ICOT Technical Memorandum: TM-0157

TM-0157

CILによる文法素性と制約条件の記述

奥西稔幸、杉村領一
三吉秀夫、向井国昭

March, 1986

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

CIL による文法素性と制約条件の記述

奥西　稔幸　　杉村　領一　　三吉　秀夫　　向井　国昭
（(財）新世代コンピュータ技術開発機構）

1. はじめに

文法・パーザに対して論理プログラミングを基にした幾つかのアプローチが行われている。ICOTでは東工大・FTL との協力によるDCG (Definite Clause Grammar)[1]をボトムアップに解析するGALOP(Grammar As Logic Programming) システム[2] の開発、また意味解析のためのGALOP 上へのLFG (Lexical Functional Grammar)の実装[3,4] 、さらにDCG 用の文法記述言語GDL0(Grammar Description Language 0)の実現[5] 等を行ってきた。これらはいずれもDCG を基本としそのメカニズムの中心はユニフィケーションである。現在は、意味・知識表現及び談話理解処理のための、Prologを拡張したプログラミング言語CIL (Complex Indeteminates based Language)[6,16]の開発を行っている。CIL の核は、連想リストとfreeze制御の導入である。これらは、従来、規則指向であった文法理論がそのデータである文法素性重視の文法記述になってきたという最近の流れに対応している。

本稿、前半では幾つかの実際の文法理論を例にとり、最近の文法理論に見られる傾向について検討してみる。後半では、それらがどのようにCIL で記述できるかを述べる。

2. 文法素性

ここでは，文法素性(feature)とは，語彙が持つ大量の固有な情報，及び語彙が（統語的に，また意味的に）組合わされ句・文を構成する際に用いられる情報であると考える。解析の際には，文法素性によって表現された各種の情報を受け渡していくものと考える。それらの情報の受け渡し(passing)を行うためのメカニズムが制約条件（constraints）と考える。

ところが，句・文を構成するのに必要な情報は，文法素性だけでなく規則（rule）にも記述されていた。例えば，支配関係(dominance relation)，順序関係(precedence relation)である。この意味で，解析に必要な言語情報は規則と文法素性とに分離して記述されていた，と言える。最近になり，このような不自然さを補うべく，文法素性の重視が見られるようになってきた。すなわち，文法素性に情報をまとめようというのである。規則中に現れるカテゴリを文法素性の集まり(feature-set)と考えるGPSG(Generalised Phrase Structur Grammar)[7]はそのような流れの先駆けとなった理論である。

また，意味処理の面からも文法素性に対する要求が起こってきた。すなわち，意味に関する情報も文法素性と同じ枠組みで扱いたい，というのである。人間が言語を理解するメカニズムを考えると当然のことであるといえる。LFG,HPSG(Head-driven Phrase Structure Grammar)[9]がこの流れにある。

このように，文法素性が記述する情報が多くなることは，計算機への実装により強力で柔軟な枠組みが必要なことを示唆する。文法素性は，一般に，属性／属性値対（attribute/value pair）で表現されている。これは属性／属性値対が，数学的に部分関数（partial-function）と考えられ，また，属性をラベル，属性値をノードと考えれば有向グラフ（directed-graph）とも言え，等式理論（ユニフィケーション）を基礎とする数学的手法の導入が容易になるからである[8,12]。

以下では，個々の文法理論における文法素性の特徴・形式などについて触れる。

[DCG] DCG には，次のような特徴がある。

- CFG 規則とホーン節を基にしたDCG 節が類似しているため対応がとりやすい，
- 各文法カテゴリがPrologの述語に対応しているため，理解が容易，

その反面，

- 文法カテゴリを項(term)として表現し，素性を引数部に展開，属性はその引数位置で属性値はそこに代入される値と考えれば，属性／属性値対の一種といえる．また，文法データを文法カテゴリの引数として持ちまわるため，修正・変更が困難であるといえる，
- 引数の記述に制限がない，平板になる，等，一般に可読性が低い，

[GDL0] この様に規則の記述には向いているDCG も、文法素性の記述には決して向いていないことがわかる。そこで、GDL0では、次のような機能を導入している。対象指向性でいうクラスの概念を導入し、カテゴリをクラスと考え、素性をそのクラス変数としている。また、クラスの継承により、GPSGなどでいうHFC(Head Feature Convention)等を実現することでデータの抽象化が図られている。（記述例参照）

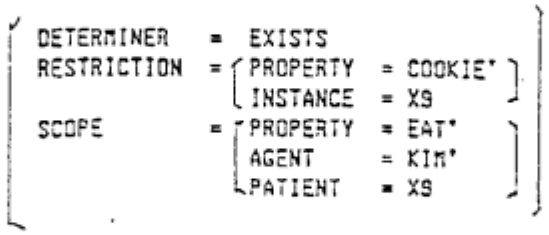
[LFG] LFG では、文の機能を表わすf-structure が属性／属性値対で実現される。f-strucureはsubj,obj,obj2 等の文法機能(grammatica function) 、predの意味形式(semantic form) 、num,per 等の素性記号(feature symbol)からなる。また表層の木構造はc-structure により表わされ、木の各ノードはsubj,comp 等の文法機能にマッピングされる。predはこれらのsubj,obj等を引数にもつ述語式(predicate-argument formula)で表わされる。

```
[ SUBJ   [ SPEC  A                                  ]
         | NUM   SG                                 |
         [ PRED  'GIRL'                             ]
| TENSE  PAST                                       |
| PRED   'HAND<( SUBJ)(  OBJ2)(  OBJ)>'            |
| OBJ    [ SPEC  THE                                 ]
         | NUM   SG                                  |
         [ PRED  'BABY'                              ]
| OBJ2   [ SPEC  A                                  ]
         | NUM   SG                                  |
         [ PRED  'TOY'                               ]
]
```

f-strucure of "a girl handed the boy a toy"

[GPSG] GPSGについては最初に提唱された後、いくつかのバージョンが発表されているが、基本的にはカテゴリを素性の集まりとしている。なお、各バージョンの比較については[14]に詳しくのべられている。

[HPSG] semantic-form として、従来の固定された引数を持つ項ではなく、複雑な情報も伝達できるように拡張性のある枠組みとしている。具体的には、functional unification metalanguage とよばれるつぎのような形式をしている。



semantics of "Kim eats a cookie"

このように、それぞれの文法理論で同じような属性／属性値対を定義している。GOLOも、クラスの概念を知識工学でのフレームの１つのパラダイムと考えれば同じ枠組みであると言える。

3. 制約条件

文法素性の占める割り合いが大きくなってきたのは前節でのべたが、当然、これにたいし、文法素性を処理して句・文を構成するための枠組み、すなわち制約条件を変える必要がある。前節より、制約条件に対し次のような情報を扱う必要であることがわかった。

①規則で扱っていた情報

②意味情報

③解析の効率向上のための情報

①、②は従来には他の枠組みで処理していた情報である。

ところが、この制約条件に対するアプローチはなるべく限定する必要がある。即ち、変形文法のように、付加、代入、削除などの操作を許すと一般には許されない文まで生成してしまう。そこで、制約条件を文法素性が表されている属性値間の等式(equation)の集まりとする[12]。単一化はその手続であるとする。②のための意味構造へのマッピングも単一化でおこなう(LFG)。

以下では各文法理論での制約条件の記述方式・処理内容についてのべる。

[DCG] extra-condition としてPrologの手続きが記述できる。このため実際にはCFG 以上の記述能力を持つ事になる。

```
記述例    規則      np(Num,Per,np(D,N)) -->
                        det(D),
                        noun(Num,Per,N).
```

[GDLO] DCGでは、カテゴリに対するルールとそれらの適用条件等を混在させており、ともすれば繁雑な文法の枠組みとなった。そこで、GDLOでは、条件の記述する箇所を一箇所に指定することで、文法の可読性を高めようとしている。文法記述の点では、これからのべるLFG,GPSGとも（結果的に）ある程度の限定はされている。

例えば、GPSGでは、条件をより一般化した原則(principle) を設けることで、構造規則と制約条件を分離している。また、LFG のように制約条件に記述できる操作を予めシステムが幾つか提供することで、条件記述の乱用を避けているのもある。

```
記述例    カテゴリ    category(np, attribute:num,per,str).
                      category(noun,head-of:np).
          規則        np --> det, noun,
                          { np!str = np(det!str, noun!str),
```

```
np <= noun:[num,per],
det!num = noun!num }.
```

[LFG] LFG では、明示的に記述できる制約条件と、素性に対して暗黙的に課せられた条件がある。
即ち、文法機能のマッピング等には次の幾つかの等式(equality)を用いて記述することができる。

'=' (equate) f-structure の部分的定義、f-strucure間の関係の記述.
'∈' (include) 集合要素の記述.
'=c'(equational constraint) 制約事項への記述.
'^' (check exist) 属性値の存在を要求する.
'~' (negation) 否定的記述
'↑', '↓' (immediate dominance) それぞれ規則中の親、娘ノードを指示する.
'⇑', '⇓' (unbounded dominance) それぞれcontroller,controllee を表す.

記述例 規則 vp --> v, np, np.
(↑obj)= ↓ (↑obj2)=↓

また、これとは別に文法素性に対して、次の３つの条件がチェックされ、非文法的な文の解析を失敗させる。

唯一性(uniqueness) 属性値は唯一つの値を持つ.
完全性(completeness) 素性predの文法機能はそのpredを含むf-structure にすべて存在する.
無矛盾性(coherence) f-structure 中の文法機能はそのpredにすべて現れる.

[JPSG(Japanese Phrase Structure Grammar)] 制約条件をより一般化することで次のような原則(principle) として提供している。[11]

FCR (Feature Co-occurrence Restriction)
カテゴリ間の共起制限の導入
<BRFL +> => <GR SBJ> 等である
HFC (Head Feature Convention)

親と主要娘部の主要素性の等価性の記述.

SFP (Subcat Feature Principle)

m=SUBCAT(M),d=SUBCAT(D),h=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 dが単一化する.

(b) 任意補語構造(adjunction): m とh が単一化する.

(c) 等位構造(coordination): m とd とh が単一化する.

FFP (Foot Feature Principle)・

再帰代名詞、穴を素性操作で処理する.

4. 意味・知識表現言語CIL

CIL 言語の開発の動機は、意味・知識表現および談話理解システムの計算モデルの記述である。このためには、知識、文脈を効率よく処理する必要がある。CIL では基本要素として連想リストとfreezeの導入を行っている。連想リストとは、属性／属性値対の集合であり、属性名でその値をアクセス出来る。これは、Prolog上で、平衡２進木を用いて実現されているため、動的な拡張が可能である。また、freezeは、変数に値が束縛されるまでその実行を凍結させる述語あり、これを用いるとプログラムが宣言型に記述できる。他の機能は、これらから組み立てられるべきである。

以下に、CIL の持つ機能・特徴を次に示す。

(1) freeze制御

(2) 拡張単一化処理を組込んだ連想リストの導入

(3) (1) をベースにした遅延実行制御メカニズム

(4) シンタックス・シュガーの導入による知識表現機能

{ α1/β1,... ,αn/βn }	n≧0	連想リスト
{ α1/β1,... ,αn/βn }!αi		スロット・アクセス
X:γ		条件付き項
X?		freeze記法
X??	T:(T=X?)の略記	
@p	X:p(X?) の略記	
X@p	X:p(X?) の略記	
X#Y	X:(X=Y) の略記	

次に、言語仕様を幾つかの例題とともに示す。

(1) スロット・アクセス

```
| ?- {a/b,c/{d/E}}!c!d = h.
E = h
```

(2) スロット名が未定のときはfreezeする。

```
| ?- X!Y=1, Y=b, write(X).
x(_778,t((b,1),_781,_782))
X = x(_778,*),
Y = b
```

(3) スロット値が変数の連想リストの単一化

```
| ?-  {a/X,b/c} = {b/1,a/Y}.
X = 1,
Y = c
```

(4) スロット・アクセスとfreeze記法の組み合せ

```
| ?-  print((X!a!b)?), X!a!b=ok.
ok          %  X!a!b に値が束縛されるまでprint の実行を待たせて
                いる.
```

(5) If-filled デモン

```
| ?-  X={a/b}, Y={a/ @print}, X=Y.
b           %  @printはA:print(A?) のシンタックス・シュガーであ
            %  り, A に値が束縛されればprint を実行。この場合,
            %  スロット値の単一化でb が束縛される.
```

(6) 遅延実行プログラミング

```
| ?- dif(X, a), print(aaaaaaa), X=b.
aaaaaaa     %  dif は遅延型否定. 2つの値が決ってからその比較を
no          %  行う.

| ?- Icon(((X=a)->(Y=b)), F), X=c,Y=a.
F=true      %  Icon(P,F) : P を評価しF にその真理値を返す. P に
            %  は等式, 論理式が記述でき, P の評価はその引数が全
            %  て束縛されるまで待つ.

| ?-  constr((X=a), F), print(pass), X=b.
pass        %  Iconが実行の結果, 値が束縛されるのに対し, constr
F=false     %  では束縛されない. constrは制約条件の記述に使える.
```

(6) のような遅延実行プログラミングに関しては、現在ライブラリを開発中であり、今後はそれらを基に談話理解システムを開発する計画である。ここでは、文法記述に必要と思われる機能の紹介にとどめた。

5. CILへのトランスレーション

現在，以上の文法理論のCIL でのインプリメント実験を行っている。素性と制約条件の部分について以下のようなことがわかった。

・連想リストは，各文法理論の文法素性を属性／属性値対としてそのまま表現できる。

・また，文法記述の面から属性／属性値対に望まれる幾つかの表現力も連想リストではサポートしている[8] 。すなわち，連想リストでは，循環的なデータ記述も許し，さらに，否定的記述や選言的記述もfreezeにより実行を遅らせることにより実現できる。

・連想リストは動的に拡張が可能であるため，HPSGのような意味表記もそのままインプリメントができる。

・LFG,JPSGなどの暗黙的な制約条件はfreeze制御により宣言的に記述することで実現できる。

・またfreeze制御は統語構造から意味表記へのマッピングを行う有効な手段であることがわかった。

・CIL は基本的にはPrologであるため集合はリストで表現し手続き的な処理を加えることで，疑似集合的に扱う必要がある。(LFGでのadjunctionの扱い，JPSGでのsubcatの扱い)

以下に，LFG,JPSGのトランスレーションの例を示す。

[LFG]

'↑'，'↓' は各カテゴリの連想リストを表す変数に対するスロット・アクセスで表現できる。

'='　これはCIL における拡張単一化と同じ機能である．

ex.)　X!pred = give(X!subj, X!obj1, X!obj2).

'∈'　Prologのリスト処理で実現．

'=c'　freezeにより実行を遅延させればよい．

ex.)　(A!spec)? == sg.　　(A!spec)? ＼== sg.

'^'　check exist

ex.)　bound(A!obj2).　　unbound(A!obj2).

否定的記述はそれぞれのオペレータで定義されている（右側）。

また，文法素性のuniquenessは，CIL の拡張単一化により保証される。即ち，1度束縛された値はバックトラックされないかぎり解放されることはない。

これらの等式をもとに，規則は次ぎのように記述できる。

```
v(v(handed)
      { tense / past,
        participate / none,
        pred / hand{Subj??, Obj??, Obj2??},
        subj / Subj,
        obj / Obj,
        obj2 / Obj2 }, B} -->        [handed]
```

LFG でのf-structure の扱いにはcompletenessおよびcoherence の制御が必要である。GALOP 上のLFG では，このチェックのために幾つかのオペレータを設けていた。CIL の場合には，freezeの制御を用いた遅延実行プログラミングで実現する必要がある。即ち，f-structure に現れる属性と，predに現れる属性は完全に一致してなくてはいけないが，その保証は連想リストに拡張性があるかぎり行なわれない。そこで，freezeによりその属性へのアクセスが起こるまでまたせる事にすればよい。上の規則に現れるfreezeがそれである。このようなf-structure へのマッピングはfreeze制御により解析と共進的に行なわれる為，無駄な計算が避けられる。

また，ここでは記さなかったが，long-distance-dependencyの扱いは，難しい問題であり，CIL ではとくに枠組みを設けているわけではない。

[JPSG]

JPSGにおける原則(principle) は，必要な素性にfreezeをかけることにより宣言的に表現できる。

FCR は，素性間同士の共起関係であるから，その真偽は双方の値がバインドされるまで遅延させる。

```
fcr(C) :- !con( (C!brfl='+' -> C!gr=sbj) ).
```

即ち，BRFLの値が'+' とならないかぎり，GRの値のチェックは行なわれない。

SFP は，subcat-featureのカテゴリ・リストに対する処理で記述できる。
例えば，補語構造に対する記述を次にしめす。

```
sfp(FC,                                          /*  補語構造  */
      {subcat/S  head},
      {subcat/S  rest}
   )
       :- subtract(FC,S  head,S  rest).          /*  指定要素の削除  */
```

ここで、subtract(X,Y,Z) はリストY からX と同じ要素を削除してその残りをZ に返すPrologの述語である。

JPSGシステムは、現在パーザをインプリメント中で、その１つのアプローチとしてGALOP システムとCIL の上で実験中である[13]。そこでは、これらの原則は、次のように句構造規則および語彙中に埋め込まれている。

```
規則    nt(d(FD,D),G] -->
              t(h(FH,H)),
             {sfp(FD,FH,FM),          % subcat feature principle
              hfc(FH,FM),             % head heature convention
              ffi(FD,FH,FM)},         % foot feature inheritance
              nt(m(FM,m(D,H)),G).

語彙    t(G,Fea#{pos / n,
              subcat / []})-->
              [ken],
              nt(I(Fea:fcr(Fea)),G).   % feature co-occurence restriction
```

6. おわりに

本稿では、最近、多くの文法理論で見られる素性と制約について、幾つかのシステムを例に挙げながら検討してみた。この試み自体は、とくに新しいものではない[10,12]。しかし、対象とした文法がHPSG、JPSG などの最新の理論であるため、文法素性、制約条件共により一般化、抽象化の傾向にあることがわかった。従来、文法素性・文法規則・制約条件からなっていた文法が、この様な流れのもとで、文法素性とそれに対する制約条件からのみ成立っていると言えるようになった。事実、JPSGにおいては、（現在のところ）句構造規則がわずか１つであるため、それを制約条件と考えると、その文法体系は文法素性とそれに対する制約条件ですべてであると見なせる。

また現在のところ、制約条件として意味解析を記述しているものは幾つか見られるが、文脈処理まで行っているのは殆ど見当たらない。そのほとんどが構文解析のレベルの記述である[5]。意味を１つの素性と考えた場合、状況意味論[15]などの最近の意味論の流れを見ると、HPSGのように動的に扱える枠組みが、今後、有望になると予想される。

CIL へのトランスレーション実験により、我々が目指している制約条件的プログラミングが、自然言語処理の基本となる文法理論に対し、非常に相性のいいことがわかった。CIL はまだまだ核の部分ができたばかりである。今後の拡張・改良により、CIL はより高度なレベルの自然言語処理プログラミング言語になると予想される。

なお、CIL の詳細な機能および開発状況については、別途、報告の予定である。

［参考文献］

[1] Pereira, F. and Warren, D., Definite Clause Grammar for Language Analysis, Artificial Intelligence, 13(1983).

[2] 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.

[3] Bresnan, J. and R. Kaplan, Lexical-functional grammar: a formal system for grammatical representaion, in J.Bresnan(ed.), The Mental Represen tation of Grammatical Relations, The MIT Press, pp.173-281, 1982.

[4] Yasukawa, H., LFG in Prolog -Toward a formal system for representing grammatical relations- ICOT TR-019, 1983.

[5] Morishita, T.,Hirakawa, H., GDLO: A Grammar Description Language Based on DCG, ICOT TM-084, 1984.

[6] Mukai, K., Horn Clause Logic with Parameterised Types for Situation Semantics Programming, ICOT TR-101, 1985.

[7] Gazdar, G., Klein, E., G. Pullum, and I. Sag, Generalized Pharse Structure, Grammar, Basil Blackwell, Oxford, 1985.

[8] Karttunen, L., Features and Values, COLING'84, 1984.

[9] Pollard, C., Lecture on HPSG, Unpublished Lecture Notes, Stanford Univ., 1985.

[10] Shieber, S.M., An Introduction to Unification-Based Approaches to Grammar, ACL'85, 1985.

[11] Gunji, T. The JPSG System(Second Revision), ICOT JPS-Working Group memo, JPSG-119, 1986.

[12] Pereira, F. and Shieber, S.M., The Semantics of Grammar Formalisms Seen as Computer Languages, COLING'84, 1984.

[13] Miyoshi, H., JPSG Parser on CIL-BUP, ICOT JPS-WG memo, JPSG-091, 1985.

[14] Harada, Y., GPSG no 3tsu-no han-ni okeru FEATURE SYSTEMS, ICOT JPS-WG memo, JPSG-071, 1985.

[15] Barwise, J. and Perry, J., Situations and Attitudes, The MIT Press,1983.

[16] Mukai, K. and Yasukawa, H., Complex Indeterminates in Prolog and its Application to Discourse Models, New Generation Computing , Vol.3, OHMSHA, LTD, and Springer-Verlag, 1985.