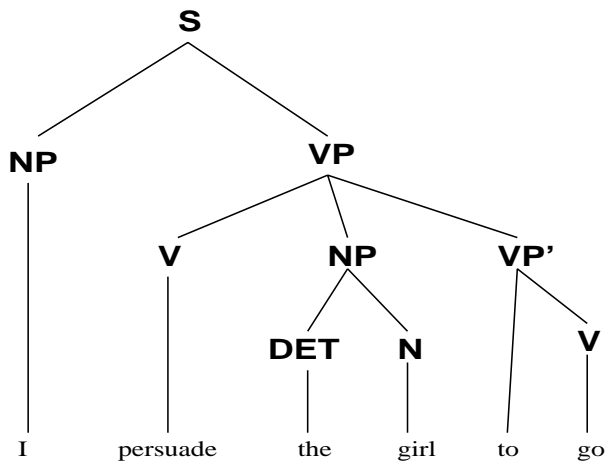


41

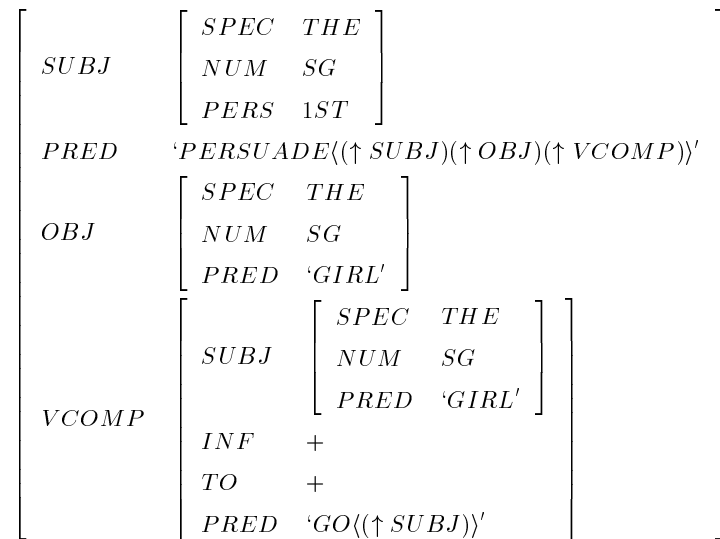


42

**LFG の c-構造と f-構造の例 (persuade の例)**



43



44

## 上の2つの例からわかること

- promise と persuade のように、構文的には同じ構造 (主語、目的語、不定詞句) をもつ動詞の内部的な意味の違いがあること
- LFG では、その違いが素性構造の内部表現によって表現することができる
- さらに、その違いは、それぞれの動詞の語彙構造の中に記述されている
- このような個々の語がもつ意味の違いは、LFG や次に示す HPSG などの語彙主導型 (あるいは、語彙化) 文法によって表現することができる

45

## f-構造の適格性条件

### 唯一性条件 (Uniqueness condition)

同じ属性は一つの f-構造中に2箇所以上現れない。

### 完全性条件 (Completeness condition)

f-構造は、その述語が持つすべての統率可能な文法機能を含むとき局所的に完全であると言う。f-構造は、自分とすべての下位 f-構造が局所的に完全なとき完全であると言う。

\* The girl handed.

*'hand((↑ SUBJ)(↑ OBJ2)(↑ OBJ))'*

46

### 一貫性条件 (Coherence condition)

f-構造は、それが含むすべての統率可能な文法機能とその述語に含まれているとき局所的に一貫していると言う。f-構造は、自分とすべての下位 f-構造が局所的に一貫しているとき一貫していると言う。

\* The girl fell the apple a dog.

*'fall((↑ SUBJ))'*

(注) 統率可能な文法機能 SUBJ, OBJ などのように、下位範疇化に用いられ得る文法機能のこと

47

## HPSG(Head-driven Phrase Structure Grammar)

すべての文法機能 (語彙, 文法カテゴリ) は素性構造で記述され, 言語普遍の原則および言語固有の原則を満たす語の列が適格な文と認定される。

HPSG で仮定されている基本的な素性構造

SYN が構文情報、SEM が意味情報を記述する

$$\left[ \begin{array}{l} \text{word/phrase} \\ \text{SYN} \left[ \begin{array}{l} \text{HEAD} \\ \text{SUBCAT} \end{array} \right] \\ \text{SEM} \end{array} \right] \left[ \begin{array}{l} \text{category} \\ \text{FORM} \quad \dots \end{array} \right]$$

48

## 基本的な素性とその役割

HEAD 品詞 や格などの基本的な文法機能

SUBCAT 下位範疇化情報を記述 (最近の理論では、主語 (SPR) と補語 (COMPS) に分けて表現される)

## HPSG の特徴

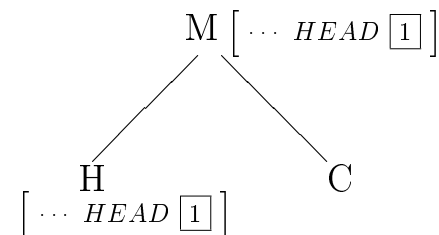
- 基本的な文法構造は主辞と補語 (または付加語) からできる 2 分木構造 .
- 個々の文法規則が存在するのではなく、文法規則全体に成り立つスキーマが存在するだけ .
- 構文論、意味論が統一的な記述法で表現されている .
- 語彙または句がもつ素性構造によってその句の構文的、意味的な性質を決定する .

49

## 主要な文法規則

実際には文法規則というよりも句がより大きな句を形作る際の一般的な構造を表す原理

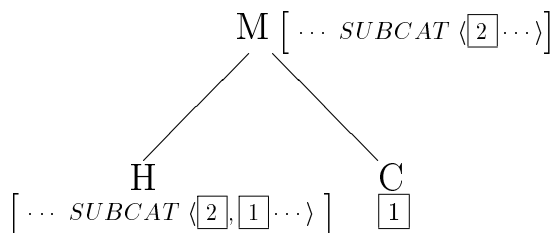
## 主辞素性原理 (Head Feature Principle)



50

## 主辞-補語規則 (Head-Complement rule)

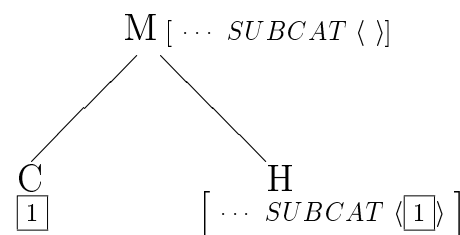
SUBCAT その句が飽和するにはどのような句と結合しなければならないかを記述 .



51

## 主辞-指定句規則 (Head-Specifier rule)

最近の理論では、主語を SUBCAT 素性から分離して SPR (Specifier) と呼び、残りの要素を補語 (COMPS) と呼ぶ



52

## 色々な動詞の SUBCAT と SEM の記述

- *tend* (Raising)      They tend to run.

$$\left[ \begin{array}{l} \text{SUBCAT } \langle \text{NP}_{\boxed{1}}, \text{VP}[\text{inf}, \text{SUBCAT } \langle \text{NP}_{\boxed{1}} \rangle] \rangle \\ \text{SEM } \left[ \text{RELATION } \textit{tend} \right] \end{array} \right]$$

- *try* (Equ)      They try to run.

$$\left[ \begin{array}{l} \text{SUBCAT } \langle \text{NP}_{\boxed{1}}, \text{VP}[\text{inf}, \text{SUBCAT } \langle \text{NP}_{\boxed{1}} \rangle] \rangle \\ \text{SEM } \left[ \begin{array}{l} \text{RELATION } \textit{try} \\ \text{TRYER } \boxed{1} \textit{ref} \end{array} \right] \end{array} \right]$$

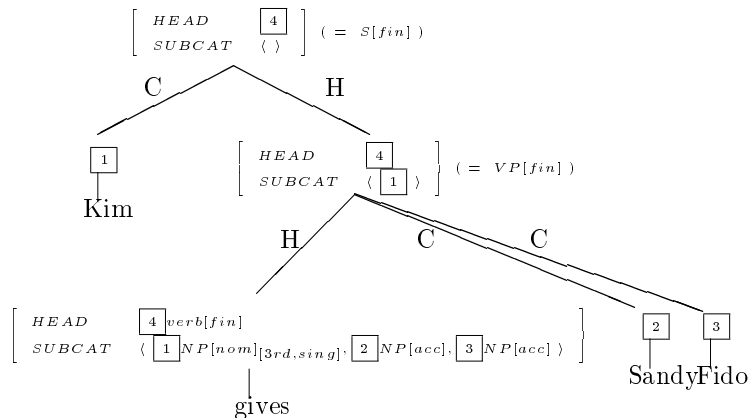
- *persuade* (Obj-Equ)      They persuade him to go.

$$\left[ \begin{array}{l} \text{SUBCAT } \langle \text{NP}_{\boxed{1}}, \text{NP}_{\boxed{2}}, \text{VP}[\text{inf}, \text{SUBCAT } \langle \text{NP}_{\boxed{2}} \rangle] \rangle \\ \text{SEM } \left[ \begin{array}{l} \text{RELATION } \textit{persuade} \\ \text{PERSUADER } \boxed{1} \textit{ref} \\ \text{PERSUADEE } \boxed{2} \textit{ref} \end{array} \right] \end{array} \right]$$

- *promise* (Subj-Equ)      They promise him to go.

$$\left[ \begin{array}{l} \text{SUBCAT } \langle \text{NP}_{\boxed{1}}, \text{NP}_{\boxed{2}}, \text{VP}[\text{inf}, \text{SUBCAT } \langle \text{NP}_{\boxed{1}} \rangle] \rangle \\ \text{SEM } \left[ \begin{array}{l} \text{RELATION } \textit{promise} \\ \text{PROMISER } \boxed{1} \textit{ref} \\ \text{PROMISEE } \boxed{2} \textit{ref} \end{array} \right] \end{array} \right]$$

## HPSG による解析例



## 参考文献

- Colmerauer, A. "Metamorphosis Grammars," *Natural Language Communication with Computers*, pp.133-189, 1978.
- Dahl, V., "On Gapping Grammars," *Proc. International Conference on Logic Programming*, pp.77-88, 1984.
- Dahl, V., "Discontinuous grammars," *Computational Intelligence*, Vol.5, pp.161-179, 1989.
- Dalrymple, M.(ed.): *Semantics and Syntax in Lexical Functional Grammar*, The MIT Press, 1999.
- Dowty, D.R., Wall, R.E. and Peters, S.: *Introduction to Montague Semantics*, Reidel, 1981.
- Fillmore, C.J., "The Case for Case," in Bach, E., and Harms, T.(eds.), *Universals in Linguistic Theory*, New York: Holt, Rinehart & Winston, 1968.
- Gazdar, G., "Phrase Structure Grammar," in P. Jacobson and G.K. Pullum (eds.), *The Nature of Syntactic Representation*, SLL 15, D.Reidel, pp.131-186, 1982.
- Gazdar, G., Klein, E., Pullum, G.K. and Sag, I.: *Generalized Phrase Structure Grammar*, Basil Blackwell, 1985.
- Joshi, A.K., "Role of Constrained Computational Systems in Natural Language Processing," *Artificial Intelligence*, Vol.103, No.1-2, pp.117-132, 1998.

- Kaplan, R.M. and Bresnan, J., "Lexical-Functional Grammar: A Formal System for Grammatical Representation," in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, Chapter 4, MIT Press, pp.173-281, 1982.
- Lambek, J. "The Mathematics of Sentence Structure," *American Mathematical Monthly*, Vol.65, pp.154-170, 1958.
- Oehrle, R., et al(eds.), *Categorial Grammars and Natural Language Structures*, Reidel, pp.297-318, 1988.
- Montague, R., "The Proper Treatment of Quantification in Ordinary English," in R.Thomason(ed.), *Formal Philosophy: Selected Papers of Richard Montague*, New Haven, Yale University Press, pp.247-270, 1973.
- Pereira, F.C.N., Warren, D.H.D., "Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks," *Artificial Intelligence*, Vol.13, No.3, pp.231-278, 1980.
- Pereira, F.C.N., "Extraposition Grammars", *American Journal of Computational Linguistics*, Vol.7, No.4, pp.243-256, 1981.
- Pratt, V., "LINGOL - A Progress Report," *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, pp.422-428, 1975.
- Sag, I.A. and Wasow, T., Bender, E.M.(Eds.): *Syntactic Theory: A Formal Introduction*, CSLI Lecture Notes, No.152, CSLI Publications, 2003.
- Schank, R.C., "Conceptual Dependency: A Theory of Natural Language Understanding," *Cognitive Psychology*, Vol.3, No.4, pp.552-631, 1972.

- Schabes, Y., "Stochastic Lexicalized Tree-Adjoining Grammars," *COLING-92*, pp.426-432, 1992.
- Schabes, Y. and Waters, R.C., "Tree Insertion Grammar: A Cubic-Time, Parsable Formalism that Lexcalizes Context-Free Grammar without Changing the Tree Produced," *Computational Linguistics*, Vol.21, No.4, pp.479-513, 1995.
- Steedman, M.: *Surface Structure and Interpretation*, Linguistic Inquiry Monograph 30, The MIT Press, 1996.
- Shieber, S.M., "The Design of a Computer Language for Linguistic Information," *COLING-84*, pp.362-366, July 1984.
- Wood, M.M.: *Categorial Grammars*, Routledge, 1993.
- Woods, W.A., "Transition Network Grammars for Natural Language Analysis," *C.ACM*, Vol.13, No.10, pp.591-606, 1970.