

TR-I-0361

形態素生成処理解説書  
-ASURA における形態素生成処理-  
Morphological Synthesis in ASURA: Reference Manual

菊井 玄一郎      渡辺 学      関 倫彦  
Gen-ichiro Kikui   Manabu Watanabe   Michihiko Seki

1993.3

梗概

音声言語翻訳システム ASURA では生成処理全体を「句構造生成」と「形態素生成」とに分け、句構造生成においては屈折変化形の違いを捨象した語彙エントリによって生成を行ない、この句構造生成結果に対して別に用意された形態素生成処理によって屈折変化を行なっている。

本報告書では ASURA の形態素生成処理で用いる規則と辞書の記述形式について説明する。さらに、英語用の形態素生成規則・辞書についても説明する。

ATR 自動翻訳電話研究所  
ATR Interpreting Telephony Research Laboratories

©(株) ATR 自動翻訳電話研究所 1993  
©1992 ATR Interpreting Telephony Research Laboratories

# 目次

1	はじめに	1
2	概要	2
2.1	メイン処理	2
2.2	MG ネット	2
2.3	屈折規則 (MG-Rule)	3
2.4	形態素生成辞書	3
3	リファレンスマニュアル	5
3.1	MG ネット記述	5
3.1.1	MG ネットの記述規則	5
3.1.2	MG ネットのアクション	6
3.1.3	辞書検索	6
3.2	屈折規則	7
3.3	形態素生成辞書	8
3.3.1	記述形式	8
3.3.2	インデックスファイルの作成について	8
3.4	ユーザ I / F	9
3.4.1	MG ネット	9
3.4.2	MG ルール	9
3.4.3	形態素生成辞書	9
4	英語形態素生成について	12
4.1	MG ネット	12
4.2	英語屈折規則	12
4.3	英語形態素生成辞書	13
4.3.1	辞書に含まれる品詞と必要な活用情報	13
4.3.2	見出し	13
5	今後の課題	15
	参考文献	16
	Appendix 1. 英語 MG ネット フローチャート	

Appendix 2. 英語 MG ネットソースファイル

Appendix 3. 英語生成のための形態素処理 - 規則変化一覧表

# 第 1 章

## はじめに

ATR 自動翻訳電話研究所では、自動翻訳電話の実現可能性を確認するため、音声認識、言語翻訳、音声合成などの要素技術を組み合わせて実際に動作する音声言語翻訳システム ASURA (Advanced Speech Understanding and Rendering System of ATR) を構築した [1][2]。このシステムの言語翻訳部は解析・変換・生成の3つの部分から成るいわゆるトランスファー方式で構成されており、このうち生成処理部は単一化に基づく宣言的な言語知識記述を用いた処理を採用している [3]。

単一化に基づく句構造文法では語彙エントリとして屈折変化済みのものを用いることになっているが、この方法では記述量が増えるなど効率性の点で問題がある。そこで、ASURA では生成処理全体を「句構造生成」と「形態素生成」に分け、句構造生成においては屈折変化形の違いを捨象した語彙エントリによって生成を行ない、別に用意された形態素生成処理によって、この句構造生成結果に対する屈折処理を行なう。

本報告書では ASURA の形態素生成処理で用いる規則と辞書の記述形式について説明する。

以下では、まず、形態素生成処理の概要について述べ、次に規則や辞書の作成者のためのリファレンスマニュアルを示す。最後に、この ASURA の英語生成処理部で実装されている英語形態素生成規則等について解説する。

なお、ASURA ではこの形態素処理系を用いたドイツ語の形態素生成処理も実装されている。ドイツ語の形態素生成処理に関しては [4] を参照されたい。

## 第 2 章

### 概要

形態素生成処理の仕事は句構造生成の結果である「素性構造の形式で表現された統語構造」を入力として、この構造に対する適切な表層単語列を生成することである。

ここで実現している形態素生成処理は、句構造生成の段階で各葉節点に対して語彙エントリーや品詞などの統語的な素性が付与されていることを前提とし、各葉節点に対して表層単語を生成し、これをつないで表層単語列とする。ここで、各節点に対する表層単語の生成においては、原則としてこの節点とこれに隣接する節点の情報のみを用いる局所的な処理を行なう。

形態素生成処理はメイン処理、MG ネット、屈折変化処理、辞書引き処理から成っている。MG ネットはメイン処理から呼びだされ、屈折変化処理と辞書引き処理は MG ネット処理から呼び出される。

#### 2.1 メイン処理

メイン処理は次の通りである。

入力の句構造の葉節点のリストを作る

foreach 節点 in 葉節点リストのリバースリスト do

    節点に対して MG ネットを呼び出す

    処理中の語が冠詞 "a" の場合、必要であれば "an" に変える

end

先頭語の頭文字を大文字にする。

#### 2.2 MG ネット

各葉節点の保持する単語の変化形はその葉節点の素性構造 (と前後の節点の素性構造) から決定する。この処理を「MG ネット」と呼ばれる弁別ネット (decision net) によって記述する。

MG ネットの各ノードにはアクセスすべき素性パスとこのパスによって示される素性値に応じて選択すべき枝が記述されている。各枝には次に進むべきノードや実行すべき語形操作などが記述されている。この語形操作には、次に述べる屈折規則の呼び出しや形態素辞書を参照する屈折変化などが含まれる。

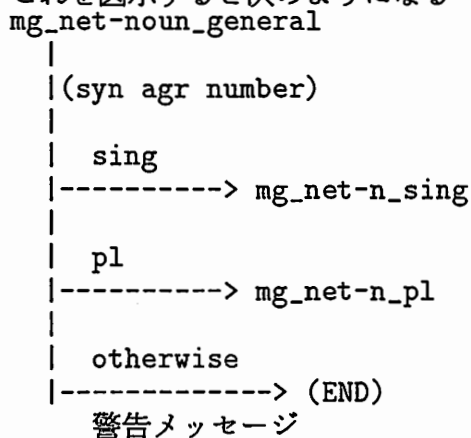
MG ネットの例を次に示す。

```

例 (def_mg_net
      :name    mg_net-noun_general
      :path    (syn agr number)
      :default (:warning "xxx")
      :body    ((sing    (:call mg_net-n_sing))
                 (pl      (:call mg_net-n_pl))
                 (otherwise (:warning "yyy"))))

```

これを図示すると次のようになる



ここで、:name はこのノードの名前である。:path がアクセスすべき素性値を得るためのパスであり、:body の部分のこの素性値に対応するアクションである。この例では、処理中の節点の (syn agr number) 素性に注目し、この素性値が sing であれば mg\_net-n\_sing、pl であれば mg\_net-n\_pl という名前のノードに遷移しそこでまた同様の処理を行なう、これ以外の素性値であれば警告メッセージ“yyy”を出力し他に遷移しないことを表している。:default は: path で指定されているパスの先に素性構造がない場合に行なうべき動作を指定する。

なお、パスの指定は処理対象としている節点の素性構造の根節点から行なうので、原則としてこの素性構造の内部の素性しかアクセスできないが、本処理系では処理対象としている節点の前後の節点の素性構造をアクセスできるようなパス記述法を与えている (詳しくはリファレンスマニュアルを参照のこと)。

### 2.3 屈折規則 (MG-Rule)

「屈折規則」という名前であるが、実際は語形を変化させるための文字列操作関数の定義である。たとえば「ED という名前の操作は、語尾に文字列“ed”を付加する操作である」といった定義を行なう。

ここで定義された「屈折規則」は MG ネットから直接適用されたり、MG ネットが次に述べる辞書記述を参照した結果適用されたりする。

### 2.4 形態素生成辞書

各語に対して適切な屈折を行なうための情報を定義する。辞書には屈折形の文字列か適用すべき屈折規則を記述する。屈折規則の定義にも依るが、たとえば、動詞“come”の過去形に関し

て文字列 "came"、動詞 "call" に関して規則名 ED などと記述する。

具体的な辞書記述は次のようになる。

```
("call"          VERB   (:ALL C S ED ED ING))
("come"         VERB   (:ALL C S "came" "come" NG))
```

内側のリストの第4要素が動詞の過去形に対応する記述である。

この辞書は MG ネットの「辞書を参照する語形変化」の実行によってアクセスされる。

```
(def_mg_net
 :name      mg_net-verb
 :path      (syn vform)
 :default   (:warning "xxx")
 :body      ((bse      (:dict :nadine91 :lex-cat 0)) ;; 原形
             (present (:call mg_net-v_present)) ;; 現在
             (past     (:dict :nadine91 :lex-cat 4)) ;; 過去
             (psp      (:dict :nadine91 :lex-cat 5)) ;; 過去分詞
             (prp      (:dict :nadine91 :lex-cat 6)) ;; ing
             (otherwise (:warning "yyy"))))
```

この例では (syn vform) 素性が、たとえば、past の時には、: nadine91 という名前の辞書に対して辞書引きを行ない、その辞書項目の語彙記述 (上の辞書記述における内側のリスト) の第4要素を用いて屈折変化を行なうべきことを示している。

## 第 3 章

### リファレンスマニュアル

#### 3.1 MG ネット記述

##### 3.1.1 MG ネットの記述規則

MG ネットを構成するノードは次の形式で記述します。

```
(def_mg_net
:name      ノードの名前
:path      素性パス
:default   デフォルトのアクション
:body      条件付きアクション)
```

ノードの名前	mg_net- で始まるシンボルで、ノードが支配するサブネットを一意に識別する名前を付けます。ただし、トップノードの名前は mg_net-main に固定します。
素性パス	ノードが参照する素性を素性パス (リスト) で指定します。素性パスは一つのノードについて必ず一通り指定します。処理対象の節点の前後の節点の素性を指定することができます。これを行なうためには特別のパス名:previous, :next を使います。:previous は直前の節点の素性構造全体に対するパスを表し、:next は直後の節点を表します。たとえば (:previous syn lex) は「直前の節点の (syn lex) 素性」を示す素性パスとなります。
デフォルトのアクション	指定された素性パスがなかった場合の動作を指定します。アクションとして指定可能な動作は全て記述可能です。また、複数のアクションを並べて記述する事が可能です。その場合、アクションは記述された順に実行されます。
条件付きアクション	( (条件1 アクションの並び) (条件2 アクションの並び)



.....  
(otherwise アクションの並び)

のように記述します。最初に条件部が満たされたアクションが実行されます。

条件部には、素性値かキーワード otherwise を記述します。素性値はアトムか (:set \*\*\*) か (:not \*\*\*) の何れかです。マッチする条件がない場合は、その形態素に対しては何も行いません。

### 3.1.2 MG ネットのアクション

アクションとして記述可能なものを以下に示します。

:nop	何もしません。
(:vowell <i>t/nil</i> )	発音が母音で始まるか否かを明示します。この情報は、冠詞 "a" の変化のために使われます。
(:warninig <i>string</i> )	<i>string</i> を警告メッセージとして表示します。 <i>string</i> が省略された場合は、デフォルトの警告のみ出力します。
:warninig	デフォルトの警告のみ出力します。
<i>string</i>	<i>string</i> を形態素として出力します。
:genitive	代名詞以外の名詞の所有格を作ります。
(:mgrule <i>rule</i> )	MGルール <i>rule</i> を適用します。
(:call <i>node</i> )	MGサブネット <i>node</i> を呼び出します。
(:dict <i>dict key position</i> )	<i>key</i> をキーとして <i>dict</i> で指定される形態素生成辞書を引き、辞書データ中の <i>position</i> で位置を指定されるカラムの情報に基づいて、屈折変化を求めます。

### 3.1.3 辞書検索

:dict のパラメータについてやや詳しく説明します。

<i>dict</i>	現在は使っていないパラメータですが、nadine91 と指定して下さい。将来、複数の辞書をサポートするようになった場合どの辞書を検索するかを指定するものです。
<i>key</i>	どの素性をキーとするかを指定するパラメータです。lex-cat か lex-cat-id の何れかが指定できます。
<i>position</i>	辞書データを使って何をするのか (3単現を求めるのか、過去完了形が欲しいのか) を、0 または 3 以上の数で指定します。この数は辞書情報中のカラムの位置を意味しています。ただし、0 を指定した場合は辞書を引いても屈折変化は行わず、発音情報だけが使われます。

### 3.2 屈折規則

屈折規則の一般形は次の通りです。

```
(def-mg-rule 規則名 削除文字数 "付加する文字列" 最後の文字列の重なり)
```

例えば dictionary を dictionaries に変化させるような規則 (操作)、すなわち、

語尾から1語削除して "ies" を付ける。

ような操作は

```
(def-mg-rule VES1 1 "ies" nil)
```

のように記述します。

### 3.3 形態素生成辞書

#### 3.3.1 記述形式

各辞書エントリーの記述形式は次の通りです。

(”単語の原形” 品詞 (語義 発音情報 活用情報))

品詞           ->NOUN, VERB, ADJ, ADV

語義           -> 語義によって活用形が異なる語以外は:ALL と記述  
                  例 "people"参照

発音情報       ->V(母音で始まる),C(子音で始まる)

活用情報       -> 活用規則名 (シンボル), または不規則変化形  
                  (文字列) の並び。

#### 3.3.2 インデックスファイルの作成について

今まで述べた記述形式によって記述された形態素辞書を形態素「辞書ソースファイル」と呼びます。実際の辞書引き処理では、このソースファイルを直接使うのではなく、このファイルをコンパイルしたファイルを使います。コンパイルの仕方は次節のユーザーインタフェースを参照して下さい。

### 3.4 ユーザ I / F

#### 3.4.1 MG ネット

(mg-load-net :file *file* :add *t/nil*)

MG ネットファイルを読み込みます。:file *file* を省略した場合は、デフォルトのファイルが読み込まれます。:add *t/nil* を省略した場合は、:add nil として動作します。

(mg-set-net *file*)

デフォルトのMG ネットファイルを設定します。 *file* を省略すると現在の設定値を表示します。

(mg-trace-net *t/nil*)

MG ネット実行のトレースを指示します。 *t/nil* を省略すると現在の設定値を表示します。

#### 3.4.2 MG ルール

(mg-load-rule :file *file* :add *t/nil*)

屈折規則ファイルを読み込みます。:file *file* を省略した場合は、デフォルトのファイルが読み込まれます。:add *t/nil* を省略した場合は、:add nil として動作します。

(mg-set-rule *file*)

デフォルトの屈折規則ファイルを設定します。 *file* を省略すると現在の設定値を表示します。

#### 3.4.3 形態素生成辞書

##### 辞書検索 / 更新

辞書を検索したり、ユーザ辞書を会話的に更新するための関数です。

(mg-use-u-dict *t/nil*)

ユーザ辞書を使うか否かを指定します。 *t/nil* を省略すると、現在の値を表示します。

(mg-start)

マスター辞書と（必要ならば）ユーザ辞書をオープンします。既にオープンされている場合は、一度クローズしてから再オープンします。辞書がオープンされるとプロンプトが "\$>" になります。

(mg-end *mode*)

マスター辞書と（オープンされていれば）ユーザ辞書をクローズします。ユーザ辞書に変更があれば実行時形式ユーザ辞書ファイルは書き替わります。 *mode* は元の実行時形式ユーザ辞書ファイルの扱いを指定します。:update なら新しいファイルのみが残ります。:backup なら元のファイルは拡張子を付加したファイル名にリネームされてバックアップファイルとして残ります。 *mode* が省略されたときのデフォルトは :backup です。

- (mg-quit) マスター辞書と（オープンされていれば）ユーザ辞書をクローズします。ユーザ辞書に対して行われた変更は無効になります。
- (mg-fetch *string*) *string* を見出し語とする辞書データをユーザ辞書・マスター辞書の順に検索して、画面に表示します。ユーザ辞書にデータがあればマスター辞書は検索しません。
- (mg-add *string pos data*) 見出し語 *string* 品詞 *pos* 形態素情報 *data* なる語彙をユーザ辞書に追加します。
- (mg-update *string pos data*) 見出し語 *string* 品詞 *pos* 形態素情報 *data* なる語彙でユーザ辞書を更新します。既にユーザ辞書にある語彙は「見出し語」と「品詞」をキーとして検索されます。もしあれば、既にある語彙は無効になります。
- (mg-del *string pos*) 見出し語 *string* 品詞 *pos* をキーとしてユーザ辞書を検索して、その語彙を無効にします。
- (mg-save) ユーザ辞書をファイルに出力します。

#### マスター辞書

- (mg-set-m-dict *file*) *file* をデフォルトのマスター辞書ファイルとして設定します。*file* を省略すると現在デフォルトとなっているファイル名を表示します。
- (mg-make-m-dict *file*) 辞書ソースファイルから実行時形式のマスター辞書ファイルを作成します。実行時形式辞書は辞書ソースファイル名に ".data" を付加したファイル名で作成されます。同時にインデックスファイルも作成されます。インデックスは、辞書ソースファイル名に ".index" を付加したファイル名で作られます。  
*file* を省略するとデフォルトの辞書ファイルを対象とします。
- (mg-make-m-source *file*) 実行時形式のマスター辞書ファイルからソース形式の辞書ファイルを作成します。*file* を省略するとデフォルトの辞書ファイルを対象とします。
- (mg-make-m-index *file*) 実行時形式のマスター辞書ファイルからインデックスファイルを作成します。*file* を省略するとデフォルトの辞書ファイルを対象とします。
- (mg-open-m-dict *file*) ファイル *file* をマスター辞書としてオープンし、プロンプトを "\$>" にします。既にマスター辞書がオープンされている場合は、それをクローズしてから改めてマスター辞書をオープンします。*file* を省略するとデフォルトの辞書ファイルをオープンします。使用中の辞書を切り替える場合などに、使って下さい。

- (mg-close-m-dict)            マスター辞書をクローズします。
- (mg-set-backup *t/nil*)        マスター辞書およびユーザ辞書に変更を加えた時に元のファイルをバックアップとして残すか否かを指定します。 *t/nil* を省略すると現在の設定値を表示します。
- (mg-set-backup-ext *string*)    バックアップファイルの拡張子を指定します。 ”.”で始まる文字列を指定して下さい。デフォルトは “.backup” です。 *string* を省略すると現在の設定値を表示します。

#### ユーザ辞書

ユーザ辞書をハンドリングするための関数です。機能や使い方はマスター辞書に対する関数と同じです。

- (mg-set-u-dict *file*)
- (mg-make-u-dict *file*)
- (mg-make-u-source *file*)
- (mg-make-u-index *file*)
- (mg-open-u-dict *file*)        使用中の辞書を切り替える場合などに、使ってください。
- (mg-close-u-dict)

## 第 4 章

### 英語形態素生成について

#### 4.1 MG ネット

APPENDIX1 の MG ネット図と APPENDIX2 の MG ネットのソースファイルを参照されたい。

#### 4.2 英語屈折規則

屈折規則ファイルは「生成のための形態素処理 - 規則変化一覧表」に基づいて、原形の末尾の削除文字数、文字を重ねるかどうか、付加する文字列を規則化したものとなっている。

以下に規則の一覧を示す。APPENDIX3 に例などの補足説明を添付する。

```
(def-mg-rule S 0 "s" nil)
(def-mg-rule ES 0 "es" nil)
(def-mg-rule IES 1 "ies" nil)
(def-mg-rule VES1 1 "ies" nil)
(def-mg-rule VES2 2 "ies" nil)
(def-mg-rule ED 0 "ed" nil)
(def-mg-rule D 0 "d" nil)
(def-mg-rule IED 1 "ied" nil)
(def-mg-rule KED 0 "ked" nil)
(def-mg-rule $$ED 0 "ed" t)
(def-mg-rule EN 0 "n" nil)
(def-mg-rule N 0 "n" nil)
(def-mg-rule $$EN 0 "en" t)
(def-mg-rule ING 0 "ing" nil)
(def-mg-rule NG 1 "ing" nil)
(def-mg-rule KING 0 "king" nil)
(def-mg-rule YING 2 "ying" nil)
(def-mg-rule $$ING 0 "ing" t)
(def-mg-rule ER 0 "er" nil)
(def-mg-rule R 0 "r" nil)
```

```
(def-mg-rule IER 1 "ier" nil)
(def-mg-rule $$ER 0 "er" t)
(def-mg-rule MORE 0 "-" nil)
(def-mg-rule EST 0 "est" nil)
(def-mg-rule ST 0 "st" nil)
(def-mg-rule IEST 1 "iest" nil)
(def-mg-rule $$EST 0 "est" t)
(def-mg-rule MOST 0 "-" nil)
(def-mg-rule ZERO 0 "-" nil)
```

### 4.3 英語形態素生成辞書

#### 4.3.1 辞書に含まれる品詞と必要な活用情報

(“単語の原形” 品詞 (語義 発音情報 活用情報))

品詞	辞書に記述されている活用情報
名詞	複数形
動詞	3人称単数現在形 過去形 過去分詞 ~ing 形
形容詞 / 副詞	比較級 最上級

#### 4.3.2 見出し

- 見出しは名詞・動詞・形容詞 / 副詞の順でファイルに記述した。
- 固有名詞は最初登録していなかったが、形態素処理の際に警告が出ることや、複数形がありうる語もあることから辞書に登録した。
- 名詞の "Higashi ku" "Kita ku"などは-(ハイフン)なしで統一。
- 副詞の分離複合見出しは、それ以上分けると品詞が決められなかったり、一般的な辞書でひとまとまりで見出しにしてあるものをそのまま登録した。

見出し語数 (atr\_made.dict8)

```
NOUN 1475
VERB 1092
ADJ 699
ADV 323
計 3589
```

```
例 ("abstract" NOUN (:ALL V S))
    ("activity" NOUN (:ALL V IES))
    ("address" NOUN (:ALL V ES))
    ("all" NOUN (:ALL V ZERO))
```



("analysis"	NOUN	(:ALL V "analyses"))
("call"	VERB	(:ALL C S ED ED ING))
("care"	VERB	(:ALL C S D D NG))
("carry"	VERB	(:ALL C IES IED IED ING))
("come"	VERB	(:ALL C S "came" "come" NG))
("control"	VERB	(:ALL C S \$\$ED \$\$ED \$\$ING))
("early"	ADJ	(:ALL V IER IEST))
("early"	ADV	(:ALL V IER IEST))
("easily"	ADV	(:ALL V MORE MOST))
("efficient"	ADJ	(:ALL V MORE MOST))
("fair"	ADJ	(:ALL C ER EST))
("fair"	ADV	(:ALL C ER EST))
("fine"	ADJ	(:ALL C R ST))
("fine"	ADV	(:ALL C R ST))
("hot"	ADJ	(:ALL C \$\$ER \$\$EST))
("hot"	ADV	(:ALL C \$\$ER \$\$EST))
("in advance"	ADV	(:ALL V ZERO ZERO))

("people"            NOUN    (people-n-1 C ZERO)(people-n-2 C S))

people は語義により people で複数扱いの場合と peoples で複数になる場合がある。名詞の複数形を出す MG ネットのアクションでは key が lex-cat-id となっているので、lex-id を参照して people-n-1 か people-n-2 かを選んでから 活用情報が決められる。

## 第 5 章

### 今後の課題

最後に形態素生成処理機構および英語形態素生成規則において改良すべき点を挙げておく。

#### 1. 規則による語形の変形

現在の処理系では語の屈折変化の仕方を全て辞書に記述している。例えば、名詞を複数形にする場合、s を付加するか、es を付加するかなどはその名詞の辞書項目に記述されている。しかし、屈折変化の仕方は変化させようとする語の文字列が参照できれば、ある程度の規則化が可能である。辞書は、この規則化から洩れる「例外的」なものに対してのみ用いれば良い。

もちろん、形容詞の級変化など、綴文字だけでは決定できない語形変化に関しては辞書に頼ることになる<sup>1</sup>。

#### 2. 英語における縮約形の決定

現在、“will not” や “it is” などから “won’t” や “it’s” のような縮約形を生成する処理は形態素生成処理で行なっている。しかし、縮約するかどうかを正しく決定するには、単に一次的な前後関係だけでなく句構造を考慮しなければならない。すなわち、句構造境界を採り入れた縮約処理を実現する必要がある。

---

<sup>1</sup>たとえば、発音のためのシラブル数などが辞書に与えてあれば、これを使ってある程度規則化された処理を行なうことはできる。

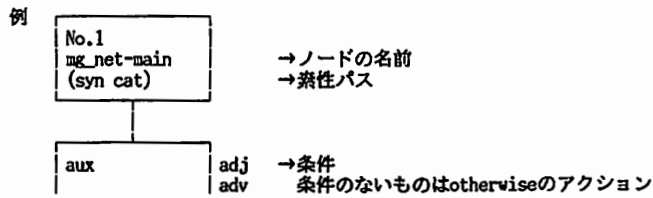
## 参考文献

- [1] 竹沢寿幸, 森元逞, 谷戸文廣, 鈴木雅実, 嵯峨山茂樹, 樽松明 (1993-03): “ATR 音声言語翻訳実験システム ASURA”, 情報処理学会第 46 回全国大会, 6B-5.
- [2] 嵯峨山茂樹, 鷹見淳一, 永井明人, ハラルド・シンガー, 竹沢寿幸, 谷戸文廣, 鈴木雅実, 森元逞, 樽松明 (1993-03): “自動翻訳電話実験システム ASURA の概要”, 日本音響学会 平成 5 年度春季研究発表会 講演論文集, 3-4-17.
- [3] Gen-ichiro Kikui(1992-8):“Feature Structure Based Semantic Head Driven Generation”, In *Proceedings of Coling-93*, pp. 32-38
- [4] Mark Seligman, (1993-03): “Morphological Facilities for German Generation in ASURA”, *ATR Technical Report*, ATR-I-0362.

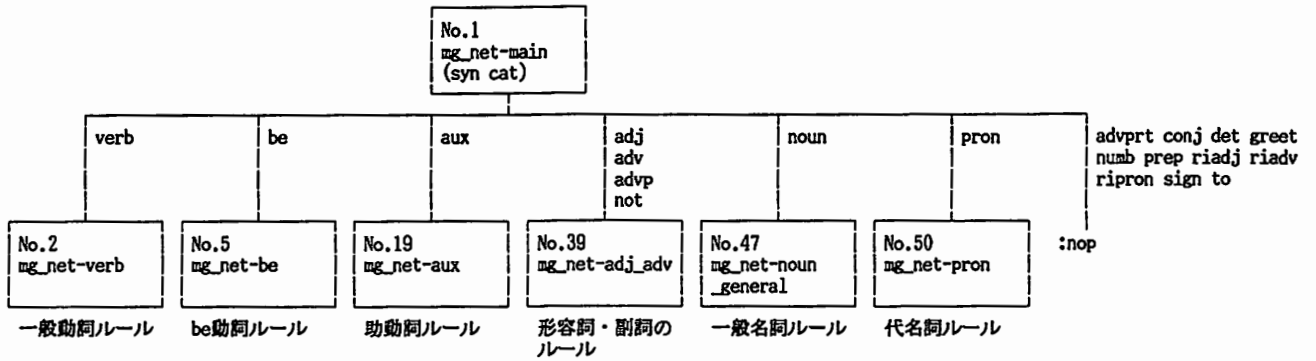
APPENDIX 1

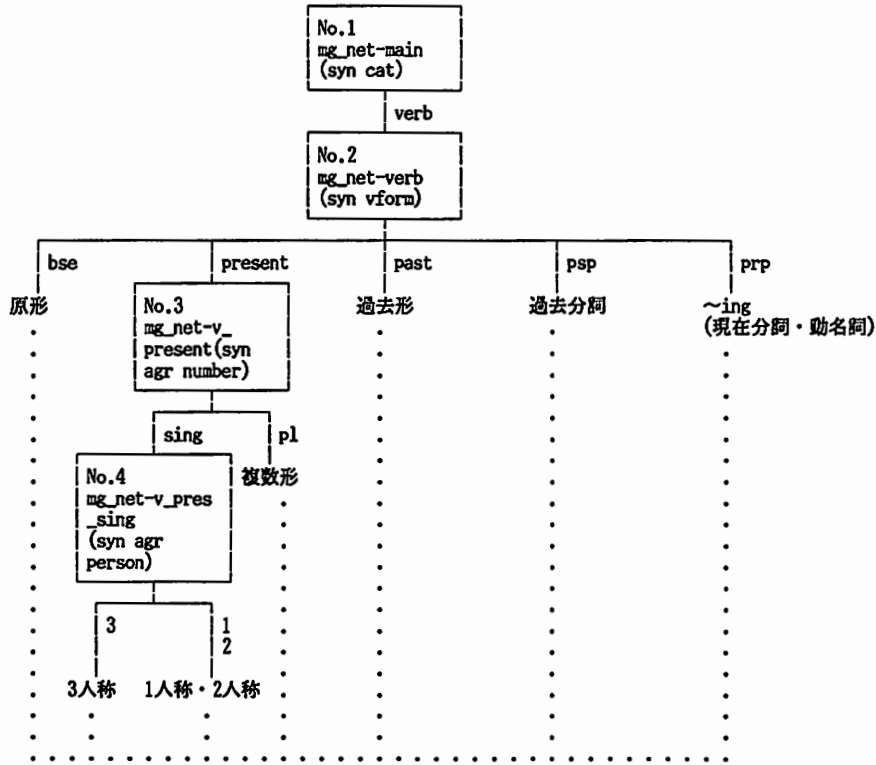
MGネットフローチャート

MGネット規則を、処理に従って以下に図式化した。

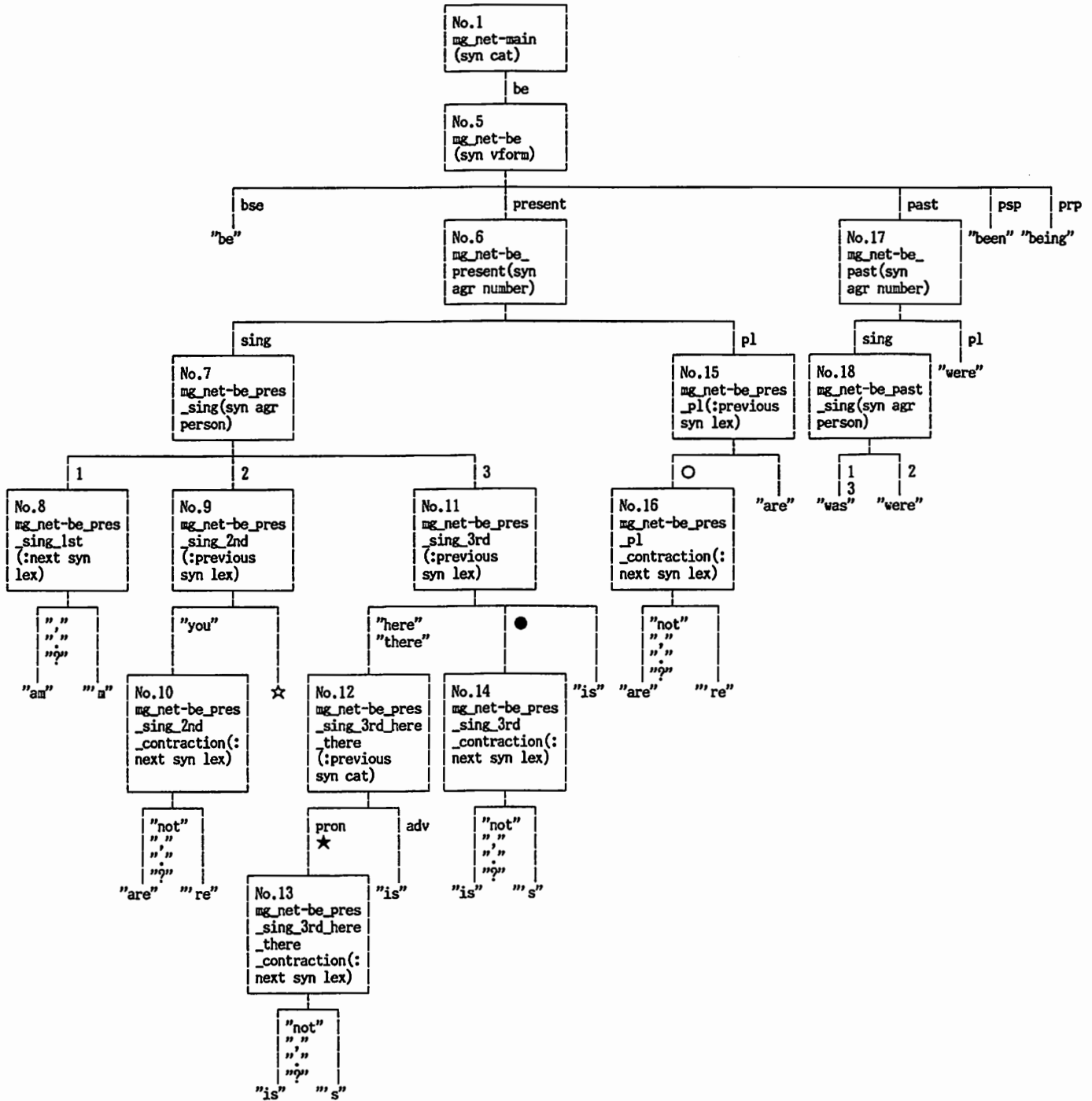


メインルール





形態素辞書を引き、指定された位置のカラムから変化形を取り出す



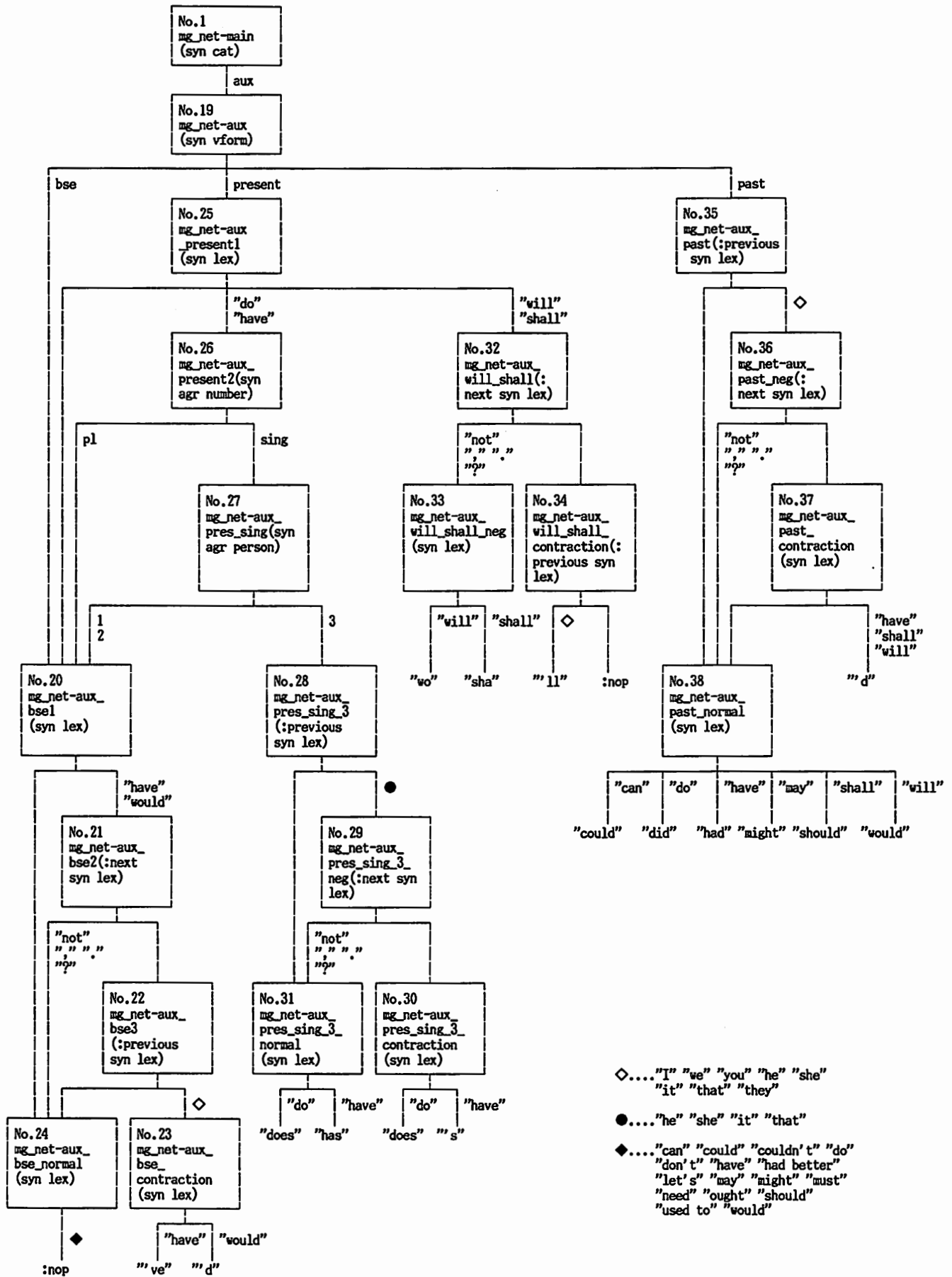
○.... "we" "you" "they"

なおHere are ~./There are ~.は縮約形にしないので、No.16のルールへは行かずに"are"を出す

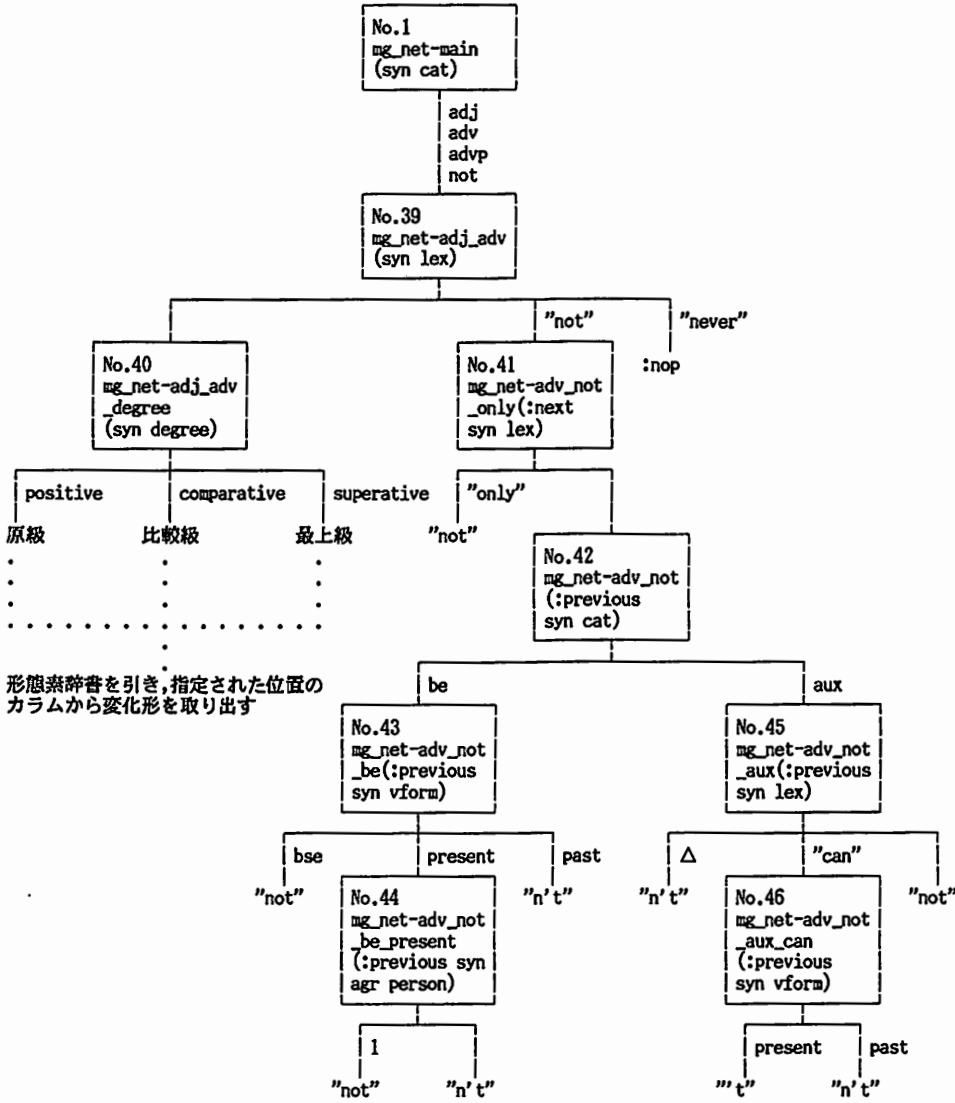
●.... "he" "she" "it" "that"

☆.... 2人称の場合、you以外の主語はありえないので警告を出す

★.... Here is ~./There is ~.構文の場合、here/thereのcatはpronになっており、それをadvのhere/thereと区別する



形容詞・副詞のルール

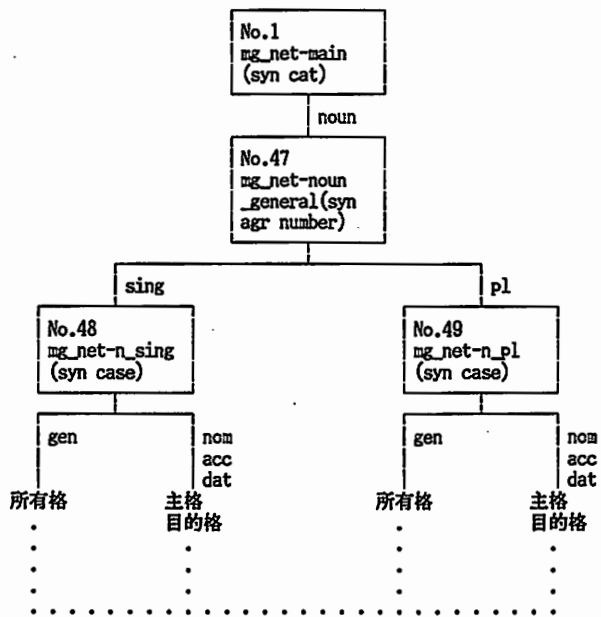


形態素辞書を引き、指定された位置の  
カラムから変化形を取り出す

Δ...."could" "do" "have" "might" "must" "shall" "should" "will" "would"

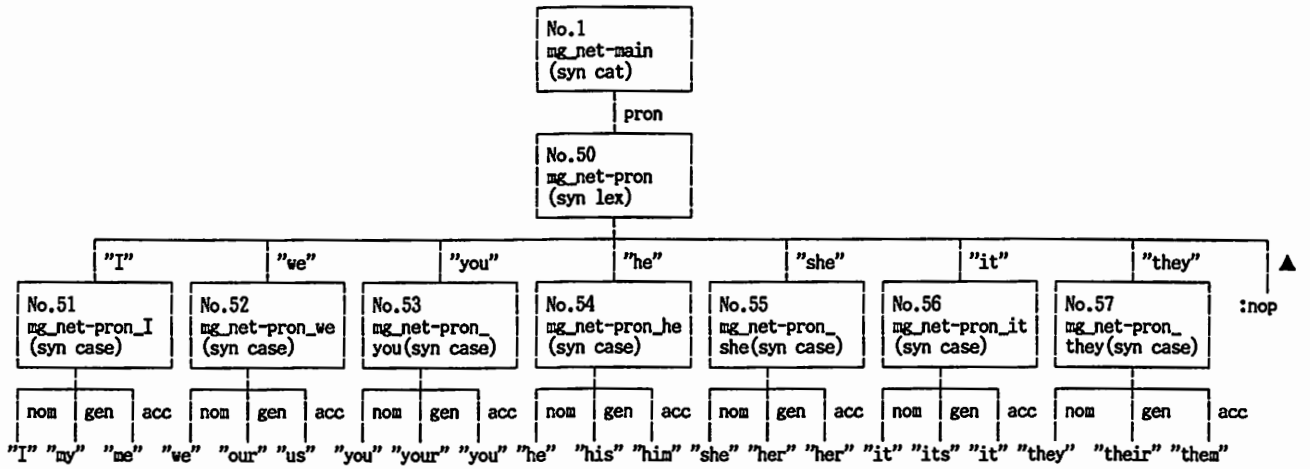


一般名詞のルール



形態素辞書を引き, 指定された位置のカラムから変化形を取り出す

代名詞のルール



▲...."anyone" "anything" "anything else" "everyone" "nothing" "some place" "someone" "something" "that" "there" "this" "yourself"(不定代名詞・指示代名詞など)

APPENDIX 2

```
(in-package "USER")

;No.1
;メインルール (catを見て品詞で分ける)
(def_mg_net
:name mg_net-main
:path (syn cat)
:default (:warning "xxx")
:body ((verb (:call mg_net-verb)) ;;動詞
      (be (:call mg_net-be)) ;;be動詞
      (aux (:call mg_net-aux)) ;;助動詞
      ((:set adj adv) (:call mg_net-adj_adv)) ;;形容詞・副詞
      (not (:call mg_net-adj_adv)) ;;not
      (advp (:call mg_net-adj_adv)) ;;副詞句
      (noun (:call mg_net-noun_general)) ;;一般名詞
      (pron (:call mg_net-pron)) ;;代名詞
      ((:set advprt conj det greet numb prep riadj riadv ripron sign to) :nop) ;;その他
      (otherwise (:warning "yyy"))))
```

```
;;;;;
; 一般動詞
;;;;;
```

```
;No.2
;動詞ルール (vformを見て変化形で分ける)
(def_mg_net
:name mg_net-verb
:path (syn vform)
:default (:warning "xxx")
:body ((bse (:dict :nadine91 :lex-cat 0)) ;;原形
      (present (:call mg_net-v_present)) ;;現在
      (past (:dict :nadine91 :lex-cat 4)) ;;過去
      (psp (:dict :nadine91 :lex-cat 5)) ;;過去分詞
      (prp (:dict :nadine91 :lex-cat 6)) ;;ing
      (otherwise (:warning "yyy"))))
```

```
;No.3
;動詞・現在のルール (agr numberを見て単複で分ける)
(def_mg_net
:name mg_net-v_present
:path (syn agr number)
:default (:warning "xxx")
:body ((sing (:call mg_net-v_pres_sing)) ;;単数
      (pl (:dict :nadine91 :lex-cat 0)) ;;複数
      (otherwise (:warning "yyy"))))
```

```
;No.4
;動詞・現在・単数のルール (agr personを見て人称で分ける)
(def_mg_net
:name mg_net-v_pres_sing
:path (syn agr person)
:default (:warning "xxx")
:body ((3 (:dict :nadine91 :lex-cat 3)) ;;3人称
      ((:set 1 2) (:dict :nadine91 :lex-cat 0)) ;;1人称または2人称
      (otherwise (:warning "yyy"))))
```

```
;;;;;
; B E 動詞
;;;;;
```

```
;No.5
;be動詞ルール (vformを見て変化形で分ける)
(def_mg_net
:name mg_net-be
:path (syn vform)
:default (:warning "xxx")
:body ((bse "be" (:vowel t)) ;;原形
      (present (:call mg_net-be_present)) ;;現在
      (past (:call mg_net-be_past)) ;;過去
      (psp "been" (:vowel nil)) ;;過去分詞
      (prp "being" (:vowel nil)) ;;ing
      (otherwise (:warning "yyy"))))
```

```
;No.6
;be動詞・現在のルール (agr numberを見て単複で分ける)
(def_mg_net
:name mg_net-be_present
:path (syn agr number)
:default (:warning "xxx")
:body ((sing (:call mg_net-be_pres_sing)) ;;単数
      (pl (:call mg_net-be_pres_pl)) ;;複数
      (otherwise (:warning "yyy"))))
```

```

;No.7
;be動詞・現在・単数のルール (agr personを見て人称で分ける)
(def_mg_net
 :name mg_net-be_pres_sing
 :path (syn agr person)
 :default (:warning "xxx")
 :body ((1 (:call mg_net-be_pres_sing_1st)) ;;1人称
        (2 (:call mg_net-be_pres_sing_2nd)) ;;2人称
        (3 (:call mg_net-be_pres_sing_3rd)) ;;3人称
        (otherwise (:warning "yyy"))))

;No.8
;be動詞・現在・単数・1人称のルール
(def_mg_net
 :name mg_net-be_pres_sing_1st
 :path (:next syn lex)
 :default "am"
 :body (((set " " " " " ") "am" (:vowel t)) ;;Yes, I am.の場合
        (otherwise "m" (:left_space nil))))

;No.9
;be動詞・現在・単数・2人称のルール (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
 :name mg_net-be_pres_sing_2nd
 :path (:previous syn lex)
 :default "are"
 :body (("you" (:call mg_net-be_pres_sing_2nd_contraction))
        (otherwise "yy" (:vowel t))))

;No.10
;be動詞・現在・単数・2人称・短縮のルール
(def_mg_net
 :name mg_net-be_pres_sing_2nd_contraction
 :path (:next syn lex)
 :default (:warning "are")
 :body (((set "not" " " " " " ") "are") ;;否定文またはYes, you are.の場合
        (otherwise "re" (:left_space nil))))

;No.11
;be動詞・現在・単数・3人称のルール (1つ前の語を見て代名詞とhere/thereとそれ以外に分ける)
(def_mg_net
 :name mg_net-be_pres_sing_3rd
 :path (:previous syn lex)
 :default "is"
 :body (((set "he" "she" "it" "that")
          (:call mg_net-be_pres_sing_3rd_contraction))
        ((set "here" "there")
          (:call mg_net-be_pres_sing_3rd_here_there))
        (otherwise "is" (:vowel t))))

;No.12
;be動詞・現在・単数・3人称・here/thereのルール (1つ前の語here/thereのcatを見てisを出す)
(def_mg_net
 :name mg_net-be_pres_sing_3rd_here_there
 :path (:previous syn cat)
 :default "is"
 :body ((pron (:call mg_net-be_pres_sing_3rd_here_there_contraction)) ;;there構文の場合
        (adv "is")
        (otherwise (:warning "yyy"))))

;No.13
;be動詞・現在・単数・3人称・here/there・短縮のルール
(def_mg_net
 :name mg_net-be_pres_sing_3rd_here_there_contraction
 :path (:next syn lex)
 :default (:warning "is")
 :body (((set "not" " " " " " ") "is") ;;否定文またはYes, here is.の場合
        (otherwise "s" (:left_space nil))))

;No.14
;be動詞・現在・単数・3人称・短縮のルール
(def_mg_net
 :name mg_net-be_pres_sing_3rd_contraction
 :path (:next syn lex)
 :default (:warning "is")
 :body (((set "not" " " " " " ") "is") ;;否定文またはYes, he is.の場合
        (otherwise "s" (:left_space nil))))

;No.15
;be動詞・現在・複数のルール (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
 :name mg_net-be_pres_pl
 :path (:previous syn lex)
 :default "are"
 :body (((set "we" "you" "they")
          (:call mg_net-be_pres_pl_contraction))
        (otherwise "are"))))

```

```

;No.16
;be動詞・現在・複数・短縮のルール
(def_mg_net
:name mg_net-be_pres_pl_contraction
:path (:next syn lex)
:default (:warning "are")
:body (((set "not" " " " " " ") "are") ;;否定文またはYes, we are.の場合
(otherwise "re"(:left_space nil))))

;No.17
;be動詞・過去のルール (agr numberを見て単複で分ける)
(def_mg_net
:name mg_net-be_past
:path (syn agr number)
:default (:warning "xxx")
:body ((sing (:call mg_net-be_past_sing)) ;;単数
(pl "were" (:vowel nil)) ;;複数
(otherwise (:warning "yyy"))))

;No.18
;be動詞・過去・単数のルール (agr personを見て人称で分ける)
(def_mg_net
:name mg_net-be_past_sing
:path (syn agr person)
:default (:warning "xxx")
:body (((set 1 3) "was" (:vowel nil)) ;;1人称または3人称
(2 "were" (:vowel nil)) ;;2人称
(otherwise (:warning "yyy"))))

;~~~~~
; 助動詞
;~~~~~

;No.19
;助動詞ルール (vformを見て変化形で分ける)
(def_mg_net
:name mg_net-aux
:path (syn vform)
:default (:warning "xxx")
:body ((bse (:call mg_net-aux_bse1)) ;;原形
(present (:call mg_net-aux_present1)) ;;現在
(past (:call mg_net-aux_past)) ;;過去
(otherwise (:warning "yyy"))))

;No.20
;助動詞・原形のルールその1 (lexを見てhave/wouldを分離する)
(def_mg_net
:name mg_net-aux_bse1
:path (syn lex)
:default (:warning "xxx")
:body (((set "have" "would") (:call mg_net-aux_bse2))
(otherwise (:call mg_net-aux_bse_normal))))

;No.21
;助動詞・原形のルールその2
(def_mg_net
:name mg_net-aux_bse2
:path (:next syn lex)
:default (:warning (:call mg_net-aux_bse3))
:body (((set "not" " " " " " ") (:call mg_net-aux_bse_normal)) ;;否定文またはYes, I would.の場合
(otherwise (:call mg_net-aux_bse3))))

;No.22
;助動詞・原形のルールその3 (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
:name mg_net-aux_bse3
:path (:previous syn lex)
:default (:call mg_net-aux_bse_normal)
:body (((set "I" "we" "you" "he" "she" "it" "that" "they")
(call mg_net-aux_bse_contraction))
(otherwise (:call mg_net-aux_bse_normal))))

;No.23
;助動詞・原形・短縮のルール (lexを見てhave/wouldを短縮する)
(def_mg_net
:name mg_net-aux_bse_contraction
:path (syn lex)
:default (:warning "xxx")
:body (("have" "ve" (:left_space nil))
("would" "d" (:left_space nil))
(otherwise (:warning "yyy"))))

;No.24
;助動詞・原形・短縮しないルール (lexを見て助動詞を出す)
(def_mg_net
:name mg_net-aux_bse_normal
:path (syn lex)
:default (:warning "xxx")
:body (((set "can" "could" "couldn't" "do" "don't" "have" "had better" "let's" "may" "might" "must" "need" "ought" "should" "used to" "would")
) :nop)
(otherwise (:warning "yyy"))))

```

```

;No.25
;助動詞・現在のルールその1 (lexを見てdo/have/will/shallを分離する)
(def_mg_net
 :name mg_net-aux_present1
 :path (syn lex)
 :default (:warning "xxx")
 :body (((set "do" "have") (:call mg_net-aux_present2))
        ((set "will" "shall") (:call mg_net-aux_will_shall))
        (otherwise (:call mg_net-aux_bsel))))))

;No.26
;助動詞・現在のルールその2 (agr numberを見て単複で分ける)
(def_mg_net
 :name mg_net-aux_present2
 :path (syn agr number)
 :default (:warning "xxx")
 :body ((sing (:call mg_net-aux_pres_sing)) ;;単数
        (pl (:call mg_net-aux_bsel)) ;;複数
        (otherwise (:warning "yyy"))))

;No.27
;助動詞・現在・単数のルール (agr personを見て人称で分ける)
(def_mg_net
 :name mg_net-aux_pres_sing
 :path (syn agr person)
 :default (:warning "xxx")
 :body ((3 (:call mg_net-aux_pres_sing_3)) ;;3人称
        ((set 1 2) (:call mg_net-aux_bsel)) ;;1人称または2人称
        (otherwise (:warning "yyy"))))

;No.28
;助動詞・現在・単数・3人称のルール (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
 :name mg_net-aux_pres_sing_3
 :path (:previous syn lex)
 :default (:call mg_net-aux_pres_sing_3_normal)
 :body (((set "he" "she" "it" "that")
          (:call mg_net-aux_pres_sing_3_neg))
        (otherwise (:call mg_net-aux_pres_sing_3_normal))))

;No.29
;助動詞・現在・単数・3人称・否定のルール
(def_mg_net
 :name mg_net-aux_pres_sing_3_neg
 :path (:next syn lex)
 :default (:warning (:call mg_net-aux_pres_sing_3_normal))
 :body (((set "not" " " " " " " ?") (:call mg_net-aux_pres_sing_3_normal)) ;;否定文またはYes, he has.の場合
        (otherwise (:call mg_net-aux_pres_sing_3_contraction))))

;No.30
;助動詞・現在・単数・3人称・短縮のルール (lexを見てdo/haveを短縮する)
(def_mg_net
 :name mg_net-aux_pres_sing_3_contraction
 :path (syn lex)
 :default (:warning "xxx")
 :body (("do" "does")
        ("have" "s" (:left_space nil))
        (otherwise (:warning "yyy"))))

;No.31
;助動詞・現在・単数・3人称・短縮しないルール (lexを見てdo/haveを3人称にする)
(def_mg_net
 :name mg_net-aux_pres_sing_3_normal
 :path (syn lex)
 :default (:warning "xxx")
 :body (("do" "does")
        ("have" "has")
        (otherwise (:warning "yyy"))))

;No.32
;助動詞・will/shallのルール
(def_mg_net
 :name mg_net-aux_will_shall
 :path (:next syn lex)
 :default (:warning (:call mg_net-aux_will_shall_contraction))
 :body (((set "not" " " " " " " ?") (:call mg_net-aux_will_shall_neg)) ;;否定文またはYes, I will.の場合
        (otherwise (:call mg_net-aux_will_shall_contraction))))

;No.33
;助動詞・will/shall・否定のルール (lexを見てn'tの前の形を作る)
(def_mg_net
 :name mg_net-aux_will_shall_neg
 :path (syn lex)
 :default (:warning "xxx")
 :body (("will" "wo")
        ("shall" "sha")
        (otherwise (:warning "yyy"))))

```

```

;No.34
;助動詞・will/shall・短縮のルール (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
 :name mg_net-aux_will_shall_contraction
 :path (:previous syn lex)
 :default :nop
 :body (((set "I" "we" "you" "he" "she" "it" "that" "they") "ll"(:left_space nil))
 (otherwise :nop)))

;No.35
;助動詞・過去のルール (1つ前の語を見て代名詞かどうかで分ける)
(def_mg_net
 :name mg_net-aux_past
 :path (:previous syn lex)
 :default (:call mg_net-aux_past_normal)
 :body (((set "I" "we" "you" "he" "she" "it" "that" "they")
 (:call mg_net-aux_past_neg))
 (otherwise (:call mg_net-aux_past_normal))))

;No.36
;助動詞・過去・否定のルール (1つ後ろの語を見てnotかどうかで分ける)
(def_mg_net
 :name mg_net-aux_past_neg
 :path (:next syn lex)
 :default (:warning (:call mg_net-aux_past_normal))
 :body (((set "not" " " " " "p") (:call mg_net-aux_past_normal)) ;;否定文またはYes, I would.の場合
 (otherwise (:call mg_net-aux_past_contraction))))

;No.37
;助動詞・過去・短縮のルール (lexを見て短縮する)
(def_mg_net
 :name mg_net-aux_past_contraction
 :path (syn lex)
 :default (:warning "xxx")
 :body (("have" "d" (:left_space nil))
 ("shall" "d" (:left_space nil))
 ("will" "d" (:left_space nil))
 (otherwise (:call mg_net-aux_past_normal))))

;No.38
;助動詞・過去・短縮しないルール (lexを見て助動詞の過去形を出す)
(def_mg_net
 :name mg_net-aux_past_normal
 :path (syn lex)
 :default (:warning "xxx")
 :body (("can" "could")
 ("do" "did")
 ("have" "had")
 ("may" "might")
 ("shall" "should")
 ("will" "would")
 (otherwise (:warning "yyy"))))

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 形容詞/副詞
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

;No.39
;形容詞・副詞のルール (lexを見てnotを分離する)
(def_mg_net
 :name mg_net-adj_adv
 :path (syn lex)
 :default (:warning "xxx")
 :body (("not" (:call mg_net-adv_not_only))
 ("never" :nop)
 (otherwise (:call mg_net-adj_adv_degree))))

;No.40
;形容詞・副詞の級のルール (degreeを見て級で分ける)
(def_mg_net
 :name mg_net-adj_adv_degree
 :path (syn degree)
 :default (:dict :nadine91 :lex-cat 0)
 :body ((positive (:dict :nadine91 :lex-cat 0)) ;;原級
 (comparative (:dict :nadine91 :lex-cat 3)) ;;比較級
 (superlative (:dict :nadine91 :lex-cat 4)) ;;最上級
 (otherwise (:warning "yyy"))))

;No.41
;副詞・not・onlyのルール (1つ後ろの語を見てonlyを分離する)
(def_mg_net
 :name mg_net-adv_not_only
 :path (:next syn lex)
 :default (:warning (:call mg_net-adv_not))
 :body (("only" "not") ;;not only A but also Bの場合
 (otherwise (:call mg_net-adv_not))))

```

```

;No.42
;副詞・notのルール (1つ前の語を見て品詞で分ける)
(def_mg_net
  :name mg_net-adv_not
  :path (:previous syn cat)
  :default "not"
  :body ((be (:call mg_net-adv_not_be)) ;;be動詞
         (aux (:call mg_net-adv_not_aux)) ;;助動詞
         (otherwise (:warning "yyy"))))

;No.43
;副詞・not・be動詞のルール (1つ前の語を見て時制で分ける)
(def_mg_net
  :name mg_net-adv_not_be
  :path (:previous syn vform)
  :default (:warning "xxx")
  :body ((bse "not") ;;原形
         (present (:call mg_net-adv_not_be_present)) ;;現在
         (past "n't" (:left_space nil)) ;;過去
         (otherwise (:warning "yyy"))))

;No.44
;副詞・not・be動詞・現在のルール (1つ前の語を見て1人称かどうかで分ける)
(def_mg_net
  :name mg_net-adv_not_be_present
  :path (:previous syn agr person)
  :default (:warning "xxx")
  :body ((1 "not") ;;1人称
         (otherwise "n't"(:left_space nil))))

;No.45
;副詞・not・助動詞のルール (1つ前の語を見てcanを分離する)
(def_mg_net
  :name mg_net-adv_not_aux
  :path (:previous syn lex)
  :default (:warning "xxx")
  :body (((set "could" "do" "have" "might" "must" "shall" "should" "will" "would") "n't" (:left_space nil))
         ("can" (:call mg_net-adv_not_aux_can))
         (otherwise "not" (:vowel t))))

;No.46
;副詞・not・助動詞・canのルール (1つ前の語を見て時制で分ける)
(def_mg_net
  :name mg_net-adv_not_aux_can
  :path (:previous syn vform)
  :default (:warning "xxx")
  :body ((present ""t" (:left_space nil)) ;;現在のときcan't
         (past "n't"(:left_space nil)) ;;過去のときcouldn't
         (otherwise (:warning "yyy"))))

;
; 名詞
;
;No.47
;一般名詞ルール (agr numberを見て単複で分ける)
(def_mg_net
  :name mg_net-noun_general
  :path (syn agr number)
  :default (:warning "xxx")
  :body ((sing (:call mg_net-n_sing)) ;;単数
         (pl (:call mg_net-n_pl)) ;;複数
         (otherwise (:warning "yyy"))))

;No.48
;名詞・単数のルール (caseを見て格で分ける)
(def_mg_net
  :name mg_net-n_sing
  :path (syn case)
  :default (:warning "xxx")
  :body ((gen (:dict :nadine91 :lex-cat 0) :genitive)
         ((:set nom acc dat) (:dict :nadine91 :lex-cat 0))
         (otherwise (:warning "yyy"))))

;No.49
;名詞・複数のルール (caseを見て格で分ける)
(def_mg_net
  :name mg_net-n_pl
  :path (syn case)
  :default (:warning "xxx")
  :body ((gen (:dict :nadine91 :lex-cat-id 3) :genitive)
         ((:set nom acc dat) (:dict :nadine91 :lex-cat-id 3))
         (otherwise (:warning "yyy"))))

```



```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; 代名詞
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;No.50
;代名詞ルール (lexを見て分ける)
```

```
(def_mg_net
 :name mg_net-pron
 :path (syn lex)
 :default (:warning "xxx")
 :body ((("I" (:call mg_net-pron_I))
 ("we" (:call mg_net-pron_we))
 ("you" (:call mg_net-pron_you))
 ("he" (:call mg_net-pron_he))
 ("she" (:call mg_net-pron_she))
 ("it" (:call mg_net-pron_it))
 ("they" (:call mg_net-pron_they))
 (:set "anyone" "anything" "anything else" "everyone" "nothing" "some place" "someone" "something" "that" "there" "this" "yourself")) :n
 op)
 (otherwise (:warning "yyy"))))
```

```
;No.51
;代名詞・Iのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_I
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "I" (:vowel t)) ;;主格
 (gen "my") ;;所有格
 (acc "me") ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.52
;代名詞・weのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_we
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "we" (:vowel t)) ;;主格
 (gen "our" (:vowel t)) ;;所有格
 (acc "us" (:vowel t)) ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.53
;代名詞・youのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_you
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "you") ;;主格
 (gen "your") ;;所有格
 (acc "you") ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.54
;代名詞・heのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_he
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "he") ;;主格
 (gen "his") ;;所有格
 (acc "him") ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.55
;代名詞・sheのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_she
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "she") ;;主格
 (gen "her") ;;所有格
 (acc "her") ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.56
;代名詞・itのルール (caseを見て格で分ける)
```

```
(def_mg_net
 :name mg_net-pron_it
 :path (syn case)
 :default (:warning "xxx")
 :body ((nom "it" (:vowel t)) ;;主格
 (gen "its" (:vowel t)) ;;所有格
 (acc "it" (:vowel t)) ;;目的格
 (otherwise (:warning "yyy"))))
```

```
;No.57
;代名詞・theyのルール (caseを見て格で分ける)
(def mg_net
  :name mg_net-pron_they
  :path (syn case)
  :default (:warning "xxx")
  :body ((nom "they") ;;主格
         (gen "their") ;;所有格
         (acc "them") ;;目的格
         (otherwise (:warning "yyy"))))
```

APPENDIX 3

生成のための形態素処理 - 規則変化一覧表

形態素語尾	説明	削除する文字数	付加する文字列	最後の文字列の重なり	例
(1) -s	原形にsが付く	0	"s"	Nil	apple → apples cat → cats book → books
(2) -es	原形にesが付く	0	"es"	Nil	fish → fishes box → boxes watch → watches dish → dishes
(3) -ies	原形末尾のyがiesに変わる	1	"ies"	Nil	lily → lilies dictionary → dictionaries library → libraries city → cities
(4) -ves	原形末尾のfがvesに変わる	1	"ves"	Nil	leaf → leaves thief → thieves
(5) -ves	原形末尾のfeがvesに変わる	2	"ves"	Nil	knife → knives wife → wives
(6) -ed	原形にedが付く	0	"ed"	Nil	regard → regarded cook → cooked look → looked
(7) -d	原形にdが付く	0	"d"	Nil	save → saved achieve → achieved love → loved
(8) -ied	原形末尾のyがiedに変わる	1	"ied"	Nil	satisfy → satisfied study → studied
(9) -ked	原形にkedが付く	0	"ked"	Nil	panic → panicked picnic → picnicked
(10) -@ed	原形末尾の子音が重なりedが付く	0	"ed"	T	wrap → wrapped shop → shopped stop → stopped
(11) -en	原形にenが付く	0	"en"	Nil	fall → fallen eat → eaten
(12) -n	原形にnが付く	0	"n"	Nil	see → seen give → given grow → grown take → taken
(13) -@en	原形末尾の子音が重なりenが付く	0	"en"	T	bid → bidden forbid → forbidden
(14) -ing	原形にingが付く	0	"ing"	Nil	cook → cooking rock → rocking look → looking
(15) -ng	原形末尾のeが取れingが付く	1	"ing"	Nil	drive → driving give → giving love → loving
(16) -king	原形にkingが付く	0	"king"	Nil	panic → panicking picnic → picnicking
(17) -ying	原形末尾のieがyingに変わる	2	"ying"	Nil	lie → lying die → dying
(18) -@ing	原形末尾の子音が重なりingが付く	0	"ing"	T	cut → cutting forget → forgetting run → running stop → stopping

(19)	-er	原形にerが付く	0	"er"	Nil	tall → taller low → lower small → smaller
(20)	-r	原形にrが付く	0	"r"	Nil	false → falser late → later nice → nicer
(21)	-ier	原形末尾のyがierに変わる	1	"ier"	Nil	lonely → lonelier lovely → lovelier happy → happier
(22)	-@er	原形末尾の子音が重なりerが付く	0	"er"	T	hot → hotter big → bigger
(23)	more	原形に more を前置する	0	"_"	Nil	beautiful → more beautiful suspiciously → more suspiciously
<hr/>						
(24)	-est	原形にestが付く	0	"est"	Nil	tall → tallest short → shortest small → smallest
(25)	-st	原形にstが付く	0	"st"	Nil	false → falsest late → latest  nice → nicest
(26)	-iest	原形末尾のyがiestに変わる	1	"iest"	Nil	lovely → loveliest lonely → loneliest happy → happiest
(27)	-@est	原形末尾の子音が重なりestが付く	0	"est"	T	hot → hottest big → biggest
(28)	most	原形に most を前置する	0	"_"	Nil	beautiful → most beautiful slowly → most slowly