複数の形態の入力に対応した選言的素性構造による辞書記述法

中野幹生[†] 島津明[†,☆]

本論文では，単一化文法による構文解析用の辞書で，ローマ字列，漢字かな混じり列，音素列，音声認識や形態素解析システムの出力などの様々な形態の入力に対応したものを記述する方法を提案する．複数の形態の入力に対応した辞書の，最も単純な構築法として，各入力形態の終端記号の組を作り，それに辞書記述を割り当てる方法が考えられる．しかし，各入力形態の終端記号どうしは1対1対応しない．そのため，ある入力形態のある終端記号の辞書項目が1つの選言的素性構造でコンパクトに記述できる場合でも，他の入力形態では複数の異なる終端記号が用いられるときには，複数の語彙項目を作らなくてはならない．これは，記述の冗長性のみならず，解析時の効率の低下を招く．この問題を解決するため，選言的素性構造の中に素性値として各入力形態の終端記号を記述することにより，1つの辞書記述で複数の入力形態に対応する方法を提案する．終端記号を記述する素性の値は選言でもよいので，終端記号が1対1に対応しなくても，1つの選言的素性構造の中で記述することができる．構文解析の際には，入力された終端記号に対し，単一化操作により辞書記述から適切な語彙素性構造を生成する．

Lexical Descriptions with Disjunctive Feature Structures for Dealing with Various Forms of Input

MIKIO NAKANO[†] and AKIRA SHIMAZU[†,☆]

This paper presents a feature-based lexical description method for dealing with various forms of input such as sequences of Roman letters, sequences of kana-kanji characters, sequences of phonemes, and outputs of speech recognizers and morphological analyzers. A simple method is to describe pairs of a lexical feature structure and a set of terminal symbols in each input form. However, there can be no one-to-one correspondence between terminal symbols in two different input forms. Therefore, even if one lexical entry with a disjunctive feature structure is sufficient for one terminal symbol in one input form, multiple lexical entries are required if multiple terminal symbols are used in another input form. This causes not only redundancy in the lexical descriptions but also inefficiency in the parsing process. This paper proposes describing terminal symbols as feature values in the disjunctive feature structures. Since those terminal symbols can be disjunctive values, even if there is no one-to-one correspondence between terminal symbols, they can be described in one disjunctive feature structure. The lexical feature structure for an input word is obtained by unification.

1. はじめに

自然言語の構文意味解析システムは，どのようなアプリケーションに組み込まれるかに応じて，様々な形態の入力を解析する必要がある．たとえば，テキストの解析を行うシステムでは，ひらがな列，漢字かな混じり列，ローマ字列などを解析することがありうる．また，形態素解析システムが出力する品詞情報付きの単語列や，音声認識システムが出力する音素列を解析することもある．

入力形態のバリエーションに応じて別々の構文意味解析システムを作るのは労力がいるため，複数の入力形態に対応した構文解析システムを作成することを考える．本論文では特に単一化文法を用いる．単一化に基づくアプローチでは，文法を宣言的に記述することが可能であり，したがって，文法の開発や修正が容易であるといった利点を持っているからである[1)]．

入力形態にかかわらず，解析のプログラムや句構造規則は共通に用いることができる．入力形態に応じて終端記号（単語）が異なるので，辞書だけを入力形態ごとに作る方法が考えられる．しかし，音素列「aruku」と漢字かな混じり列の「歩く」のように，異なる入力形態の終端記号が同じ語彙素性構造を持つので，記述

† NTT 基礎研究所
NTT Basic Research Laboratories
☆ 現在，北陸先端科学技術大学院大学
Presently with Japan Advanced Institute of Science and Technology, Hokuriku

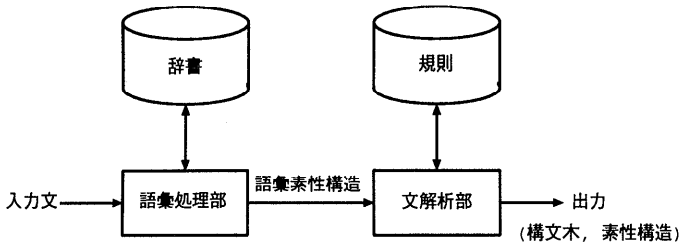


図 1 単一化に基づく構文意味解析システム
Fig. 1 Unification-based parsing system.

が冗長になってしまう．そこで，1つの辞書記述で複数の入力形態に対応する方法が望まれる．

最も単純な方法として，各入力形態の終端記号の組を作り，それに辞書記述を割り当てる方法が考えられる．たとえば，⟨aruku，歩く⟩に対して，語彙素性構造を与える．

この方法では，選言の値を持つ素性構造（選言的素性構造）を用いた場合に問題が生じる．選言を用いると，辞書記述中の冗長性が減り[5),8)]，構文解析の効率が向上するため[2),3),6),10),11),19)]，できるかぎり選言を用いて辞書記述を行うことが望まれる．一例として，「はかる（図る，計る，測る，量る）」のようにヲ格の名詞の意味範疇に応じて意味と漢字が変わる動詞を考える．音素列入力のみに対応した辞書では，下位範疇化素性（格フレーム）のヲ格の意味素性とこの動詞の意味とを選言で記述すれば，辞書記述もコンパクトになるうえ，構文解析も冗長な処理を行わなくてすむので高速になる．しかし，漢字かな混じり列に対応するためには，別々の辞書記述が必要となる．したがって，入力形態の終端記号の組は，⟨hakaru，図る⟩，⟨hakaru，計る⟩，⟨hakaru，測る⟩，⟨hakaru，量る⟩の4つ必要になる．結果として，音素列「hakaru」は語彙項目を4つ持つことになるため，選言による構文解析の効率化が得られない．フレーム構造による辞書記述において，選言で表現された動詞の格フレームを入力形態に応じて変化させる研究[18)]はあるものの，選言的素性構造による辞書記述に関しては，この問題は解決されていない．

本論文では，選言的素性構造の中に素性値として各入力形態の終端記号を記述することにより，1つの辞書記述で複数の入力形態に対応する方法を提案する．本方法では，辞書は，選言的素性構造で表された語彙項目の集合と，終端記号と語彙項目の対応表からなる．この表を用いて，入力された終端記号に対応する語彙項目を得て，単一化操作により適切な語彙素性構造を生成する．

本論文の構成は以下のとおりである．2章では，複数の入力形態に対応した辞書記述を行う際の問題点を述べる．3章では，本論文で提案する，複数の入力形態に対応した選言的素性構造による辞書記述法について述べる．4章では，本方法の実現について述べる．

2. 辞書記述を複数の入力形態に対応させる際の問題点

図1のような単一化ベースの構文意味解析システムを考える．入力文の各単語に対し，語彙処理部が辞書から語彙素性構造を取り出し，文解析部に送る．文解析部は規則を用いて構文意味解析を行い，構文木を作って文全体の素性構造を得る．

本論文では，辞書と規則を合わせて文法と呼ぶことにする．辞書は終端記号と語彙素性構造の対の集合である．語彙素性構造は，終端記号の構文意味情報を素性構造の形で表現したものである．素性構造は素性名と値のペアの集合である．以下に動詞「買う」の語彙素性構造の例を示す．

$$\begin{bmatrix} \text{POS} & v \\ \text{SUBCAT} & \left\{ \begin{bmatrix} \text{POS} & p \\ \text{CASE} & ga \\ \text{SORT} & human \\ \text{SEM} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{POS} & p \\ \text{CASE} & wo \\ \text{SORT} & thing \\ \text{SEM} & \boxed{2} \end{bmatrix} \right\} \\ \text{SEM} & \begin{bmatrix} \text{ACTION} & buy \\ \text{AGENT} & \boxed{1} \\ \text{OBJECT} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

たとえば，POS（part of speech，品詞）が素性名で v（verb，動詞）が値である☆．値は，シンボルだけで

☆ v のほかに，n（noun，名詞），p（particle，助詞）などが用いられる．

表 1 従来の辞書記述例
Table 1 Example of previous lexical descriptions.

音素列	ひらがな列	漢字かな混じり列	語彙素性構造
kau	かう	買う	[POS *v*, SUBCAT {[POS *p*, CASE *ga*, SORT *human*, SEM [1]], [POS *p*, CASE *wo*, SORT *thing*, SEM [2]]}, SEM [ACTION *buy*, AGENT [1], OBJECT [2]]]
kau	かう	飼う	[POS *v*, SUBCAT {[POS *p*, CASE *ga*, SORT *human*, SEM [1]], [POS *p*, CASE *wo*, SORT *animal*, SEM [2]]}, SEM [ACTION *raise*, AGENT [1], OBJECT [2]]]
kiqpu	きっぷ	切符	[POS *n*, SORT *thing*, SEM *ticket*]

はなく，素性構造や素性構造の集合である場合もある．SUBCAT（subcategorization，下位範疇化）素性は，その動詞に何格のどのような句がかかるかを表すものであり，その値は素性構造の集合である．CASE 素性は助詞の格を，SORT 素性は意味カテゴリを表す．SEM 素性値は意味表現であり，ACTION 素性が動作を，AGENT 素性が動作の行為者を，OBJECT 素性が動作の対象を表す．[1]，[2] などは変数を表す．この素性構造を用いた構文意味解析中に，[1] が *taro* に，[2] が *ticket* になったとすると，SEM 素性値の

[ACTION *buy*, AGENT [1], OBJECT [2]]

は，

[ACTION *buy*, AGENT *taro*, OBJECT *ticket*]

になる．これは，*taro* が行為者で *ticket* が対象物である *buy* という行為を表している．単語列が入力されると，語彙処理部は各終端記号の語彙素性構造を辞書から得て文解析部に送る．文解析部は構文意味解析を行い，文全体の語彙素性構造を得る．

このような構文意味解析システムの辞書を，音素列，ひらがな列，漢字かな混じり列など，n 種類の入力形態の入力に対応させるためには，終端記号と語彙素性構造の 2 つ組を，n 種類の入力形態の終端記号と語彙素性構造を対応付ける $n+1$ 個組に変更すればよい．具体例を**表 1** に示す．音素列中の q は促音の音素で

ある．上記の音素列，ひらがな列，漢字かな混じり列などの入力形態は，1対1対応しない．たとえば，漢字かな混じり列で「飼う」と表される動詞も，ひらがなでは「かう」，音素列では「kau」である．「飼う」と「買う」はSUBCAT素性が異なるので，語彙項目を2つ用いて記述する．

このとき，もし，ひらがな列の入力に「かう」があれば，「かう」の語彙素性構造が2つあるとして構文意味解析が行われる．「買う」も「飼う」も，動詞としての性質と，SUBCAT素性のうちガ格に関する部分は同じなので，入力が「わたしがかう」のようにヲ格がない文のときには，ほとんど同じ処理が2度行われることになり，効率が悪い．「買う」の構文意味情報と「飼う」の構文意味情報のうち，共通部分を括り出して1つの辞書記述にしておき，共通でない部分は，必要になったときにだけどの選択肢が使えるか調べるようにすれば，解析時間が増加しなくてすむ．たとえば，構文意味情報を，次のように記述すればよい．

$$
\left[\begin{array}{ll}
\text{POS} & v \\
\text{SUBCAT} & \left\{\begin{array}{l}
\left[\begin{array}{ll}
\text{POS} & p \\
\text{CASE} & ga \\
\text{SORT} & human \\
\text{SEM} & \boxed{3}
\end{array}\right] \\
\left[\begin{array}{ll}
\text{POS} & p \\
\text{CASE} & wo \\
\text{SORT} & \boxed{1} \\
\text{SEM} & \boxed{4}
\end{array}\right]
\end{array}\right\} \\
\text{SEM} & \left[\begin{array}{ll}
\text{ACTION} & \boxed{2} \\
\text{AGENT} & \boxed{3} \\
\text{OBJECT} & \boxed{4}
\end{array}\right]
\end{array}\right]
$$

$$
(\boxed{1}, \boxed{2}) = (thing, buy) \vee (animal, raise)
$$

この構造で，$(\boxed{1}, \boxed{2}) = (thing, buy) \vee (animal, raise)$ は，$\boxed{1}$ と $\boxed{2}$ の値が，*thing* と *buy* の組合せか，*animal* と *raise* の組合せのどちらかでなくてはならないことを意味している．このように選言の値を持つ素性構造を選言的素性構造[5),8)]と呼ぶ．選言的素性構造どうしの単一化を，できるだけ選言を展開せずに行うメカニズムを用いることにより，構文解析の効率が向上する[2),3),6),10),11),19)]．

しかしながら，「kau」や「かう」に対しては，上記の選言的素性構造を用い，「買う」，「飼う」に対しては，表1の素性構造を用いようとすると，辞書記述のセットを入力形態の数だけ用意する必要がある．この場合，辞書記述の各セットには共通部分が多いにもかかわらず，別々の記述を行わなくてはならないので，辞書の記述や修正に要する労力が多くなる．そこで，様々な入力形態の入力に対応し，かつ，入力形態の増加によって解析時間が増加しない辞書記述法が望まれる．

なお，表1の辞書記述から，素性構造のgeneralization の手法[9)]を用いて，選言を含む語彙素性構造を自動的に得ることが可能であると考えられる．しかしながら，表1の辞書記述は共通部分が多く冗長であるため，記述に労力が必要となるという問題がある．

3. 選言的素性記述による複数の入力形態への対応

上記の問題への解決法として，本論文では，選言的素性構造の中に素性値として各入力形態の終端記号を記述することにより，1つの辞書記述で複数の入力形態に対応する方法を提案する．HPSG[16)]やJPSG[4)]などの言語理論では，素性構造の中に音素列や漢字かな混じり列などの情報を含めているが，本方法はそれを選言的素性構造による辞書記述に応用するものである．

本方法では，辞書は，選言的素性構造で表された語彙項目の集合と，終端記号と語彙項目の対応表からなる．語彙項目は，入力形態 $M_1 \ldots M_n$ を素性として持ち，その値を各入力形態の終端記号とする．終端記号と語彙項目の対応表は，3つ組 $\langle w, M, Item \rangle$ の集合で，各 $\langle w, M, Item \rangle$ は，語彙項目 *Item* が入力形態 M の終端記号 w を素性値として持つことを示す．入力形態 M の語 w が入力されたとすると，語彙処理部は，終端記号と語彙項目の対応表の w の列から語彙項目を取り出す．次にこの語彙項目と

$$
\left[\begin{array}{ll} M & w \end{array}\right]
$$

を単一化することにより語彙素性構造を得る．この語彙素性構造が文解析部に送られ，文法を用いて構文意味解析され，入力文の構文木と，文全体の素性構造が出力される．

例を用いて説明する．まず，語彙項目の例を**表2**に示す．ここで，PHON，KANA，KANJIは，音素列，ひらがな列，漢字かな混じり列の入力形態を表す素性である．次に，終端記号と語彙項目の対応表の例を**表3**に示す．

漢字かな混じり列「買う」に対しては，語彙処理部は，1番の語彙項目と

$$
\left[\begin{array}{ll} \text{KANJI} & \text{買う} \end{array}\right]
$$

を単一化することにより，次の語彙素性構造を得る．

表 2 辞書記述の例
Table 2 Example lexical descriptions.

番号	語彙項目
1	PHON kau KANA かう KANJI [0] POS *v* SUBCAT { [POS *p*, CASE *ga*, SORT *human*, SEM [3]], [POS *p*, CASE *wo*, SORT [1], SEM [4]] } SEM [ACTION [2], AGENT [3], OBJECT [4]] ([0], [1], [2]) = (買う, *thing*, *buy*) ∨ (飼う, *animal*, *raise*)
2	PHON kiqpu KANA きっぷ KANJI 切符 POS *n* SORT *thing* SEM *ticket*
3	PHON o KANA を KANJI を POS *p* CASE *wo*

表 3 終端記号と語彙項目の対応表の例
Table 3 Correspondences between terminal symbols and lexical items.

終端記号	入力形態の種類	語彙項目番号
kau	PHON	1
かう	KANA	1
飼う	KANJI	1
買う	KANJI	1
kiqpu	PHON	2
きっぷ	KANA	2
切符	KANJI	2
o	PHON	3
を	KANA	3
を	KANJI	3

PHON kau
KANA かう
KANJI 買う
POS *v*
SUBCAT { [POS *p*, CASE *ga*, SORT *human*, SEM [3]], [POS *p*, CASE *wo*, SORT *thing*, SEM [4]] }
SEM [ACTION *buy*, AGENT [3], OBJECT [4]]

ひらがな列「かう」に対しては語彙処理部は，1番の語彙項目と，

KANA かう

を単一化することにより，次の語彙素性構造を得る．

PHON kau
KANA かう
KANJI [0]
POS *v*
SUBCAT { [POS *p*, CASE *ga*, SORT *human*, SEM [3]], [POS *p*, CASE *wo*, SORT [1], SEM [4]] }
SEM [ACTION [2], AGENT [3], OBJECT [4]]

([0], [1], [2]) = (買う, *thing*, *buy*) ∨ (飼う, *animal*, *raise*)

このように,「買う」に対しては内部に選言のない語彙素性構造が,「かう」に対しては選言のある語彙素性構造が出力される．

これらの語彙素性構造は文解析部の入力となり，**図 2** に示すような文法を用いて解析される．この文法は，PATR 形式 [17] で書いたものである．ここで，記号 VP1, VP2, PP, NP, P は，カテゴリ名ではなく，素性構造の identifier である．各規則において，親カテゴリの PHON, KANA, KANJI 素性値は，子カテゴリの値を連結したものである．この制約は，意味表現を求めることには無関係であるが，本方法によって各入力形態での文全体の表記が求められることを示すために用いている．

「わたし が きっぷ を かう」という単語列が入力された場合を考える．まず，語彙処理部で語彙素性構造の列に変換される.「きっぷ」と「を」については表 2 の 2，3 の語彙項目がそのまま語彙素性構造として出

(1) VP1 → PP VP2
⟨VP1 PHON⟩ = append(⟨PP PHON⟩, ⟨VP2 PHON⟩)
⟨VP1 KANA⟩ = append(⟨PP KANA⟩, ⟨VP2 KANA⟩)
⟨VP1 KANJI⟩ = append(⟨PP KANJI⟩, ⟨VP2 KANJI⟩)
⟨VP1 POS⟩ = *v*
⟨PP POS⟩ = *p*
⟨VP2 POS⟩ = *v*
⟨VP1 SEM⟩ = ⟨VP2 SEM⟩
⟨VP2 SUBCAT⟩ = ⟨PP⟩ ∪ ⟨VP1 SUBCAT⟩
(2) PP → NP P
⟨PP PHON⟩ = append(⟨NP PHON⟩, ⟨P PHON⟩)
⟨PP KANA⟩ = append(⟨NP KANA⟩, ⟨P KANA⟩)
⟨PP KANJI⟩ = append(⟨NP KANJI⟩, ⟨P KANJI⟩)
⟨PP POS⟩ = *p*
⟨P POS⟩ = *p*
⟨NP POS⟩ = *n*
⟨PP CASE⟩ = ⟨P CASE⟩
⟨PP SORT⟩ = ⟨NP SORT⟩
⟨PP SEM⟩ = ⟨NP SEM⟩

図 2 文法の例
Fig. 2 Example grammar.

力され,「かう」については，上述の語彙素性構造が出力される.「わたし」と「が」についても同様に語彙素性構造が得られるとする．これらに文法を適用することにより，**図 3** のような構文構造が得られる（この構文木での記号 VP，PP，V，N，P は対応する素性構造の簡略形である）.

文全体の素性構造は，次のようになる.

$$\begin{bmatrix} \text{PHON} & \text{watashigakiqpuokau} \\ \text{KANA} & \text{わたしがきっぷをかう} \\ \text{KANJI} & \text{私が切符を買う} \\ \text{POS} & v \\ \text{SUBCAT} & \{\} \\ \text{SEM} & \begin{bmatrix} \text{ACTION} & \textit{buy} \\ \text{AGENT} & \textit{speaker} \\ \text{OBJECT} & \textit{ticket} \end{bmatrix} \end{bmatrix}$$

切符の SORT 素性は *animal* ではないので，構文意味解析中に「かう」の意味が *raise* である解はなくなり，*buy* の解のみが残る.

終端記号と語彙項目の対応表は，語彙項目の集合から次のようにして自動的に生成することができる．すべての語彙項目について，各入力形態の素性が持ちうるすべての値を調べ，その各々と入力形態および語彙項目番号の 3 つ組を作ればよい．したがって，辞書記述者は，語彙項目を記述するだけでよい.

この例では，漢字かな混じり列の終端記号の曖昧性のみを選言によって記述したが，逆に,「あく」と「ひらく」など，同じ「開く」という漢字かな混じり表記を持つ単語もある．本方法では，特定の終端記号に関

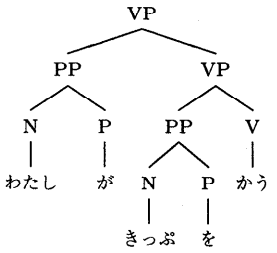


図 3 構文木の例
Fig. 3 Example parse tree.

する曖昧性の選言的記述だけではなく，1 つの選言的素性構造の中に，複数の種類の終端記号の曖昧性の記述を行うことができる．この例の場合には,「開く」,「ひらく」,「あく」，さらには,「あく」の別の漢字かな混じり表記の「空く」や「飽く」なども 1 つの選言的素性構造の中に記述することができる．それらすべてに関して，動詞としての性質は共通に記述することができる．ただし，格助詞の「に」と数詞の「に」のように，品詞の異なるものに関しては，語彙記述の中身が大きく異なるため，曖昧性を選言で記述することによるメリットがほとんどないと考えられる．したがって，同じ品詞の単語の辞書記述に関しては，1 つの終端記号が複数の語彙項目に現れないように選言的素性構造によるパッキングを行うという基準を用いることにより，不必要なパッキングを回避することができると考えられる．また，この基準を用いることにより，1 つの終端記号に関する情報が複数の語彙項目に現れることを回避することができると考えられる☆.

4. 実　　現

上記の辞書記述法を用いて，話し言葉解析実験システム [14),15)] の文法を構築した．この解析システムでは，選言的素性構造の単一化は，制約射影と呼ぶ論理制約の変換法によって，選言を展開することなく効率良く行うことができる [11),12)]．文解析は上昇型チャート解析 [7)] によって行う．文法は JPSG[4)] に基づいており，受動文，使役文，および，話題化，等位接続，連体修飾，接続助詞を含む文など，日本語の基本的な文型の文を扱うことができる．さらに，言い直し，つなぎ語などの話し言葉特有の現象をカバーしている [13),15)]．入力形態は，ローマ字列，漢字かな混じり列，音声認

☆ この基準を用いると，語彙項目によっては選言的素性構造が非常に大きくなってしまい，解析効率が悪くなる場合がありうるが，そのときには適切な大きさの語彙項目に分割する必要がある．しかしながら，分割をどのように行うかの基準を決定することは今後の課題である.

識システム ECLAIR2[20] が出力するカテゴリ付き単語列の3つを用いることができる.

この文法は文献14)の実験で用いられているが,その実験では,あらかじめ選言を展開した文法と,選言を保持したままの文法とを用いて構文解析の比較実験を行い,選言を保持したままの文法を用いた解析の方が効率が良いことが示されている.この実験は終端記号に関する選言のみを扱ったものではないが,この実験結果と,日本語には同音異漢字語が非常に多いことを考え合わせると,本論文の辞書記述法が,構文解析の効率の低下を防ぐのに有効であると考えられる.

5. おわりに

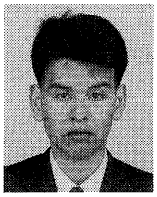
本論文では,選言的素性構造の中に,各入力形態の終端記号を素性値として記述することにより,1つの辞書記述で複数の入力形態に対応し,かつ,入力形態の増加による構文意味解析の効率の低下を避ける方法を提案した.

謝辞 日頃ご指導いただく石井健一郎情報科学研究部長,川端豪グループリーダ,および,討論していただいた対話理解研究グループの皆様,実験システムの実現を手伝っていただいた井上みづほ氏,今井豊氏,静洋一郎氏に感謝します.本研究に際し,NTTヒューマンインタフェース研究所音声情報研究部で開発した音声認識サーバシステム ECLAIR を使用しました.同システム使用についてご支援をいただいた嵯峨山茂樹氏,山田智一氏,井本貴之氏に感謝します.

参考文献

1) Allen, J.: *Natural Language Understanding* (second ed.), Benjamin/Cummings (1995).
2) Dörre, J. and Eisele, A.: Feature Logic with Disjunctive Unification, *Proc. 13th COLING*, Vol.2, pp.100–105 (1990).
3) Eisele, A. and Dörre, J.: Unification of Disjunctive Feature Descriptions, *Proc. 26th ACL*, pp.286–294 (1988).
4) Gunji, T.: *Japanese Phrase Structure Grammar*, Reidel, Dordrecht (1987).
5) Karttunen, L.: Features and Values, *Proc. 10th COLING*, pp.28–33 (1984).
6) Kasper, R.T.: A Unification Method for Disjunctive Feature Descriptions, *Proc. 25th ACL*, pp.235–242 (1987).
7) Kay, M.: Algorithm Schemata and Data Structures in Syntactic Processing, Technical Report CSL-80-12, Xerox PARC (1980).
8) Kay, M.: Parsing in Functional Unification Grammar, *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, Dowty, D.R., Karttunen, L. and Zwicky, A.M. (Eds.), pp.251–278, Cambridge University Press (1985).
9) 小暮 潔:増進的複製による型付き素性構造汎化手法,情報処理学会論文誌, Vol.34, No.9, pp.1993–1930 (1993).
10) Maxwell, J.T. and Kaplan, R.M.: A Method for Disjunctive Constraint Satisfaction, *Current Issues in Parsing Technology*, Tomita, M. (Ed.), pp.173–190, Kluwer (1991).
11) Nakano, M.: Constraint Projection: An Efficient Treatment of Disjunctive Feature Descriptions, *Proc. 29th ACL*, pp.307–314 (1991).
12) 中野幹生,島津 明:論理的制約の射影演算を用いた単一化に基づく構文解析,情報処理学会論文誌, Vol.36, No.1, pp.22–31 (1995).
13) 中野幹生,島津 明:言い直しを含む発話の解析,情報処理学会研究報告, SLP16-1, pp.1–6 (1997).
14) 中野幹生,島津 明:選言情報を含む単一化文法記述から論理的制約に基づく内部表現への変換法,情報処理学会論文誌, Vol.38, No.3, pp.490–499 (1997).
15) Nakano, M., Shimazu, A. and Kogure, K.: A Grammar and a Parser for Spontaneous Speech, *Proc. 15th COLING*, pp.1014–1020 (1994).
16) Pollard, C.J. and Sag, I.A.: *Head-Driven Phrase Structure Grammar*, CSLI, Stanford (1994).
17) Shieber, S.M.: *An Introduction to Unification-Based Approaches to Grammar*, CSLI (1986).
18) 島津 明,内藤昭三,野村浩郷:構造予測を用いた日本語文の意味解析法,情報処理学会論文誌, Vol.27, No.2 (1986).
19) Tsuda, H., Hasida, K. and Sirai, H.: JPSG Parser on Constraint Logic Programming, *Proc. 4th European Chapter of ACL*, pp.95–102 (1989).
20) 山田智一,野田喜昭,井本貴之,嵯峨山茂樹:クライアント・サーバ構成の HMM-LR 連続音声認識システムとその応用,情報処理学会研究報告, SLP5-6, pp.39–46 (1995).

(平成9年12月3日受付)
(平成10年9月7日採録)



中野 幹生（正会員）

1965年生．1988年東京大学教養学部基礎科学科第一卒業．1990年同大学大学院理学系研究科修士課程修了．同年日本電信電話（株）入社．以来，同社基礎研究所において，自然言語処理，対話理解の研究に従事．人工知能学会，言語処理学会，日本ソフトウェア科学会，ACL各会員．

島津 明（正会員）

1948年生．1971年九州大学理学部数学科卒業．1973年同大学大学院修士課程修了．同年日本電信電話公社武蔵野電気通信研究所入所．1997年北陸先端科学技術大学院大学情報科学研究科教授．自然言語処理の研究に従事．工学博士．言語処理学会，計量国語学会，電子情報通信学会，人工知能学会，ACL，ACM各会員．