選言的素性構造による複数の形態の入力に対応した辞書記述

中野幹生　　島津 明
NTT 基礎研究所

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．例えばテキストの解析を行うシステムでは，平仮名列，漢字仮名混じり列，ローマ字列などを解析することが有り得る．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．さらに，意味表現の論理形式を語順の自由な言語の文とみなして構文解析することによって文を生成する方法[4),9)]では，論理形式も入力形態の一つと考えることができる．このような入力形態のバリエーションをモダリティと呼ぶことにする．

モダリティ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つくるのは労力がいるため，複数のモダリティに対応した構文解析システムを作成することを考える．本稿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は，文法を宣言的に記述することが可能であり，従って，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．

モダリティに関わらず，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．モダリティに応じて終端記号(単語)が異なるので，辞書だけをモダリティ毎に作る方法が考えられる．しかし，音素列「aruku」とかな漢字混じり列の「歩く」のように，異なるモダリティの終端記号が，同じ語彙素性構造を持つ場合があるため，記述が冗長になってしまう．そこで，一つの辞書記述で複数のモダリティに対応する方法が望まれる．

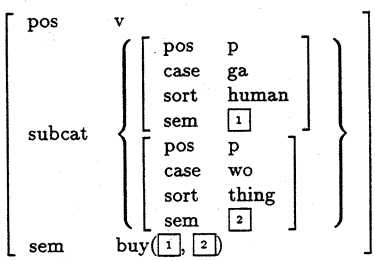
最も単純な方法として，各モダリティの終端記号の組をつくり，それに辞書記述を割り当てる方法が考えられる．例えば，⟨aruku, 歩く⟩に対して，語彙素性構造を与える．

この方法だと，選言の値を持つ素性構造(選言的素性構造)[2),3)]を用いた場合に問題が生ずる．選言的素性構造を用いると，辞書記述中の冗長性が減り，構文解析の効率が向上する．例えば，「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と，音素列入力用の辞書では，下位範疇化素性(格フレーム)のヲ格と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上，構文解析も冗長な処理を行わなくてすむため高速になる．しかし，漢字入力用の辞書では別々の辞書記述が必要である．したがって，モダリティの終端記号の組は，⟨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つ持つこととなるため，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モダリティに応じて変化させる研究[10)]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．

本稿では，選言的素性構造の中に素性値として各モダリティの終端記号を記述することにより，一つの辞書記述で複数のモダリティに対応する方法を提案する．本方法では，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．

2 辞書記述を複数のモダリティに対応させる際の問題点

次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．文法は辞書と規則からなる．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．以下に例を示す．



例えば，pos(part of speech, 品詞)が素性名でvが値である．値は，シンボルだけではなく，素性構造や素性構造の集合である場合もある．subcat(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る．[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

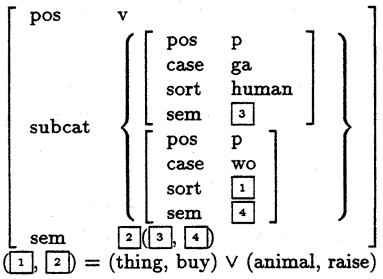
音素列	平仮名列	漢字仮名混じり表記	語彙素性構造
kau	かう	買う	pos v subcat { [pos p, case ga, sort human, sem 1], [pos p, case wo, sort thing, sem 2] } sem buy(1, 2)
kau	かう	飼う	pos v subcat { [pos p, case ga, sort human, sem 1], [pos p, case wo, sort animal, sem 2] } sem raise(1, 2)
kiqpu	きっぷ	切符	pos n sort thing sem ticket

図 1: 従来の辞書記述例

ticket になったとすると，意味素性値の buy(1, 2) は buy(taro, 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，平仮名列，漢字仮名混じり表記など，N 種類のモダリティ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 種類のモダリティ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．具体的を図 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，平仮名列，漢字仮名混じり表記などのモダリティは，1 対 1 対応しない．例えば，漢字仮名混じり表記で「飼う」と表される動詞も，平仮名では「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，平仮名列の入力に「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 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時には，殆んど同じ処理が 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済む．例えば，構文意味情報を，次のように記述すればよい．



この構造で，(1, 2) = (thing, buy) ∨ (animal, raise) は，1と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 [2), 3)] と呼ぶ．選言的素性構造同士の単一化を，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 [1), 2), 5)]．

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，図 1の素性構造を用いようとすると，辞書記述のセットをモダリティ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関わらず，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，様々なモダリティの入力に対応し，かつ，モダリティ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．

3 選言的素性記述による複数のモダリティへの対応

上記の問題への解決法として，本稿では，選言的素性構造の中に素性値として各モダリティの終端記号を

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 [2]([3], [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

図 2: 辞書記述の例

記述することにより，一つの辞書記述で複数のモダリティ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．語彙項目は，モダリティ $M_1, ..., M_n$ を素性として持ち，その値を各モダリティの終端記号とする．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 $Item$ がモダリティ M の終端記号 w を素性値として持つことを示す．モダリティ 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

$$[\ M \quad w\ ]$$

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，語彙項目の例を図 2に示す．ここで，phon, kana, kanji は，音素列，平仮名列，漢字仮名混じり表記のモダリティを表す素性である．次に，終端記号と語彙項目の対応表の例を図 3に示す．

終端記号	モダリティの種類	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

図 3: 終端記号と語彙項目の対応表の例

漢字仮名混じり表記「買う」に対しては語彙処理部は，1 番の語彙項目と，

$$[\ \text{kanji} \quad \text{買う}\ ]$$

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 買う
pos v
subcat { [pos p, case ga, sort human, sem [3]], [pos p, case wo, sort thing, sem [4]] }
sem buy([3], [4])]

平仮名列「かう」に対しては語彙処理部は，1 番の語彙項目と，

$$[\ \text{kana} \quad \text{かう}\ ]$$

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 [2]([3], [4])
([0], [1], [2]) = (買う, thing, buy)
∨ (飼う, animal, raise)]

(1) VP → PP V
⟨VP phon⟩ = append(⟨PP phon⟩, ⟨V phon⟩)
⟨VP kana⟩ = append(⟨PP kana⟩, ⟨V kana⟩)
⟨VP kanji⟩ = append(⟨PP kanji⟩, ⟨V kanji⟩)
⟨VP pos⟩ = v
⟨PP pos⟩ = p
⟨V pos⟩ = v
⟨VP sem⟩ = ⟨V sem⟩
⟨V subcat⟩ = ⟨PP⟩ ∪ ⟨VP 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 4: 文法の例

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4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[8]で書いたものである．各規則において，親カテゴリの phon, kana, 漢字仮名混じり表記素性値は，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，本方法によって各モダリティ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」については図2の 2, 3 の語彙項目がそのまま語彙素性構造として出力され，「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，以下のような構文構造が得られる．

[[[わたし]$_{NP}$[が]$_{P}$]$_{PP}$
[[[きっぷ]$_{NP}$[を]$_{P}$]$_{PP}$[かう]$_{V}$]$_{VP}$]$_{VP}$

文全体の素性構造は，次のようになる．

$$\begin{bmatrix} \text{phon} & \text{watashigakiqpuokau} \\ \text{kana} & \text{わたしがきっぷをかう} \\ \text{kanji} & \text{私が切符を買う} \\ \text{pos} & \text{v} \\ \text{subcat} & \{\} \\ \text{sem} & \text{buy(speaker, ticket)} \end{bmatrix}$$

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，各モダリティの素性が持ち得る全ての値を調べ，その各々とモダリティ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．

4 おわりに

選言的素性構造の中に，各モダリティの終端記号を素性値として記述することにより，一つの辞書記述で複数のモダリティに対応し，かつ，モダリティ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．本方法は，話し言葉解析実験システム[7]の中の構文意味解析モジュール[6]の語彙処理部として実現している．

参 考 文 献

1) A. Eisele and J. Dörre. Unification of Disjunctive Feature Descriptions. In *ACL-88*, pp. 286–294, 1988.

2) R. T. Kasper. A Unification Method for Disjunctive Feature Descriptions. In *ACL-87*, pp. 235–242, 1987.

3) M. Kay. Parsing in functional unification grammar. In D. R. Dowty, L. Karttunen, and A. M. Zwicky, editors, *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, pp. 251–278. Cambridge University Press, 1985.

4) M. Kay. Chart generation. In *ACL-96*, pp. 200–204, 1996.

5) 中野，島津． 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, 36(1):22–31, 1995.

6) M. Nakano, A. Shimazu, and K. Kogure. A grammar and a parser for spontaneous speech. In *COLING-94*, pp. 1014–1020, 1994.

7) 中野，島津，小暮. 分散協調処理による自然言語解析. コンピュータソフトウエア, 12(5):33–44, 1995.

8) S. M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*. CSLI, 1986.

9) S. M. Shieber. A Uniform Architecture for Parsing and Generation. In *COLING-88*, pp. 614–619, 1988.

10) 島津，内藤，野村. 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, 27(2), 1986.