

# 音声対話タスクのための高速なキーワードラティスパーザ BTH

## BTH:an Efficient Parsing Algorithm for Keyword Spotting

屋野 武秀  
Takehide YANO

(株) 東芝 研究開発センター マルチメディアラボラトリー  
Multimedia Lab., Corporate Research & Development Center, Toshiba corp.  
takehide.yano@toshiba.co.jp

笹島 宗彦  
Munehiko SASAJIMA

(同 上)  
munehiko.sasajima@toshiba.co.jp

河野 恭之  
Yasuyuki KONO

奈良先端科学技術大学院大学 情報科学研究科  
Graduate School of Information Science, Nara Institute of Science & Technology  
kono@is.aist-nara.ac.jp

**keywords:** spoken dialogue system, lattice parser

### Summary

In this paper, we propose BTH, a parsing algorithm which is able to efficiently parse keyword lattice that contains large number of false candidates. In BTH, the grammar is written in template form, and then, it is compiled into a hash table. BTH analyzes the lattice without unfolding to keyword sequences, by propagating acceptable templates among the linked keywords and filtering through the hash table in each keywords. It has a time bound proportional to  $n^2$  (where  $n$  is the number of keywords in the lattice), although the number of false candidates increases exponentially. Simulation results shows that BTH can parse lattice which contains over 100 billion false candidates within 0.35 sec, with grammar which is corresponding to 2 million of templates, on a notebook-PC(PentiumII 266MHz).

### 1. はじめに

近年に見られる計算機の性能の飛躍的な向上に伴い、音声や画像といったマルチメディア情報がより簡便かつ安価に扱えるようになってきている。これにより、従来は計算量の点で実現困難であった音声やジェスチャなどの人間にとって自然なコミュニケーションチャンネルによるシステムへの入力も可能となりつつある。利用者にとって扱いやすいこれら入力方法の利用を促進することで、より良いインターフェースとなることが期待される。特に音声は人間にとって基本的かつ自然なコミュニケーションチャンネルであり、音声による対話機能をもったシステムの実現が求められている。また音声対話機能はハンズフリーかつアイズフリーであるために、利用者が「ながら」利用する状況、例えばカーナビゲーションシステムなどへの適用が特に望まれている。

このような音声対話システムには音声認識結果に曖昧性があり結果を一意に得られないという問題がある。音声認識結果から文法で受理可能な複数の候補を探索し、それらに対して意味解釈およびユーザに回答すべき正解の選択を行う必要がある。

これに加えて音声対話システムには、特に「ながら」利用されるという状況下では利用者の発話が話言葉になるという問題がある。話言葉入力は書き言葉とは異なり、「えーと」などの不要語や「サ、サービスエリア」などの言い淀みといった特有の現象を持つため、その扱いが重要な課題となる。更に利用者システムとの対話を快適なものにするという観点から音声対話システムには即応性が求められる。

話言葉特有の現象の取扱いに有利な音声認識方式としてキーワードスポッティング方式 [金沢 93] がある。キーワードスポッティング方式は認識対象語彙 (キーワード) に該当する音声信号区間を入力音声信号から検出し、その区間と対応するキーワードの組みを出力するものである。このため不要語などの影響を受けずにキーワードを検出でき、話言葉の影響が軽減されるという特徴を持つ。しかしこの方式によって与えられる音声認識結果はその曖昧性により一次元にキーワードが並んだ系列とはならず、キーワードラティス (以下ラティス) と呼ばれるグラフ状の構造となる (図 1)。ラティス上で開始ノードから終了ノードに至る道がたどるキーワードの系列は、ユーザが発話入力した文の候補 (以下、文候補) に対応してい



売という規模の小さいものに限定し、音声による実時間対話を可能としている。この構文解析は拡張 LR パーザ [Tomita 86] に基づき、認識結果のラティスから文法を利用した枝刈りを行ないつつキーワード系列に展開している。しかしながら、大きなタスクに適用した場合、認識対象語彙の増加に伴い湧き出し誤りが増加し、ラティスのノード数が増加する。これによりラティスに含まれるキーワード系列数は指数関数的に増加することになる。またタスクに対応する文法規模の拡大に伴いノード間の接続可能性が高まるため、キーワード系列展開時の枝刈り効率が劣化する。

部分系列の尤度が上位のもののみを残し探索を進めるビームサーチなどの手法がこの計算量を削減するものとして挙げられる [Manning 99]。音声認識結果の尤度を参照することでキーワードラティスの構文解析にビームサーチを適用できると考えられるが、現状では認識結果の尤度計算方法に一般的なものは無くアプリケーションやタスクに応じて適切な方法を選択して調整し適用する必要がある [河原 00]。尤度計算方法をタスクやドメインに応じて調整すればビームサーチによる高速化が可能と考えられるが、タスク規模が大きくなるにつれてこの調整作業は非常に困難なものとなる。

また例えばカーナビ音声対話システムのように話者の状況が時間とともに変化するようなシステムでは、文候補に対して音響的な評価だけでなく意味的な評価が重要となる。例として「出口までどれくらい」という文候補は自車が高速道路上などで出口に向かって走行している状況では意味を成すが、そうでない場合には意味を成さない。このようなタスクにおいて音響スコアのみを参照したビームサーチを行うことは真の話者意図を抽出できない結果を招く。筆者らが試作したカーナビ音声対話システム上での予備実験からも音響スコアのみでなく意味スコアによる評価を加えることでより話者意図を反映した対話が可能になる結果が得られている [屋野 98]。

音声入力のシステムではラティス全体を考慮した構文解析を行い結果として複数候補を出力する必要がある。複数の文候補それぞれに意味的な情報を加味してそのうえで例えば N-best 処理 [田中 99] などで最適なものを選択する方式が望ましい。しかし従来 TOSBURG-II [竹林 94] などで用いられてきたラティスから一旦キーワード系列に展開する構文解析では、受理可能性が不明であるキーワード系列の仮説全てに対してその検証を行わねばならない。このためラティス・文法規模の拡大による計算量の増加は膨大なものになってしまう。

以上のような条件の下で対話システムとしてのインターフェースの即時性を保つためには、多くのキーワードを含む大規模なラティスに対しても高速に解析する構文解析器が必要である。

### 3. 文テンプレートハッシュ方式 (BTH)

文テンプレートハッシュ方式 (Bun(文) Template Hash:BTH) はラティスからキーワード系列に展開することなく解析し、文法で受理可能なキーワード系列を全て探索することが可能なラティス構文解析方式である。以下、3.1 節で BTH が対象とする文法形式について述べ、3.2 節、3.3 節で BTH のアルゴリズムについて説明する。最後に 3.4 節でその計算量について述べる。

#### 3.1 対象とする文法

書き換え規則やネットワーク形式の文法記述では、再帰的に同じ規則を適用することにより無限に長い文を受理することが可能となる。しかし人間が一度に話せることには限界があるため、無限に長い文を受理する必要性は小さい。そこで BTH に適用する文法を次のように限定する。

有限個のテンプレート形式に基づく文法 入力として考えられる個々の言いまわしのパタンを非終端記号 (例えば品詞) の系列のテンプレートで定義する。以下このテンプレートを文テンプレートと呼ぶ。文法は有限個の文テンプレートの集合と定義される。

書き換え規則やネットワーク形式での文法からでも、それから非終端記号系列まで展開することで文テンプレート集合を得ることが可能である。ただしその展開時には規則を再帰的に使用する回数を限定するなどして、最終的に有限個の文テンプレートとなるようにする必要がある。この制限は無限に長い文を受理しないように文法を限定するが、展開後の文法を使用する場合における実用上の問題は少ないと考えられる。

#### 3.2 アルゴリズム

BTH は 2 つの処理からなる。第一はラティスの左から右に受理可能な文テンプレートの種類を絞り込む処理 (絞り込み処理) であり、第二に最終的に絞り込まれた文テンプレート集合に該当する系列を抽出する処理である。絞り込み処理は各ノードに対して (1) その前方に接続されたノードで受理可能な文テンプレート集合を受け取り、(2) そのノードが受理しうる文テンプレート集合との積集合を計算して考える文テンプレートの種類を絞り込み、(3) 絞り込まれた文テンプレートをその後続ノードに伝播する、を繰り返すものである。この処理によって、ラティスから受理可能な文テンプレートを高速に探索することが可能となる。以下その詳細について述べる。

##### §1 BTH の構成

BTH は品詞辞書・文テンプレート辞書・文テンプレートハッシュ辞書・処理中ノードリストを参照する。

- 品詞辞書： キーワードとそれに対応する非終端記号 (例えば品詞) との関係性を記述した辞書。

- **A:** (経由地) (順序相対) (順序絶対) (施設) (WHERE)  
- 「出口 前 最後 サービスエリア どこ」
- **B:** (経由地) (順序相対) (施設) (WHERE)  
- 「インター 次 サービスエリア どこ」
- **C:** (経由地) (WHERE)  
- 「出口 どこ」
- **D:** (経由地) (順序相対) (順序絶対) (経由地) (WHERE)  
- 「出口 後 最初 交差点 どこ」
- **E:** (経由地) (順序相対) (経由地) (WHERE)  
- 「出口 次 交差点 どこ」

図 3 文テンプレート辞書の例

出現位置	1	2	3	4	5	6
経由地		順序相対	順序絶対	施設	WHERE	終了
	<b>A,B,C, D,E</b>	<b>A,B,D,E</b>	<b>A,D</b>	<b>A</b>	<b>A,D</b>	<b>A,D</b>
		WHERE	施設	経由地	終了	
		<b>C</b>	<b>B</b>	<b>D</b>	<b>B,E</b>	
			経由地	WHERE		
			<b>E</b>	<b>B,E</b>		
			終了			
			<b>C</b>			

図 4 文テンプレートハッシュ辞書の例

- 文テンプレート辞書： 受理可能な文テンプレートと対応する ID (文型 ID) とを記述した辞書。図 3 に一例を示す。図 3 において “( )” で囲まれている文字列が非終端記号を表し、左端のアルファベットが文型 ID を表している。
- 文テンプレートハッシュ辞書： 辞書に登録されている各非終端記号について、文中の出現箇所 (順序) と所属する文型 ID の集合への対応関係を要素とする表。つまり「ある非終端記号に属すキーワードがある順序に現れる場合、そのキーワードを使用した文章はどの文型 ID の文テンプレートに所属しうるか」という記述を要素に持つ。文テンプレートハッシュ辞書は文テンプレート辞書をコンパイルすることによって構成される。

文型 ID が  $i$  である文テンプレートを文テンプレートハッシュ辞書に登録する際には、 $i$  中の全ての非終端記号について、その記号と出現位置によって特定されるハッシュ辞書上の文型 ID 集合に文型 ID  $i$  を追加する。文テンプレート辞書にある各文テンプレートにつきこの登録作業を進めることで文テンプレートハッシュ辞書を構成する。例えば図 3 の文型 IDC を登録する場合は、ハッシュテーブル上で次の 3 つの非終端記号とその出現位置が交わる点に文型 IDC を登録する。

非終端記号: 出現位置  
(経由地) 1 番目  
(WHERE) 2 番目  
(終了) 3 番目

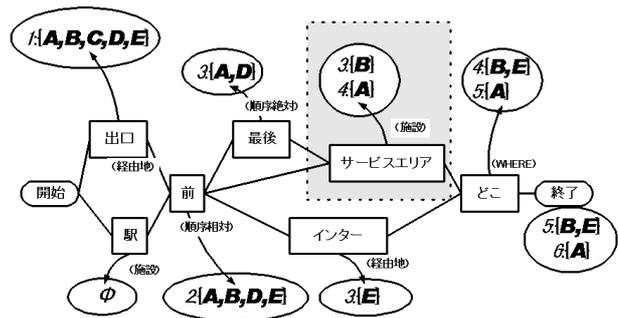


図 5 絞り込み処理の例

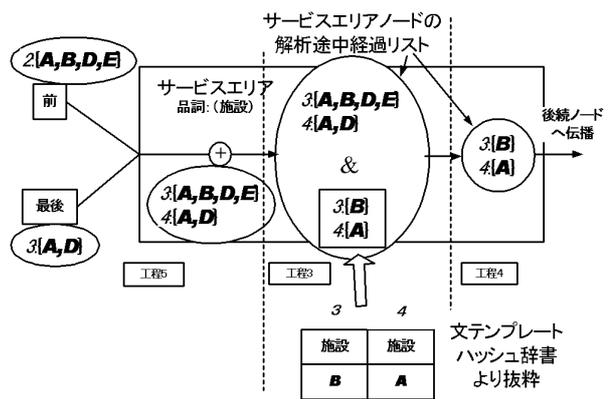


図 6 「サービスエリア」ノードにおける確定処理

図 4 は図 3 の文テンプレート辞書から生成される文テンプレートハッシュ辞書である。

BTH に文候補即ちキーワードの系列が入力されると、その系列中のキーワードの出現位置と対応する非終端記号をキーとして文テンプレートハッシュ辞書からある非終端記号が文中の特定の位置に出現する特徴を備えた文型の集合を抽出することができる。例えば、ある文候補の 3 番目のキーワードが非終端記号「(順序絶対)」に属しているなら、図 4 のハッシュ辞書より、その文候補を受理可能な文型 ID の集合は {A, D} となる。

- 処理中ノードリスト： 現在絞り込み処理対象とする入力ラティス上のノードのリスト。

ラティス上の各ノードには未処理前ノードリストと解析途中経過リストを持たせる。

- 未処理前ノードリスト： 各ノードの前方に接続されているノード (前ノード) のうち、絞り込み処理を終了していないノードのリスト。
- 解析途中経過リスト： そのノードを利用し、かつ受理可能性のある文型 ID の集合を、そのノードの出現順序別に記録するリスト。今後この内容を『[(順序: { 文型 ID 集合 }, ...)]』の形式で表記する。

## § 2 構文解析処理

[初期設定]

BTH は構文解析の初期設定として以下の手順を実行する。

- (1) 全てのノードの未処理前ノードリストをその前ノードのリストとする。
- (2) 系列の先頭となりうるキーワードに対応するノードを処理中ノードリストに追加する。
- (3) 処理中ノードリストに追加したノードの解析途中経過リストに初期値を設定する。初期値は文テンプレートハッシュ辞書から順序=1 と該当する非終端記号をキーとして検索した結果を用いて  $[(1:\{\text{検索結果}\})]$  とする。

#### [絞り込み処理]

初期設定が終了すると絞り込み処理を行う。絞り込み処理では、処理中ノードリストが空になるまで以下の工程を繰り返す。

1. 処理中ノードリストから一つのノード (X) を選択する。
2. X の未処理前ノードリストが空であるかを確認する。空でなければ 1 に戻る。
3. X に関する確定処理を行う。確定処理とは、解析途中経過リストに存在する順序の文型 ID 集合とその順序に対応する文テンプレートハッシュ辞書の要素との積集合を順序別に計算する処理である (図 6 中央)。
4. 確定処理によって得られた積集合群を X の解析途中経過リストに代入し、X の後続ノードに伝播 (図 6 右側) する。X を処理中ノードリストから除く。
5. X の後続ノードを処理中ノードリストに加える。また後続ノードでは、伝播されてくる解析途中経過リストと後続ノードの解析途中経過リストとの和集合を順序別に計算 (図 6 左側) し、X を未処理前ノードリストから除く。1 に戻る。

こうして上記 5 が終了した時点で処理中ノードリストが空になると全ノードの確定処理がなされたことになり、文テンプレート絞り込みが終了する。この時点で終了ノードの解析途中経過リストにある文型 ID 集合が、このラティスで受理可能な文テンプレートの種類となる。

先頭のノードの解析途中経過リストに与える初期値は「順序=1 で受理可能な全ての文テンプレートの集合」と見ることができる。これが伝播されるノードの解析途中経過リスト内順序=2 の要素は、「順序=1 である前ノードの全ての道を考慮して考えられる文テンプレートの集合」と等価になる。確定処理でその文テンプレート集合と順序=2 であるハッシュ辞書の要素との積集合を計算することによって、「順序=2 のそのノードを含めた全ての道を考慮して受理可能な文テンプレートの集合」を算出している。これをその後続ノードに伝播し確定処理を続けることにより、あるノード (X) に伝播される文テンプレート集合は「X までの全ての道を考慮して受理可能な文テンプレートの集合」と等価となる。X に関する確定処理は「X を含めた全ての道を考慮して受理可能な文テンプレートの集合」を算出することにほかならない。よって終了ノードの確定処理を終了した時点でのその解析途中経過リストは「終了ノードまでの全ての道を考慮して受理可能な文テンプレート集合」と等価になる。

最終的なキーワード系列を得る際には次のように系列をラティスからスキャンする。まず、系列探索の対象となる文型 ID を絞り込み処理結果の文型 ID 集合から選択する。続いて、開始ノードを 0 番目として系列探索を開始し、解析途中経過リスト中の必要とする順序の要素にその文型 ID を含むノードを順に探索しつつ、系列を構成してゆく。終了ノードまで探索された系列がその文型 ID に該当するキーワード系列となる。この処理を絞り込み処理結果中の全ての文型 ID について実行するとそのラティスで受理可能なキーワード系列が収集される。

#### [系列抽出]

最終的なキーワード系列を得る際には次のように系列をラティスからスキャンする。まず、系列探索の対象となる文型 ID を絞り込み処理結果の文型 ID 集合から選択する。続いて、開始ノードを 0 番目として系列探索を開始し、解析途中経過リスト中の必要とする順序の要素にその文型 ID を含むノードを順に探索しつつ、系列を構成してゆく。終了ノードまで探索された系列がその文型 ID に該当するキーワード系列となる。この処理を絞り込み処理結果中の全ての文型 ID について実行するとそのラティスで受理可能なキーワード系列が収集される。

### §3 動作例

図 5 は図 3、図 4 を利用した絞り込み処理の簡単な例である。初期設定により、開始ノードのみに接続しているノード「出口」「駅」の解析途中経過リストは確定処理終了状態になる。その解析途中経過リストを後方に伝播し、伝播されたノード「前」での解析途中経過リストは  $[(2:\{ABCDE\})]$  となる。ここでノード「前」における確定処理を行ない、ノード「前」時点で受理可能な文テンプレートの集合を計算している。

図 6 は図 5 中のノード「サービスエリア」における確定処理の詳細である。このノードの前方に接続しているノード「前」から伝播されてきた解析途中経過リスト  $[(2:\{ABDE\})]$  とノード「最後」からの解析途中経過リスト  $[(3:\{AD\})]$  との和集合を計算する。ただしノード「前」「最後」の順序がそれぞれ 2,3 番目と異なるためそれらの直後のノード「サービスエリア」の順序も 3,4 番目となり、解析途中経過リストの各々の順序毎に和集合を計算すると  $[(3:\{ABDE\}), (4:\{AD\})]$  を得る。そしてノード「サービスエリア」の未処理前ノードが空になったため、非終端記号「施設」に対応する文テンプレートハッシュ辞書の要素との積集合を計算する。この時もそれぞれの順序毎に対応するハッシュ辞書を引き、別個に積集合を計算する。結果としてノード「サービスエリア」時点で受理可能な文テンプレートとして  $[(3:\{B\}), (4:\{A\})]$  を得る。これを更に後続ノードである「どこ」に伝播する。このようにして最終的に終了ノードの解析途中経過リストから得られる文型 ID 集合 (ABE) が図 3 の文法を適用して受理可能な文型 ID の集合である。

受理可能な文型 ID の集合を取得すると、続いて対応するキーワード系列を探索する。探索対象となる文型 ID を A とし、図 5 の開始ノードから順に A が含まれるノードを探索する。開始ノードの後続ノード「出口」「駅」の中で解析途中経過リスト中順序=1 の要素に A が含まれるノードは「出口」であり、これをキーワード系列候補として記憶する。ノード「出口」の後続ノード「前」に

は解析途中経過リスト中順序=2の要素に  $A$  が存在するため、この時点の系列は『出口 前』となる。同様の処理を終了ノードまで繰り返し、文テンプレート  $A$  に該当するキーワード系列は『出口 前 最後 サービスエリア どこ』となる。文型 ID  $B, E$  についても同様の処理を実行し、対応するキーワード系列として『出口 前 サービスエリア どこ』『出口 前 インター どこ』が追加される。

### 3.3 階層化文法への対応

BTH に適用される文テンプレートは非終端記号系列の形式で表されなければならない。この形式では、同じ意味をなす異なる部分系列が存在する場合、文テンプレートが指す意味が同じであってもそれぞれ別の文テンプレートを定義する必要がある。このために文法管理・保持のコストが増大する可能性がある。

我々は部分系列を一つのグループとして定義<sup>\*1</sup> し、そのグループの ID を一つの非終端記号として使用する階層化文法に BTH を対応させた<sup>\*2</sup>。これにより各グループに対する文テンプレートハッシュ辞書が構成されるが、グループ間の依存関係の順に BTH を複数回実行させることにより階層化された文法に対しても構文解析が可能となっている。

### 3.4 BTH の計算量

キーワードラティス上で開始ノードから終了ノードに至る道がたどるキーワードの系列は、ユーザが発話した文の候補(文候補)に対応している。本研究において構文解析とは文候補から受理可能な候補のみを抽出することである。ラティスが包含する個々の文候補に対して受理可能性を確認する構文解析方式を音声認識結果に適用した例としては、TOMITA による拡張 LR パーザ [Tomita 86] に基づくもの [Niedermaier 92, 坪井 94, 中野 98] や、トップダウンパーザに基づく MIT の TINA [Senef 89] などがある。これらは受理可能性の不明なものを含めた個々の文候補について文法適合性を検証し枝刈りを行う方式である。しかし文候補数は音声認識されたキーワード数  $n$  の増加と共に指数関数的に増加し、ノード数  $n$  のラティスが包含する文候補数は  $O(a^n)$  ( $2 \geq a \geq 1$ ) となる<sup>\*3</sup>。個々の文候補に対して受理可能性を確認するこれらの方式では、枝刈りがあるとしてもその計算量は  $O(a^n)$  となる。このため実用的な実行時間で動作させるためには構文解析中にビームサーチなどで探索幅を限定し解析対象の文候補数を削減する必要がある。

\*1 ここで言う部分系列は句にあたるものだと考えることができる。例えば、カーナビゲーションタスクでは場所に関する様々な言い回しを「場所句」として定義できる。

\*2 BTH の文法は、その展開結果が有限個のテンプレート形式の文法と等価にならなければならないため、グループ間の依存関係が再帰している文法には対応していない。

\*3 ラティスが包含する文候補数のオーダーについては付録 A を参照のこと

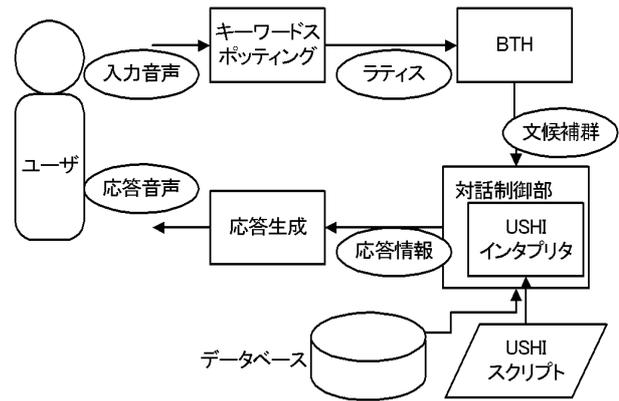


図 7 MINOS の構成

一方 BTH はキーワード間の辺の数に比例する回数の集合演算で、そのラティスで受理可能な全ての文テンプレートの情報を算出できる。 $n$  個のキーワードから成るラティスを構成する辺の数は  $O(n^2)$  であるため、BTH の計算量は  $O(n^2)$  にとどまる。その後受理可能な単語系列を生成する処理は受理可能な文テンプレートを指定したうえでラティス上をスキャンするだけなので、その計算量は  $O(n)$  となる。このため BTH では構文解析中に探索幅を限定しなくとも実用的な実行時間で構文解析することが可能であると言える。

## 4. 実験

### 4.1 MINOS:BTH を利用した対話システム

我々は構文解析器に BTH を使用した対話システムとして、カーナビゲーション対話を扱う音声対話システム MINOS (Mobile Interactive NavigatiOn Speech-system) を試作した (図 8, 図 9)。MINOS は予め設定された経路上の交差点・サービスエリア・店舗などの位置についての質問と、2 地点間の距離や所要時間に関する質問の 2 種類に対して応答を行なう音声対話システムである。動作画面では地図<sup>\*4</sup>上にユーザが設定した経路と自転車位置が表示され、あらかじめ指定された緯度経度データの系列を再生することにより自転車位置移動をシミュレートしている (図 8)。この自転車位置に基づき利用者からの入力に対応する問題を解決する。

MINOS は図 7 のような構成であり、次のように動作している。ユーザ発話がキーワードスポッティングエンジンに入力され、キーワードスポッティングエンジンは入力されたユーザ発話に対するラティスを出力する。次に BTH パーザによってラティスから文候補の集合が抽出される。対話制御部は上位スコアの候補から問題解決を実行し、ヒューリスティックに基づき各候補をリスクアする。リスクアされたものから最上位の候補を応答情

\*4 地図ソフトは (株) アルプス社のプロアトラスシリーズを用いている。



図 8 MINOS 画面例



図 9 MINOS 使用の様子

報として選択され、回答文が生成される。MINOS ではシステムの動作を記述する汎用スクリプト USHI を利用してタスク依存である問題解決手続きを記述している [Sasajima 99]。回答文は TOS Drive TTS [Akamine 98] によって音声合成され、スピーカから出力される。以上の音声認識・問題解決・音声合成の全てが一台のノート PC (Pentium II 266MHz) 上に実装されている (図 9)。

#### 4.2 解釈速度計測実験

我々はこの MINOS を利用し、音声入力に対する解釈速度計測実験を行った。以下にその詳細について述べる。

MINOS における音声認識エンジンが対象とする語彙は全 305 単語である。その内訳は道路近辺の施設名称など一般的なものが約 150 単語、それ以外は設定されているルート上の固有名詞に関するものとなっている。

MINOS が受理する文法は以下の 2 種類からなる階層化文法である。

場所句 場所を表す表現 (文テンプレート数: 240, パープレキシティ: 33.87)

文 場所・区間について問い合わせる全体表現 (文テンプレート数: 89, パープレキシティ: 16.52)

表 1 平均キーワード数と平均構文解析時間

キーワード数	構文解析時間
25.4	25.1(msec)

表 2 BTH から出力された系列数

最大系列数	最小系列数	平均系列数
2238	0	64.40

「文」は「場所句」に依存しており、場所に関する表現の詳細として「場所句」を利用している。この文法は展開すると全部で 200 万個以上の文テンプレートに相当し、その中で最長の文テンプレートの系列長は 18 となっている。BTH は「場所句」について構文解析した後に「文」を対象とした構文解析を実行する。文法は倒置・省略などを考慮して作られており、より口語的な発話も受理可能なものとなっている。例えば次のような発話を受理することが可能である。

- コンビニは?
- 目的地の手前の駐車場がある食べる所はどこ?
- インターまでどれくらい?
- 吹田インターまでの最後のガソリンスタンドから目的地の手前の駐車場がある食べる所までどれくらい?

この実験の被験者は 3 名 (男性 2 名, 女性 1 名) で合計 149 発話を入力した。発話は文法に適合したものであり、対応する文テンプレートの系列長は 1 から 14 であった。この時の認識結果のキーワード数・ラティスに含まれる系列数 (以下包含系列数と表記)・BTH 構文解析時間を計測した。尚、包含系列数を計算する際には各キーワード間のリンクの有無をそれらの時間的な重なり・離れの大きさを条件としてしきい値処理することで決定した。具体的にはキーワード同士の重なりを 160 ミリ秒以下かつ各々のキーワード出現時間長の 20% 以下に、離れを 640 ミリ秒以下とする制約を与えた。

表 1 に認識結果として出力されたラティスに含まれるキーワード数の平均とその構文解析時間の平均を示す。発話した文の系列長が最大 14 であるので、湧き出し誤りによりキーワード数が増加していることがわかる。包含系列数については、その最大値は  $2.71 \times 10^{11}$  であり主に数百～数十万に分布している。このような条件においても BTH の平均構文解析時間は 25 ミリ秒と高速に動作している。また、BTH から構文解析の結果として出力された系列数について表 2 に示す。その最大値は 2238、平均は 64 となった。

BTH による構文解析時間と相関関係の強いパラメータはラティスが包含するキーワードの数と BTH から出力される解析結果の系列数である。表 3 はキーワード数、包含系列数、構文解析時間の最大値を記録した事例である。構文解析時間最大の事例については、BTH 出力系列数も最大となっている。いずれの条件もラティス規模は爆発的に増加しているが、BTH は全て 0.35 秒以内に構

表 3 最大値を記録した事例

パラメータ	単語数	包含系列数	出力系列数	解析時間
キーワード数	99	$1.68 \times 10^5$	537	261(msec)
包含系列数	72	$2.71 \times 10^{11}$	0	56(msec)
構文解析時間	52	$1.28 \times 10^8$	2238	331(msec)

単語数：入力ラティスが包含するキーワードの数

出力系列数：BTH が解析した結果受理された文の数

文解析を終了している．ラティスが包含する全ての系列に対して構文解析を行うという問題に他の方式を適用した場合，系列数の爆発が計算時間に影響すると考えられるが，BTH はラティスの規模によらず安定して高速に解析を行っていることがわかる．

## 5. 考 察

BTH が受理可能な文法は一般の文脈自由文法やネットワーク文法に対し無限に続けられる再帰的な規則の適用を有限回にするという限定を与えたものである．言い換えれば正規文法のサブクラスに属する文法を対象とするためその受理範囲は小さくなる．しかしながらこの制限は音声対話システムに対する入力という見方においては影響が少ないと考えられる．特に話言葉の発話では細かい条件設定などの長い文を一度に言わずに「さぼった」発話をする傾向がある [中川 98]．省略しない発話をする利用者であっても一度に発話できる内容は限られる．このような条件下では上記のような再帰的ルールを数多く適用する発話は少ない．MINOS に対する発話についても対象文法が無限の再帰を許さないことに起因する問題は少なく，システム使用時の受理能力は，対象のドメイン・タスクで良く発話される内容を十分に反映させた文法かどうかに関係すると思われる．BTH で使用する文法は言わば「実用的」な部分を対象とする限定を加えたものと言える．

しかし対話システムに対する入力に複雑な文がないという保証はなく，文テンプレート形式のような制限を与えることによって聞き落しが発生する場合もありうる．このような場合でも信頼性のある対話システムを構築するためには，例えば聞き取れた内容を利用者にフィードバックし，複数の発話の統合によって複雑な文と等価の処理が可能となる対話処理機構の整備が重要となる．

構文解析をよりロバストにするために，構文解析結果としてより正当性の高い候補を出力する方式が検討されており，例えば音響尤度のみを判定基準とせず文法の生起確率などの言語的尤度を利用する方式がある [Manning 99, Senef 89]．BTH の文テンプレート形式の文法に対しても，次のようにすれば同等の処理が可能となる．文テンプレート作成時に言語尤度に基づくスコア計算を行い，文テンプレート毎にそのスコアを保持したテーブルを BTH の構成に加える．BTH の構文解析結果であるキーワード系列の音響尤度に基づくスコアを計算した後に，文テンプレートに付与されたスコアとの総合スコアを計算する．これにより音響尤度と言語尤度とを加味し

たより正当性の高い順位付けで文候補を出力できる．

## 6. ま と め

本論文ではラティス規模によらず高速に動作するラティスパーザである文テンプレートハッシュ方式 (BTH) を提案した．BTH は有限個の文テンプレートからなる文法を対象とすることで，ラティスからキーワード系列への仮説を立てること無く受理可能な文法を探索することが可能となっている．このため，個々の系列への仮説を立てる構文解析方式では計算量がノード数の指数関数的に増加する系列数のオーダーになることに対して，BTH ではノード数の 2 乗のオーダーにまで削減した．

また，BTH を使用したカーナビ対話システム MINOS を試作した．MINOS は対象語彙数約 300，受理可能な文テンプレート数 200 万以上に相当する文法を対象とし，ノート PC (PentiumII 266MHz) 上で音声認識 - 問題解決 - 音声合成の全ての処理が動作している．この MINOS に対して音声入力実験を行ったところ，BTH は最大で系列数  $2.7 \times 10^{11}$  となるラティス群に対しても全て約 0.3 秒以内で構文解析を終了することを確認した．

今後は BTH を他タスクに適用し，有限個のテンプレートとなる文法制限の妥当性を確認する予定である．

## ◇ 参 考 文 献 ◇

- [Akamine 98] Akamine, M. and Kagoshima, T.: Analytic Generation of Synthesis Units by Closed Loop Training for Totally Speaker Driven Text to Speech System (TOS Drive TTS), *ICSLP'98*, pp. 1927-1930 (1998).
- [金沢 93] 金沢, 坪井, 竹林: 雑音下の連続音声からのキーワード検出, 信学論, Vol. J76-D-II, No. 3, pp. 427-435 (1993).
- [河原 00] 河原達也: こまできた音声認識技術, 情報処理, Vol. 41, No. 4, pp. 436-439 (2000).
- [Manning 99] Manning, C. D. and Schutze, H.: *Foundations of Statistical Natural Language Processing*, The MIT Press (1999).
- [中川 98] 中川聖一: 音声対話システム構築の課題, 日本音響学会誌, Vol. 54, No. 11, pp. 783-790 (1998).
- [中野 98] 中野, 宮崎, 平沢, 堂坂, 川端: 多重文脈を用いた逐次的な発話理解, 情報研報, Vol. 98-SLP-22, pp. 21-26 (1998).
- [Niedermaier 92] Niedermaier, G. T.: Linguistic Modelling in the Context of Oral Dialogue, *ICSLP '92*, pp. 635-638 (1992).
- [Sasajima 99] Sasajima, M., Yano, T., and Kono, Y.: EUROPA: A Generic Framework for Developing Spoken Dialogue Systems, *Eurospeech99*, pp. 1163-1166 (1999).
- [Senef 89] Senef, S.: TINA: A Probabilistic Syntactic Parser for Speech Understanding Systems, *ICASSP '89*, pp. 711-714 (1989).
- [竹林 94] 竹林洋一: 音声自由対話システム TOSBURG-II, 信学論, Vol. J77-D-II, No. 8, pp. 1417-1428 (1994).
- [田中 99] 田中穂積監修: 自然言語処理, 電子情報通信学会 (1999).
- [Tomita 86] Tomita, M.: An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition, *ICASSP 86, TOKYO*, pp. 1569-1572 (1986).
- [坪井 94] 坪井, 竹林, 橋本: キーワードラティスの LR 解析による自由発話理解, 情報処理学会論文誌, Vol. 38, No. 2, pp. 260-269 (1994).
- [屋野 98] 屋野, 笹島, 河野: 文テンプレートハッシュ方式によるラ

ティス解析方式と質問 - 応答システム, 情処研報, Vol. 98-HI-80, pp. 71-76 (1998).

〔担当委員：溝口理一郎〕

2001 年 6 月 20 日 受理

◇ 付 録 ◇

A. ラティスが包含する文候補数

ここではラティス (図 1) が持つ文候補の総数 (包含系列数) を概算する. ラティスは始端ノードを左端, 終端ノードを右端に配置し, 左から右に方向が限定された有向グラフ  $G$  としてみることが出来る. 2 章で述べたようにラティスのノード間の辺はそれらが配置されている時区間の条件で決定され, 包含系列数はこの辺の多さに依存する. ここでは  $G$  にある任意の 2 個のノード  $N_i$  から  $N_j (i > j)$  に辺が張られる平均確率を  $p (0 \leq p \leq 1)$  と仮定する. ラティスの包含系列数はこのような有向グラフ  $G$  が持つ道の総数を計算することで概算できる.

まず  $G$  のノード数が 2 個の場合  $G_2$  について検討する.  $G_2$  には二つのノード  $N_2$  (左側),  $N_1$  (右側) が配置され, それらが辺を張る確率が  $p$  である. この時  $G_2$  が持つ道の総数の期待値  $C_2$  は次のようになる.

$$C_2 = 1 \times p = p$$

ここに新たにノード  $N_3$  を  $N_2$  の左側\*5 に追加しノード数 3 個のグラフ  $G_3$  を作成する.  $N_3$  からは  $N_2, N_1$  への辺を張る可能性があり,  $N_3$  から辺を張った  $N_2$  からは  $C_2$  本の道がある. このため  $G_3$  が持つ道の総数の期待値  $C_3$  は次のようになる.

$$C_3 = p + pC_2 = p + p \sum_{i=2}^{3-1} C_i$$

このように左側にノードを追加する作業を繰り返し,  $k$  個目のノード  $N_k$  を追加したグラフ  $G_k$  が持つ道の総数の期待値  $C_k$  について次の式が成立したとする.

$$C_k = p + p \sum_{i=2}^{k-1} C_i$$

更に  $G_k$  にノード  $N_{k+1}$  を左側に追加しノード数が  $k+1$  個のグラフ  $G_{k+1}$  を作成する.  $N_{k+1}$  から辺を張ることの可能なノードは  $N_1 \dots N_k$  であり,  $N_1$  までの道の総数の期待値は  $p$  で, それ以外の各ノードからは  $C_i (i = 2 \dots k)$  本の道が出現する.  $N_1 \dots N_k$  の各々へ辺を張る確率を  $p$  であることを考慮すると  $C_{k+1}$  は次のようになる.

$$\begin{aligned} C_{k+1} &= p + pC_2 + \dots + pC_k \\ &= p + p \sum_{i=2}^k C_i \end{aligned}$$

よって数学的帰納法により  $n$  個のノードからなる上述のグラフ  $G_n$  が持つ道の総数の期待値は次の式となることが証明された.

$$C_n = p + p \sum_{i=2}^{n-1} C_i$$

ここでこの式を変形すると次のようになる.

$$\begin{aligned} C_n &= p + p \sum_{i=2}^{n-2} C_i + pC_{n-1} \\ &= C_{n-1} + pC_{n-1} = (1+p)C_{n-1} \\ &= (1+p)^{n-2} C_2 = p(1+p)^{n-2} \end{aligned}$$

\*5  $N_2$  と重なる位置 (湧き出し誤りに相当) でも良い.

故にグラフ  $G_n$  が持つ道の総数は  $O((1+p)^n)$  である.

ここで最も左にあるノード  $N_n$  を始端ノード, 最も右にあるノード  $N_1$  を終端ノードとして見れば,  $G_n$  は  $N_2 \dots N_{n-1}$  が音声認識結果の各キーワードとしたラティスと見ることが出来る. よってラティスの包含系列数も  $O((1+p)^n)$  であると言える.

著 者 紹 介



屋野 武秀

1994 年神戸大学工学部電子工学科卒業, 1996 年神戸大学自然科学研究科博士前期課程修了, 同年 (株) 東芝に入社. 同社関西研究所を経て現在同社研究開発センターマルチメディアラボラトリーに勤務. 音声対話理解の研究に従事. 電子情報通信学会会員.



笹島 宗彦 (正会員)

1992 年大阪大学基礎工学部情報工学科卒業. 1997 年同大学大学院基礎工学研究科博士後期課程修了. 博士 (工学). 同年 (株) 東芝入社. 同社関西研究所を経て現在同社研究開発センターマルチメディアラボラトリーに勤務. 物理システムのオントロジーに基づくモデル構築と推論, 音声対話研究に従事. 1994 年人工知能学会研究奨励賞, 1996 年人工知能学会設立 10 周年記念論文賞受賞. 情報処理学会会員.



河野 恭之 (正会員)

1989 年大阪大学基礎工学部情報工学科卒業. 1994 年同大学大学院基礎工学研究科博士後期課程修了. 博士 (工学). 同年 (株) 東芝入社. 同社関西研究所研究主務などを経て, 2000 年 4 月奈良先端科学技術大学院大学情報科学研究科助教授. 知的 CAI, マルチモーダル理解, 音声対話 HI, 知的インタフェースの研究に従事. 電子情報通信学会, 情報処理学会, ヒューマンインタフェース学会, ACM 各会員.