自 然 言 語 処 理 102－3
（1994．7．21）

形態素解析を用いた全文検索システムとその応用

増市博, 山浦富久美, 小山剛弘, 舘野昌一

富士ゼロックス(株)システムコミュニケーション研究所

{masuichi, yamaura, koyama, tateno}@rst.fujixerox.co.jp

電子文書の大容量化に伴い、必要な情報を即座に引き出す技術として全文検索の要求が高まりつつある。本報告では形態素解析を前提とする全文検索システムについて述べる。システム中の形態素解析から全文検索に至る処理モデルは、一貫して有限オートマトンで記述されている。有限オートマトンは、冗長性を取り除く理論的最適化手法が確立されており、情報を効率的に格納するための優れた表現方法であるといえる。本システムでは、形態素解析用データおよび全文検索用語彙データに有限オートマトンの最適化手法を適用し、さらに、独自に考案した直列記録フォーマットを用いることにより小容量化を図る。

Morphological Analysis Approach for Free Text Search System and Its Applications

Hiroshi Masuichi, Fukumi Yamaura, Takahiro Koyama and Masakazu Tateno

Systems and Communications Lab., Fuji Xerox Co. Ltd.,
Godo-cho 134, Hodogaya-ku, Yokohama-shi, 240 JAPAN

This paper describes the Free Text Search System on the basis of the morphological analysis. The feature of the system is that the finite automaton is used for its data moodel through the whole system. The finite automaton can preserve the data compactly, because the method to optimize its data structure was established. Based on this advantage of the finite automaton, the system can preserve both the morphological analysis data and the index data for Free Text Search in a small size with the same optimal method.

1 はじめに

電子文書の普及に伴い、電子化された大量の文書から必要な情報を即座に引き出す全文検索の要求が高まりつつある。今後、携帯電子情報機器等の小型システム上においても、全文検索機能を実現することが重要な技術課題となると思われる。このような背景に基づいて、本報告では、システムに必要となるデータの小容量化に重点をおいた全文検索システムについて述べる。

全文検索の実現方法には、Signature 法や凝縮テキストなど文書に含まれる文字によりインデックスを作成して該当する文書を絞り込む方法や、ニューラルネットに基づくパターンマッチによる方法、ハードウェアによる高速化手法などが存在する[1][2]。本全文検索システムでは、単語の活用形に依存することなく検索が可能な、形態素解析を前提とする手法を採用した。本システムの特徴は、形態素解析から全文検索に至る処理モデルが一貫して有限オートマトンで記述されている点である。この特徴により、システム中のデータ全般に対して統一的な取り扱いができるとともに小容量化が可能である。

以下本報告では、2 章において形態素解析用のデータの作成方法を、3 章において形態素解析手法および全文検索用インデックスファイルの作成方法を、それぞれ有限オートマトンを基本処理モデルとして説明する。4 章では、本システムの応用事例について述べる。

2 形態素解析用有限オートマトンの作成

2.1 形態素解析用データの有限オートマトン表現

本システムでは、形態素解析時に必要となる形態素データを有限オートマトンで表現する。有限オートマトンは、構造に含まれる情報の冗長性を取り除く理論的最適化手法が確立しているため[3]、優れた表現方法であるといえる。有限オートマトンは、状態の有限集合 Q、ラベルの有限集合 Σ、Q の元である初期状態 q_0、Q の部分集合である最終状態 F、Q×Σ から Q への写像である遷移関数 δ の 5 つ組で定式化される有向グラフである。以下有限オートマトンを図示する場合は、状態を丸印で表現し、ラベル付きの矢印(弧)によって遷移関数を表現する。また、初期状態は図中に明示し、最終状態は二重丸で示す。

有限オートマトンは、状態 q (∈ Q)で記号 a (∈ Σ)を読み取ると、次の状態 δ(q, a)に遷移する。この動作を初期状態 q_0 から始めて、記号列 w(∈ Σ*)の先頭記号から順に繰り返し、w の終端記号による遷移後の状態が最終状態 F の元となり得る場合、有限オートマトンは記号列 w を受理するという。

図 1 は、文節(記号列)「およんで」「およんだ」「およいで」「およいだ」「およばない」「およばず」「およがない」「およがず」を受理する有限オートマトンを図示したものである。

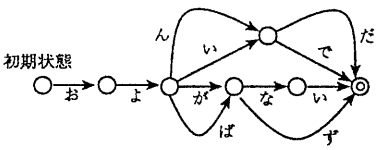


図 1 有限オートマトンの例

2.2 文節を受理対象とする有限オートマトン

有限オートマトンによって形態素データを表現する場合、有限オートマトンの受理対象として以下の選択肢が考えられる。

(a)1 語を受理対象とする
(b)1 文節を受理対象とする
(c)1 文を受理対象とする

それぞれすべての 1 語、すべての 1 文節、すべての 1 文を網羅する有限オートマトンを構成することが必要である。

1 語を受理対象とする場合、単語間の接続情報を有限オートマトンの外部に保存する必要がある。本研究では、システムに必要なデータを有限オートマトン表現に統一し、データの取り扱いの共通化を図るとともにデータの小容量化を目指す。したがって、単語間の接続情報は、有限オートマトン中に含めることとする。

1文を受理対象とする有限オートマトンは、1文節を受理対象とする有限オートマトンのすべての最終状態から初期状態へεをラベルとする弧を付加することによって構成可能である。ここでεは空入力を表し、入力記号なしに状態遷移を許す記号である。

本システムでは1文節を受理対象とする有限オートマトンを使用し、形態素解析に必要な情報をすべて有限オートマトン中に含める。

2.3 Two-level Morphology

ラベルに記号の対を持つことを許し、状態遷移中に一方の記号から他方の記号への変換を行なうことが可能な有限オートマトンを特にFST (Finite State Transducer)と呼ぶ[3]。記号対は「:」で区切り、左側のラベルを語彙層記号、右側のラベルを表層記号と呼ぶことにする。語彙層記号と表層記号が同一の場合は、単一のラベルでで表現する。FSTでは、入力記号列に対して表層記号の側に注目して状態遷移を行ない、その際に対応する語彙層記号を出力することによって、入力記号列を語彙層記号列に変換することが可能である。

本研究では、語彙層記号により形態素を表現する。活用のある語に関してはその終止形を形態素とみなして終止形を構成する文字を語彙層記号としてラベル化する。また、語幹とそれに接続可能な活用語尾を表層記号としてラベル化する。活用のない語に関しては、語を構成する文字そのものを単一のラベルとして表現する。このように、活用語に関しては終止形以外の形態素を認めず、語尾変化は語と語が接続する際に生ずる表層的変化とする立場に基づいた研究は、L. Karttunen, K. Koskenniemi, R. Kaplan らによって行なわれている[4]。この研究はTwo-level Morphologyと呼ばれ、Phonologyの知見を基礎として、主に欧米語を対象としている。

図2にTwo-level Morphologyに基づいた日本語FSTの例を示す。このFSTに対して、文節「およいで」を入力し、表層記号に注目して語彙層記号列に変換することにより、終止形で記述された形態素解析結果「およぐ/て」が得られる。

Two-level Morphologyに基づいたFSTの特徴として、形態素解析と形態素生成を同一の構造で実現できる点が挙げられる。図2では、「およぶ/て」を入力記号として、語彙層記号の側に注目して状態遷移を行ない、その際に対応する表層記号を出力することにより「およんで」が得られる。すなわち、形態素生成が実現できる。

図2では、図の簡略化のために、語彙層の形態素の切れ目を、状態に斜線を入れることによって示した。実際のFSTでは、「およぐ」に続いて、これが動詞であることを示す品詞情報を与えることにより形態素の切れ目を示す。表層記号がε、語彙層記号が「{動詞}」の記号対を用いた状態遷移によってこれを実現する。図3に品詞情報を出力するFSTの例を示す。表層がεの記号対で表現される付随情報は、品詞情報の他に、コストの情報（3.1節参照）や単語間の遷移情報がある。

なお、実際のFSTでは、ラベルに数ビット単位の記号を対応させている。日本語部分は2バイトコードであるため数回の状態遷移を伴うが、本報告書では簡略化のため1回の状態遷移で表現した。

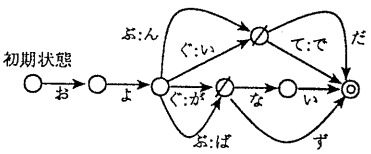
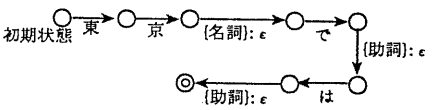


図2 Two-lebel Morphologyに基づいたFSTの例



表層	東京では
語彙層	東京{名詞}で{助詞}は{助詞}

図3 品詞情報を持ったFSTの例

2.4 音韻ルールとレキシコン

文節を表現するFSTを生成するためには、す

べての活用語の活用語尾とその終止形との対応関係を弧のラベル上に定義しなければならない。これを実現するために、文献[5]に基づいて、レキシコンおよび音韻ルールと呼ばれる2種の記述形式に分けて文節情報を定義し、これらの定義からFSTの自動生成を行なった。各記述形式の概略は以下の通りである[5]。

・レキシコン

接続に関して同種の特徴を持つ語をまとめて複数の語彙集合とし、各語彙集合が次にどの語彙集合に接続可能であるかを記述する。

・音韻ルール

音韻の不規則な変化を記述する。音韻変化が発生する箇所を、語彙層記号列および表層記号列の部分文字列を指定することによって特定し、語彙記号と表層記号の対応関係を定義する。

図4に、レキシコンおよび音韻ルールの記述例を示す。図4の記述から図2のFSTが自動生成される。なお図4では、付随情報の記述は省略している。

レキシコン

```
LEXICON ROOT
およぐ Verb;
およぶ Verb;

LEXICON Verb
て #;
た #;
ない #;
ず #;
END
```

音韻ルール

```
"Rule1"
  ぐ:が <=> お よ _ [ な い | ず ] ;

"Rule2"
  ぶ:ば <=> お よ _ [ な い | ず ] ;

"Rule3"
  ぐ:い <=> お よ _ [ て: | た: ] ;

"Rule4"
  ぶ:ん <=> お よ _ [ て: | た: ] ;

"Rule5"
  て:で <=> お よ [ ぐ: | ぶ: ] _ ;

"Rule6"
  た:だ <=> お よ [ ぐ: | ぶ: ] _ ;
```

図4 レキシコンおよび音韻ルール記述例

2.5 有限オートマトンの最適化[3]

レキシコンおよび音韻ルールから作成されるFSTは冗長な構造を含んでいるため、これを最適化する必要がある。文節を表現するFSTの最適化は主に、ε除去、決定化、最小化の3種の過程から成る。以下、それぞれの最適化過程について説明を行なう。

・ε除去

レキシコンおよび音韻ルールから作成されるFSTでは、ラベルがεの弧を多く含んでいる。これらの弧では、入力文字列の消費、語彙層記号の出力ともにないままに状態遷移を引き起こしてしまう。これは、形態素解析時の文節受理判断の効率の点で不利である。したがって、εラベルを持つ有限オートマトンと等価な(受理記号列集合が等しい)εラベルを持たない有限オートマトンへ変換する。ε除去は、状態qからラベルaによって遷移可能な状態集合($\delta(q, a)$)を、qから「ε*aε*」で表現される記号列の集合によって遷移可能な状態集合で置き換えることにより可能となる。図5にε除去の具体例を示す。ラベルがεの弧は除去されるが、他のラベルを持つ弧が増大することが分かる。

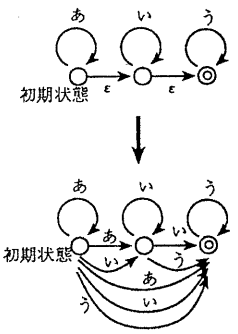


図5 ε除去の例

・決定化

ラベルの等しい弧を2本以上発する状態が存在する有限オートマトンを非決定性有限オートマトンと呼び、2本以上発する状態が存在しない場合を決定性有限オートマトンと呼ぶ。非決定性有限オートマトンでは、入力記号列に対して状態遷移を引き起こす弧が一意に決定できない。これは、文節受理判断の効率の点で不利である。したがって、非決定性有限オートマトンを等価な決定性有限オートマトンへ変換する。決定化は、ある入力記号により遷移可能な状態の集合を1つの状態として新たに付加することによって可能とな

る。文節を受理する有限オートマトンでは、決定化により非決定性を解消するとともに、弧の数を減少させる効果がみられた(図8参照)。図6に、図5において示したε除去後の非決定性有限オートマトンに対して、決定化を施した決定性有限オートマトンを示す。

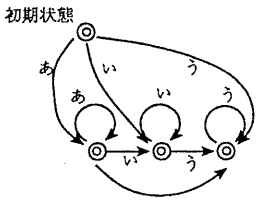


図6 決定化の例

・最小化

決定性有限オートマトンを、同値な(あらゆる入力記号列に対して受理の可否が等しい)状態の集合を一つの状態へと縮退することにより、等価で状態数が最小の決定性有限オートマトンを得ることができる。これは、有限オートマトンの小容量化のために重要である。図7に、図6において示した決定化後の有限オートマトンに対して、最小化を施した結果得られた有限オートマトンを示す。最小化により状態数が減っていることが分かる。

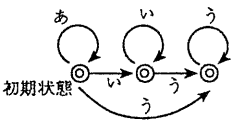


図7 最小化の例

図8に、形態素解析用データの一部(単語間接続情報)を表現する有限オートマトンの最適化過程での状態数、弧数およびε弧数の変化を示す。

	状態数	弧数	ε弧数
初期有限オートマトン	3654	7188	2631
ε除去後	3654	64729	0
決定化後	4917	9724	0
最小化後	1518	5020	0

図8 最適化過程の状態数、弧数の変化

2.6 有限オートマトンの記録フォーマット

本システムでは、小容量化のために、最適化を施した有限オートマトンを独自に考案した記録フォーマットで保存する。有限オートマトンの保存の際には、記録するポインタの数により容量が決まる。通常の記録フォーマットでは、弧から状態あるいは状態から弧を指定するために、有限オートマトンに含まれる弧の数に等しいポインタが必要となる。ここでは、以下の規則に基づいた記録フォーマットによって、保存に必要なポインタの数を減少させた。この記録フォーマットを直列記録フォーマットと呼ぶことにする。

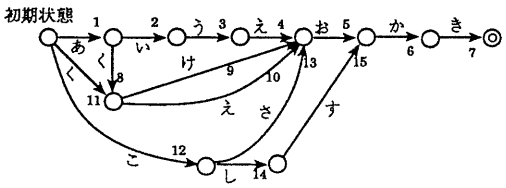
(1)初期状態を除き、弧とそれに続く状態を記録単位とする。一般に、各記録単位から発する記録単位(子記録単位)は複数存在する。これらの記録単位には順序付けがなされており、ある記録単位の次に順序付けられた記録単位をその記録単位の弟記録単位と呼ぶ。

(2)初期状態から発する記録単位から、深さ優先の順で記録単位を保存する。ただし、子記録単位がすでに保存されている場合、子記録単位へのポインタ(子ポインタ)を格納する。

(3)弟記録単位を持つ記録単位の直後に弟記録単位が保存されない場合、弟記録単位へのポインタ(弟ポインタ)を格納する。

図9に直列記録フォーマットによって有限オートマトンの保存を行なった例を示す。図中の有限オートマトンに振られた数字は記録単位の保存順序(深さ優先順)を表し、表中の数値に対応する。表の一行は一記録単位に対応している。「最終状態」の列は記録単位の状態が最終状態であるか否かを示す。「-」は記録の必要がないことを示す。

直列記録フォーマットでは格納すべきポインタが子ポインタと弟ポインタのみであるため、通常のフォーマットに比べて、有限オートマトンを小容量で保存可能である。



	最終状態	ラベル	弟ポインタ	子ポインタ
1	×	あ	11	-
2	×	い	8	-
3	×	う	-	-
4	×	え	-	-
5	×	お	-	-
6	×	か	-	-
7	○	き	-	-
8	×	く	-	-
9	×	け	-	4
10	×	-	-	4
11	×	-	-	8
12	×	こ	-	-
13	×	さ	-	4
14	×	し	-	-
15	×	す	-	5

図 9 直列記録フォーマット例

2.7 形態素解析用データ作成についてのまとめ

本章では、形態素解析用データの作成法について述べた。

(1) Two-level Morphology による有限オートマトンの生成
(2) 有限オートマトンの最適化
(3) 直列記録フォーマットによる有限オートマトンの保存

の各技術の連係によって、13 万語の辞書を含む文節表現有限オートマトンを、約 800 キロバイトの容量で保存することが可能であった。

3 全文検索用インデックスファイルの作成

3.1 コスト最小法による形態素解析

全文検索の検索対象となる自立語を切り出すために、検索対象文書に対して形態素解析を行なう。一般に形態素解析では、複数の文節区切り候補が存在する。本システムでは、候補の絞り込みを行なうための方法としてコスト最小法を採用した。コストは、形態素、形態素間の接続および文節のそれぞれに対して有限オートマトン中に埋め込まれている(2.4 節参照)。これらのコストの合計値を文節のコストとし、コストが最小の文節を解析結果として出力する。

図 10 に形態素解析精度の評価結果を示す。評価対象文書は、「現代用語の基礎知識」(自由国民社)から 500 文を抜き出したものである。システムの出力結果の形態素のうち、人手によって解析した結果と等しいものの割合を解析精度とした。

テキストジャンル	形態素数	解析精度(%)
原子力	2256	99.1
生物	3189	99.4
地震	3297	98.8
都市問題	2872	99.0
土地問題	4046	98.9
合計	15660	99.2

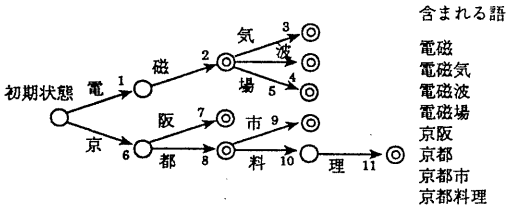
図 10 解析精度評価結果

3.2 有限オートマトンによる検索語彙記述

本システムでは、形態素解析処理によって形態素(自立語)を切り出す際に、形態素が存在する文書中の位置(形態素が含まれる章、段落の位置)を得て、インデックスファイルを作成する。インデックスファイルを用いることによって、検索対象である自立語から、その自立語が存在する文書中の位置を高速に検索することが可能となる。

検索対象となる文書中のすべての自立語を有限オートマトンとして表現し、決定化処理(2.5 節参照)を行なうことによって Trie 構造[6]を生成する。すなわち、決定化によって、検索対象となる自立語集合を木構造の有限オートマトンとし、各最終状態に対して文書中の位置情報を格納する(3.3 節参照)。

検索対象となる自立語集合は、有限オートマトンとして表現したため、2.6 節で述べた直列記録フォーマットによって保存可能である。図 11 に、直列記録フォーマットによる自立語集合の保存例を示す。ここで保存する有限オートマトンは木構造であるため、子ポインタ(2.6 節参照)を記録する必要はない。記録単位の状態が最終状態であれば、その記録単位には位置情報が設定される。



	最終状態	ラベル	弟ポインタ	位置情報
1	×	電	6	-
2	○	磁	-	X1
3	○	気	-	X2
4	○	波	-	X3
5	○	場	-	X4
6	×	京	-	-
7	○	阪	-	X5
8	○	都	-	X6
9	○	市	-	X7
10	×	料	-	-
11	○	理	-	X8
12	○	-	-	X9

図 11 語彙集合の直列記録フォーマット例

異なり語数	58,291
状態数	202,759
弧数	202,758
葉数	44,712
弟記録単位数(子記録単位のない記録単位の弟記録単位は含まない)	33,346

図 12 語彙集合有限オートマトン構成例

2.6 節でも述べたように、有限オートマトンの保存容量は、保存に必要なポインタの数に依存する。図 12 に、1,966 件の特許文書を対象として得られた語彙集合有限オートマトンの構成を示す。木構造の場合、通常の記録フォーマットでは、ポインタの数は弧の数から葉の数を引いた数に等しい。この例では 158,046 必要である。直列記録フォーマットでは、ポインタの数は弟記録単位の数と等しい。この例では 33,346 となり、通常のフォーマットの約 5 分の 1 に減少している。

3.3 インデックスファイルの間接参照記述

本システムでは、小容量化のために、検索対象となる自立語集合を表す有限オートマトンの最終状態に自立語の存在位置を直接格納せず、間接参照の構造をとった。図 13 は、検索構造の概略図である。図中の upos-table は、文書中の章、段落(unit)の位置(upos: unit position)を収めたテーブルである。uid-table は upos-table への間接参照テーブルである。uid-table に格納される値(uid: unit id)は、upos と 1 対 1 に対応する。間接参照とすることにより upos 値を重複して保存することを避けることができる。

実際の検索方法を図 11、13 を例に説明する。図 11 の有限オートマトンに対して「電磁場」が入力されたとする。入力文字列に従って、有限オートマトンを 1, 2, 5 の順に状態遷移し、位置情報 X4 を得る。その後、X4 の後続に存在する最初の位置情報 X5 を得る。X4, X5 は図 13 の offset1, offset2 に対応し、この値から uid-table, upos-table を介して文書中に「電磁場」が存在する位置を特定できる。

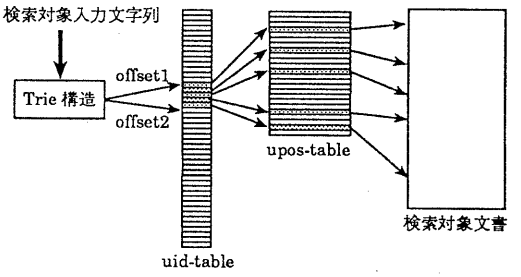


図 13 インデックスファイルの構造

3.4 インデックスファイル作成についてのまとめ

本章では、全文検索用インデックスファイルの作成法について述べた。

(1) コスト最小法による形態素解析
(2) 検索語集合の有限オートマトン化
(3) 有限オートマトンの最適化
(4) 直列記録フォーマットによる有限オートマトンの保存
(5) 間接参照記述による位置情報の保存

の各技術の連係によって、インデックスファイルの小容量化が可能となった。図 14 に、図 12 で扱った特許文書を対象として得られたインデックスファイルの容量を示した。インデックスファイルを元ファイルの約 10%の容量で保存可能であることが分かる。

(a)元文書容量	39,505[キロバイト]
(b)Tri ファイル容量	531[キロバイト]
(c)uid-table 容量	3,382[キロバイト]
(d)upos-table 容量	245[キロバイト]
(e)インデックスファイル容量 (=(b)+(c)+(d))	4,158[キロバイト]
(f)インデックスファイル比率 (=(e)/(a))	10.5[%]

図 14 インデックスファイル容量例

本節における(3), (4)の技術は、2.7 節の(2), (3)の技術と同一のものである。このように、システムのデータモデルとして有限オートマトンを一貫して使用することによって、表現内容の異なるデータに対して同一の手法を適用することが可能となった。

4 全文検索システムの応用事例

本全文検索システムは、特許文書および構造化文書エディタマニュアルを対象として動作を確認済みである。特許全文検索システムは、特許庁から配付される CD-ROM 公報の SGML 記述に従う特許文書を対象として全文検索を行なうシステムである。特許本文にはタグ付けが行なわれ構造化されているため、任意の書誌的事項内での検索が可能である。図 15 に特許全文検索システムの画面イメージを示す。

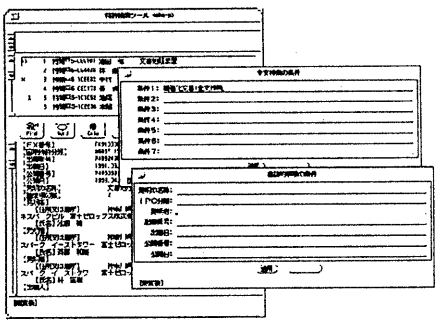


図 15 特許全文検索システムの画面イメージ

5 まとめと今後の課題

本報告では、形態素解析を用いた全文検索システムについて述べた。システムのデータモデルを有限オートマトンで統一することにより、形態素解析用データおよび検索用語彙データ双方に対して、有限オートマトンの最適化手法および直列記録フォーマットを共通に使用することができた。これにより、両データの小容量化を実現した。

今後の課題として、

(A) 形態解析用データの作成、保守の効率化

(B) 全文検索用語彙データの作成、保守の効率化

が挙げられる。いずれも、有限オートマトンの状態遷移アルゴリズムの改良、最適化アルゴリズムの改良により実現されるものである。したがって、有限オートマトンに関する単一のアルゴリズムの改善により、(A), (B)に対して同時に効率化の効果が期待できる。

謝辞

本システムの設計、実装に対して、適切なコメントおよびご協力を頂いた富士ゼロックスの関係各部門の方々に感謝致します。

参考文献

[1] 藤沢浩道, 絹川博之: 情報検索における自然言語処理, 情報処理, Vol. 34, No. 10, pp. 1259-1265 (1993).

[2] 根岸正光: フルテキスト・データベースの応用動向, 情報処理, Vol. 33, No. 4, pp. 413-420 (1992).

[3] J. ホップクロフト, J. ウルマン著(野崎他訳): オートマトン言語理論 計算論 I, サイエンス社 (1984)

[4] Koskenniemi, K.: A General Computational Model for Word-form Recognition and Production, *COLING 83*, pp. 178-181 (1983)

[5] Karttunen, K.: KIMMO: A General Morphological Processor, *Texas Linguistic Forum*, pp. 165-186 (1983)

[6] 青江順一: トライとその応用, 情報処理, Vol. 34, No. 2, pp. 224-251 (1993).