

# 文脈自由文法を基礎とするマルチモーダル入力対応文法と その分散解析法

若梅友則      田野俊一  
電気通信大学 情報システム学研究科

## 1 序論

近年、コンピュータの高性能化・マルチメディア化に伴いヒューマン・コンピュータインタフェースの分野において、ユーザとの対話に音声・手書き文字・ジェスチャ等の利用が可能なマルチモーダルユーザインタフェースの研究が盛んである。我々も視線や動作パターンを用いたシステムの研究を進めている[7]。

そこで、現在のマルチモーダルインタフェースの一般的なシステム構成をみると、個々のモダリティ毎に対応する認識エンジンを使ってそれぞれの入力を認識し、それらの認識結果を統合し、入力の意味を把握する。そして、この把握した結果と対話過程の文脈などからユーザの意図を推定している。

また、現在までのマルチモーダルインタフェースは応用システムの開発に焦点が置かれており、それらのシステムのマルチモーダル入力の解釈のためのマルチモーダル入力を表現する方法は、システムによって異なりどのようなシステムにも適用できるマルチモーダル入力を表現する方法はほとんど無い。さらに、このように応用システムに依存する表現方法を使用したマルチモーダル入力の解釈能力は、高いとは言えず、汎用的でもない。しかし、モダリティ間で共通して使用できる知識もあり、この共通の知識を表現可能な記述方法も必要となる。したがって、マルチモーダル入力を共通して表現する、汎用的な方法とそれを使用した統合解析システムが必要である。

そこで、本研究では、以下の二つを提案し、これらを使用したシステムを試作し、検証を行なう。

- マルチモーダル入力を表現するための汎用的な文法
- マルチモーダル入力の解析アルゴリズム

## 2 従来手法の問題点と必要な性質

### 2.1 システム構成上の問題点と必要な性質

図1に手書き文字と音声のそれぞれの代表的な認識システムを示す。

既存のシステムの大部分が取っている、このようなシステム構成においては、個々のモダリティにおいて、特徴抽出を行ない、とりあえず個々のモダリティの認識結果を出力する。個々のモダリティからの出力が全て行なわれてから、それらの出力を用いて統合解析を行ない、マルチモーダルシステム全体としての結果を出力している。

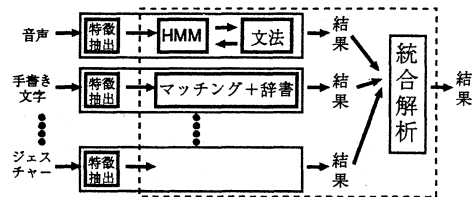


図1: マルチモーダル入力システム

しかし、このようなシステム構成では、個々のモダリティの結果が出そろった時点でシステム全体の結果出力も、統合解析の実行も困難である。だが、全てのモダリティの結果が出そろわなくても、それまでに個々のモダリティが認識した結果から、全体の結果を得られる方が有益である。また、全てのモダリティは、並列的に入力され、どのモダリティにも優先順位などはない。したがって、統合解析を行なうためには、それぞれのモダリティは並行的に処理される必要がある。さらに、既存のシステムにおいては、個々のモダリティの認識結果の粒度(単音、単語、生データなど)は単語レベルや単音レベルなど個々のモダリティ用の認識システムの設計時に決定されている。これらの出力を用いた統合解析では、個々のモダリティの認識出力結果の粒度より小さい粒度を必要とする情報統合は困難である。したがって、同一アルゴリズムを用いて全てのモダリティを認識すれば粒度に無関係に統合が可能である。

そこで、本研究では図1の点線で囲った部分を、マルチモーダル入力の表現とそれを使用した解析パーザで構成した。そして、システム全体として以下の4つを満たすアルゴリズムにした。

- システム全体としてはいつでも結果を得られるアルゴリズム (Anytime 性)
- 個々のモダリティには優先順位がなく、同時並行で処理されるアルゴリズム (分散並列性)
- 分散並行処理時に、相互参照可能できるアルゴリズム (相互参照性)
- 複数粒度を扱えるアルゴリズム (複数粒度性)

### 2.2 マルチモーダル入力の表現の問題点と必要な性質

現在までに多くのマルチモーダルシステムが実現されているが、それら従来のマルチモーダル入力の統合のための表現形式は、以下の3種類に分類される。それぞれについて問題点を述べて行く。

1. モダリティに依存した手法 (Put-That-There [1])
2. アドホックな手法 (melting pot [2])

### 3. ロジカルな手法 (MM-DCG [3])

モダリティに依存した手法とは、「場所の指定は必ずマウスで指定する」「色や形は音声で指定する」などの、特定の操作を行なう時は特定のモダリティを使うと限定する手法であり、この指定はプログラム中に埋め込まれており、知識として具体的な表現はされていない。このようにモダリティに依存した手法は、汎用性が低く、マルチモーダル入力への表現としては使えない。

アドホックな手法とは、問題に依存し、その問題の解決のためだけに作成した表現手法であり、既存の多くの実用的なシステムが取っている方法である。モダリティに依存した手法に似ているが、こちらは知識としてプログラムからは分離して表現されている。しかし、このようなシステムのそれぞれの手法は実用性を求めているため、その問題解決に適した表現がされている。したがって、このような表現は他の問題のためには使用できず、望ましい表現ではない。

ロジカルな手法とは、意味を深く考慮してマルチモーダルを解釈するための表現手法である。これは上記2つと異なり、汎用性を持った表現であるが、1つの情報に多数の意味情報を付加するので、マルチモーダルを表現する記述量が多く、また、時間関係や深い意味を考慮するので、処理量も多く、システムが重くなり望ましくない。

以上まとめると、マルチモーダル入力の表現としては

- マルチモーダル入力を記述できる表現
- モダリティに依存せず、どのようなモダリティ構成のシステムにも使用可能な、汎用的な表現
- 高速処理可能な表現

ができる手法が望ましい。

## 3 マルチモーダル対応文法

### 3.1 マルチモーダル対応文法的设计

本研究では、高速解析可能で汎用性を必要とするため文脈自由文法を基礎として用いる。

そして、この文脈自由文法にマルチモーダルを表現できるように変更を加える。これは異なるモダリティの同じ情報を示すものを結合する事で充分である ([4])。つまり、

1. 終端記号同士の結合
2. 終端記号と非終端記号の結合
3. 非終端記号同士の結合

の結合を表現出来れば充分である。そこで新たな記法として、この結合を表現する記号を導入し、記号としては、同一の内容を示していることを明記するために“=”を使用する事とした。

### 3.2 記法

CFGに、結合の文法記述である、“=”記号を導入する。つまり、表現として図2の4つを許すことにする。本研究で試作したシステムでは、これに加え、図3に示す記法が、許された記法全てである。

以下の説明においては、それぞれのイベントがどのモダリティで起こったのかを明記するために“イベント/モダリティ”のように記述する。たとえば、“私/音声”は“私”と発音することを示している。

非終端記号 = 非終端記号.

終端記号 = 非終端記号.

非終端記号 = 終端記号.

終端記号 = 終端記号.

図2: 同一内容の記法

- @ モダリティ名.
- \* 開始記号.
- 非終端記号 - 非終端記号 +.
- 非終端記号 : 終端記号 +.

図3: マルチモーダル対応文法の“=”以外の記法

### 3.3 マルチモーダル対応文法の解析木

マルチモーダル対応文法を使用して、マルチモーダル入力を統合解析した結果として得られる解析木は、普通の解析木の中に“=”で結合した表現を含んでいる木である (図4)。以下、このような木を統合解析木と呼ぶ。この“=”の結合を除けば、普通の CFG の解析木と同じである。つまり、一つの開始記号から葉まで枝で全部つながっており、途中“=”で結合した枝が存在する木である。

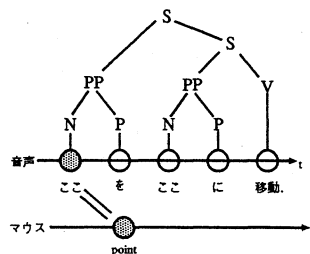


図4: マルチモーダル対応文法を使った解析木

さらに、認識したすべてのモダリティのすべての終端記号を1つの解析木にまとめあげる必要はなく、無視する終端記号が存在しても良い。ただし、使用する終端記号は1つの解析木において1回しか現われてはならない。

### 3.4 文脈依存文法への拡張

以上は CFG を基礎としたマルチモーダル入力情報の解析手法であったが、例えば以下のように記述したい場合も存在する。

$aNb = cMd.$

これは以下のように記述できる。

$aNb \rightarrow A.$   
 $cMd \rightarrow B.$   
 $A = B.$

これはすなわち、基本となる文法を CFG から文脈依存文法 (CSG) にすることにより“=”は自然に文脈依存の結合を表現できるようになることを示している。

## 4 マルチモーダル入力統合解析アルゴリズム

### 4.1 マルチモーダル対応文法の解析法

マルチモーダル対応文法は文脈自由文法を基礎としているので、構文解析アルゴリズムとしては文法規則適用に関する曖昧さが無く、バックトラック無しで無駄無く構文解析できる富田法を基礎として用いる。富田法は、入力がある毎にボトムアップ的に解析木を作って性質を持つため、Anytime 性と相性が良い。

まず、分散並列性について考えて行く。“=”を含んだマルチモーダル対応文法から LR パーズ表(以下解析テーブルと呼ぶ)を作成する必要がある。そこで、分散並列性を考慮し、複数の解析テーブルを用いた解析法を考えた。この解析法はマルチモーダル対応文法を、モダリティ毎の文法に分割し、それぞれの文法に対して1つの解析テーブルを作成し、マルチモーダル入力を解析する。

次に文法の分割方法だが、まず、“=”の記法を除いた部分を対象に、終端記号、非終端記号が1つの文法だけに現れる文法に分割する。多くの場合モダリティ毎の文法に分割されると考えられる。残された“=”の文法の処理は“=”の式に関わる物が出現したら“=”の反対側の記号を見て、その記号を含んでいる他の文法が、その記号に関しては解析していると仮定し、モダリティ内での解析を続行する。

最終的な解析木をつくるには相手の文法の表を見る。それぞれの1つの文法の解析には従来の解析法が使え“=”の文法導入に起因するアルゴリズムの変更を小さくすることができる。

以上の手法では、文法を分割した事により分散並列性が、富田法のボトムアップ型解析法により Anytime 性が、“=”の処理の導入により相互参照性が実現されている。

### 4.2 基本解析アルゴリズム

富田法に、マルチモーダル対応文法の“=”の解析手法を付け加えたアルゴリズムについて説明する。

パーズ表の作成アルゴリズムは富田法と全く同じで変更は加えていない。ただし、“=”の文法規則を記録する表を、パーズ表とは別に作成する。

富田法の全てのスタックの状態において解析失敗になる場合、解析失敗とする前に、まず、“=”の文法規則が適用できないかを確認する。“=”の文法規則が適用可能であり、解析失敗した入力が解析可能になる場合、とりあえず、“=”の文法を使用して解析を行ない、現在の入力の再解釈を行なう。また、受理する前に入力が終了してしまった場合は、最後に“=”の文法規則が適用可能か確認する。適用可能な場合は適用を行なう。

そして“=”の反対側を確認するのは、解析時ではなく、結果を出力する時点である。この時点で、“=”の反対側のモダリティでの構文解析を行なっているシステムにその記号の解析が終了しているかを問い合わせる。終了していると返答があった場合は、その出力を木に加え出力する。ないと返答があった場合は、その木を結果から削除する。

“=”以外の文法規則は富田法をそのまま適用し解析する。以上でマルチモーダル対応文法は解析できる。

### 4.3 誤り対応

誤り対応を行なうために、基本解析アルゴリズムに誤り対応のアルゴリズム(一般 LR 構文解析法におけるエラー処理 [5]) を付け加える形にした。パーズ表に関しては、前節と同じく変更を加えない。

変更を加えたところは、基本解析アルゴリズムと同じくパーズ表を引いた結果が全てのスタックの状態において対応するアクションが無く失敗した時の対処である。この対処機能は記号置換、記号飛び越し、記号挿入と呼ばれており、全て並列に行なわれる。これはスタックを複数持つ富田法の特徴を利用している。それぞれ、記号置換は、現在の入力を受け入れられる品詞全てに置き換える。記号飛び越しは、現在の入力を飛び越し、次の入力を得る。記号挿入は、現在の入力の前に、受け入れられる品詞を挿入し解析の続行をする。

## 5 マルチモーダル入力統合解析システム

図5にマルチモーダル対応文法を使用したマルチモーダル入力統合解析システムの構造と処理の流れを示す。矢印が情報の流れを示している。実線の四角は独立的に動作するプロセスであり、他のプロセスとの情報のやりとりを行なう必要がある。

個々のエージェントは、マルチモーダル入力統合解析アルゴリズムで解析動作を行なう。

システム構成の設計時において、Anytime 性、分散並列性、相互参照性、複数粒度性を考慮している。

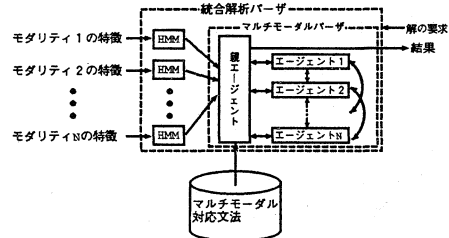


図5: マルチモーダル統合解析システムの構造

## 6 統合例:”う冠に木”

マルチモーダル統合解析システムの動作例を挙げる。この状況を図6に、文法を図8に統合解析結果を図7に示す。これは漢字入力をしている時に、漢字の一部を言葉で発音する事により補完できる例である。既存のシステムではこのような状況を記述し、さらに実現するのは困難であるが、本研究で提案したマルチモーダル対応文法で非常に単純に記述でき、かつ、ユーザインタフェースとして非常に有用に動作すると考える。

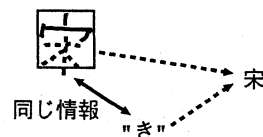


図6: うかんむりに”き”の状況

実際にシステムを動作させた結果を以下に挙げる。ここにおける入力は

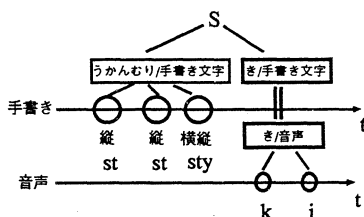


図 7: "う冠に木"の解析木状況

* S.		
@ VOICE.		
S	-	UKANMURI/V KI/V.
UKANMURI/V	-	UKANMURI.
KI/V	-	KI.
U	:	u.
K	:	k.
A	:	a.
N	:	n.
M	:	m.
U	:	u.
R	:	r.
I	:	i.
@ STROKE.		
S	-	UKANMURI/S KI/S.
UKANMURI/S	-	ST ST STY.
KI/S	-	SY ST SH SM.
ST	:	st.
STY	:	sty.
SY	:	sy.
SH	:	sh.
SM	:	sm.
UKANMURI/S	=	UKANMURI/V.
KI/S	=	KI/V.

図 8: "う冠に木"の文法定義

#### • "st st sty k i"

であり、これは"うかんむり"と書いた後、"き"と言った事に相当する。(ここで st sty はストロークに対応する。)

```
S_0_2 - (UKANMURI/V_0_0) KI/V_0_2 .
(UKANMURI/V_0_0) : UKANMURI/V = UKANMURI/S_0_3 - ST_0_1 ST_1_2 STY_2_3 .
ST_0_1 : "st" .
ST_1_2 : "st" .
STY_2_3 : "sty" .
KI/V_0_1 : K_0_1 I_1_2 .
K_0_1 : "k" .
I_1_2 : "i" .
```

```
S_0_4 - UKANMURI/S_0_3 (KI/S_3_4) .
UKANMURI/S_0_3 - ST_0_1 ST_1_2 STY_2_3 .
ST_0_1 : "st" .
ST_1_2 : "st" .
STY_2_3 : "sty" .
(KI/S_3_4) : KI/S = KI/V_0_1 : K_0_1 I_1_2 .
K_0_1 : "k" .
I_1_2 : "i" .
```

音声、手書き文字、共に1つの解析結果の木を得た。つまり、漢字の一部を言葉で補うことができた。

さらに、同じ辺やつくりを使用する漢字が増えても、辺やつくりについての音声は、1つの記述だけで済む。この様に、漢字などの複雑な物の一部を他のモダリティで示すことは、比較適簡単に行なえるので、このような記述は大変有意義である。

## 7 まとめ

本稿では、マルチモーダル対応文法と、その統合解析法について述べた。

マルチモーダル対応文法はCFGに新しい記号"="を導入し同じ情報を示していると言う結合を表現した。これは単純に結合を示しているだけであるため、形式的で汎用的な記述が出来る。また、個々のモダリティ毎にまとめて記述しているので、個々のモダリティ用の文法に分割でき分散的な記述も可能である。

そして、マルチモーダル対応文法を使用した統合解析法では、エージェント動作で分散並列性を持たせ、結果の相互補完を行なった。また、任意の時点で不完全ではあるが解析木を出力することができAnytime性を実現した。さらに、音素レベルや、単語レベルだけでなく、音素レベルの物と単語レベルの結合など、粒度が異なる入力も処理が可能とした。

しかし、問題点として、解析法の基礎とした富田法は探索の爆発を避けるために圧縮木を用いているが、試作システムでは実際にエージェントが解析木を出力する際、この圧縮を解いてしまうので、文法記述や入力が増えれば、組合せ増え、探索時間が増大している。これを避けるためには、圧縮木の利用することに換え、意味を考えた誤り対応処理を行なうことにしたり、出力する解析木を選択する必要がある。

## 参考文献

- [1] Richard A.Bolt,"Put-That-There": Voice and Gesture at the Graphics Interface", ACM Computer Graphics Vol.14 No.3 pp.262-270, 1980
- [2] Laurence Nigay, Joelle Coutaz, "A Generic Platform for Addressing the Multimodal Challenge", CHI'95 MOSAIC OF CREATIVE pp.98-105, 1995
- [3] 島津 秀雄 高島 洋典,"マルチモーダル Definite Clause Grammar(MM-DCG)", 電子情報通信学会 Vol.J77-D-II No.8 pp.1438-1446, 1994
- [4] 若梅 友則 田野 俊一,"形式文法によるマルチモーダル入力の表現と解析", 1997 年度人口知能学会全国大会 (第 11 回) 19-04 pp.392-395, 1997
- [5] 斎藤 博昭,"一般 LR 構文解析法におけるエラー処理", 情報処理学会論文誌 Vol.37 No.8 pp1506-1513, 1996
- [6] Tomita, M. and Carbonell, J.G,"The Universal Parser Architecture for Knowledge-Based Machine Translation", Technical Report Center for Machine Translation, Carnegie Mellon University, Pittsburgh, 1987
- [7] 龍田 幸拓 田野 俊一,"推定モデルと DP マッチングを用いたマルチモーダル入力からの戸惑い検出アルゴリズムの提案と評価", 電子通信学会信学技法 PRMU98-3, 1998