正規文法に基づく日本語形態素解析

丸山 宏† 荻野紫穂†

日本語の形態素解析は正規文法で行えるとされてきたが，今までの形態素解析システムでは，文法は接続表の形で記述されてきた．本論文では，文法を正規文法の生成規則として記述する形態素解析システムについて述べ，その利点と限界について議論する．

Japanese Morphological Analysis Based on Regular Grammar

Hiroshi Maruyama† and Shiho Ogino†

It is said that Japanese morphological analysis can normally be done by using a Regular Grammar. In many Japanese morphological analysis systems, grammar is described not in the form of a Regular Grammar, but in a matrix of the connectability between two words. We propose a Japanese morphological analysis system whose grammar is described in the production form of Regular Grammar. We also discuss its advantages and the limit of its descriptive power.

1. はじめに

日本語の形態素解析は，自然言語処理の分野では比較的成功している技術であり，多くのシステムが報告されている[7),15),17)-20),23),24)]．これらのシステムでは，文法は単語と単語（または単語のカテゴリとカテゴリ）の接続可能性を表す接続表の形で与えられている．

いくつかの報告では，これらの文法体系は正規文法（正規文法）と等価であることが指摘されていた[18),21)]．我々は，形態素解析の文法を，接続表の形ではなく，チョムスキーの正規文法の生成規則そのままの形で与える，形態素解析システムについて述べる．

接続表に比べて，生成規則の形で記述された文法は可読性と柔軟性に富み，文法のメインテナンスや特定分野向けのカスタマイズが容易である．また，正規文法としてきちんと定式化することによって，既知のアルゴリズムや定理などを適用することができ，システムの可能性と限界をより精密に把握することが可能となる．

本論文では，まず第2章で，形態素解析文法を正規の生成規則で書き下す方法について述べ，既存の，接続表による文法記述と比較する．第3章では，正規文法の解析に適用可能な既知のアルゴリズムについて考察し，我々が選んだ実現手法を紹介する．文法を正規で記述することによる理論的な限界については，第4章で議論する．

† 日本アイ・ビー・エム(株)東京基礎研究所
IBM Research, Tokyo Research Laboratory

2. 正規生成規則による形態素解析文法

正規文法は，チョムスキーの句構造文法のハイアラキーのうち，生成規則が，下記のいずれかの形をしているものである．

$$A \rightarrow aB \quad (1)$$

$$A \rightarrow B \quad (2)$$

$$A \rightarrow a \quad (3)$$

ここで，A，B は，非終端記号，a は終端記号を表す．

日本語の形態素の並びは，少数の例外を除いて，ほとんど直前の1語によってのみ規定されるので，正規文法，すなわち有限状態オートマトンでうまく近似できる．例えば，文節の先頭には名詞（例えば“魚”）が来うるが，それは，「文節頭」という状態と，「名詞」という状態との間の遷移として，

文節頭→“魚” 名詞

という形で生成規則として書き下すことができる（**図1**(a)）．

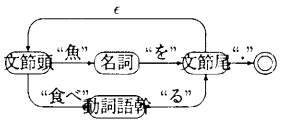
正規文法は，非決定性オートマトン（NFA）と等価であることが知られていて，非終端記号は等価なNFAの状態と読み変えることができる．例えば，生成規則

$A \rightarrow aB$

は，NFAが状態 A にいる時に，「文字 a という入力で，状態 B に遷移する」と読むことができ，したがっ

文節頭 → “魚” 名詞
文節頭 → “食べ” 動詞語幹
名詞 → “を” 文節尾
動詞語幹 → “る” 文節尾
文節尾 → 文節頭
文節尾 → “．”

(a)



(b)

図 1 (a) 正規文法による日本語形態素解析用文法 (b) そのオートマトン
Fig. 1 (a) A grammar for Japanese morphological analysis described in the form of Regular Grammar; (b) the automaton equivalent to (a).

て，図1(a)の文法は1(b)のような非決定性オートマトンと等価である．

このように，日本語形態素解析を正規文法として定式化する際，問題となるのは，2文字以上からなる語の扱いと，辞書項目の扱いである．それぞれについて我々の対応を説明する．

辞書項目の扱い

多くの自然言語処理システムでは，辞書と文法とを区別している．すなわち，「人」，「家」などの名詞をすべて NOUN という終端記号で代表させ，文法ルールの爆発を押さえている．これは，一般に，文法規則が大きくなると解析の効率が落ちるのに対して，辞書の検索は高速に行えるからである．

しかしながら，文法モデルの観点からは辞書と文法規則を区別する必要はない．辞書を文法規則と別の形でインプリメントするのはあくまでも実現上の都合である．したがって，我々は辞書と文法規則を区別しないことにする．例えば，すべての名詞について，

文節頭→“家” 名詞 (4)
文節頭→“人” 名詞 (5)
⋮ (6)

のように生成規則があるものと考える．記述の効率性を考え，このように全く同じ形をした大量の生成規則がある場合，我々のシステムではこれらを，一つの**マクロルール**，例えば

文節頭→NOUN 名詞 (7)

と，その NOUN を “家”，“人” のような個々の語に展開する**辞書データ**で，効率良く表現する．

既存の形態素解析システムでは，自立語は自立語辞書に入れることによって附属語と区別しているものがあるようだが（例えば文献15)），我々はこのような区別をしない．特殊な接続をする自立語もあるからである．あくまでも，同じ接続をするのかどうかという観点だけから語をまとめ，それを辞書に入れる．自立語/附属語の別などの，文法的な情報は，後で述べるルール属性として記述され，形態素解析結果とともに出力される．

2文字以上からなる語の扱い

日本語の形態素は必ずしも1文字でない．ほとんどの名詞（「国家」とか「計算機」）は，2文字以上からなる語である．これらの語を終端記号にしようとすると，正規文法に拡張を加えなくてはならなくなる．しかし，これらの語を複数の生成規則に分解すれば，正規文法そのままで定式化することが可能である．例えば，名詞 “国家” に関する生成規則

文節頭→“国家” 名詞 (8)

があったとしよう．我々は，これを

文節頭→“国” S0034 (9)
S0034→“家” 名詞 (10)

のように二つの生成規則の別記法と考える．ここで，S0034はこの文法のほかの部分に現れない非終端記号（＝状態）である*．

コストの付加

形態素解析の結果は一意でないために，発見的規則を用いて解の候補を絞ることが一般的である．このために，**最長一致法**，**文節数最小法**[23]，**コスト最小法**[16),24)] などのヒューリスティックスが用いられる．最小一致法と文節数最小法はコスト最小法の特殊な場合であることが指摘されている[7]．コスト最小法においても，形態素にコストをつける場合（例えば文献24)）と，遷移にコストをつける場合があり得る．一般性が高いのは，遷移にコストをつける方法である．したがって我々は，

カ行5段→“か” 動詞5段未然ナイ接続
cost＝300

のように，各生成規則にコストを割り当てる．このコストは，後に述べるグラフ探索時に使用されるのみであり，文法の生成力には影響を与えない．

2.1 既存の手法との比較

日本語形態素解析は伝統的に接続表の検定によって

* 解析システムの実現上は，これらの状態は，文法ルールの TRIE 木のインデックスとして表され，NFA の状態集合には陽には現れない．

行われてきた．接続表の例として，**図2**に，EDR の接続表を示す．各語は，左連接属性と右連接属性を持ち，表で許される（○のついている）左右の連接属性を持つ語同士が接続できる．このように接続表は，語による状態から状態への遷移と考えることができる．

我々の正規文法による文法の記述（**図3**）は，以下の3点において，接続表による方法より可読性と柔軟性に富む．

	JLNN1 名詞	JLNN2 固有名詞	JLNN3 人称名詞	JLNN4 サ変名詞	JLNN5 形式名詞	JLNN6 数詞	JLNN7 時詞	JLNN8 記号	JLV1 動詞	JLA1 形容詞	JLM1 形動	JLDD12 副詞	JLR1 連体詞	JLCC13 接続詞	JLT1 感動詞
JRN1 名詞	○	○	○	・	・	○	・	○	・	・	・	・	・	○	・
JRN2 固有名詞	○	○	・	・	・	○	・	○	・	・	・	・	・	○	・
JRN3 人称名詞	○	○	・	・	・	○	・	○	・	・	・	・	・	○	・
JRN4 サ名詞	○	○	○	・	・	○	・	○	・	・	・	・	・	○	・
JRN5 人称代名詞	・	・	・	・	・	○	・	○	・	・	・	・	・	○	・
JRN6 指示代名詞	・	・	・	・	・	○	・	○	・	・	・	・	・	○	・
JRN7 形式名詞	・	・	・	・	・	○	・	・	・	・	・	・	・	○	・
JRN8 数詞	○	・	○	・	・	○	○	○	○	○	○	・	・	○	・
JRN9 時詞	○	○	○	○	・	○	○	○	○	○	○	・	○	○	・
JRNA 記号	○	○	○	○	・	○	○	○	○	○	○	・	○	○	・
JRNB 単位	○	・	○	・	・	○	○	○	○	○	○	・	・	○	・
JRV1 一段語幹	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
その他の動詞の語幹	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・

図 2 EDR の接続表の一部
Fig. 2 A part of EDR's connectability matrix.

```
/*++++++++++ 動詞 ++++++++++*/
カ行5段 -> "か" [pos=4,kow=aux_v1,fe={causative}] サ行5段使役 cost=400;
カ行5段 -> "" [phrase=[type=not_yet,pre1_modified=yes,pre1_type=pred_obj]]
          "け" [pos=12,kow=aux_v1,fe={can}]       1段可能 cost=400;
カ行5段 -> "か" [pos=26,kow=v_infl]                5段未然ナイ cost=300;
カ行5段 -> "こ" [pos=27,kow=v_infl]                5段未然ウ cost=300;
カ行5段 -> "き" [pos=28,kow=v_infl]                5段連用マス cost=300;
カ行5段 -> "い" [pos=29,kow=v_infl]                5段連用テ cost=300;
カ行5段 -> "く" [pos=31,kow=v_infl]                5段終止 cost=300;
カ行5段 -> "け" [pos=32,kow=v_infl]                5段仮定 cost=300;
```

図 3 我々の文法記述の一部
Fig. 3 A part of our grammar description.

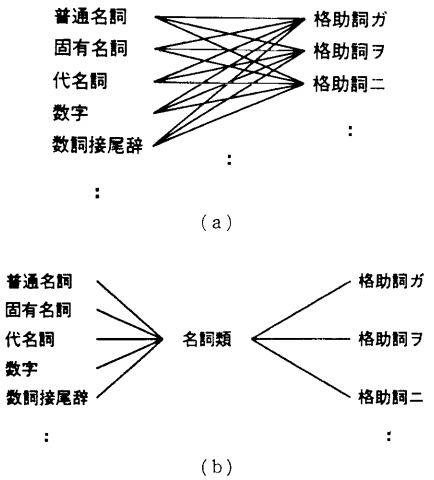


図 4 (a) 空遷移を使わない記述 (b) 使った記述
Fig. 4 Grammar description (a) without ε-transition; (b) with ε-transition.

空遷移の扱い

接続表の各連接属性はフラットな構造をもっており，互いに階層構造をなしていない．したがって，共通な性質を持つ属性（状態）をグループ化することは困難である．

例えば，普通名詞や代名詞などの名詞類は皆，ガやヲなどの格助詞を後方に伴うことができる．ところが，これらの右連接属性はそれぞれ別のものであるので，「名詞類は格助詞に接続する」という一般的なルールを記述するには，普通名詞や代名詞などのカテゴリから個々の格助詞への遷移を，すべて独立の遷移として記述しなければいけない（**図4**(a)）．本質的に一つである規則が，類似した複数の規則（または接続表のエントリ）に分散されて記述されるため，可読性とメインテナンス性が良くない．

我々の記述方式では，空遷移（右辺に終端記号のない生成規則）を使って，

普通名詞→名詞類
代名詞　→名詞類
⋮　　→名詞類
名詞類　→“が” 格助詞ガ

のように，普通名詞や代名詞などの後方接続属性を，まず空遷移を使って名詞類という非終端記号（状態）にまとめてから，個々の格助詞への遷移を書くことができる（図4(b)）．記述は単純であり，可読性もよい．

多重遷移の扱い

非常に緊密な関係にある複数語間の遷移を記述する場合，接続表はそれらを1語として扱うことがある．例えば「に関する」という連なりは，本来「〜に/関/する」と分けて解析されるべきだが，そのためには，「に/関」と「関/する」という二つの特別な接続用項目を作成しなければならず，表が不必要に大きくなってしまうためである．

```
名詞      →  "に"
             "関"    サ変語尾
サ変語尾  →  "し"    サ変連用
サ変語尾  →  "する"  サ変連体
```

図 5 多重遷移を使った記述
Fig. 5 Grammar description with multiple transitions.

我々の方法では，こういった遷移を，**図 5** のように，多重遷移として記述することができる．これは，先に述べた 2 文字以上からなる語の扱いと同様に処理することができ，正規文法そのままで，定式化できる．

文法情報の付加

正規文法の解析の出力は，形式的には遷移の並びである．単語の切れ目に関する情報は，この遷移の並びを解析することによって得られる．しかしながら，形態素解析の目的は，単に単語の切れ目を求めることだけではない．例えば機械翻訳などにおいては，より高度な言語学的な情報も同時に得られることが望ましい．

このため，我々の文法記述では，各遷移に素性構造を付加しておき，これを出力情報に含めることができる．図 3 の例では，pos=… の部分で，形態素連接属性を，kow=… の部分で，その語の品詞情報を，fe=… の部分でその語の構文的な情報を表している．また，phrase=[…] は，この遷移が文節の始まりであり，その文節の構文的な素性は […] の中に与えられることを示す*．

これらの素性は解析のパスには影響を与えないが，形態素解析の出力時に出力の遷移列に付加されて出力される．**図 6** に出力の例を示す．ここでは，各行が一つの遷移に対応しており，source=… が辞書情報を，s=… がこの遷移の出発点である状態名を，str=… がこの遷移でカバーされる表層の文字列を表す．str=… がない遷移は，空遷移である．

```
[source="f",s=文頭];
[fs=[phrase=[type=not_yet]],source="f",s=文節頭];
[str="今日",fs=[pos=102,kow=fukushiteki_meishi],source="system",s=副詞的名詞];
[source="f",s=文節尾];
[source="f",s=読点付文節尾];
[str="、",fs=[pos=100,kow=touten],source="f",s=文節末];
[fs=[phrase=[type=not_yet]],source="f",s=文節頭];
[str="学校",fs=[pos=19,kow=meishi],source="system",s=中名詞];
[source="f",s=名詞];
[source="f",s=体言類];
[str="に",fs=[pos=78,kow=kaku_joshi],source="f",s=格助詞ニ];
[fs=[phrase=[type=not_yet]],source="f",s=格助詞ニ付動詞];
[str="行",fs=[pos=2,kow=doushi],source="f",s=カ行５段特殊];
[str="っ",fs=[pos=29,kow=v_infl],source="f",s=５段連用テ];
```

図 6 形態素解析の出力例
Fig. 6 An output of morphological analysis.

この出力列から，fs=[phrase=…] のつく遷移を文節の区切りとみなし，fs=[pos=…] のつく遷移を単語の区切りとみなすと，通常の形態素解析の出力結果「今日/，（文節区切り）学校/に/行/っ/…」が得られる．

このように，各々の遷移ごとに属性を記述できるため，辞書で 1 語として扱われている語に対して，前方に接続する語に応じて複数の遷移を用意し，それらに別々の情報を付加するなどの処理が可能である．これによって，異なる文法的情報を持つ付属語の情報の制御などを，容易に行うことができる．

また，語の単位と遷移の単位が必ずしも一致していなくてもよいため，例えば複数の遷移で解析される数字列を一つにまとめて 1 語として取り出すことができる．これは特に，数字列，英字列，あるいは部品名など特別なフォーマットを持つ記号列の解析に特に有効である．

形態素解析の文法は決してユニバーサルなものではない．解析したい特定の文章の性質に応じて文法をチューニングすることは，より高い精度を求める上で必須のことである．端的な例では，特定のフォーマットに従った文字列（たとえば，*XXX-XXXX* という英数字列）が部品番号であるような文章に対しては，その部品番号を認識する遷移列を組み込むことで，全体の精度の向上を図ることができる．このような，状態数の増える拡張は，接続表ではやりにくいであろう．

また，解析対象が，敬語を含む/含まない，口語を含む/含まないなどの情報があらかじめわかっていれば，それなりのチューニングが可能で，対象文章に応じたきめの細かい文法を用意することができる．このように，形態素解析文法においては，可読性と柔軟性が非常に重要であると我々は考える．

3. 正規文法の解析アルゴリズム

文法を正規文法として定式化するもう一つのメリットは，正規文法に対してよく知られているアルゴリズムがただちに適用可能なことである．

---

* 空遷移に情報を付加することもできる．この場合は，終端記号として空語 " " を記述する．

正規文法について一般に知られたアルゴリズムは，認識アルゴリズムであり，解析アルゴリズムではない．認識アルゴリズムを解析アルゴリズムに拡張するのは，認識アルゴリズムのトレースをグラフの形で保存しておき，そのグラフから解析結果を生成するのが一般的である．

3.1 非決定性オートマトンのシミュレーション[2]

正規文法は，非決定性のオートマトンに変換することができ，その認識の手間は $O(nM)$ である．ただし，n は文の長さ（文字数），M は，非決定性オートマトンのサイズである．これは，文法のルール数にオーダ的に等しい．

非決定性オートマトンをシミュレートするには，各入力文字位置について状態集合を保持し，各遷移について，入力文字による遷移の行き先をこの状態集合に入れていけばよい．この際，遷移先の状態がすでに状態集合に入っているかどうかを $O(1)$ で調べる必要がある．これには，スタックとビットベクトルを用いればよい[1]．

非決定性オートマトンは理論的には決定性オートマトンに変換することが可能である．しかし，その大きさがもとの非決定性オートマトンのサイズの指数関数の大きさになることがあるため，自然言語の解析には実用的ではない*．

3.2 トレリスの探索

非決定性オートマトンのシミュレーションの結果は，**トレリス**と呼ばれるデータ構造で表現される．これは，すべてのパス（全部で入力の長さの指数関数個になり得る）をパックした表現であり，**グラフ構造スタック**[18]などと等価である．

形態素解析システムとしては，この中から一つまたは少数個のパスを解析の候補として出力しなければならない．遷移にコストが割り振られているコスト最小法の場合，このトレリスは，エッジにコストのついた有向グラフであるから，有向グラフの探索アルゴリズムを用いることができる．よく知られているのはダイクストラの方法であり，このアルゴリズムは，各エッジに非負のコストがつけられたグラフについて，$O(E+N\log N)$ で最適パスを求めることができる[5]（Eはグラフのエッジ数，Nはグラフのノード数）．トレリスが一般には非巡回グラフであることを利用すれば，最短パスの探索は $O(E)$ で終わることが知られている[3]．

通常，トレリスのノード数N，エッジ数Eは，文の長さnに比例すると考えることができるから，トレリスの探索にかかる手間は，文の長さに比例する．

グラフの探索に，A* などの発見的探索を用いる試みもある[19]．しかし，これら動的計画法を用いない探索法は，最悪の場合は文の長さに関して指数関数の手間がかかるため，一般的に高速であるとはいえない．もちろん，コスト関数の割り当て方によっては，非常に高速になる可能性もある．

3.3 2位以下の解候補の生成

形態素解析は常に正解を出すとは限らないので，2位，3位の解候補が必要になる場合がある．グラフの最短パスを短いものからk個取り出す問題は，k最短経路問題と呼ばれ，Dreyfus のアルゴリズムがよく知られている[4]．我々は形態素解析のために，別のk最短経路アルゴリズムを開発した[9]．このアルゴリズムはある条件の下で，Dreyfus のアルゴリズムよりも小さい計算量を持つ．Hisamitsu ら[7]は非巡回グラフのk最短経路アルゴリズムを示したが，このアルゴリズムよりも，非巡回グラフにおける Dreyfus のアルゴリズムのほうが，時間計算量に関して高速である．

3.4 コストの学習

すべての非終端記号Xについても，もし，Xを左辺に持つルールのコストの和が1であれば，それは**確率的正規文法**である[6]．確率的正規文法に対しては，ある学習データに対して，その学習データの出現確率を極大にするような反復計算アルゴリズムが知られていて大量のデータからの学習が可能である[22]．これも正規文法として形態素解析を定式化するメリットの一つといえる．

4. 正規文法の限界

括弧の対応のような埋め込みを扱う文法は，正規文法では記述できないのはよく知られている．我々の文法開発においても，いくどか，正規文法をはずれた記述をしたいことがあった．

現実問題としては，括弧が無制限に入れ子になることはありそうもない．したがって，括弧の入れ子の数を制限して，生成力を正規文法のそれに合わせることも考えられる．例えば，入れ子を三重に制限するとすれば，すべての状態 s について，s_0, s_1, s_2, s_3 という新たな状態を作り，それぞれ，自分の左側に閉じられ

* 入力ストリングが与えられた時に，必要な部分だけ決定性オートマトンに動的に展開して処理を行うアルゴリズムも提案されている[1]．このアルゴリズムを日本語形態素解析に応用することも今後の課題として考えられるだろう．

ていない括弧が何個あったかを記憶しておけば良い．

ところが，括弧は丸括弧だけではない．また，覚えておきたい情報は括弧の状態だけではない．例えば，「決して…でない」などの係り結びを形態素解析文法で扱おうとすると同様の問題を生じる．これらをすべて扱おうとすると，形態数および遷移数がこれら「覚えておきたい状態」の種類の指数関数で増加する．したがって，このようなことを行うと，状態数と遷移数の爆発を招くので我々は得策ではないと考える．

素性の束で状態を表現することは一つの可能性である．この方法を実現するのは，状態集合に新たに状態を追加する際の，状態の等価性の評価の問題を解決しなければならない．

そもそも形態素解析を初めから文脈自由文法で行う試みもあり[11]，効率の面での問題さえなければ，そのほうが良いであろう．これは，今後の日本語形態素解析の進むべき方向の一つと我々は考える．

5. 日本語形態素解析の評価

エラーの数え方

日本語の形態素解析には，2種類のエラーがある．一つは，「北-大西洋」を「北大-西洋」と切ってしまうような，分割誤りである．

もう一つは，誤った品詞を付与してしまう，品詞誤りである．品詞誤りのレートは，品詞をどのくらい細分化しているかに依存する．現時点で我々の形態素解析は48個の自立語品詞と，71個の附属語品詞（活用語尾も含む）を持つ．**図7**は，我々の形態素解析システムでの，それぞれの細分化の目安を示す．

我々は，（正解の）全形態素数に対するエラーの数を数えるが，その前に，もう少し定義を厳密にしなければならない．ある原因（例えば未知語）が複数の形態素に影響するかもしれないからである．我々は，正解であるべき個々の形態素についても，もし，それと同じものが解析結果になければ，エラーが一つと数える．

自立語				
名詞	動詞	形容詞／形容動詞	副詞	その他
13	17	4	6	10

附属語				
活用語尾	助詞	助動詞語幹	その他	合計
41	24	4	2	71

図7 品詞体系
Fig. 7 POS system.

表1 形態素解析の誤り率
Table 1 Error rate of morphological analysis.

カテゴリ	エラー出現回数	エラー率
分割誤り	364	1.25%
品詞誤り	323	1.11%
total	687	2.36%

例えば，「AABBCC」が，本来，「AA-BB-CC」と3語に解析されるべきだったとする．もし，「BB」が辞書になかったとすると，「AAB-BCC」と解析されるかもしれない．この場合，エラーの原因は一つであるが，本来解析されるべき「AA」，「BB」，「CC」という三つの形態素がどれも正しく得られていないので我々は，エラーが3回出現したと数える．

新聞記事からとった1,016文（29,024語）について，我々の形態素解析の精度を**表1**に示す．このエラー出現回数には，258個の未登録語（辞書に見出しはあったが，必要な品詞が登録されていなかったものも含む）に起因するものも含まれている．

形態素解析の精度についてはいくつか報告されている数字があるが*，テストの対象文の選び方や，エラーの数え方がそれぞれ異なり，あるいは精密な条件/基準が報告されていないために簡単に比較することはできない．共通なコーパスの上で，公開された基準で比較できる環境が整えられることが望まれる．

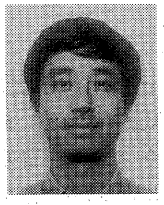
6. おわりに

日本語はわかち書きをしない言語であるため，単語を切り出す形態素解析のプログラムはさまざまな自然言語処理の基本となる技術であり，我々の形態素解析プログラムも，機械翻訳[8),12)]だけでなく，情報検索，校正支援[13)]，OCR後処理[14)]などに用いられている．また，言語データの収集[10)]にも必要不可欠な技術であり，現在我々は，この形態素解析システムを用いて大量の言語データの解析を行っている．形態素解析システムとしては，今後は，これらの多くの応用にきめ細かに対応できるシステム作りが求められていくであろう．

* 文献19)では，平均38.5文字の280文について，「文全体の解析に成功した」精度について，60.8%と報告している．また，文献15)では，形態素分割の精度が，98.4%，品詞割当を含めた精度が98.2%と報告している．

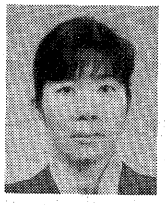
参考文献

1) Aho, A. V., Sethi, R. and Ullman, J. D.: *Data Structures and Algorithms*, Addison-Wesley, Reading, Massachusetts, pp. 163-169 (1983).
2) Aho, A. V., Sethi, R. and Ullman, J. D.: *Compilers—Principles, Techniques, and Tools*, Addison-Wesley, Reading, Massachusetts, pp. 126-127 (1986).
3) Cormen, T. H., Leiserson, C. E. and Rivest, R. L.: *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts (1990).
4) Dreyfus, S. E.: An Appraisal of Some Shortest-Path Algorithms, *Operations Research*, Vol. 17, pp. 395-412 (1969).
5) Fredman, M. L. and Tarjan, R. E.: Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms, *J. ACM*, Vol. 34, pp. 596-615 (1987).
6) Fu, K. S.: *Syntactic Methods in Pattern Recognition*, Academic Press, New York (1974).
7) Hisamitsu, T. and Nitta, Y.: A Uniform Treatment of Heuristic Methods for Morphological Analysis of Written Japanese, *Proc. of 2nd Japan-Australia Joint Workshop on NLP*, pp. 46-57 (1991).
8) Maruyama, N., Morohashi, M., Umeda, S. and Sumita, E.: A Japanese Sentence Analyzer, *IBM Journal of Research and Development*, Vol. 32, No. 2, pp. 238-250 (1988).
9) Maruyama, H.: A New *k*-th Shortest Path Algorithm, *IEICE Trans. Inf. & Syst.*, Vol. E 76-D, No. 3, pp. 388-389 (1993).
10) Maruyama, H., Ogino, S. and Hidano, M.: The Mega-Word Tagged Corpus Project, *Proc. of 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, Kyoto, pp. 15-23 (1993).
11) 相澤，徳永，田中：一般化 LR 法を用いた形態素解析と統語解析の統合，電子情報通信学会技術研究報告，NLC 93-2, pp. 9-16 (1993).
12) 荻野，丸山：日英機械翻訳システム JETS における日本語解析，情報処理学会自然言語処理研究会報告，84-17, pp. 127-134 (1991).
13) 奥村，脇田，金子：日本語校正支援システム FleCS の新聞社における実用化，情報処理学会自然言語処理研究会報告，91-5, pp. 33-40 (1992).
14) 伊藤，丸山：OCR 入力された日本語文の誤り検出と自動訂正，情報処理学会論文誌，Vol. 33, No. 5, pp. 664-670 (1992).
15) 木谷：固有名詞の特定機能を有する形態素解析処理，情報処理学会自然言語処理研究会報告，90-10, pp. 73-80 (1992).
16) 小松，安原：コスト最小法形態素解析のコストルールの作成方法，情報処理学会自然言語処理研究会資料 85-1, pp. 1-8 (1991).
17) 坂本：日本語形態素解析の基本設計，情報処理学会自然言語処理研究会資料，38-3 (1983).
18) 杉村，赤坂，久保：論理型形態素解析 LAX, *Proc. of Logic Programming Conference*, pp. 213-222 (1988).
19) 関根，菅野，長尾：形態素解析システムにおける新聞記事の調査とシステムの評価，情報処理学会自然言語処理研究会資料，77-1, pp. 1-8 (1990).
20) 中村，今永，吉田：接続コスト最小法による日本語形態素解析の評価実験，電子情報通信学会技術研究報告，NLC 91-1, pp. 1-8 (1991).
21) 日高：自然言語理解の基礎-形態論，情報処理，Vol. 30, No. 10, pp. 1169-1175 (1989).
22) 丸山，荻野，渡辺：確率的形態素解析，第5回ソフトウェア科学会全国大会予稿集，pp. 177-180 (1991).
23) 吉村，日高，吉田：文節数最小法を用いたべた書き日本語文の形態素解析，情報処理学会論文誌，Vol. 24, No. 1, pp. 40-46 (1983).
24) 吉村，武内，津田，首藤：未登録語を含む日本語文の形態素解析，情報処理学会論文誌，Vol. 30, No. 3, pp. 294-301 (1989).



丸山　宏（正会員）

1958 年生．1981 年東京工業大学理学部情報科学科卒業．1983年同大学大学院修士課程修了．同年日本アイ・ビー・エム(株)入社．東京基礎研究所に勤務．自然言語理解，論理型言語，日英機械翻訳の研究に従事．



荻野　紫穂（正会員）

1986 年3月東京女子大学文理学部日本文学科卒業．1988年3月同大学大学院文学研究科修士課程修了．同年日本アイ・ビー・エム(株)入社，東京基礎研究所に勤務．日英機械翻訳の研究に従事．計量国語学会，言語処理学会各会員．