

論理型言語処理と関係データベースの統合環境

佐野 洋, 佐々木博司†, 小野寺浩†, 小原雅己‡

(財) 新世代コンピュータ技術開発機構 (ICOT)

† 富士通 F-I-P 株式会社

‡ 東芝情報システム株式会社

概要

本稿は、特定の処理方式に依存しない汎用データベースシステム上での辞書構築の実際と、論理型プログラミングという特定の処理方式を与え、自然言語処理の枠組の中で汎用データベースを運用したシステムについて報告する。解析時にデータベースシステムを実時間で運用した結果を報告する。

ICOT で開発されている知識ベース言語について説明するとともに、こうした知識ベースに対応する DCG シンタクスを用いた文法の意味表現について述べ、自然言語処理の枠組の中で知識ベースとの統合の可能性について述べる。

LINGUA:

A Logic-Based Integrated Environment for Language Analysis

SANO, Hiroshi

e-mail : sano@icot.or.jp

Institute For New Generation Computer Thecnology (ICOT)

4-28-1 Mita, Minato-Ku, Tokyo 108, Japan.

and

SASAKI, Hiroshi, ONODERA, Hiroshi

FUJITSU FACOM INFORMATION PROCESSING CORPORATION

Daisan Kyodo Bldg., 4-14, Kamiyamacho, Shibuya-Ku, Tokyo 150, Japan.

and

OHARA, Masami

TOSHIBA INFORMATION SYSTEMS (JAPAN) CORPORATION

Toshiba Information Systems Bldg., 2-1, Nisshincho, Kawasaki-ku, Kawasaki 210, Japan.

Abstract

The central concern of this paper on the issue of combining tools from a parser for LUG (Localized Unification Grammar), a development system of LUG grammar rules, a data base of lexical entries and from a dictionary management system. These are computational tools being developed in the framework of logic programming. The main design objectives for the integrated computer environment, called LINGUA, are to provide LUG grammar development tools that are useful, and that also make it easier to write large grammars and to construct huge data base using existing techniques based on logic programming.

This paper describes the design and implementation of the LINGUA system and give an outline of some of the problems have been encountered.

1 はじめに

辞書の自然言語処理において占める位置は大きい。語彙検索と情報提供という狭義の機能に限定せず、一般的には言語知識ベースとなる。大規模化の要求の高まりにもかかわらず、その使用目的や適用の分野の違い、そして、処理方法の相違から、いくつもの辞書構築のために莫大な努力が払われ、膨大な数の辞書が存在している。言語処理のレベルの差異は、その記述内容に影響を与えており、このことが辞書内容の共通化に否定的な見通しを余儀無くしている。言語処理レベルに応じてその目的に適う辞書を特定の辞書から作りだし利用することは、少なくとも辞書構築に伴う労力の節減につながる。計算機用日本語基本辞書 [3, 4]、EDR 電子化辞書 [1] は、そうした基本の辞書を提供しようとするものである。

これまでも辞書構築系に関するインプリメンテーションの議論はあった。その多くは、固有で特殊なデータベースシステムが仮定されていたために、辞書内容であるデータを演算対象としたデータ変換に関する形式的な枠組との関連が極めて不透明である。基本の辞書から、さまざまな用途に適する辞書への変換や構築については、その方法論は発展途上にある。

本稿は、特定の処理方式に依存しない汎用データベースシステム上での辞書構築の実際と、論理型プログラミングという特定の処理方式を与え、自然言語処理の枠組の中で汎用データベースを運用したシステムについて報告する。解析時にデータベースシステムを実時間で運用した結果を報告し、汎用知識ベース言語との結合のための記述形式の提案を行う。

次節では、論理型プログラミングと自然言語処理について若干の説明と非正規型データベースの辞書記述への応用について述べる。3 節は、論理型プログラミング上で統合化した処理システムについて説明する。統語化システムの設計指針が 4 節で述べられ、実際の運用形態についても言及する。5 節では、最近のデータベース研究の動向と ICOT で開発されている知識ベースについて若干述べるのと同時に 6 節では、こうした知識ベースに対応する DCG シンタックスを用いた文法の意味表現について述べ、知識ベースとの統合の可能性について述べる。まとめでは、単に辞書という枠組でなく、言語知識ベースとしてこれら汎用データベースシステムを運用してゆくことを提案する。

2 論理型プログラミングと自然言語処理

2.1 文の解析と Prolog

論理型プログラミング言語である Prolog の基本処理機構をそのまま解析エンジン (パーザ) として用いる文法記述形式に DCG [5] がある。宣言的な記述が行え、一階述語論理に基づくために理解しやすい性質をもつ。文法は確定節に変換され、この節集合の計算過程が文の解析に対応している [11, 12]。項表現を用いて適切な形式化を行えば文脈自由文法を越える能力をもつ文法の開発が容易に行える。

近年の文法理論は、その基本操作に単一化が仮定されており、それら文法理論が設定する原理や原則を直接表現するデバイスはないものの DCG シンタックスとの整合性はよい。引数の位置や数が固定される項の表現能力は、部分項を扱う CIL などにより改善されている。

単一化ベースの文法であることの利点は、単一形式のデータ表現にある。例えば、形態単位を明確にした文法体系 [7] を与えてやることで、形態の単位と構文の単位に区別がなくなる。従来の形態素解析と構文解析の区別が不要となる。単一の文法によって文の解析が可能である。

2.2 辞書記述と非正規関係データベース

辞書に限らず自然言語処理においては、表現対象を属性とその値の対による記述で示す。辞書はその典型である。辞書記述では、その記述対象は語彙であり、一般に語彙はその形態や文法情報だけでなく、意味的な素性によって特徴づけられている。こうした、素性とその束によって対象を表現する方法は、辞書に限らず単一化文法の枠組にも採用されている。

一方、データベースでは、表現レベルの拡張と処理効率の向上の点から非正規関係を扱う枠組が開発されている。非正規関係データベースでは、属性の抽象化が可能であり、その値もアトミックシンボルに限定されずデータタイプも豊富である。データタイプの豊富さは辞書記述との整合性を向上させている。

我々は、ICOT で開発された、非正規関係データベースである Kappa を辞書記述のための汎用データベースシステムとして採用した。

Kappa (Knowledge Application Oriented Advanced Database Management System) は知識情報処理の応用を

意識し開発された汎用のデータベース管理システム¹ (Database Management System; DBMS) である。

人工知能を始めとする、知識情報処理技術分野では、実験的なシステムの研究開発から、実用規模を目指したシステムの研究開発の段階に入っている。大規模のデータや複雑な知識を効率的に蓄え、かつ管理してゆくことは重要である。さらに、そうしたデータや知識に一貫した整合性を与えることが必要である。Kappa はそうした要請に沿った汎用データベースシステムである。

Kappa の特徴は幾つかあるが、(1) データタイプの拡張によって項表現やリストをデータとすることが可能で、Prolog におけるデータ構造をそのままデータベース化し運用することができる。システム上の特徴として (2) プログラムから Kappa をアクセスするコマンド・インタフェースが拡張機能を持っているので、応用に適したコマンドを定義することができる。関係演算を用いて、新しいデータベース、すなわち新しい辞書を容易に構築することができる。(3) 端末から Kappa を簡便に扱うためのウィンドウ・インタフェースが用意されていて、これを応用することで辞書編集系を比較的容易に構築できる。

一般に、辞書編集系は特定のデータ構造やシステムに依存することが多い。汎用データベースシステム上でそのデータを操作対象とする編集系があれば、データベース上で関係演算を用いて作られた新しい辞書に対しても編集、変更、検索などの操作が可能となる。この点においても汎用データベースシステムを運用する利点は大きい。

3 統合環境 LINGUA システム

我々はすでに論理型プログラミング環境上に自然言語分析支援システム LINGUIST[8] を構築した。DCG シンタックスの文法記述を確定節に変換する BUP トランスレータ [6] を備えた文法開発環境で、文法実行過程を追跡するビジュアルトレーサをもつ。文法編集系はスクリーンエディタ・ベースの仕様の上に文法の部分変換機能を備えるため、文法開発が効率よくおこなえる。

このシステムが文法開発を第一義の開発目的としていた

¹1985 年、Kappa は、その研究開発が始まり、最初のシステム (Kappa-I) は、データモデル、処理効率、データベースとしての言語の能力などを評価する試作システムであった。逐次型推論マシン (PSI) 上で実現された。その第二版の研究開発が行われ 1989 年に、Kappa-II 第 1 版の開発が完了している。

のに対し、LINGUA システムは、解析そのものに力点を置く。実用規模の文法開発に伴い語彙の規模拡大が行われた。語彙の集積に伴う保守や管理などの問題が生じている。抽象記述の言語データの蓄積が進み、さまざまなデータ形式を持つ辞書記述を受け入れなければならない。LINGUA システムは解析・エンジン (パーザ) とデータベース・エンジン (辞書) を用いて文法の実行環境と辞書の記述環境を統合したものである。PSI-II の持つ Prolog マシンとしての機能とデータベースマシンとしての機能を合わせたもので、使用者から見ると 1 つの自然言語処理環境として機能する。

3.1 設計指針

LINGUIST システムで得た開発経験を基に、LINGUA システムは次の設計指針を持って開発された。

1. さまざまな記述形式を持つ複数の辞書に対する操作を可能とする
2. データベースを単にエンジンとして使用する実行面だけでなく、データベースの持つ演算機能を用いて辞書変換の研究の側面を重視する
3. 計算機に特有の知識なくして操作できる環境である
4. データベースシステムを解析用の辞書として直接運用する

第 1 点に沿って、Kappa の持つ端末インタフェースを利用し汎用の辞書編集系を作成した。データベースに対する基本コマンドは、低いレベルのデータ操作である。Kappa には、これら基本コマンド列を、抽象度の高い高機能コマンドとして登録、実行するメカニズムをもつ。辞書編集系は、自然言語の辞書に固有のコマンド列を予め用意したほか、この高機能コマンドの利用者による設定を容易に行えるよう改良している。この結果、利用者は辞書データベースから、目的に合う別の辞書データベースを構築する際に Selection²あるいは Join³といった拡張関係代数演算を特に意識することなく作業を行なうことができるのである。第 3 点は、LINGUIST システムでの経験を生かした。第 4 点に関して、一般には、解析時の実行速度の点から辞書アクセスのために工夫が施されるが、我々は特に意識をせず、Kappa の提供する機能を用いた。

²Selection はテーブルのレコードの選択を行う操作。

³Join は 2 つのテーブルをある属性について結合すること。

3.2 システム構成

LINGUA システムのシステム構成を図 1 に示す。

図 2 には、LINGUA システムの実行画面を載せた。これはエンジン部分で、BUP トランスレータによって文法が確定節に変換され解析エンジンが動作可能にあることを示している。また、Kappa システムに対してデータベースを使用可能とする処理や、使用するデータベースに関する情報を提供している。統合環境はユーザに対し視覚的に情報を与え、操作はダイレクト・マニピュレーションを基本としている。

図 3 は、実際に解析を行っている様子を示している。インデックスサーチのために、解析・エンジンとデータベース・エンジンのインタフェース部分に Trie 辞書を設けている。このインタフェースは ESP で記述され、LINGUA システムのユーティリティの 1 つである。

4 データベースから知識ベースへ

前節で述べた統合環境は解析・エンジンとデータベース・エンジンを備えた自然言語処理システムである。狭義の意味処理やあるいは文脈処理を目指す場合に、語彙は単なる辞書記述の枠組では捉えることができない。ここに知識ベースの必要性がある。

一方、データベースの分野においても、従来の関係データベースモデルに基礎を置くデータベースでは表現することのできなかつた複雑な構造の扱いやデータベースへの問合せ処理の高度化が進んでいる。データベース発展の一方向として演繹データベースがある。演繹データベースはデータから異なるデータを構成する、いわゆる推論に相当する規則も蓄えることができる。こうした機能の向上は、複雑なデータ構造を扱えることと、それらデータへの柔軟な問合せ処理にあり、知識情報処理において要求される演繹や推論を支援するものである。

言語情報処理の高度化に伴ってデータベースは知識ベースへと移行する。本稿は、第五世代コンピュータ・プロジェクトにおいて開発されている代表的な知識ベース言語である *QUIXOTE* との統合について述べる。

5 知識ベース言語 *QUIXOTE*

QUIXOTE[9] は Kappa の上位に位置する知識ベース言語・知識表現言語であり次の設計思想のもとに開発されている。

(1) (制約) 論理型言語の枠組に基礎が置かれ、演繹、制約解消といった推論能力を持つ。そして、(2) 複雑な記述対象の構造を表現するために、複合構造や部分情報などの表現が可能である。

QUIXOTE は、演繹オブジェクト指向データベース [13] にその基礎が置かれ、オブジェクト識別性、複合オブジェクトなどが導入されている。オブジェクトの持つ性質は、オブジェクト識別性に付随する属性として記述できる。これによって自然言語の意味表現における素性構造が扱えるほか、オブジェクト識別子間には順序関係が定義されて、属性の継承(多重継承、例外継承)を扱う。このようにして規則とデータが定義されるとともに、モジュール化の概念によって知識の区分や仮説的推論をも支援する能力を持っている。

QUIXOTE の基本要素は拡張項である。拡張項は頭部とラベルとその値の集合により示される。この基本オブジェクトの性質を示すために属性項と呼ばれる記述が許され、各オブジェクト間には、順序関係があり、属性が順序関係に従って継承される。こうしたオブジェクトはモジュール単位で記述できる。モジュールは識別子を持ち、そうしてモジュール間の関係の記述が可能である。

LINGUA システムでは、Kappa の持つデータタイプ(項)によって文法は比較的単純な DCG シンタクスによる形式で自然言語処理における統合化が図れた。強力な表現能力を持つ知識ベース言語に対応するため、解析のための文法記述形式は抽象化した記述を求めなければならないが、DCG シンタクスの枠組は有用であり Prolog ベースの記述枠組を設定することで、知識ベースとの統合を図ることを目指す。次節では、DCG シンタクス上で実現する意味表現のための LUG 記述形式について述べる。

6 The LUG system

LUG 記述形式は句構造ベースの文法記述形式である。

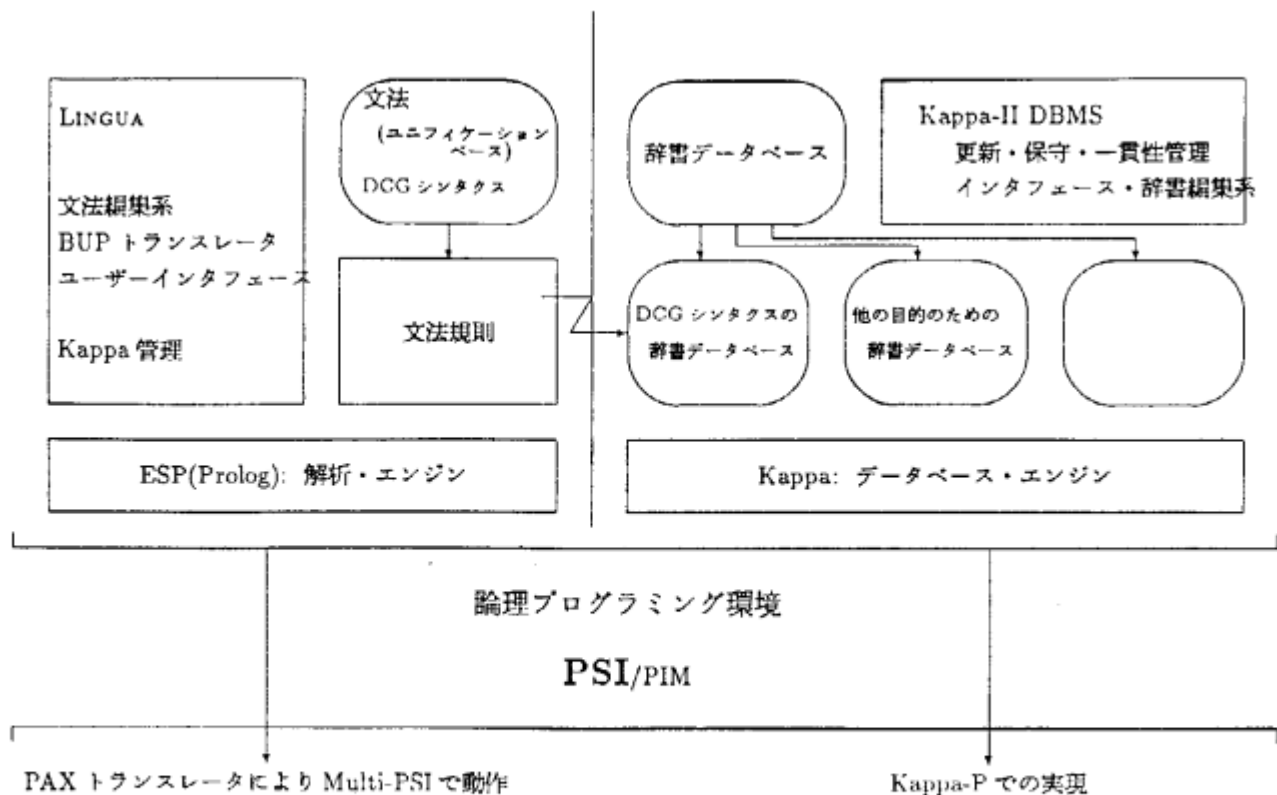


図 1: システム構成図

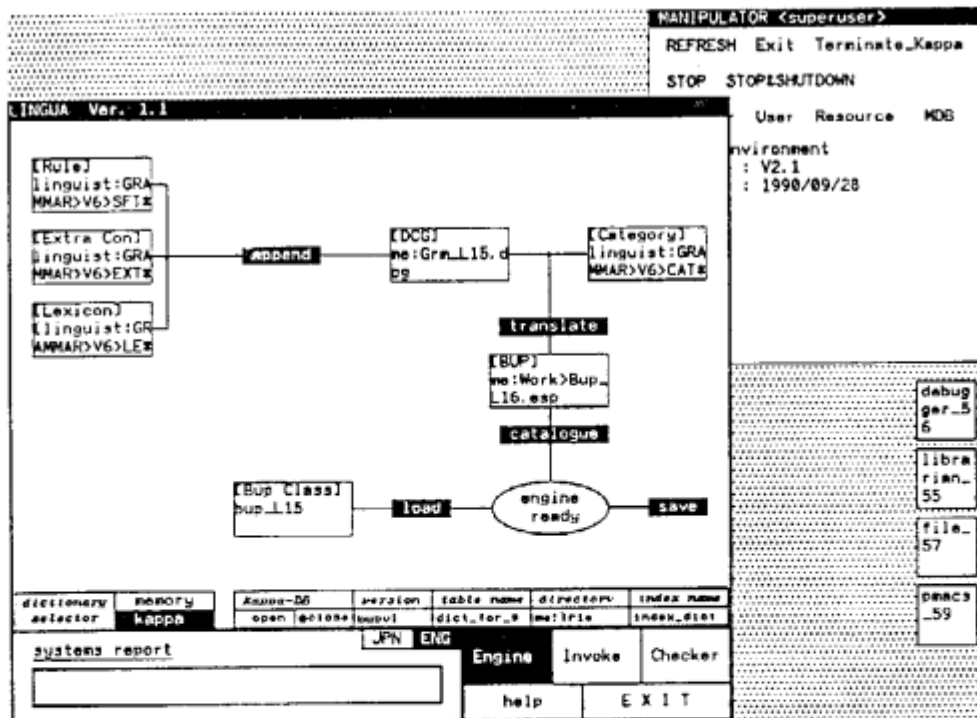


図 2: LINGUA システム 実行画面 (1)

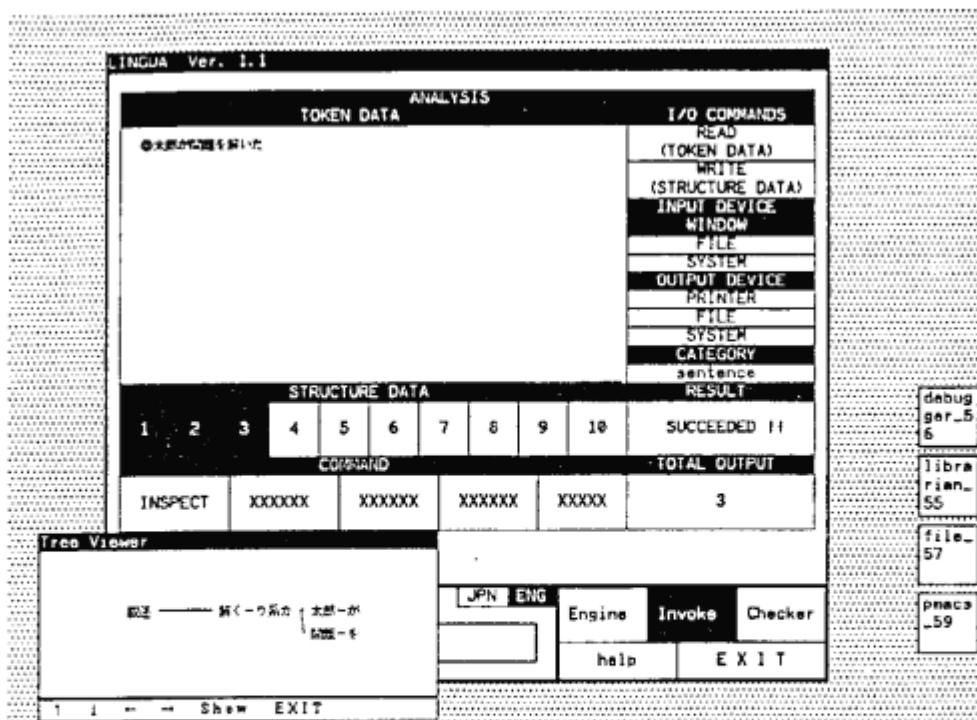


図 3: LINGUA システム 実行画面 (2)

LUG 記述の文法規則は、(1) 文字列の連合操作の点から、形式的には文脈自由のクラスの文法に属する。しかしながら、文法規則の範囲 (非終端) は、Atomic Symbol に限定されず、属性 (Attribute)- 値 (Value) の組みの束 (集合) で示される。(2) 情報の融合操作に仮定される操作は、単一化であることから、文法能力は、文脈自由のクラスを越えている。従って、統語分析におけるオブジェクトは、その範囲の名前で唯一に区分されない。そのため、LUG では、オブジェクト変数を導入し、素性をオブジェクトを識別子とすることで、統語オブジェクトと意味オブジェクトを表現する。統語オブジェクトと意味オブジェクトは、同じオブジェクト変数を共有するので、統語オブジェクトは意味オブジェクトを内在する。

以上の点から (3) LUG 記述形式は文法記述形式であって、同時に統語表示の形式でもある。その統語表示には、意味オブジェクトを含む。統語表示において、範囲名が主導的な機能をせず意味オブジェクトを含むために、統語表示を特徴づける多くの情報は意味表現に極めて近く、QUIXOTE との整合性が高い。更に、(4) LUG 記述形式は、極めて簡便なシンタックスを持つ。文法規則は DCG シンタックスを採用し、単一化操作は Prolog の単一化を利用して、そうして、統語オブジェクトと意味オブジェクトは Prolog のデータ構造を使って表現する。

6.1 オブジェクト構成子

6.1.1 オブジェクト変数

オブジェクト変数は、それ自身で内部構造を持たないオブジェクトである。LUG では、Prolog の変項を割り当てる。但し、他の変数との区分の目的から、識別子 '指標' を項とする述語表現で示す。指標 (X) で示される変数 X はオブジェクト変数である。

6.1.2 素性記述

素性構造は、オブジェクトの性質を、属性とその値の対の束 (集合) によって特徴づける。属性- 値の組みを LUG 記述形式では述語表現を使って表し、Attribute (Value) は素性を示し、素性群をリストを使って束 (集合) で示す。

Attribute は Atomic Symbol であり、Attribute の集合が導入される。あるオブジェクトの構造を規定する素性は (1) Value の値が Atomic Symbol である Simply-Valued Feature と (2) オブジェクト変数を基にした素性記述であって Value の値に Atomic Symbol とオブジェクトあるいはオブジェクト変数が許す Complex-Valued Feature の 2 種類がある。

$$\text{Simply valued Features} ::= \left[\begin{array}{l} \text{Attribute}_1(\text{Value}_1), \\ \dots, \\ \text{Attribute}_n(\text{Value}_n) \end{array} \right]$$

$$\text{Complex valued Features} ::= \left[\begin{array}{l} \text{Attribute}_1(O_{\text{variable}}, \text{Value}_1), \\ \text{Attribute}_2(O_{\text{variable}}, \text{Object}_1), \\ \dots, \\ \text{Attribute}_{n-1}(O_{\text{variable}}, \text{Object}_m), \\ \text{Attribute}_n(O_{\text{variable}}, \text{Value}_n) \end{array} \right]$$

特に、オブジェクト変数に関する記述を持つ構造を基本オブジェクト (*Basic Object*) と呼んでいる。

$$\text{Basic Object} ::= \left[\begin{array}{l} \text{Attribute}_1(O_{\text{variable}}, \text{Value}_1), \\ \text{Attribute}_2(O_{\text{variable}}, \text{Object}_1), \\ \dots, \\ \text{指標}(O_{\text{variable}}), \\ \text{Attribute}_{n-1}(O_{\text{variable}}, \text{Object}_m), \\ \dots, \\ \text{Attribute}_n(O_{\text{variable}}, \text{Value}_n) \end{array} \right]$$

6.1.3 制限記述

素性記述は、あるオブジェクトを特定し区分するの記述である。LUG 記述形式には、オブジェクトの性質を制限する制限記述 (*Restrictional Expression*) がある。あるオブジェクトに対する制約を示す記述が可能である。

$$\text{Restrictional Expression} ::= \text{Attribute}(O_{\text{variable}}, \text{Operator}, \text{Value})$$

Operator は *Atomic Symbol* のデータタイプを持ち、*Value* は *Atomic Symbol* に限る。この記述を用いることでいわゆる用言の格要素に対する選択制限を記述できるほか、体言や用言自身の性質記述などが可能となる。

6.2 統語オブジェクト

LUG で導かれる構成素構造は統語オブジェクト (*Syntactic Object*) に対応する。文のレベルに限らず、語のレベルや句を構成単位としても同様の統語オブジェクトが存在する。統語オブジェクトは $\{SYN, REL, F\}$ の形式を持つ。特に、これを基礎三組 (*Basic Triplet*) という。

$$\text{Syntactic Object} ::= \{SYN, REL, F\}$$

SYN は形態統語 (*Morph syntactic*) に関する情報を表し、*Symply valued Feature bundles* である。REL は、統語意味 (*Syntax semantic*) に情報を示し、*Basic Object* の構造を持つ。F は、ある統語オブジェクトが支配する他の統語オブジェクトをオブジェクト変数を使って明示する。論理的に直接指定されるオブジェクトが、日本語の体系下でその要素に課される統語制限と形態制限とともに記載される。統語上、支配することができなかった統語オブジェクト

については、その痕跡を示す機能を持っている。論理表現と自然言語表現とのインタフェース部分として機能する。

語彙構造を持つ述語構造を仮定する述語表現は REL に反映される。直接下位範疇化要素の形態あるいは統語特徴は F に記述する。

F は差分リスト構造を持ち、それぞれのリストに含まれる要素は、(MOR, GRF, OBJ) である。MOR と GRF はそれぞれ、形態指標と統語関係子と呼ばれ、前者が「が」、「を」、「に」…、後者が *subj*, *comp*, … といった Atomic Symbol である。OBJ はオブジェクト変数である。

6.2.1 Tag feature

Complex valued feature において、その *Value* がオブジェクト変数あるいは内部構造を持つオブジェクトである素性 (*feature*) の *Attribute* を *Tag* と呼び、その素性を *Tag feature* という。

あるオブジェクトに対して *Attribute* は属性名や性質名を示し、*Tag* は統語上の依存関係名を示すのである。*Attribute* と同様に *Tag* も、その集合が導入される。LUG 記述形は、単に文法記述の形式であるばかりか、オブジェクトの構造を明確にすることで、文の統語表示のみならず、意味表示能力をも持っているのである。

6.3 意味オブジェクト

LUG 記述の統語オブジェクトから意味オブジェクトを構成的に得ることができる。意味表現のための意味オブジェクトは、基本オブジェクトが、それに相当する。

一般の記述を示す。

$$\text{Semantic Object} ::= \left[\begin{array}{l} \text{Attribute}_1(O_{\text{variable}}, \text{Value}_1), \\ \dots, \\ \text{Tag}_1(O_{\text{variable}}, \text{Object}_1), \\ \dots, \\ \text{指標}(O_{\text{variable}}), \\ \dots, \\ \text{Tag}_k(O_{\text{variable}}, \text{Object}_k), \\ \dots, \\ \text{Attribute}_n(O_{\text{variable}}, \text{Value}_n), \end{array} \right]$$

統語オブジェクトからの意味オブジェクトへの変換は、直感的には REL の抽出になる。但し、*Attribute* と *Tag* が属する集合を構成する領域が統語領域から意味領域に変わる。統語オブジェクトにおいて、*Tag* の含まれる集合は {動作主語, 補語, 並列, …} といったものであるが、意味オ

ブジェクトにおいて、Tag は {agent, object, cause, time, sequence, ...} という集合の要素となる。

6.3.1 QUIXOTE 表現への変換

QUIXOTE では、あるオブジェクトを識別するためにオブジェクト項を用いる。LUG 記述の意味オブジェクトはほぼ、QUIXOTE のオブジェクト項に対応する。

6.3.2 オブジェクト項と属性項

LUG 記述における統語オブジェクトとそのオブジェクトから導くことのできる意味オブジェクトは述語表現を用いて素性を表現している。素性記述の制約は等値性とそれらの連言だけに限られている。同じ述語表現を用いる制限記述は、厳しい等値制約ではなく等価であることやサブクラスであることを表現する。QUIXOTE における属性項に制限記述が対応する。

制限記述表現は QUIXOTE では属性項に変換されるから、QUIXOTE 処理系の持つ継承の機能なりが利用でき、知識ベースへの問合せ処理がそのまま意味処理に対応することになる。

7 おわりに

汎用データベースシステムである Kappa を用いて、辞書をデータベース上に構築しただけでなく、論理型プログラミングという実際の処理方式を与え、自然言語処理の枠組の中で汎用データベース運用したシステムについて報告した。解析・エンジンとデータベース・エンジンは、PSI-II 上で統合システムに管理されて動作する。この場合、PSI-II は自然言語処理マシンとして機能するかに見えるのである。

現在 Kappa は、Kappa-II の機能整備のほかに、並列推論マシン (PIM) 上の並列 DBMS (Kappa-P) の研究開発が行われている。Kappa は、ICOT で研究開発している知識ベース管理モジュールの中で、データベース・エンジンの位置を占めている。ICOT で研究開発中の知識表現言語・知識ベース言語 QUIXOTE によって、分子生物学データベース、判例データベースなどの構築が予定されている。

知識ベースシステムに対応する DCG シンタックスを用いた文法の意味表現の記述形式を述べた。紙面の都合、詳細説明に至ることができなかったが、解析結果のオブジェク

は知識ベースへの問合せ処理に対応するだろう。

謝辞

データベースシステムについての多くのコメントと示唆をいただいた横田一正氏 (ICOT)、河村元夫氏 (ICOT) に感謝します。福島秀顕氏 (ICOT) には、草稿に対しコメントを頂きました。ここに併せて感謝します。

参考文献

- [1] EDR TECHNICAL REPORT: TR-011, TR-012, TR-013, TR-014, TR-015, TR-016, 日本電子化辞書研究所, 1989.
- [2] Kappa 説明書 (2.2 版), ICOT, 1990.
- [3] 「計算機用日本語基本形態辞書 IPAL」, 情報処理進行事業協会技術センター編, 1990.
- [4] 「計算機用日本語基本動詞辞書 IPAL」, 情報処理進行事業協会技術センター編, 1987.
- [5] Pereira, F et al: *Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks*, Artificial Intelligence, No.13, pp.231-278 (1980).
- [6] Yuji Matsumoto, Masaki Kiyono, Hozumi Tanaka: *BUP toransreta*, Densou-ken-i-hou, Vol.47, No.8, 1983.
- [7] 佐野洋, 福本文代: 「文の様相区分の文型投射への試み」, 「自然言語における統語」シンポジウム論文集, 電子通信学会・日本ソフトウェア科学会共催, 1991.
- [8] 佐野洋, 田中裕一, 佐々木博司, 小野寺浩, 木下聡: 「論理型文法の枠組における言語分析支援環境」, 自然言語処理研究会, 情報処理学会, 1990.
- [9] 「知識ベース / 知識表現言語 QUIXOTE」, 成果報告書, ICOT, 1991.
- [10] Yukihiro MORITA, Hiromi HANIUDA, Kazumasa Yokota: *Object Identity in QUIXOTE*, データベース・人工知能合同研究会, 情報処理学会, 1990.
- [11] 田中穂積, 松本裕治: 「自然言語処理における Prolog」, 情報処理学会誌, NO.12, VOL.25, 1984.
- [12] 田中穂積, 新田義彦: 「ロジック・プログラミングと計算言語学」, 情報処理学会誌, NO.8, VOL.27, 1986.
- [13] 横田一正, 西尾幸治郎: 「演繹・オブジェクト指向データベース」, 情報処理学会誌, NO.2, VOL.31, 1990.
- [14] 吉田将: 「辞書構築における諸問題」, 情報処理学会誌, NO.8, VOL.27, 1986.