

談話理解実験システム DUALS 第3版における 自然言語処理

Natural Language Processing of Experimental Discourse Understanding System DUALS-III

杉村 領一^{*1}
Ryoichi Sugimura

田中 裕一^{*}
Yuichi Tanaka

橋田 浩一^{*}
Koichi Hashida

向井 国昭^{*}
Kuniaki Mukai

* (財)新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology

1989年2月10日 受理

Keywords: natural language processing, discourse, logic programming.

1. はじめに

ICOTにおける自然言語処理の研究は、計算機との知的なソフトウェアインタフェースを目的としていることから、談話理解の研究開発に重点が置かれている。

前期3年の談話理解の研究開発目標は、それまでの自然言語処理要素技術を論理プログラミングの上に再構築するとともに、そこに生じる問題点と可能性を見いだすことであった。具体的な研究開発としては、談話理解実験システム DUALS 第1版を、計算機 DEC 2060 上の論理型プログラミング言語 Prolog を用いて作成し、前期のまとめである1984年11月の国際会議 FGCS '84 において、デモンストレーションを行った。このシステムは、小学校3年生の国語のテキスト18文/200単語を読解し質問に答えたが、システムを構成するパーザ、文法、辞書、問題解決機構などの作りは極めて未熟であり、ほとんど拡張性がなかった。しかし、談話理解システムの要素技術が Prolog の上で無理なくまとまり、自然言語処理システム構築における論理プログラミングの見通しの良さが理解できた。

以上の経緯から、中期4年の目標は第1版を全面的に見直し、パーザ、文法、辞書、問題解決機構などを一般性のあるものに作り変えるとともに、意味・談話処理の機能を研究し⁽¹⁷⁾、これらを統合してシステム

を実際に試作することとした。

まず使用していた計算機を、メモリ容量などに制限のある DEC2060 から、メモリ容量も十分にあり、さらにインタフェース機能などが充実した逐次論理型計算機 PSI-I に移行させた。

中期の2年目には、この PSI-I 上に談話構造記述に適し、Prolog に部分項などを加え拡張した論理型言語 CIL や、高速に構文解析を実行するパーザ SAX のプロトタイプを開発し、これらを用いて DUALS 第2版を完成させた。そして、これらプロトタイプを汎用性のあるツール群「汎用日本語処理系 (Language Tool Box : LTB)」⁽¹⁹⁾ として PSI-II 上に構築する作業を開始した。そして中期末を目標に、この汎用日本語処理系上に、例文数が200文2000単語の規模の、一般質問を許すシステムの構築を始めた。これが DUALS 第3版である。

DUALS 第3版では処理すべき文が多いために、例題に依存したアドホックなソフトウェア設計は通用しない。このことは、DUALS 第3版は一般性のあるシステムにならざるを得ないことを意味している。言うまでもないが、談話理解に大がかりに挑んだ例は世界的に見てもほとんどなく、設計の基本方針を決定することから開発は始まった。

まず、DUALS 第3版の各処理系の設計にあたっては、談話処理に関する一般的な事情、「形態素処理、構文処理、談話処理とレベルが上がるにつれて、これらに必要な処理が複雑になる」ということを考慮した。つまり処理レベルの上昇とともに、処理系をより汎用

*1 1989年4月より松下電器産業(株)技術本部情報システム研究所勤務

性があるものにして、処理規則をより自由に記述できるようにした。具体的には、形態素処理系としては自由度は低いが高速に処理を行える3型(正規文法)のものを、構文処理系には前者よりは自由度の高い2型(文脈自由文法)中心ものを、そして問題解決・談話処理系にはさらに一般的で自由度の高い処理が行える制約伝搬機構を、デバッガなど開発環境とともに、すべて新しくLTB上に開発した。

次に、前記の処理系で用いる日本語に関する情報を中心とした知識ベースの設計においては、下位の処理系で扱うべき知識はできる限り下位の処理系で処理するようにして知識レベルの区分化を進めた。形態素意味知識としては、形態的な情報を手がかりとして、できるかぎり従来よりも深い意味処理に適した知識の構築を行った。次に構文知識としては、原形開発の容易な依存構造文法(係り受け)を基本に構文・意味知識の蓄積を進めた。そして、談話処理・問題解決では、談話の知識をすべて制約として状況理論/意味論⁽¹¹⁾に基づいて記述することを試みている。

以上の処理系や知識ベースを用いてDUALS第3版は作成され、国際会議FGCS'88においてそのデモが行われた。そして、現在もDUALSの研究開発は続いている。

2. DUALS 第3版の構成

DUALS第3版は、その下位の処理を行う汎用日本語処理系LTBと上位の処理を行う問題解決機構などから成り、その構成は図1に示すようなものになっている。

DUALSの各モジュールは、LTBの各ツールの動きを制御するシェル⁽⁵⁾と呼ばれるプロセスによって制御され、データをやり取りするための通信路を介して図中の細い矢印で示したように接続されている。DUALSは、図2へ示したDUALS画面左上の文入力部ウィンドウから文を読み込み、通信路を介して形態素解析LAX、構文解析SAX、問題解決、文生成プランニング、表層文生成で処理を行い、画面下のウィンドウに各処理の途中結果と、処理時間をそれぞれ表示し、左中央のウィンドウへ応答する。

3. 言語知識ベース

3.1 言語知識ベースの概要

DUALSの辞書⁽²¹⁾には、解析、生成の各局面で利用される知識すなわちマスタ辞書やシソーラスと、問題解決の中で利用される意味的な知識とが記述されて

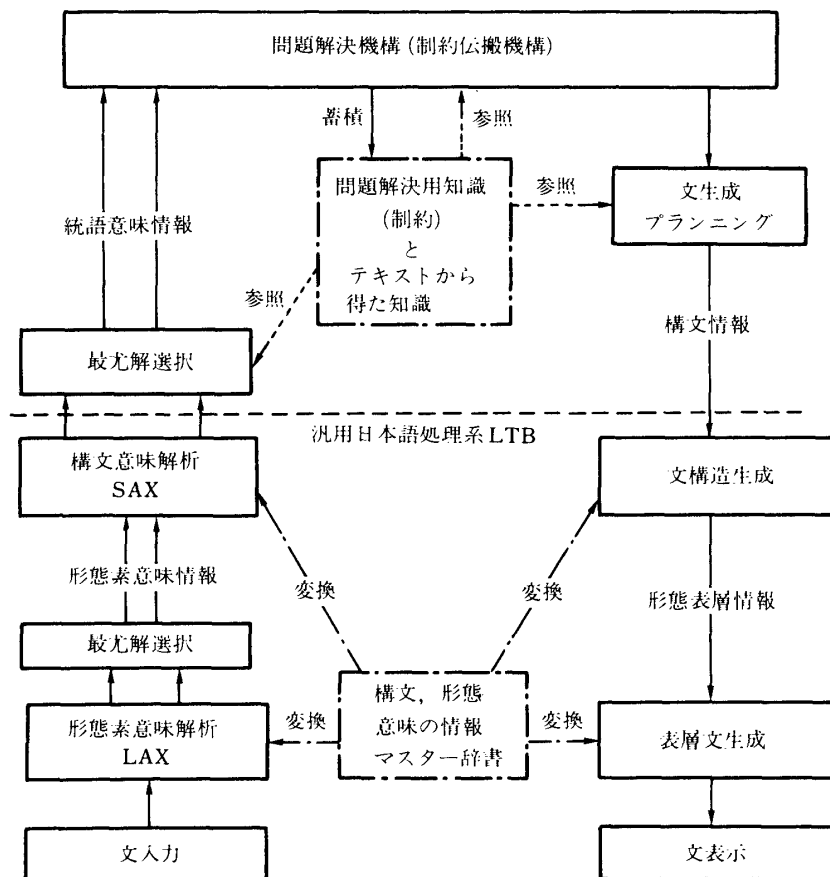


図1 DUALS第3版の構成

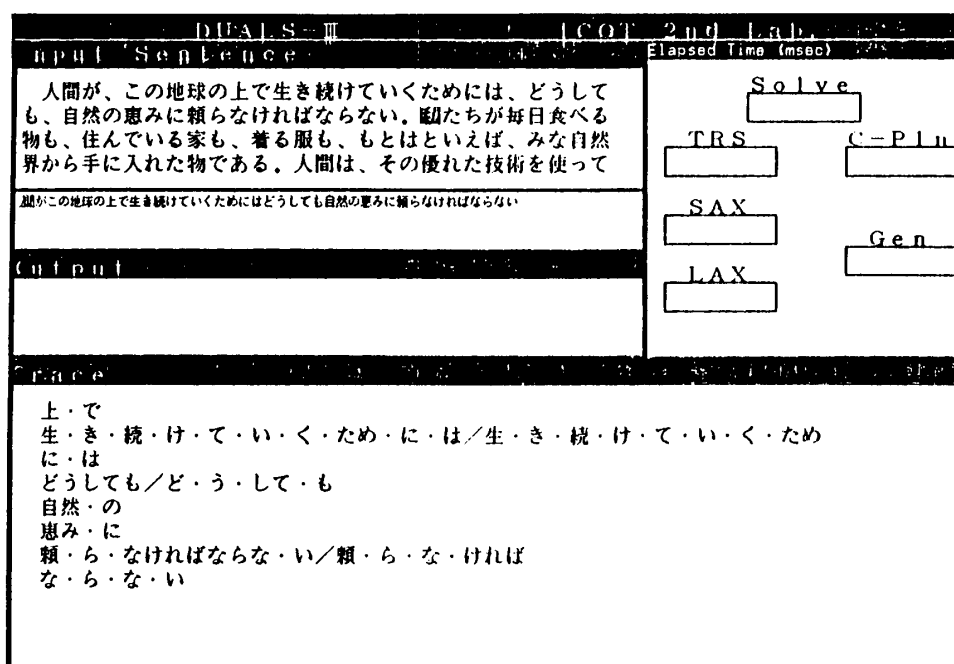


図 2 DUALS の画面

おり、日本語文法とともに一つの言語知識ベースを構成している。

マスタ辞書には、単語の意味を表現する意味構造と、それを文の中で他の単語と組み合わせるための意味構成規則が含まれており、これらはこれまでに順次実験検証を経ながら蓄積されてきたものである。これらは、形態素解析、構文解析、表層文生成のデータとして予め変換され用いられている。

問題解決に使われる知識は、入力文の情報と予め与えられている情報との総和である。後者は言語知識ベースの独立した要素である。

これらマスタ辞書の知識と問題解決用の知識を言語知識ベースとして統合的に記述することにより、記法の統一と知識の一貫性が保たれる統合的なシステムの実現が可能になる。

3.2 マスタ辞書

LTB のマスタ辞書は、見出し語ファイルと語義ファイルからなっている。前者は個々の語に対する見出し語レコードの集合であり、後者は個々の語義に対する語義レコードの集合である。一般に一つの見出し語レコードには図 3 に示す例のように、複数個の語義レコードが対応する。

3.3 問題解決用知識

DUALS 第 3 版において問題解決で用いた知識は、主に指示代名詞の指し示す対象を同定するために必要な照応についての知識と、ある状況と状況の間に成り

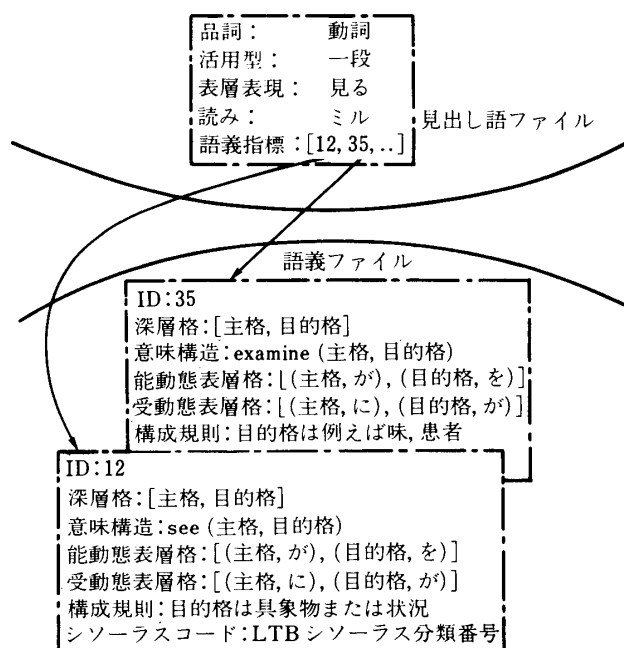


図 3 見出し語レコードと語義レコード

立つ『前提』, Presupposition に関する知識である。ほかに、例えば、談話における話者の意図と発話の間に成り立つ制約や、文中で述べられている事象の時間順序などに関する制約や、敬語表現と話者や聞き手の間の社会的関係との制約など、蓄積すべき知識は多数あり、今後の研究課題である。

3.4 言語知識ベースの今後の課題

DUALS 第 3 版では、形態素、構文解析レベルの処理をソフトウェアとして確立し、その上で文脈解析

レベルの処理を少しずつ実現することを目指した。しかし、この上位レベルの処理が求める情報を下位レベルの処理が提供できるためには、言語知識ベースについて、次のような見直しが必要となっている。

まず、マスタ辞書には解析系と生成系とで必要とする最低限の形態的な情報はそろっているが、DUALSの談話処理能力の向上に伴い、各処理系の必要とする情報とのずれが出てきている。今後、完全に共用できるよう双方が独自に蓄積した統語的、意味的な情報を再度反映させ、その内容を改良していく必要がある。

知識の記述に関しては、まだ記述形式に十分納得のいくものが得られていない。今後 DUALS での実験により、必要十分な規模と内容を蓄積する必要がある。これに加えて知識の増大を図るために、DUALS を利用して学習による知識獲得を行う必要がある。この場合、言語知識ベースとしての一貫性の保持が重要な課題となるであろう。

4. 意味記述言語

4.1 CIL の概要

DUALS において CIL は、①言語情報記述⁽¹⁾⁽¹⁵⁾、②意味処理記述、③システム記述、の三つの役割を担う。①の言語情報記述については、問題解決以外の各モジュールで用いられている「部分項」の有用性が確認されている。②の意味処理記述については、部分項を操作するのに有効な機構を準備している。③のシステム記述については、部分項の言語情報管理用のプログラム環境を一部提供している。

4.2 部分項とその操作

部分項は、各種解析や意味処理の記述をより宣言的に、かつ、容易にするために Prolog に追加されたもので、項としてネットワーク状の構造を表せるレコード構造である。また、値に構造的な矛盾がない限り、中身を無限に拡張できる。CIL では、項は次のような属性と値のペア (属性/値) の集合で記述される。

```
{名前/DUALS, 版/3, 住所/ICOT}
```

CIL は属性や値に関するさまざまな述語を準備している。まず、意味解析処理では、単語の持つ種々細かな意味を順次組み上げて文や文章の大きな意味構造を構成するが、このための基本的な機能である単一化が下記のように用意されている。

[単一化の実行例]

```
入力>X={名前/DUALS, 版/3},
      Y={デモ/{会議/FGCS88, 場所/東京}},
```

```
X=Y.
```

(以下、単一化の実行結果)

```
出力>X={名前/DUALS, 版/3,
        デモ/{会議/FGCS88, 場所/東京}}
      Y={名前/DUALS, 版/3,
        デモ/{会議/FGCS88, 場所/東京}},
      yes
```

また、組み上げた部分項の内容を順次取り出して処理するための下記のような述語も用意されている。

[部分項述語の実行例]

```
入力>getRole ({名前/DUALS, 版/3}, A,B).
出力>A=名前,      B=DUALS;
      A=版,       B=3
      yes
```

また、非常によく用いられる部分項操作述語については、下記のような簡単な表現 (マクロ) を用意している。

[部分項 X の Y 属性の値 Z を取り出す述語 role (X, Y, Z) の略記法]

```
入力>{名前/DUALS, 版/Z}! 版=3.
出力>Z=3
      yes
```

4.3 意味記述言語 CIL の今後の課題

以上のように、部分項は有用な表現形式と操作述語を備えている。また、部分項のモデルである PTT 領域 (Partially Tagged Tree 領域) は、部分等式理論に関して、コンパクトかつ充足完全である。これに基づき、CIL の SLD 導出などの完全性や健全性が証明されている⁽¹⁶⁾。つまり CIL は、まさしく Prolog の拡張になっていることが理論的に確認されている。

現在は前記 CIL の三つの役割の統一的な枠組として、データの表現形式とその操作を統一して定義することを可能にする「型理論と型システム」の調査を行い、このうちの幾つかの型システムをツールとして提供している。今後は、制約言語としての機能充実を図る計画である。

5. 形態素意味解析

5.1 形態素解析システム LAX の概要

形態素解析プログラムは、マスタ辞書を変換して得られた形態素解析辞書を高速に検索しながら、逐次処理と並列処理双方に適したレイヤードストリーム型アルゴリズムにより、日本語の入力文字列を語の並びに変換する。このプログラムは、そのデバッグ環境や辞書の変換機能を具備した LAX システム⁽³⁾を用いて

開発される。

5.2 形態素意味知識の記述形式

一つの形態素は、図4の例のように、その書き表し方(表層)と、左方接続素性および右方接続素性と呼ばれる情報を持っている。左方接続素性は形態素の分類

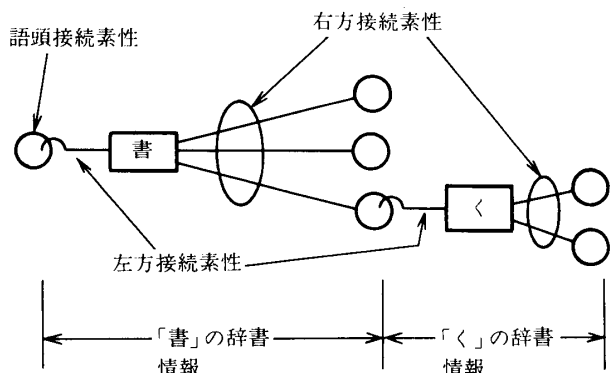


図4 形態素解析のしくみ

カテゴリであり、右方接続素性は、後ろにつながりうる形態素カテゴリの集合である。二つの形態素は、先行形態素の右方接続素性の中に、後方形態素の左方接続素性がただ一つある場合に接続できる。例では「書」と「く」は接続できる。そして、図4に示したようないわば形態素のチェインングを行った後で意味を構成する。

5.3 形態素意味知識の特徴

形態素意味知識としての文法⁽²⁾は、国語学の森岡文法⁽¹⁰⁾を参考に行っている。形態素意味解析例を図5に示す。例では形容詞「高い」、名詞「高さ」、他動詞「高める」、自動詞「高まる」の各々の形態と意味を、これらに共通した形態と意味を持つ語基「高」から構成している。この方法によれば、「高」という語基の意味を一つだけ記述しておけば、図中の破線で囲まれた派生規則を用いるだけで「高い」、「高さ」、「高める」、「高まる」の意味を構成でき、意味構成規則の開発効率が上がる。また、『何かが高まったなら、これは高い状態にある』という問題解決用推論規則も本方式では不要になり、真に問題解決に用いるべき推論規則の輪郭が明瞭になる。また、中心的な意味(例では『高』)を一箇所で記述するので、派生語(例では「高い」、「高さ」、「高める」、「高まる」)ごとに似たような意味を記述する従来の手法に比べて、意味記述における一貫性が向上する。

以上のような手法によりながら形態素解析処理を行い、この後にアルゴリズムが平易で効果の期待できる文節数最小法と、文節内形態素最小法からなるヒューリスティクスを利用して、複数の解析結果から最も

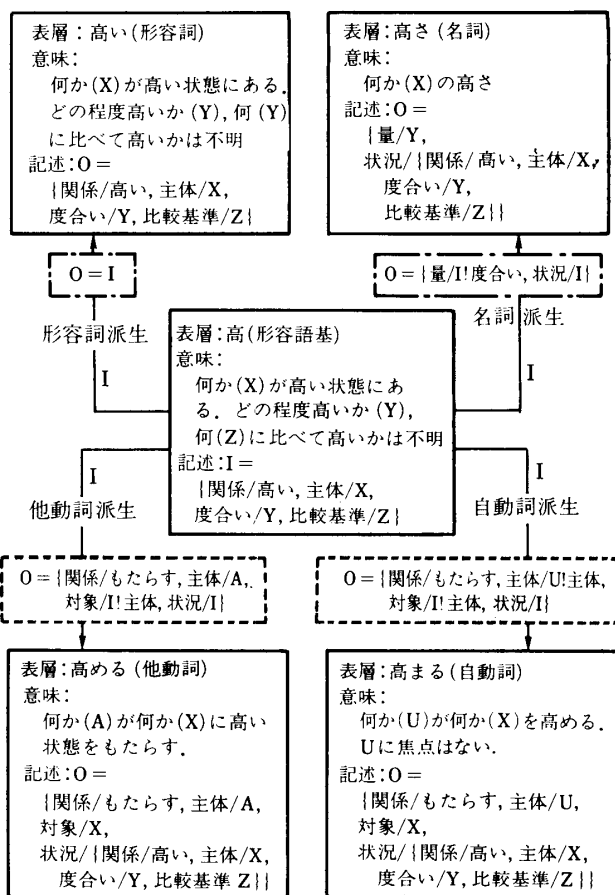


図5 語基による形態素意味解析例

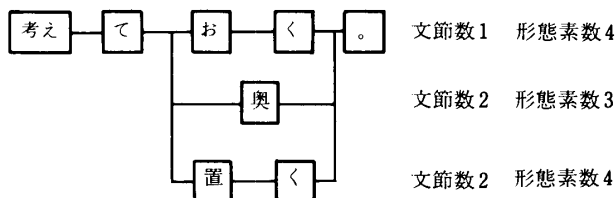


図6 最尤解釈選択例

もっともらしい(尤度の高い)解釈を選択する。例えば、図6の場合には最上段の解釈が優先される。候補が絞りきれず曖昧さが残る場合は例のようなグラフをそのまま構文解析へ送る。

従来は、形態素解析の曖昧さを構文解析へ伝搬するのは、それ専用のメカニズムを必要としたが、DUALS第3版の用いる汎用日本語処理系の形態素解析、構文解析は、双方がレイヤードストリームを基本とした解析を行っているので、このストリームで曖昧さを表現することにより、自然に両者をつなぐことができるようになってきている。

5.4 形態素解析の課題

形態素解析プログラムは、解析結果に曖昧さがある場合でも、30文字/1文あたりの処理時間が平均0.1秒以下と十分使用に耐える速度を実現している。しか

し、曖昧さの尤度を計算するメカニズムが現在は作り付けであるので、これを自由度のあるものに拡張する必要がある。また、形態素知識の記述形式の自由度を増す(2型も記述できるようにする)ことも検討することが必要である。

データに関しては、本方式による形態的な分類が有効であることが確認されたので、今後は辞書の語彙を60000語に増やして、形態的な知識の質、量双方のレベルアップを行う予定である。意味構成の部分は、その有効性がDUALS第3版により確認されているが、形態素が多数の意味を持つ場合の意味構成処理がまだ不十分であり、これらの知識の蓄積を図りながら意味を構成する部分の研究を進める必要がある。

6. 構文意味解析

6.1 構文解析システムSAXの概要

構文意味解析プログラムは、形態素解析結果(語の並び)を受け取り、語と語の係り受け関係を決定して意味構造を作る。このプログラムはSAXシステムを用いて開発され、前記LAXと同様にレイヤードストリーム型のアルゴリズム⁽¹⁴⁾により動作する。

6.2 構文意味知識の記述形式

SAXの文法記述形式は、DCGと呼ばれる図7のような木構造を表現するのに適した記述形式を拡張したものである。DCGの規則は、文法規則右側のボディと呼ばれる部分と左側のヘッドと呼ばれる部分からなり、ヘッドには文中における文法的な役割を示す分類記号を1個書き、ボディにはこれを構成する役割を示す分類記号を記述する。

DCGは、記号を1個だけヘッドに記述する2型と呼ばれる規則になっているが、SAXでは図8の(1)に示すように複数個の記号をヘッド部に記述できる0型文法へ記述形式が拡張されている。この拡張は、後述する係り受け文法を記述する際に必要である。また、一般に構文解析は複数の解釈を出すので、その中からもっともらしい解釈を選択する必要があり、そのもっともらしさ⁽⁹⁾を計算するための規則を{pref_rule}の部分に書くことができる。また、解析実行時に実行すべきプログラムを{extra}の部分に、解析終了後に実行してよいプログラムは{delayed_extra}に、それぞれ記述できる。

6.3 構文意味知識の特徴

DUALS第3版では、例文200文と一般質問文を

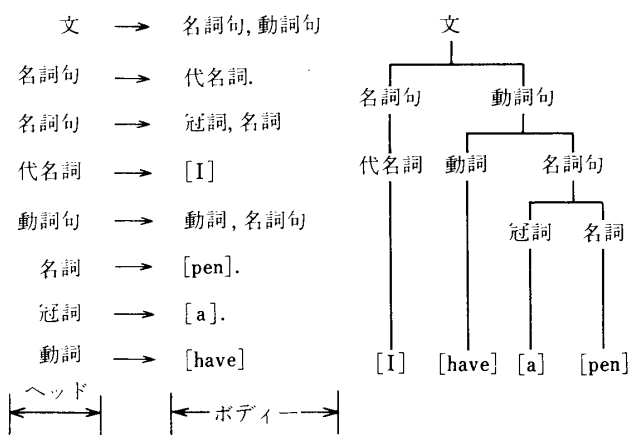


図7 DCGによる文法記述

- head₁, head₂, ..., head_m (1)
- body₁, {extra₁} :: {pref_rule₁}, (2)
- (3)
- body_n, {extra_n} :: {pref_rule_n}, (4)
- & {delayed_extra}, (5)
- && {pref_rule}. (6)

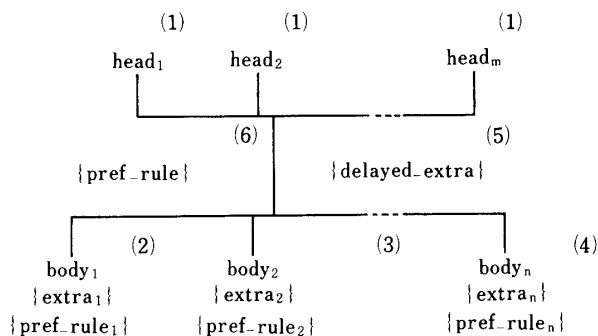


図8 SAX文法記述形式

構文意味解析する必要があり、早急に文法の原型を開発する必要があった。そこで文法開発者の直感に訴えやすく、仕組が単純な係り受け文法を、構文意味解析処理に用いている。

係り受け文法は、その解析結果がグラフになり、木構造を現すDCGなどの2型の規則では記述できない。そこで、SAXの文法記述形式を係り受け文法開発に先立って6.2で述べた形式に拡張し、この記述形式を用いて図9のように規則として係り受け文法⁽¹⁸⁾を記述できるようにした⁽²⁰⁾。

この規則は、隣接した係りと離れた係り双方を示す。図中上段(2)の語が(3)の語を修飾する場合は下段(1)のように書き換えを行う。修飾の可能性は係り決定述語modifyだけで調べられているので、文法開発者はこれだけを構文意味知識として記述すればよい。これにより、知識が分散せずに蓄積されるので開発効率が向上する。

6.4 曖昧さの処理

以上のような係り受け文法を用いて開発された構文

- 語 (Args1), 係り, 語 (Args2) (1)
- 語 (Args1), ((係り; []), 語-係り-列; []), (2)
- 語 (Args2), {modify (Args1, Args2)} (3)

係り受けが成り立つか
modify (Args1, Args2) で
調べる

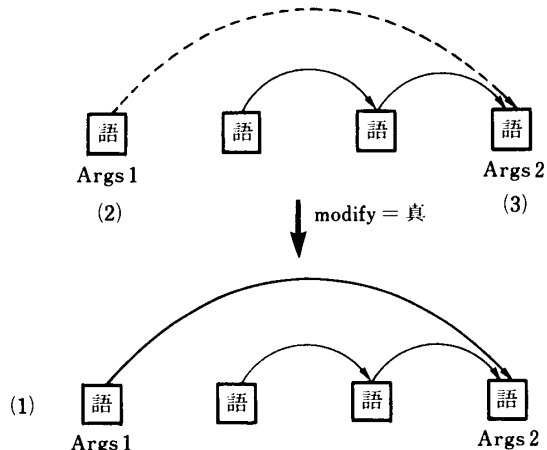


図 9 係り受けの検査

意味解析部は、形態素解析結果の語の並びを入力すると、係り受け決定述語 modify を用いながらその係り受けを決定する。次に、文中でかけ離れた語どうしの係り受けを持つ構造を解釈から除外するために、図 10 にあるような係り受け構造にもっともらしさを示す点数 (尤度) を付け、この点数の高い構造を選択する。点数に差がない場合には、これに対応する図 11 のような曖昧な意味どうしの関係に注目し、談話の情報と制約論理プログラミングの手法を用いて解釈をしぼる⁽¹⁸⁾。例えば図 11 において、「花子は誰かに約束をしてもらっていた」ことが談話情報としてわかっていたとすると、「B=花子」を含む解釈(4)と(5)が選択され、構文意味解析結果として問題解決へ出力される⁽⁴⁾⁽¹³⁾。

問題解決は、この曖昧さを順次制約を蓄積しながら決定することができる処理系になっている。従来は曖昧さを残したまま意味処理を行うことができなかったので、解析処理は曖昧さが出るとその場でこれを文の本来の意味とは関係の薄い種々の方法 (ヒューリスティクス) を用いて絞り込もうとしていた。しかし、制約伝搬を用いることが可能になったため、解析部は従来のように無理に解釈を絞る必要がなくなった。

6.5 構文意味解析の今後の課題

まず、処理系については、0型文法を記述できるようになり、文法規則がより自由に記述できるようになった。しかし、0型文法用のプログラム開発環境がまだ不十分であり、今後これを充実させる必要がある。

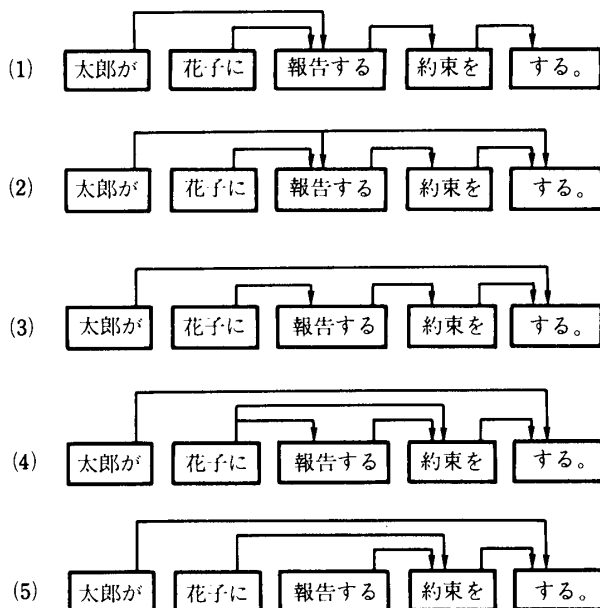


図 10 複数の係り受け構造

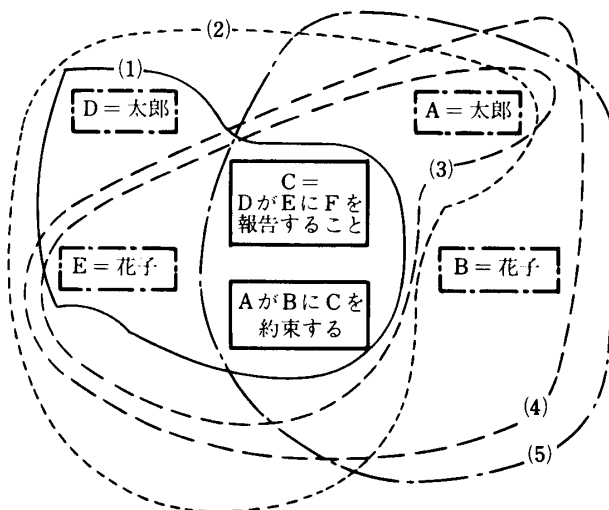


図 11 係り受けから得られる文の意味

また、解析過程などを表示するユーザインタフェース部分にも具備すべき機能が多く残っているので、これらは順次追加していく計画である。

解析用の知識に関しては、対象テキストを順次増やして、問題解決や談話理解のメカニズムで必要となる意味構造が出力できているか否かを検討しながら、一般性のある文法規則を蓄積する必要がある。

なお、SAX の並列版 PAX が並列推論マシン Multi-PSI で動作しており、今後解析システムを順次並列推論マシンで動作させていく予定である。

7. 問題解決

7.1 問題解決の概要

問題解決部は、JPSG⁽¹²⁾などの研究から生れた依

存伝搬処理プログラム⁽⁶⁾を用い、指示代名詞などの指し示す対象を同定する照応処理や問題解決などに用いる制約の統一的な処理を目指す。ここに制約とは、何らかの対象が持ち得る構造に関する情報である。そして、談話構造に対する制約処理が談話処理であるとなし、談話における意味論的、語用論的な制約の多くを同一記法で記述しているため、照応処理や問題解決専用のプログラムは必要としない⁽⁷⁾。

7.2 問題解決知識の記述形式

詳細な技術内容は文献(6)にゆずり、ここでは、敬語によって表現される人間関係、いわゆる待遇関係を例に説明を行う⁽¹⁷⁾。

ある状況において、以下の会話があったとしよう。

状況1：話者=太郎、聞き手=花子

発話：「二郎様が来られたと知子が言っていました」

発話者と聞き手の待遇関係は、発話の言い切りの形に反映されるという制約が日本語にあるので、「言いました」という太郎の発話から、太郎より花子は目上であることがわかる。この情報を得るには、形態素解析で「ました」という形態素が待遇関係を示すものであることを解析し、その結果を構文解析へ出力し、構文解析が、「言いました」が文末の言い切りであることを解析し、この結果を問題解決へ出力することにより得られる。

発話者と話中の人物との待遇関係は、その人物を指し示す言葉づかいに反映されるという制約があるので、「知子が」という太郎の発話より、太郎は知子と同等もしくは上位であることがわかる。この情報も、前記の手順で得られる。

次に、「二郎様が来られた」が、知子が直接発話した言葉(直接話法)なのか、太郎の言葉(間接話法)なのかで得られる待遇関係が異なってくる。直接話法と間接話法の可能性があることは構文的な曖昧さとして構文解析において解析され、構文解析は二つの解釈を問題解決へ出力する。問題解決では、知子の発話であるとすると、知子よりも二郎が目上であることがわかるが、太郎と二郎の関係はわからない。太郎の発話だとすると、太郎より二郎が目上であることがわかるが、知子と二郎の関係はわからない。そこで、待遇関係を決定せず曖昧さを残したまま次の会話を処理する。

状況1：話者=太郎、聞き手=花子

発話：「二郎のやつ来たのですね。」

この発話と前の発話から、図12のように、太郎よりも二郎が目下であることがわかると共に、知子より

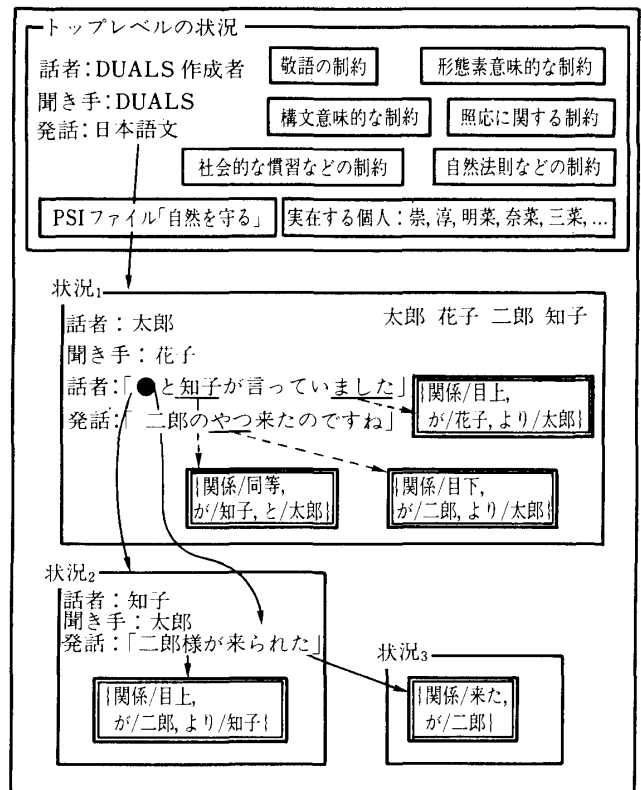


図12 制約による待遇関係の決定

も二郎が目上であることがわかる。

制約伝搬とは、この例のように、ある構造に関する制約(敬語に関する制約)を用いて、その構造(待遇関係)を順次決定するようなメカニズムである。技術的には、どのような構造を用いて制約を示すかが重要な研究テーマとなっており、ブール代数などについても近年多くの成果が得られつつある。

従来はこのように順次制約を蓄積しながら構造を決定することができなかったが、一般に文は前章の図11にも示したように多くの意味を持っているので、これらを重なり合わせて処理できない限り、文や文章の本当の意味を得ることはできないのである。制約伝搬はこれを可能にしており、談話理解を目指すDUALS第3版にとって必須の技術となっている。

制約伝搬をDUALS第3版で用いる場合には、処理すべき制約が複数与えられているので、問題解決機構は制約に重みを付けて、その適応順序を制御できるようになっている。

7.3 問題解決機構の特徴

DUALS第3版の問題解決は、文の意味は制約であるとする状況理論に従って、談話構造を記述する。すなわち文の意味を、求めるべき談話構造に関する制約として制約伝搬により順次蓄積し⁽⁴⁾⁽¹³⁾、談話構造を決定していく。

この談話構造は、発話そのものや会話の参加者のモデルも包含する⁽¹⁷⁾。また、発話を人間の行う行為ととらえ、この発話行為などに関する制約も、談話構造に対する制約として、統語的制約や意味的制約と同様に扱う。

平叙文はその意味が、事象の集まりである現実の状況(状況1と示す)において成り立つという制約を生成する。

質問文は質問の答えにあたる発話を行う義務を聞き手が負っているという制約を生成する。すなわち、質問を受けた談話処理システムは、それに答えねばならないという制約を処理することになり、質問応答が生ずる。

例えば、「人間が地球で生きつづけるには自然が必要だ。」という平叙文に対応して図13の構造が作られる。「人間はどこで生きているか」という質問が入ると、人間が生きている事象が状況1にあるか探索し、何かXという行為を「し続ける」状況は、何かXを「している」状況があることを前提としているという一般的な制約を用いて人間が地球で生きている事象を状況1に生成し、図中先頭の構造を文生成部へ出力する。

この例のように、込み入った談話処理も制約伝搬によればすべて制約の処理として記述することが可能になる。また状況理論は、これら制約をすべて理論的に包含する強力な枠組となっているので、制約伝搬と状況理論とを用いることにより、理論的にも工学的にも

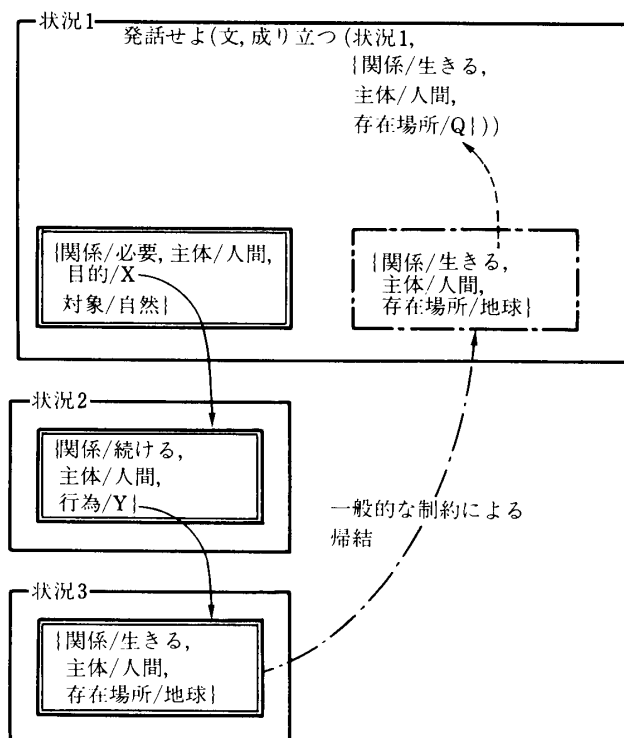


図13 問題解決例

談話処理への見通しが従来に比べて良くなっている。

7.4 問題解決の今後の問題

制約伝搬プログラムは、談話構造についての制約を順次蓄積しながらその構造を決定することを可能にした。これらの制約は、現在 Prolog プログラムの形式(ホーンクローズ)である必要があるので、より自由に制約が記述できるように、つまり Prolog のプログラム形式を包含する一階述語が記述できるようにする予定である。また、談話に関する制約の読み込み手順も現在のトップダウンの読み込み以外も可能にして、柔軟性を確保する予定である。

談話構造に対する制約の蓄積については、二つのアプローチの併用を考えている。一つは、現在の DUALS の枠組を利用して順次知識の蓄積を行うものである。このアプローチでは、従来の種々の自然言語処理系の談話処理における限界を明確にすると共に、制約生成系としてどのような機能を取り込めばよいかを研究することができる。もう一つは、解析生成を含むすべての知識をいったん、本問題解決機構の記述形式に置き換え、談話理解メカニズムをそのまま制約処理として実現するものである。

談話の種々の知識の蓄積には、知識の記述に対する自由度が高く見通しも良い後者のアプローチが適している。後者のアプローチによって談話処理に関する知識の蓄積を行い知識の質を実験を通じて検証する。そして、蓄積された知識を、前者のアプローチにより順次 DUALS へ組み込み、処理速度など含めた総合的な評価を行う。最終的に評価が終わった段階で汎用日本語処理系 LTB へこれらの知識を移植する計画である。

8. 文生成

8.1 文生成の概要

文生成部は、問題解決部から送られた談話構造を入力し、解答文を生成する。談話構造には、表層文の構造に直接対応するものや、文要素の省略・代名詞化の手がかりとして間接的に文構造を決定するものがある。このような構造から表層文を生成するために、文生成は図14に示す処理を行う。

- (1) 談話構造中で、解答の生成に必要な生成対象を探す。
- (2) 表層文の文型を決定し、表層文中の単語の選択などを行う。結果を「中間表現」なる、意味表現と表層表現との中間的な表現として出力する。

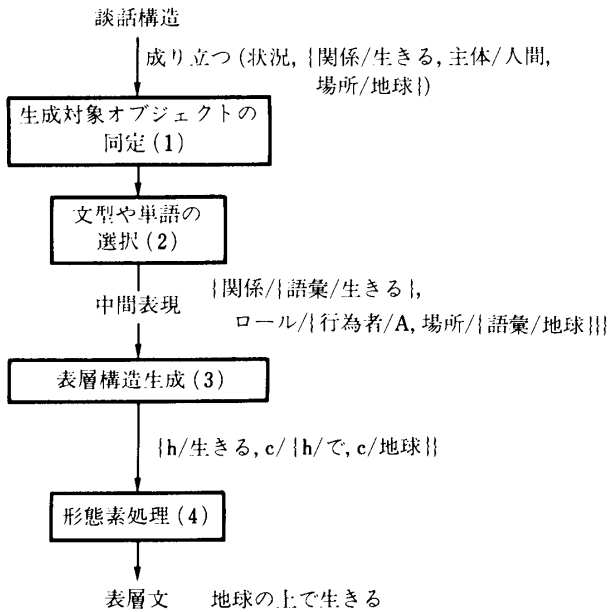


図 14 文生成部の処理

(3) 中間表現の情報を詳細化し、表層に近い表現を生成する。

(4) 単語の活用処理を行い、表層文を生成する。

(1)は、問題解決部の推論の結果得られる談話構造をもとに問題解決部が求める。(1), (2)が DUALS 専用の機能でありプランニングと呼ばれる。文脈情報を用いない処理である(3), (4)は LTB 文生成部⁽⁸⁾の機能を流用して行っている。

8.2 プランニング

文型の決定・単語の選択の際には、発話に関する制約を用いる。例えば、下記の制約は、談話構造と発話文に関するものである。

(1) 質問と似た文型で答える。今までの質問応答で指示されたオブジェクトを再び指示するときには、以前に用いた単語を用いたほうがよい。

【例】 問：何に頼る必要があるか？

答：自然に頼る必要がある。O

答：自然に頼らなければならない。X

(2) 直前に発話された文要素は、省略するほうが自然である。話しの焦点にあたり、省略できない場合は、代名詞で置き換えたほうがよい。

【例】 問：人間は何に頼る必要があるか？

答：自然に頼る必要がある。O

答：人間は自然に頼る必要がある。X

DUALS 第3版では、発話の履歴を談話構造の中に制約の形で取り込んでおり、これを生成の文型・単語の決定に用いている。生成部内では、上記のようなヒューリスティクスを CIL の部分項を用いて記述しており、それを文型・単語の決定に利用している。

8.3 表層文生成

LTB 文生成では、まず入力中間表現に明示的に表されていない情報を明示的に付加していくとともに表層文の構造を生成する。このとき以下のような処理を行う。

(a) 深層格から表層格への変換

(b) 格要素などの、表層文における出現順序の決定

(c) 受動・使役などの態変化が起きる場合の格変換

(d) 単語の派生処理など

これらの処理は、CIL の PST の展開規則を実行することにより行われる。また、このとき単語の持つ性質などの情報は、辞書を参照して得る。

次に、表層文の構造に含まれる単語の形態素処理を行い、表層文を生成する。その処理は、形態素解析時に用いている形態素文法をそのまま用いて行われる。

8.4 文生成の今後の課題

文生成は、その中核となる処理メカニズムがまだ明確になっていない。この部分を例えば問題解決に用いた制約伝搬機構を用いたりしながら、談話処理との関連を順次解き明かす必要がある。

また、言語データベースの章でも述べたが、生成が独自に持つ知識があるので、それらを順次マスタ辞書へフィードバックする予定である。

9. おわりに

1988年11月末の国際会議 FGCS '88において、我々は DUALS 第3版のデモンストレーションを行った。この国際会議は ICOT 中期のしめくりにあたり、自然言語処理研究開発グループにとって、DUALS 第3版は、ICOT 後期の自然言語処理技術を考える上で重要な位置を占める。

DUALS 第3版で制約伝搬処理系が組み込まれたことにより、談話理解に必要なすべての処理系はそろった。また、各処理系は言語データベースを用いて、下位レベルで出力すべき情報は、そのレベルで十分に出力できるようになっており、この結果、上位レベルで真に必要な談話処理用の知識と下位レベルの知識の混在が避けられ、談話処理用知識の輪郭を明確にできる見通しが立った。

複雑に絡み合った談話処理のメカニズムを解明し、その知識を蓄積してゆくには、まず、記述自由度の高い制約プログラミングと状況理論の枠組を用いることが有効であることが DUALS 第3版により確認でき

た。そこで、今後は問題解決に用いる依存伝搬技術の確立を進めながら、DUALS第3版では少数の制約の実装にとどまっている照応処理などの談話処理技術をより発展させる必要がある。そして、これにより得られた知識を、順次評価を行いながら汎用日本語処理系へ移植し、計算機との日本語インタフェースとして実際に用いて磨きをかけていく計画である。

また、知識の増大に伴い、計算機パワーが不足することが予測されるので、汎用日本語処理系の各処理系を順次、並列推論マシンPIMの上へ移植し、総合的な自然言語処理能力の向上を図りたい。

後期3年間も、対話をより柔軟にこなすDUALSを目指し努力を続けたい。

謝 辞

いろいろ、ご指導、ご意見をいただく ICOT 所長 淵一博氏、研究所次長古川康一氏、第二研究室室長内田俊一氏、第二研究室室長代理吉岡勉氏に感謝する。いろいろ、ご助言をいただく、自然言語ワーキンググループ主査東京工業大学教授田中穂積氏、ワーキンググループ委員京都大学大型計算機センター助教授松本裕治氏、KDD 上福岡研究所主査瀧塚孝志氏に感謝する。また、日本情報科学研究所弘田直人氏をはじめとする諸氏に感謝する。最後に、本論文執筆にあたって協力をいただいた、ICOT 研究員、久保幸弘氏、佐野洋氏、幡野浩司氏、赤坂宏二氏、奥西稔幸氏に感謝する。

◇ 参 考 文 献 ◇

- (1) 奥西稔幸, 向井国昭: CIL における言語情報の構造と操作のモデル, IPSJ 第 37 回全国大会, pp. 1076-1077 (1988. 9).
- (2) 佐野 洋, 赤坂宏二, 久保幸弘, 杉村領一: 語構成に基づく形態素解析, IPSJ 全国大会 (1988).
- (3) 杉村領一, 赤坂宏二, 久保幸弘, 松本裕治: 論理型形態素解析 LAX, LPC '88, pp. 213-222 (1988. 4).
- (4) 杉村領一: 論理型文法における制約解析, JSAI 全国大会第 2 回論文集, pp. 427-430 (1988. 7).
- (5) 瀧塚孝志, 杉村領一: LTB シェルの構成, IPSJ 第 37 回全国大会, pp. 1074-1075 (1988. 9).
- (6) 橋田浩一, 白井英俊: 条件単一化, コンピュータソフトウェア, Vol. 3, pp. 28-38 (1986).
- (7) 橋田浩一: AI とは何でないか—情報の部分性について—, bit, Vol. 20, No. 8 (1988. 8).
- (8) 幡野浩司, 福島秀顕, 小島 量, 池田光生, 重永信一: LTB 文生成部の構成, IPSJ 第 37 回全国大会, pp. 1082-1083 (1988. 9).
- (9) 松本裕治, 杉村領一: 構文解析システム SAX のための文法記述言語, JSSST 第 5 回大会, pp. 77-80 (1988. 9).
- (10) 森岡健二: 語の構成, 現代語研究, Vol.1, 明治書院(1987).
- (11) Barwise, J. and Etchemendy, J.: The Liar; An Essay on Truth and Circular Propositions, MIT Press (1987).
- (12) Gunji, T.: Japanese Phrase Structure Grammar, Dordrecht D. Reidel (1987).
- (13) Hasida, K.: Dependency Propagation; A Unified Theory of Sentence Comprehension and Generation, Proc. of the 10th IJCAI, pp. 664-670 (1987).
- (14) Matsumoto, Y. and Sugimura, R.: A parsing system based on Logic Programming, Proc. of IJCAI 87 (1987).
- (15) Mukai, K.: Unification over Complex Indeterminates in Prolog, ICOT, Technical Report, No. 113 (1985).
- (16) Mukai, K.: Partially Specified Term in Logic Programming for Linguistic Analysis, in Proc. of FGCS '88, ICOT (1988).
- (17) Sugimura, R.: Japanese Honorifics and Situation Semantics, Proc. of the 11 th Int. Conf. on Computational Linguistics COLING '86, pp. 507-510 (1986).
- (18) Sugimura, R., Miyoshi, H. and Mukai, K.: Constraint Analysis on Japanese Modifying Relations, Natural Language Understanding and Logic Programming II, North-Holland, pp. 93-106 (1987).
- (19) Sugimura, R., Hasida, K., Akasaka, K., Kubo, Y., Hatano, K., Okunishi, T. and Takizuka, T.: A Software Environment for Research into Discourse Understanding Systems, FGCS '88, ICOT, pp. 285-295 (1988).
- (20) Sugimura, R.: Logical Dependency Grammar and its Constraint Analysis, ICOT, TM-No. 679 (1989).
- (21) Tanaka, Y. and Yoshioka, T.: Overview of the Development of Dictionary and Lexical Database, FGCS '88, ICOT (1988).

著 者 紹 介



杉村 領一 (正会員)

1980年京都大学工学部情報工学科卒業。同年、松下電器産業(株)入社。1985年より(財)新世代コンピュータ技術開発機構に転出。1989年4月より、松下電器産業(株)情報システム研究所勤務。この間、かな漢字変換、並列形態素/構文解析、談話理解実験システムなどの研究開発に従事。情報処理学会、日本ソフトウェア科学会、ACM、IEEE各会員。



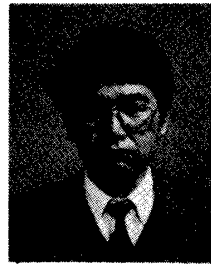
田中 裕一 (正会員)

1979年東京大学教養学部基礎科学科卒業。1984年富士通(株)入社。現在、(財)新世代コンピュータ技術開発機構に転出中。自然言語処理に関する研究開発に従事。情報処理学会、ACL各会員。



橋田 浩一

1981年東京大学理学部情報科学科卒業。1986年同大学院理学系研究科博士課程修了。理学博士。同年、電子技術総合研究所入所。パターン情報部推論システム研究室勤務。1988年(財)新世代コンピュータ技術開発機構に出向。認知言語学の研究に従事。人工知能、絵画、料理などに興味を持つ。AIUEO、認知科学会、情報処理学会、ソフトウェア科学会、ACL各会員。



向井 国昭

1971年東京大学理学部数学科卒業。同年、三菱電機(株)入社。計算機言語処理、自然言語処理などに従事。1982年(財)新世代コンピュータ技術開発機構に出向。自然言語処理に従事。情報処理学会、ソフトウェア科学会各会員。