TM-1273

自然言語の制約ベース文法理論への DOOD 風アプローチ

津田　宏、横田　一正

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5

Institute for New Generation Computer Technology

自然言語の制約ベース文法理論へのDOOD風アプローチ

津田 宏　　横田 一正
(財) 新世代コンピュータ技術開発機構

自然言語の文法理論では, 最近HPSGやJPSGといった制約ベース文法という理論が提唱されている. これらは, (ソート付き) 素性構造という部分情報を取り扱うデータ構造を用い, 文法を単一化や制約を用いて宣言的に記述するところに特色がある. 本稿ではDOOD言語である *Quixote* の高いモデル化能力 (オブジェクト, 継承, 制約) を用いて, 制約ベース文法のデータ構造, および辞書や文法記述によるその実現を試みる. 従来の自然言語意味記述における *Quixote* の有効性と合わせて, 統合的な自然言語の記述枠組として *Quixote* を位置付ける.

A DOOD Apporach to Constraint-Based Grammar

Hiroshi Tsuda　　Kazumasa Yokota
Institute For New Generation Computer Technology (ICOT)
1-4-28 Mita, Minato-Ku, Tokyo 108, Japan
E-mail: tsuda@icot.or.jp

Constraint-based grammars such as HPSG and JPSG are recently advocated in computational linguistics. They are constructed on the (typed) feature structure, with constraints in terms of unification. This paper treats feature structures, lexicons, and grammars in the constraint-based grammar with a DOOD language *Quixote*. *Quixote* can offer such useful features for grammar descriptions as objects, property inheritance, and constraints. In addition to the previous work about semantic description, *Quixote* gives a total description scheme of constraint-based grammar formalisms.

1 はじめに

自然言語処理とデータベースとは従来から関連が深く，辞書 / 文献データベース，様々なデータ表現，自然言語によるデータベース問い合わせ等の分野で，多くの研究が続けられている．

自然言語処理と一言で括っても，応用により種々の方法論がある．しかし，大きくは知識ベースアプローチと経験的アプローチに分けることができる．知識ベースアプローチは，辞書や文法，意味，文の構造を記述可能なものとしてその記述枠組を研究していく．一方，経験的アプローチは，例文 (コーパス) 等のデータを用いて統計的な手法を用いる研究方法である．こちらは 1950 年代に流行ったものの，芳しい成果が少なくしばらく忘れ去られていた．しかし経験的アプローチは，最近の計算機の進歩や，大規模なコーパスに基づいた自然言語処理 [4] や知識獲得等の点から見直されており，データベースへの応用 [18] 等も考えられている．

本稿では知識ベースアプローチを取る．まず，計算機言語学 (computational linguistics) における最近の文法理論を取りあげる．それらの多くは，統一的なデータ構造を使い，また制約の考えを用いて宣言的に文法を記述するところに特徴がある．一方，演繹オブジェクト指向データベース (DOOD) は，データベースに高いモデル化および推論能力を加えるという動機づけから研究が進められている．DOOD の特徴は，最近の文法理論におけるデータ表現や辞書 / 文法の実現にどう生かされるだろうか．

近年の文法理論を分類する軸は二つある．一つは，文の構造のレベルをいくつか仮定し，それらの間の変形操作により文法を構築する変形文法か，あるいは変形操作を持たない非変形文法であるかという軸である．変形文法にはチョムスキーによる GB (Government and Binding) 理論があり，非変形文法の中で最も重要なのは単一化文法 (unification-based grammar)[13] である．これらの代表的なものは，LFG (Lexical Functional Grammar)[1], GPSG (Generalized Phrase Structure Grammar)[6], HPSG (Head-driven PSG)[9, 10], JPSG (Japanese PSG)[7] である．単一化文法は主として素性構造をノードとする句構造文法であり，この特徴から論理型プログラム言語と相性が良く自然言語処理の文法記述としても重要である．

もう一つの軸は，言語の文法をルールで手続き的に記述するか，制約 / 原理で宣言的に記述するかというものである．前者には GPSG や LFG があり，後者には GB 理論，HPSG, JPSG がある．手続き的な処理だけを考えれば，前者の方が容易である．しかし，後者の方は形態素，構文，意味が統一的な制約という枠組でとらえているため，より柔軟で拡張性のある記述が可能である．最近ではもっぱら後者の方に研究の主流は移っている．

これらの文法理論の実装プログラム言語は，ルールベースの単一化文法については，FUG, PATR-II[13] など研究は多い．しかし，制約 / 原理ベースの文法記述を自然に実装できるプログラミング言語の研究は遅れている．ソート付き素性構造の処理エンジン ALE[2] のようなツールは最近ようやく現れてきているが，統合的な実装環境というものはまだない．筆者らは [16] にて，制約論理型言語 *cu-Prolog* を提唱し，自然言語の制約を制約として処理することで制約ベースの単一化文法 (制約ベース文法: constraint-based grammar[14]) の一つの実装枠組を与えた．しかしながら，cu-Prolog は宣言的にはホーン節論理と同等であるため，文脈に依存するような意味の記述，オブジェクトや継承概念に基づいたモデル化能力という点では不満が残った．

本稿では，演繹オブジェクト指向データ

ベース言語 QUIXOTE を利用して制約ベース文法の (ソート付き) 素性構造の表現, それを利用した辞書や文法の記述を試みる. 文法理論として具体的には HPSG または JPSG を取り上げる.

従来 QUIXOTE を自然言語に応用した研究としては, [15] のように, QUIXOTE のモジュールを状況と見なし, 状況理論に基づいた意味記述や推論に関連したものが多かった. しかしながら, DOOD 言語としての QUIXOTE の特徴であるオブジェクトによる高いモデル化能力は, 自然言語の辞書や文法のデータベースを実現するという応用においてこそ力を発揮する. 本稿では主として自然言語の構文的な側面への応用を考える. そして, QUIXOTE を自然言語の制約ベースの統合的な記述枠組として位置付けていこうというのが筆者の目標である.

2 制約ベース文法

HPSG, JPSG のような制約ベース文法は, 素性構造と呼ばれるデータ構造をノードとする句構造文法として定式化される. 本節ではその簡単な紹介を行なう.

2.1 素性構造

素性構造 (*feature structure*) は, 部分情報を表現するデータ構造で, 単一化文法 [13] で共通に使われる.

素性構造は, 素性から値への部分関数であり, 素性と値の対の集合である AVM (attribute-value matrix) 記法により表されることが多い. (1) はその一例であり, 素性 number から *singular* へ, person から *third* への関数を表している [13]. 他の素性の値については未定義である.

$$\begin{bmatrix} \texttt{number} : \textit{singular} \\ \texttt{person} : \textit{third} \end{bmatrix} \qquad (1)$$

素性構造は, 素性をリンク, 値をノードとする, DAG(directed acyclic graph) 構造としても定式化される [13]. また, [11] では, 素性構造とデータベースにおける complex objects との関連について述べている.

2.2 ソート付き素性構造

HPSG の最新の定式化 [10] では, 素性構造を拡張したソート付き素性構造 (*sorted feature structure*[10]) [1] を基本的なデータ構造として用いている. ソート付き素性構造は, 各構造がソート記号によりラベルづけされている素性構造である. ソート記号の間には半順序関係による階層があり, 上位のソートの情報が下位のもの (サブソート) に継承される. このため効率良く自然言語の記述を行なうことができる

ソート付き素性構造は, (2) のように AVM 記法にソート記号を加えて表現される. これは, HPSG 風に単語 "run" を記述したものである. (2) のソートは word であり, CAT は文法カテゴリー, PH は音素, SUBCAT はこの動詞が取る補語 (の集合) を表す. SUBCAT の値は, phrase でラベル付けされているソート付き素性構造である. なお, HPSG では word および phrase は, いずれもソート sign のサブソートである.

$$\begin{bmatrix} \textbf{word} \\ \texttt{CAT} : [\textbf{vp}] \\ \texttt{PH} : [\textbf{run}] \\ \texttt{SUBCAT} : \begin{bmatrix} \textbf{phrase} \\ \texttt{CAT} : [\textbf{np}] \end{bmatrix} \end{bmatrix} \qquad (2)$$

2.3 ルール, 構造原理

制約ベース文法では, 文法は句構造の枝分かれルールと, 枝分かれにおける複数の素性構造の間の局所的な制約として与えられる. 一般にはルールの数は非常に少なく, 例えば JPSG では $M \rightarrow DH$ という一つだけのルール (左の子が D, 右の子が H, 親が M) から

[1] [3] では, 型付き素性構造 (typed feature structure) とも呼ばれる.

成る．文法のほとんどの情報は局所的制約にあり，これを構造原理と呼ぶ．[2]

構造原理は複数の素性構造の素性の間に成り立つ制約で，宣言的に記述される．以下に幾つかJPGSの構造原理の例を挙げよう．

主辞素性原理 M の主辞素性の値は，H の同じ素性の値と単一化する．

SUBCAT素性原理 M のSUBCAT素性の値は，H の同じ素性の値から D と単一化したものを除いたものと単一化する．

ここで主辞素性とはpos, gr, semのような素性，SUBCAT素性とはsubcat, adjacentのような素性のことを言う．このように，単一化のような処理の方向によらない操作を中心に記述しているのが特徴である．[3]

(1)はJPSGにより「健が走る」という文を解析した簡単な例である．SUBCAT素性原理により，「走る」の変数 X が ken に単一化しているところに注意されたい．

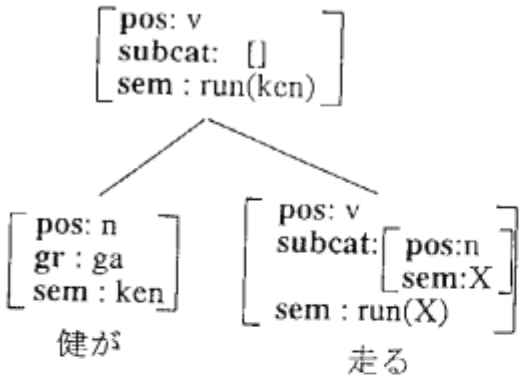

図 1: JPSGの解析例: 「健が走る」

3 辞書の記述

DOODと自然言語処理とのまず第一の接点は辞書であろう．制約ベース文法の辞書記述においては以下のようなポイントが問題となる．

- 素性構造の記述
- 効率性 (継承による効率の良い辞書記述)
- 制約を用いた曖昧性のパッキング

以降順に *Quixote* による記述を試みる．

3.1 素性構造の記述

3.1.1 素性構造

Quixote では，属性項により情報の部分表現が可能である．例えば(1)は，素性構造自体を指すオブジェクトXを与えて，それを頭部とする属性項にて次のように表現される．

```
X/[number=singular, person=third]
```

素性構造の単一化操作は，頭部のオブジェクトの単一化および，関連するドット項制約の制約解消に対応する．

関連研究としてはプログラミング言語CIL[8]におけるPST(partially specified term)が挙げられる．例えば(1)は，次のPST

```
{number/singular, person/third}
```

により表現される．ただし，*Quixote* の属性項ではさらに制約も記述することができるため，CILに比べて記述力はまさっている．

実際の自然言語システムへの応用では，しばしば全ての素性があらかじめ全て決まっていることが多い．それに対しては，アリティの決まったデータ構造を用いて処理速度の向上を計ることが考えられる．例えば，CILにおけるTST (totally specified term)は，PSTと同じ構文で表され，実行時にはPrologの項に変換され高速な単一化が可能になる．*Quixote* における複合オブジェクト項は，ラベルつきグラフ構造 (labeled graph structure) として定式化されており[17]，固

[2] 制約ベース文法の名前の由来である．
[3] 単一化文法の名前の由来である．

定アリティの素性構造と自然に対応づけできる. 例えば (1) は

```
fs[number=singular, person=third]
```

という複合オブジェクト項により表すことができる. この場合, 素性構造の単一化は複合オブジェクト項のミート操作に対応する.

3.1.2 ソート付き素性構造

Quixote における基礎オブジェクト項と, その包摂関係 $< Bobj, \sqsubseteq >$ により, ソートおよびそれらの継承階層 (inheritance hierarchy) は自然に実現できる. ソート付き素性構造自体は, 素性構造と同様に *Quixote* の属性項と, ソートラベルづけに対応する制約で表現できる. *Quixote* における属性継承は, まさにソート付き素性構造間の情報の継承に対応する.

一番外側の素性構造を Y, SUBCAT の引数の素性構造を Z なるオブジェクトにて表すと, (2) は次のように, 包摂制約と二つの属性項により表現できる.

```
Y=<word, Z=<phrase
Y/[cat->vp,ph->run,subcat=Z]
Z/[cat=np]
```

この場合, *Quixote* の属性継承機構により Y=<word という制約から, 任意のラベル l について, Y.l =< word.l なる関係が成立する. これは word にて定義されている属性が自然に Y にも継承されるという, ソート付き素性構造のサブソートへの情報の継承に対応する.

3.2 継承による効率的な辞書記述

ソート付き素性構造に見られるように, 最近の計算機言語学では, 継承の記述・処理は重要なトピックの一つである. 学会誌 *Computational Linguistics* でも 1992 年第 2,3 号にわたり継承の特集を行ない, [5, 12] といった研究を紹介している.

3.2.1 自然言語における継承

[5] は, 自然言語処理に現れる継承を, 構文論, 音韻論, 意味論という分野毎に分類している. 以下に示すように, 大別すると継承には二つの分類がある: 一つは単一継承であるか / 多重継承であるか, もう一つは単調な継承であるか非単調であるかである.

	単調	非単調 (デフォルト)
単一継承	×	
多重継承	HPSG[9]	ELU[12]

単調で単一継承のシステムでは, 自然言語の辞書記述においてはたちまち破綻を来たす. 英語の不規則動詞や, 日本語の不規則変化のようにすぐに例外が出てきてしまうためである. 例外に対処する手段としては, 非単調な継承を認めるか, 多重継承を導入することになる. 最近の言語理論の構文論における趨勢としては, 多重継承に基づくものが多いようである. HPSG は [9] では基本的に単調で, デフォルトという制限した形で非単調性を入れようとしている. また, [12] は ELU という自然言語処理システムの特徴である非単調な継承を辞書記述に応用している.

3.2.2 *QUIXOTE* による継承を用いた辞書記述

まず, 単調な単一継承の例 [5] を *Quixote* で記述してみよう. ソート verb のサブソートに transitive_verb, intransitive_verb があり, その下に love, expire 等の動詞があるとする. 各ソートにおいて, category, transitive などの素性が定義されている.

```
&subsumption;;
verb >= {transitive_verb,
        intransitive_verb};;
love =< transitive_verb;;
expier =< intransitive_verb;;
&rule;;
```

```
verb/[category=verb, past=ed];;
transitive_verb/[transitive=yes];;
intransitive_verb/[transitive=no];;
love/[form=love];;
expire/[form=expire];;
```

次に, この辞書に beat という不規則変化の他動詞を付け加える. まず, 多重継承を採用すると新たにソート ed_verb, en_verb を定義することで次のようになる (以下, 主要部分のみ記述することにする).

```
&subsumption;;
verb >= {ed_verb,en_verb,
        transitive_verb,
        intransitive_verb};;
love =< {transitive_verb, ed_verb};;
beat =< {transitive_verb, en_verb};;
&rule;;
verb/[category=verb];;
en_verb/[past=en];;
love/[form=love];;
beat/[form=beat];;
```

また, 非単調性を用いて例外を記述する方法もある. これも *Quixote* の複合オブジェクト項における属性継承の例外で記述できる. 例えば,

beat =< transitive_verb かつ
transitive_verb.past = ed
であっても, beat[past=en].past = en である.

```
&subsumption;;
verb >= {transitive_verb,
       intransitive_verb};;
beat =< transitive_verb;;
expier =< intransitive_verb;;
&rule;;
verb/[category=verb, past=ed];;
transitive_verb/[transitive=yes];;
intransitive_verb/[transitive=no];;
beat[past=en]/[form=beat];;
```

以上, *Quixote* の属性継承機構によって簡単な例外の記述を示した. ただし *Quixote* の問題点は, 多重継承における名前の衝突の回避機構がないことである. 従って, この方法では, 上位の情報が衝突した時にどちらかを選択する prioritized inheritance は扱えず, 上位ソートがすべて独立な orthogonal inheritance のみに限られる. 例えば有名な「ニクソンはクエーカー教徒で共和主義者である. 一般にクエーカー教徒は平和主義者で, 共和主義者はそうではない」という例では, クエーカー教徒と共和主義者のうち, より具体的なもの (前者であろうか) の方の素性が優先されるというような情報を, ニクソンについて記述する枠組 (例えば ELU[12]) が必要である.

3.3 制約を用いた曖昧性のパッキング

[16] では, cu-Prolog の組合せ制約を生かした一例として, 同音異義語の辞書記述を挙げている. 同音異義語や多義語については辞書エントリをそれぞれ分けると, 処理の上で何度も辞書引きを行ない効率が悪くなる場合がある. そのような語は一つのエントリにして, 意味のバリエーションを組合せ制約の形で記述するのが良い. 曖昧性の解消は, 構文解析の処理の中で自動的に制約解消により実現されるのである.

例えば, 助動詞「れる」が, 五段, サ変動詞の未然形に接続するというのは *Quixote* では以下のように表現できる.

```
れる/[adjacent=reru_aj];;
reru_aj/[pos=v,infl->{vs,vs1}];;
```

adjacent 素性は直前の動詞を表す. この場合は素性構造 reru_aj で, その infl(活用形) の値は vs(五段) もしくは vs1(サ変) であることを意味している.[4]

[4] この場合 infl 属性をドット項制約で表すと reru_aj.infl &in {vs,vs1} である.

これは選言的素性構造の value disjunction を記述することに対応する．ただし，*Quixote* は cu-Prolog と違って，一般に複数の変数やドット項の取り得る値の組合せを制約で書くことができないため general disjunction は記述できないのが問題である．

4 文法の記述

制約ベース文法は 2.3 節で紹介したように，ルールと複数の素性構造の素性の間の制約から成る．JPSG におけるルールを M <= L,H;; なる *Quixote* のルールで表すと，制約はオブジェクト M,L,R の属性の関係である．

JPSG で pos, gr のような主辞素性の従う主辞素性原理は，以下のようにルールに制約の形で埋め込んで記述する．

```
M/[pos=P] <= L,H/[pos=P] ||
    {M=<phrase,L=<phrase,H=<phrase};;
```

subcat 等が従う SUBCAT 素性原理については，*Quixote* の制約の領域を越えているため，制約のみでは記述できない．しかし，例えば subcat の値をリスト (list[car=H, cdr=L] なるオブジェクト項) で実現するならば，以下のように表現することが考えられる．

```
M/[subcat=MS] <=
      L,H/[subcat=HS],
      member[in=HS,one=L,rest=MS] ||
      {M=<phrase,L=<phrase,H=<phrase};;
member[in=[X],one=X,rest=[]];;
member[in=[X|Y],one=O,rest=[X|R]] <=
    member[in=Y,one=O,rest=R];;
```

[11] も指摘しているが，現行のデータベース言語は制約言語としての側面が弱い．制約ベースの自然言語処理や，状況意味論に基づく文脈処理では文法や情報は処理の方向によらずに宣言的な制約として書かれるところに特徴がある．そのため同一の辞書や文法が，解析や生成の両方向に用いることができるのである．制約の記述及び処理は，*Quixote* においても更に考えていくべき点の一つであると思われる．

5 おわりに

本稿では，DOOD 言語 *Quixote* による自然言語の制約ベース文法の辞書記述などを通して，自然言語の情報を取り扱う枠組を検討した．現在の *Quixote* では，継承の例外や組合せ制約の記述にはまだ不十分な面があるものの，オブジェクトの属性記述による素性構造 (部分情報) の表現，属性継承による効率的な辞書記述などの点で，十分自然な記述が可能であることを示した．従来研究されてきた状況意味論の記述と併せて，*Quixote* は制約ベースの自然言語の統合的な記述枠組と評価することができよう．

今後の発展の一つの方向としては，これらの情報を使って，如何に効率の良い，もしくは質的に面白い処理を行なうかということがある．今後の *Quixote* 上の自然言語処理の展開として，例えば以下のようなものが考えられる．

- *Quixote* のモジュール，ルール継承機構を使った，辞書の分割記述
- 辞書・文法の両方向性 (一つの辞書や文法が文解析 / 生成両方向に使われる)．
- *Quixote* の abductive な推論機能を使い，制約が不足している非文を仮定つきで解析する．
- コーパス (例文) データベースからの知識獲得．継承 / 例外に基づいた辞書の構築を学習によりある程度自動化できないか．

また，前述のように，本稿では自然言語処理に対し知識ベースアプローチという観点から DOOD の有効性を述べた．しかし，最近のデータベースの大規模化 [4] ということを

考えると，経験的アプローチも見直すべきであり，両者の特徴を合わせ持った自然言語処理が望まれている．その実現枠組としてのDOODという研究方向も十分に考える意義がある．

参考文献

[1] J. W. Bresnan, editor. *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass, 1982.

[2] B. Carpenter. ALE – The Attribute Logic Engine User's Guide. Anonymous FTP (CMU), December 1992.

[3] B. Carpenter. *The Logic of Typed Feature Structure.* Cambridge University Press, 1992.

[4] K. W. Church and R. L. Mercer. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics*, 19(1):1–24, 1993.

[5] W. Daelemans, K. D. Smedt, and G. Gazdar. Inheritance in Natural Language Processing. *Computational Linguistics*, 18(2):205–218, 1992.

[6] G. Gazdar, E. Klein, G. K. Pullum, and I. A. Sag. *Generalized Phrase Structure Grammar.* Basil Blackwell, England:Oxford, 1985.

[7] T. Gunji. *Japanese Phrase Structure Grammar.* Reidel, Dordrecht, 1986.

[8] K. Mukai and H. Yasukawa. Complex Indeterminates in Prolog and its Application to Discourse Models. *New Generation Computing*, 3(4):441–466, 1985.

[9] C. Pollard and I. A. Sag. *Information-Based Syntax and Semantics, Vol.1 Fundamentals.* CSLI Lecture Notes Series No.13. Stanford:CSLI, 1987.

[10] C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar.* University of Chicago Press and CSLI Publications, 1993. (to appear).

[11] B. Rounds. Situation-Theoretic Aspects of Databases. In G. P. Jon Barwise, Jean Mark Gawron and S. Tutiya, editors, *Situation Theory and Its Applications, Vol.2*, pages 229–255. Stanford University, 1991.

[12] G. Russel, A. Ballim, J. Carroll, and S. Warwick-Armstrong. A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons. *Computational Linguistics*, 18(3):311–337, 1992.

[13] S. M. Shieber. *An Introduction to Unification-Based Approach to Grammar.* CSLI Lecture Notes Series No.4. Stanford:CSLI, 1986.

[14] S. M. Shieber. *Constraint-Based Grammar Formalisms.* MIT Press, A Bradford Book, 1992.

[15] S. Tojo and H. Yasukawa. Situated Inference of Temporal Information. In *Proc. of FGCS92*, pages 395–404, 1992.

[16] H. Tsuda. cu-Prolog for Constraint-Based Grammar. In *Proc. of FGCS92*, pages 347–356, 1992.

[17] H. Yasukawa and K. Yokota. Labeled Graphs as Semantics of Objects. In *Proc. SIGDBS and SIGAI of IPSJ*, October 1990.

[18] 北川高嗣, 清木康. 意味の数学モデルとデータベースへの応用可能性について. 電子情報通信学会データ工学研究会 *DE92-6*, pages 43–50, 1992.