

TM-0165

日本語文法記述 — JPSG

三 吉 秀 夫

JPS ワーキンググループ

April, 1986

©1986, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

日本語文法記述 — JPSG*

三吉秀夫 (ICOT)

JPSワーキンググループ

1. 概要

一般に自然言語理解システムに於いて、文法は辞書や常識などの知識ベースと同様に、非常に重要なコンポーネントであり、その良し悪しがシステム全体の性能に大きく影響する。きちんとした言語理論的基礎の上に作り上げられた文法体系は、あらゆる自然言語処理応用分野に有用なデータを提供するものであり、その計算機ツールを開発することは有意義な研究テーマである。以上のような観点から本プロジェクトではGazdarらの提唱するGPSG(Generalized Phrase Structure Grammar)[Gazdar 85a]をベースにした新しい日本語の文法体系(これをJPSG(Japanese Phrase Structure Grammar)と呼ぶ)とそれに基づいた処理系(構文・意味解析システム)を開発することになった。GPSG理論は文脈自由文法(CFG)をベースにした自然言語構文論であり、基本的な計算メカニズムとしてunificationを用いている。また構文論としての枠組みが明確であり、意味論や語用論に対するモジュラリティが高い。従って我々の進めているロジックプログラミングとの相性が非常に良い。開発期間は60年度より始まった中期の4年間である。前半2年間で日本語の基本的な構文とそれに対するセマンティクスを決める。また既存のシステムを参考にしてパーザのプロトタイプを作成する。後半2年間ではそれらを更に改良拡張し、実用に耐え得る規模にする。最終目標として小学校国語の教科書に現れる文の85%を正しく構文・意味解析する日本語の文法体系と処理系を開発することをめざしている。ここで開発されるJPSGシステムは、機械翻訳や自然言語によるQAなどの応用システムに於ける構文解析部や文生成部などに広く利用される。また言語学者にとっても自分の理論の正しさを実証したり、曖昧性の検証を行ったりするための有用なツールとなるであろう。

60年度は、文法班とパーザ班に分かれて活動を行った。文法班では日本語の基本構文に

* 本稿は次の委員、オブザーバから構成される JPSワーキンググループの昭和60年度の活動成果をまとめたものである。

委員：郡司隆男(主査：大阪大)、白井英俊(玉川大)、有働真理子(神戸大)、
外池俊幸(都立大)、新田義彦(日立)、橋田浩一(東大)、
原田康也(早大)

オブザーバ：五十嵐義行(上智大)、伊藤博樹(FIP)、
ウイセ・ウォラスチャ(東工大)、奥村学(東工大)、向後秀二(松下)、
今野聡(東芝)、東条敏(MRI)、野本忠司(上智大)

対する記述の試みを行った。また小学校六年生の国語教科書 [石森 84] の文章の分析を行った。パーザ班はJPSGの各種の機能を実現するための検討と、2つのアプローチによる実装検討を行った。またユーザインタフェースとしての文法記述言語が持つべき機能について検討を行った。本節ではこれらの検討内容について報告する。

2. 概念設計

2.1 文法体系

1で述べたように、本格的な日本語文法体系の開発が本テーマの目標である。文法体系を構築するにあたり、それを評価するための基準としては次のようなものが考えられる [Shieber 85]。

(c1) Linguistic Felicity ……言語学者の言わんとすることがどの程度記述できるか。

(c2) Expressiveness ……どのクラスの文法を記述することができるのか。

(c3) Computational Effectiveness ……計算効率。

我々はこれらの条件を満たす第1近似としてGPSGを考える。何故ならGPSGは、

- (1) CFG ベースであるため(c3)については問題がない。また(c2)についても自然言語の大部分はCFG で記述できる事が分かっており [Pullum 82, Gazdar 85b] 日本語についてもCFG で十分だという見通しが得られている。
- (2) 各種の Feature Instantiation Principles やメタルールにより、いろいろな言語現象が言語学的な一般化をとらえた形で記述し易い枠組みが提供されている。従って(c1)についても問題はない。

その他、

(3) 属性の束としての統語範疇。

(4) 構文規則と意味規則の対。

などの特徴により、記述性、汎用性が高められている。しかし日本語には、語順の任意性や、助詞の使用など特有の現象があるため、英語を対象にしたオリジナルのGPSGをそのまま適用するのは問題がある。そこでJPSGではHG(Head Grammar) [Pollard 84, 85] 風の素性の導入や日本語特有の素性、語彙規則 (Lexical Rules) の導入を考えている [Gunji 86]。

2.2 パーザ

このJPSGプロジェクトは、言語学者と計算機科学者の共同プロジェクトであるから、何よりもまず、言語学者にとっても使い易いパーザであることが要求される。それには辞書や文法規則の入力および修正、解析過程や結果の表示等、人間とのインタフェースを工

夫するのみならず、文法記述で用いられる概念や道具をいかに実現するかが重要と考えられる。特に語彙規則により基本となる語彙項目を基にして、新たな（例えば、英語においては受動態や現在形などの）語彙項目を体系的に作り出す、という概念は興味深いものである。

また、JPSGの統語範疇は素性のリストとして特徴付けられる構造を持ち、豊富な情報を記述する。文法規則は動詞句とか名詞句というような個々の局所的構造を一つ一つ記述するのではなく、より一般的な形に表される。文法構造は、部分を構成している統語範疇それぞれがもつ情報に基づき、この文法規則が「展開」されて定まる。そのため、数少ない規則で文法を規定できるという特徴がある。これを可能にしている概念がユニフィケーションである。Prologのユニフィケーションと似た概念ではあるが拡張されており、これを効率よく実現する事が一つの鍵である。

パーザはまた、文法規則の個数が少ない、という事をいかした効率的な解析アルゴリズムを備えるべきである。これはまた、パーザの動作に対する見通しの良さを提供する。従来の手法は品詞に基づいて規則を分類し、それぞれ固有の手続きを付随させていたが、JPSGではHFCやFFP等の原則及び意味の合成を一般的な手続きとして記述するからである。意味規則との相関も考慮しなければならない重要な点である。

2.3 文法記述言語

(1) 文法記述言語

文法記述言語は言語学者がシステムに対して文法記述を与えたり、保守したりする上で、言語学者とパーザとのインタフェースとなる部分である。従って、記法などに関しては言語学者になじみやすく、かつPrologないしCILに翻訳することが容易であることが望ましい。

(2) 文法記述

現在検討されているJPSGの枠組みでは、日本語などの個別言語の文法記述は次のような要素からなることになる。

- (a) 統語素性の宣言：これは統語素性名、その素性が取りうる値の範囲の指定、その素性の省略時の値の指定からなる。
- (b) 統語素性の集合の宣言：これは集合名とその要素の指定からなる。
- (c) 省略記法宣言：使用すべき省略記法とその解釈からなる。
- (d) 統語規則：これは、一般的な句構造規則として表される。
- (e) 統語素性共起制限：これは、複数の統語素性指定間の含意関係として表される。
- (f) 統語原則：これは、関係する統語範疇の素性指定の満たすべき条件として表される。
- (g) 語彙項目：現在のところ、簡単のために、統語範疇の中で特に指定されたものを

語彙項目と考える。

(h) 語彙規則：これは、複数の語彙項目間の含意関係として表される。

(3) 統語原則を記述するための基本的な関係

統語原則を記述する上で使用すべき関係としては、現在のところ以下のものが必要であると考えられる。

(a) 「ユニファイ（または条件付きユニファイ）」：素性指定間の「同一性」を定義する基本的な関係である。

(b) 「（部分集合として）含む」、および「（要素として）含む」といった集合に対して定義される関係：SUBCAIという素性はその値として統語範疇の集まりを取るため（ただし、集まりを「リスト」として実現する場合には、集合に対する関係を「リスト」に対する関係に翻訳する必要がある）。

3. 機能設計

3.1 文法

3.1.1 文法体系

2.1で述べたように、GPSGの枠組みをそのまま日本語に適用するには問題点も多いので、郡司による日本語の分析[Gunji 85]をベースにHGの枠組みの導入や、本プロジェクトでの検討を通じて、独自の新しい日本語文法体系の検討を行った。次に示す第1版では、補語構造のほか主題化、総記化、関係節などの日本語の基本構文や、再帰代名詞の束縛などの現象を扱う。

(1) 素性(Features)

統語範疇は素性(features)の集合であり、素性はその値により次のように分類される。

(a) 二値素性(+, -)

PAS, BRFL,

(b) 多値素性

素性名	素性値
POS	{V, N, P, Q, AD,} (いわゆる品詞)
ORTH	正書法による記述
STEM	ローマ字表記による語幹
VFORM	{ROOT, CONJ, REL, IMP} (活用形)
PFORM	{ga, wo, ni, no, de, e, kara, made,} (助詞の形)
INFL	{VC, VV, ADJ, DA, NIL} (活用型)
GR	{SBJ, OBJ} (文法関係)
ISF	ローマ字表記の終端記号素性
SEM	意味表現

(c) 統語範疇の集合を値として持つ素性

SUBCAT, SLASH, REFL(P[SBJ]のみ)

集合タイプのSUBCAT素性により日本語特有のscramblingを扱うことが可能になる。

(2) 句構造規則

句構造規則としては、「親は右側にくるヘッドと左側にくる他の1個の要素から成る」ことを示す次の1個のみである。

$M \rightarrow D H$

(3) HFC(Head Feature Convention)

SEM, SUBCAT, SLASH, REFL以外の素性はヘッド素性と呼ばれ、親とヘッドのヘッド素性は値が等しい。

(4) FCR(feature Cooccurrence Restriction)

次のような共起制限を導入する。

・ $\langle \text{BRFL} \rangle \rightarrow \langle \text{GR SBJ} \rangle$

・ 全てのカテゴリX に対して

$P[\text{SBJ}; \text{BRFL}] \in \text{SUBCAT}(X) \setminus \text{SLASH}(X) \rightarrow \text{REFL}(X) \cdot \{\}$

など。

(5) FFP(Foot Feature Principle)

$m' = \text{SLASH}(M), d' = \text{SLASH}(D), h' = \text{SLASH}(H)$

$m'' = \text{REFL}(M), d'' = \text{REFL}(D), h'' = \text{REFL}(H)$ とする時、

$m' \subseteq d' \setminus h' \subseteq m' \cup \{P\}$

$m'' \subseteq d'' \setminus h'' \subseteq m'' \cup \{P[\text{SBJ}]\}$

(6) SFP(SUBCAT feature Principle)

$m = \text{SUBCAT}(M), d = \text{SUBCAT}(D), h = \text{SUBCAT}(H)$ とするとき次のいずれかが成立つ。

(a) 補語構造(Complementation):

h と $m \setminus d$ がunify する。

(SUBCAT Precedence: $V \in h$ ならば $\text{POS}(D) = V$)

(b) 任意補語構造(Adjunction):

m と h がunify する。

(c) 等位構造(Coordination):

m と d と h がunify する。

(7) 語彙規則(Lexical Rules) と意味規則

伝統的な変形文法でRaising やEquiと分析されてきたような現象を扱うために、いくつかの語彙規則を導入している。次に例を示す。

LR1a: SUBCATの強化($u \in x$ なら"Equi", その他の場合は"Raising" に対応する)

$X[\text{SUBCAT } x \cup \{Y[\text{SUBCAT } y]\}] \Rightarrow$

$$X[\text{SUBCAT } x \cup u \cup \{Y[\text{SUBCAT } y \cup v]\}]$$

但し、 u と v は PFORM を除いて unify する。

意味規則は構成的意味論 (Compositional Semantics) を用いており、親の意味は子の意味から計算される。計算の方法は句構造規則のタイプによってことなるが、通常の補語構造の場合は次のように計算される。

$$\begin{aligned} \text{SEM}(H) &= \text{SEM}(H)(\text{SEM}(D)) && \text{または} \\ &\text{SEM}(D)(\text{SEM}(H)) && \text{または} \\ &\text{SEM}(H) \end{aligned}$$

意味構造は現在モンタギュ流の内包論理式を用いて記述しているが、将来状況意味論などの新しい意味論による記述の導入も考えられ、具体的な記述形式は最終的には決まっていない。

3.1.2 国語教科書の調査

今後、大規模な文法を開発するには、実際の文章の中でどのような言語現象がみられるかを分析しておく必要がある。そこで小学校六年生用の国語教科書中に現れる文章に対して、品詞分類あるいは用語に対して SUBCAT 素性を割当てて試みを行った。その結果、

- ・複合動詞の扱い、
- ・助動詞の作用範囲 (scope) の記述、
- ・生産的な語尾とみなせるもの (「する」, 「あう」など) の扱い、
- ・形容詞や形容動詞に対する素性の割当て、

のような問題点があきらかになった。これらの問題を更に深く調査することにより、前述の文法体系の詳細化、改善などを行う予定である。

3.2 パーザ

5.4.3.2.1 条件付ユニフィケーションによるアプローチ

(1) 条件付ユニフィケーションの概念

JPSG におけるユニフィケーションは Prolog のそれと異なり、拡張されている。統語範疇は素性とその値とのリストから構成されているが、その値のかわりに値として取りうる範囲が指定されていてもよく、また、異なる値同志の間に何らかの関係が成り立っていてもよい。これを Prolog で表すには単なるユニフィケーションでは不十分であり、従来、手続き付加によって実現されてきた。しかし、この「手続き付加」による方法では幾つかの明らかな欠点がある。たとえ、ある変数の値を制限するために付随している手続きをいつ評価すべきかは一般には分らない。この事は、その変数に関係ないユニフィケーションが行われた時でさえも、いつも手続きを実行しなければならない事を意味する。また、手

続き付加は情報の流れが一方向だけに行われた時のみに効果がある。将来、統語規則を生成と解析の両方に用いるという事を考えた時、この事は障害になるであろう。

そこで我々が提案するのが「条件付ユニフィケーション」〔橋田 85〕である。ユニファイされるパターンそれぞれにはそのパターン中の変数に対する条件の集合が付随している。条件付ユニフィケーションはパターン同志が付随する条件の下でユニファイするか否かを判定し、先の制約に従った新たな条件を作り出す。ただし、次の制約がある：「パターン中の変数に付随する条件はただ一つに限られる。即ち、各変数は、付随する条件の中に高々一回しか現れない。これは、条件定義に対しても適用される制約である。」もしもそのような条件がなければ、ユニファイは失敗する。これはパターンについての情報蓄積のプロセスとみなすことができよう。条件付きユニフィケーションの述語は、

`unify(Pat1, Pat2, Condition, NewCondition)`

で定義される。Pat1, Pat2 はユニファイされるパターン、Condition はそれぞれのパターンに付随するすべての条件のリスト、NewConditionはパターンのユニフィケーションによって作られる新たな条件である。例をあげて示そう。

例：c1, c2, c3は以下のように定義されているとする（Prologと同様の記法をここでは用いている。大文字は変数である）：

`c1(a, b). c1(c, U). c2(a, U). c2(b, a).
c3(U, [U, a, V]). c3(a, [b, U, V]) :- c1(U, V).`

ここで、`unify([A, [B, a, C], D], [X, Y, Z], [c1(A, B), c2(C, D), c3(X, Y)], NewC)`

は成功し、ユニファイの結果、次の条件が求まる：

`NewC = c4(A, B, C, D).`

ただし、`c4(a, b, b, a). c4(c, c, U, V) :- c2(U, V).`

現在、この手続きはProlog上で実現されている。また先の制約は、JPSGにおいては本質的な制約ではない。

(2) 条件付ユニフィケーションを用いた構文解析の一手法

①語彙項目

すべて次のような条件付パターンと考えている。

`lexicon(Entry, cat(head(POS, FORM, INFL, GR, TSF, SEM), Subcat, Slash, Refl), Cond).`
cat の内容は日本語に対して提案されている統語素性の並びであり、これは条件付パターンである。Condはそのパターンに含まれる変数の「条件」のリストである。

②解析アルゴリズム

この条件付ユニフィケーションは、解析アルゴリズムとは独立である。即ち、どのような解析アルゴリズムにおいても用いることができる。ここではDCG的なトップダウンのアルゴリズムを例にあげよう。日本語における補語規則は次のように表されよう（ただしSlash とReflは省略してある）。

`rule(cat(Head, RestSubcat), NewConds, first, last) :-`

```

rule(Cat__of__Complement, Condition__of__Complement, First, Middle),
rule(cat(Head, [Complement | RestSubcat]),
      Condition__of__Head, Middle, Last),
append(Condition__of__Head, Condition__of__Complement, Conds),
unify(Cat__of__Complement, Complement, Conds, NewConds).

```

現在、トップダウンおよびボトムアップそれぞれのアルゴリズムを用いた条件付ユニフィケーション・パーザ(CUP)の構築を進めている〔白井 86〕。

3.2.2 既存の枠組みによるアプローチ

ロジックプログラミングによる文法記述体系および処理系としては、DCG[Pereira 80]、BUP(GALOP)[Matsumoto 83]、MG[Colmerauer 78]、XG[Pereira 83]、GG[Dahl 84]等が既に提案されている。中でもDCGは明瞭な枠組みを持ち記述性の高い文法記述言語である。反面DCGはトップダウン、バックトラックベースで解析を行うため自然言語でよくみられる左再帰規則に対しては無限ループに陥るという欠点がある。BUP(GALOP)はDCGの記述力を損うことなくこの欠点を改良した、ボトムアップに左から右へ解析を行うパーザである。JPSGの句構造規則は唯一個であるというものの本質的には左再帰的であり、ここではBUP(GALOP)の拡張によるJPSGパーザの実装手法を考える。実装言語はPrologの拡張言語であるCIL[Hukai 85]を用いる。基本的な考え方は次のとおりである。

(1) 統語範疇：

素性の集合としての統語範疇は、CILの複合不定項(Complex Indeterminate、以後CIと記す)で実現する。各素性はCIのルールに対応する。

(2) 素性共起制約(ICR)：

ICRは個々の統語範疇内の素性値の関係なのでCIの条件部に記述する。

(3) 統語原理：

SFP, HFP, FFPは親と子の範疇間の関係であるので句構造規則中に埋込む。

これらの考えを基にCILで書いたプログラム例を示す。

・素性の集合としての統語範疇

```
feature __set( (C where fcr(C)) ) <- true.
```

・句構造規則

M → **C** **H** という句構造規則は次のようなBUP(GALOP)節に変換される。

```

nt(c(FC, C), G, X0, X) <-
    t(h(FH, H), X0, X1) &
    sfp(FC, FH, FH) &
    hfc(FH, FH) &
    ffp(FC, FH, FH) &
    nt(m(FH, m(C, H)), G, X1, X).

```

ここでsfp,hfc,ffp はそれぞれSFP,HFC,FFP に関するユニフィケーションを行う述語である。

・語彙項目

語彙項目もCIL のホーン節で実現する。次に「起す」という他動詞の例を示す。

```
l(G,[okosu|X0],X) <-  
    nt(l(@lvp,lvp(okosu)),G,X0,X).  
lvp((@feature __set with n:='-',  
                                     v:='+',  
                                     subcat:=(@pp with gr:=obj),  
                                     (@pp with gr:=sbj))))  
    <- true.
```

現在、このような枠組みによりJPSGのリブセットの試作実験を行っているところである。PrologでなくCIL を用いることにより、次のような利点がある。

- (1) 素性の集合としての統語範疇をPrologで実現する方法としては、リストあるいは複合項の引数が考えられる。いずれにしても、その順番はユーザが管理しなければならず、素性の数が増えるに従ってそれは困難な事になる。CIL ではCIのロールとして表現された素性の順番を気にすることなく、特定の素性に対してアクセスすることができる。
- (2) CIL の拡張ユニフィケーション機能により、個々の範疇内の素性値のチェックは、ユーザがいちいち呼び出す必要がなく自動的に行われる。またフリーズ機能により、値が決まるまでチェックをリスペンドすることが可能である。
- (3) Prologの拡張言語なので、既に開発されている手法をそのまま移植する事が可能である。

これらの実験結果は文法グループにフィードバックされ、今後の詳細検討の材料となる。

[参考文献]

- [石森 84] 石森延男ほか, 創造国語六上, 希望国語六下, 光村図書, 1984.
- [白井 86] 白井英俊, 条件付単一化を用いたパーサーについて, ICOT JPSワーキンググループ資料, JPSG-126, 1986.
- [橋田 85] 橋田浩一, 条件付きUnification, ICOT JPSワーキンググループ資料, JPSG-061, 1985.
- [Colmerauer 78] Colmerauer, A., Metamorphosis Grammars, Natural Language Communication with Computers, Bolc, L.(ed.), Springer-Verlag, 1978.
- [Dahl 84] Dahl, V. and Abramson, M., On Gapping Grammars, Proc. of 2nd

- International Conference on Logic Programming, 1984.
- [Gazdar 85a] Gazdar, G., Klein, E., Pullum, G. and Sag, I., Generalized Phrase Structure Grammar, Oxford, Basil Blackwell, 1985.
- [Gazdar 85b] Gazdar, G. and Pullum, G.K., Computational Relevant Properties of Natural language and their Grammars, New Generation Computing, Vol.3, No3., OHMSHA, LTD. and Springer-Verlag, 1985 .
- [Gunji 85] Gunji, I., A Phrase Structural Analysis of the Japanese Language, 1985.
- [Gunji 86] Gunji, I., The JPSG System(Second Revision), ICOT JPS-Working Group memo, JPSG-119, 1986.
- [Matsumoto 83] Matsumoto, Y., Tanaka, H., Hirakawa, H., Miyoshi, H. and Yasukawa, H., BUP ; A Bottom-Up Parser Embedded in Prolog, New Generation Computing, Vol.1, No.2, OHMSHA, LTD. and Springer-Verlag, 1983.
- [Mukai 85] Mukai, K., Horn Clause Logic with Parameterized Types for Situation Semantics Programming, ICOT TR-101, 1985.
- [Pereira 80] Pereira, F. and Warren, D., Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence, 13(1983).
- [Pereira 83] Pereira, F., Logic for Natural language Analysis, Technical Note 275, SRI International, 1983.
- [Pollard 84] Pollard, C., Generalized Phrase Structure Grammars, Head Grammars, and Natural language, Ph.D. diss., Stanford Univ., 1984.
- [Pollard 85] Pollard, C., Lecture on HPSG, Unpublished Lecture Notes, Stanford Univ., 1985.
- [Pullum 82] Pullum, G.K. and Gazdar, G., Natural languages and Context-free languages, Linguistics and Philosophy, Vol.4, 1982.
- [Shieber 85] Shieber, S.H., Criteria for Designing Computer Facilities for Linguistic Analysis, Linguistics Vol.23-2, 1985.