

TR-IT-0310

TDMT 解析変換知識作成の手引
(日独版)

Instructions for building
TDMT's transfer knowledge
(Japanese → German)

Michael Paul 山崎 優子 *
平成 11 年 8 月

Abstract

TDMT 用解析変換知識データ (日独) 作成を目的とする、解析変換知識の構成とその仕様ならびにドイツ語生成システムを、以下の章ごとに説明する。

ATR Interpreting Telecommunications Research Laboratories

©1999 ATR Interpreting Telecommunications Research Laboratories

* 株式会社コングレ

Contents

1	TDMT 用データの構成と仕様	1
1.1	翻訳訓練対象テキスト	1
1.1.1	バイリンガル会話	1
1.1.2	基本表現集	2
1.1.3	CSTAR II 用会話集	2
1.2	日本語形態素タギングデータ	3
1.3	辞書	4
1.3.1	日本語意味コード辞書	4
1.3.2	日独変換辞書	4
1.3.3	ドイツ語生成辞書	6
1.4	解析知識	6
1.4.1	replace-word	7
1.4.2	lexical-transformation	7
1.4.3	local-transformation	15
1.4.4	total-transformation	18
1.5	変換知識	19
1.5.1	概要	19
1.5.2	原言語パターン/目的言語パターン/用例	20
1.5.3	パタンの種類	21
1.5.4	カテゴリ	21
1.5.5	下部構造の制限	23
1.5.6	ヘッド	25
1.5.7	ローカル辞書	27
1.5.8	生成情報	28
1.5.9	主語補完	29
1.5.10	カテゴリ別変換パターン定義の基本方針とパターン例	30
1.6	各種知識の統計データ	41
2	解析変換知識作成手順	43
2.1	意味距離計算を使わないで訓練する場合	43
2.1.1	解析変換知識作成手順	43
2.1.2	例1 "所要時間は六時間三十分になります"	44
2.1.3	例2 "インド料理かタイ料理の店があれば"	48
2.2	意味距離計算を使って訓練する場合	53
2.2.1	解析変換知識作成手順	53
2.2.2	例3 "わたくしは実は野球のことは本当によくわからないんですね"	54
2.3	解析変換知識作成時に使用するツール類	58
2.4	精度を確認するためのツール類	59
3	懸案事項	62
	付録	69
A	生成モジュール	69
B	生成マーカ使用例	74
C	分割 + 部分翻訳方式	99

1 TDMT 用データの構成と仕様

TDMT 用解析変換知識データの構成とその仕様を、以下の章ごとに説明する。

各種データの書式に関する注意事項

- [] ... 省略可能を示す

例 (((TARGET-PATTERN) ([GENERATION-INFO]))

- 英小文字 ... そのまま記述することを示す
- 英大文字 ... 該当する内容を記述することを示す

例 (define-transfer-dic :j-g t
((POS JAPANESE-STRING) TARGET)
)

1.1 翻訳訓練対象テキスト

1.1.1 バイリンガル会話

[目的]

旅行会話をトピックとし、通訳を介して収録したバイリンガル模擬会話。翻訳訓練を行なう際の訓練文として選ばれ(システム管理者が /usr/local/TDMT/tdmt-multi-dev/build/text-definitions.lisp(文単位) または /usr/local/TDMT/tdmt-multi-dev/build/text-definitions-uttr.lisp(発話単位) で指定)、解析変換知識作成者はこのコーパスから対訳データを収集した上で調整を行ない作成する。現在は発話単位が使われている。

[管理者] データベース管理者およびシステム管理者

[格納場所]

/DB/LDB/JG/JGTEXT/9511/*JG.JGTEXT
/DB/LDB/JE/EJTEXT+/{DEN94B, FURU94B, SOBA93, SOBA94A}/*.EJTEXT
/DB/SLDB/LNG/EJTEXT+/*.EJTEXT

* JGTEXT のみ日独バイリンガル会話

[例] /DB/LDB/JG/JGTEXT/9511/B12006JG.JGTEXT

G|Verkehrsamt Frankfurt.|Guten Tag.

J| はい、こちらフランクフルト観光局でございます。

J| [あの] 今、こっちに着いたばかりなんですけど、今晚泊まれるホテル、探してるんですが。

G|Ich bin gerade hier angekommen und ich suche ein Hotel für heute nacht.

G|Was für ein Hotel möchten Sie?

J| どのようなホテルをご希望でしょうか。

J| [あの一] 安くてきれいなところならいいんですけど。
G|Ich möchte ein günstiges und sauberes Hotel.

G|Okay. |Und wie teuer kann es sein?
J| はい、それでご予算の方はどれくらいをご希望ですか。

1.1.2 基本表現集

〔目的〕 広範囲な旅行会話を網羅するために作成された、対訳付き基本表現集。
以下、1.1.1 バイリンガル会話に同じ。

〔管理者〕 データベース管理者およびシステム管理者

〔格納場所〕

/DB/LDB/JG/JGTEXT/9803_kihon/*.JG3
/DB/LDB/JEK/JEKTEXT/{C9608, S9503, S9510}/{*.JETEXT, *.JEKTEXT}

* JGTEXT のみ日独対訳。JEKTEXT のうち C9608 は日英対訳。S9503, S9510 は日英韓対訳。

〔例〕 /DB/LDB/JG/JGTEXT/9803_kihon/A4JP_S9503.JG3

J|XX|AU4JP001| 今晚のホテルを予約したいのですが。
G|XX|AU4JP001|Ich möchte gerne für heute nacht reservieren.

J|XX|AU4JP002| 何名様ですか。
G|XX|AU4JP002|Für wie viele Personen?

J|XX|AU4JP003| 一名です。 | シングルを一つお願いします。
G|XX|AU4JP003|Eine Person. |Ich hätte gerne ein Einzelzimmer.

J|XX|AU4JP004| はい、ご用意できます。
G|XX|AU4JP004|Selbstverständlich, das können wir für Sie arrangieren.

*詳細は [2] を参照のこと。

1.1.3 C S T A R II 用会話集

〔目的〕 日米韓独の四ヶ国間の接続実験のために用意されたテキスト。概して長文が多く含まれる。以下、1.1.1 バイリンガル会話に同じ。

〔管理者〕 データベース管理者およびシステム管理者

〔格納場所〕

/DB/CstarII/WEARABLE/JTEXT/*.JTEXT
/DB/CstarII/Demo/July99/JTEXT/*.JTEXT
/DB/CstarII/Demo/May99/JTEXT/*.JTEXT
/DB/CstarII/Demo/Feb99/JTEXT/*.JTEXT
/DB/CstarII/JJ/LNG/JTEXT/*.JTEXT
/DB/CstarII/JK/LNG/JTEXT/*.JTEXT

TZS のファイル (/DB/CstarII/JJ/LNG/JTEXT/TZS*.JTEXT) に関しては以下のファイルのみトレーニングを行った。

TZS80011 TZS80012 TZS80021 TZS80025 TZS80031 TZS80032 TZS80041

対訳： /DB/CstarII/JJ/LNG/GTEXT

〔例〕 /DB/CstarII/JJ/LNG/JTEXT/TBS80073.JTEXT

旅行者： [えー] 実は <wa> 十月三日か、四日、いずれか、ヤンキースタジアムで
ヤンキース対ドジャーズの試合を観戦したいと思ってるんですが、その
期間中、ホテルは <wa>、あいてますでしょうか。

代理店： [えー] ワールドシリーズのおかげで、どこも予約がいっぱいですが、 [えー]
お薦めのホテルとしましては <wa> ザニューヨーカーホテル、カールトンホテル、
ウイリントンホテル、パークセントラルホテルが、あいております。

/DB/CstarII/JJ/LNG/GTEXT/TBS80073.GTEXT

Tourist: Ich werde entweder am dritten oder vierten im Yankee Stadium
ein Spiel der Yankees und Dodgers sehen. Ist in der Zeit noch
ein Zimmer zu buchen?

Agentur: Eben wegen der World Series sind die meisten Hotels ausgebucht.
Zu empfehlen sind noch das New Yorker Hotel, das Carlton Hotel,
Wellington Hotel und Park Central Hotel. In diesen Hotels sind
noch Zimmer frei.

1.2 日本語形態素タギングデータ

〔目的〕 べた書き文字列の文を単語単位（形態素）に分解し、品詞や活用形などの
情報を付与したデータ。翻訳訓練の際は形態素解析を行わず、あらかじめ
解析済みのこのタギングデータをもとに各解析変換知識を作成する。タギ
ングデータは、訓練対象テキストそれぞれに対して、システム管理者が
/usr/local/TDMT/tdmt-multi-dev/build/text-definitions.lispまたは
/usr/local/TDMT/tdmt-multi-dev/build/text-definitions-uttr.lispにて指定
する。

〔管理者〕 タギングデータ管理者およびシステム管理者

〔格納場所〕

/DB/LDB/TDMT/JMOR/LDB-JG/9511/*.JMOR
/DB/LDB/TDMT/JMOR/LDB-JE/{DEN94B, SOBA94A}/*.JMOR
/DB/LDB/TDMT/JMOR/LDB-JEK/{C9608, S9503, S9510}/*.JMOR
/DB/LDB/TDMT/JMOR/SLDB/*.JMOR

〔書式〕

発話 ID| 発声 ID| 文節 ID| 単語 ID| 表記形| 読み| 正規形| 品詞| 活用型| 活用形| 音便|

〔例〕 /DB/LDB/TDMT/JMOR/LDB-JG/9511/AT120012JG.JMOR

10|0010|10|10| はい|ハイ|はい|感動詞| || ||
10|0010|20|20| シュロスホテルボン|シュロスホテルボン|シュロスホテルボン|普通名詞
|| ||
10|0010|20|30| です|デス|です|判定詞|特殊サ|基本 ||
10|0010|20|40|. ||.|記号| || ||
20|0020|30|50| 週末|シュウマツ|週末|普通名詞| || ||
20|0020|30|60| の|ノ|の|連体助詞| || ||
20|0020|40|70| 宿泊|シュクハク|宿泊|サ変名詞| || ||
20|0020|40|80| 割引|ワリビキ|割引|サ変名詞| || ||
20|0020|40|90| について|ニツイテ|について|格助詞| || ||
20|0020|50|100| お|オ|お|接頭辞| || ||
20|0020|50|110| 教え|オシエ|教える|本動詞|一段|た ||
20|0020|50|120| ていただき|テイタダキ|いただく|助動詞|五段カ|連用 ||

20|0020|50|130|たい|タイ|たい|助動詞|形容詞|基本||
20|0020|50|140|ん|ン|ん|準体助詞||||
20|0020|50|150|です|デス|です|判定詞|特殊サ|基本||
20|0020|50|160|が|ガ|が|接続助詞||||
20|0020|50|170|。||。|記号||||

*詳細は[4]を参照のこと。

1.3 辞書

1.3.1 日本語意味コード辞書

〔目的〕 角川書店「類語新辞典」の三桁の数字をもとに、形態素ごとに意味コードを付与した辞書。システムが意味距離計算により、類似用例を検索する際に参照する。また、翻訳訓練の際に意味距離計算を行ないながら知識を作成することもできる（「2.2.1 解析変換知識作成手順」参照）。

〔管理者〕 意味コード辞書管理者

〔格納場所〕

/usr/local/TDMT/tdmt-multi-dev/j-morph/dic/jma-atr-sem-code.text

〔書式〕 (REG-EXP POS SEM-CODE)

- REG-EXP ... 日本語形態素の正規形
- POS ... 日本語品詞シンボル
- SEM-CODE ... 意味コード（複数指定可）

〔例〕

("扱い" 普通名詞 "382")
("扱う" 本動詞 "380" "382" "489" "787")
("宛" 普通名詞 "109c")
("宛先" 普通名詞 "109c")
("安い" 形容詞 "171a" "693")
("安まる" 本動詞 "405")
("安心" サ変形容名詞 "494" "694")
("安心する" 本動詞 "494" "694")

*詳細は「日本語意味コード辞書作成の手引」を参照のこと。

1.3.2 日独変換辞書

〔目的〕 日本語形態素に対応するドイツ語訳を記述する辞書。一語につき、一対訳が登録できる。原言語に対しては (POS JAPANESE-STRING) が一単位となるが、目的言語 (TARGET) に対しては生成情報 (品詞または格) を伴った形態素列を記述する。さらに、その形態素列をいくつかまとめて生成情報を付けることもできる。ここで記述された目的言語は、変換知識のパタン中で別の対訳に変更することができる。詳細は「1.5.7 ローカル辞書」参照のこと。また、話者情報が利用できる場合は、話者によって対訳を変えることができる（詳細は「TDMT 解析変換知識作成の手引（英日版）」参照）。

[管理者] 解析変換知識作成者

[格納場所]

/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/dic/japanese-to-german.lisp

[書式]

```
(define-transfer-dic :j-g t
  ((POS JAPANESE-STRING) TARGET)
)
```

- define-transfer-dic ... 変換辞書の定義を宣言
- :j-g ... 翻訳の方向。この場合は日独翻訳を示す。
- t ... 辞書データをロードする際にハッシュテーブルを作り直す
- POS ... 日本語品詞シンボル
- JAPANESE-STRING ... 日本語正規形または表層形
- TARGET ... 目的言語の対訳と生成情報のリスト

[例]

```
(define-transfer-dic :j-g t
  ((代名詞 "いくら") (WH (FRAGEADVERB "wie") (ADJEKTIV "teuer")))
  ((普通名詞 "一部") (NP {UNBESTIMMT} (NOMEN "Teil")))
  ((普通名詞 "からすま京都ホテル") (EIGENNAME (NOMEN "Hotel") "Karasuma Kyoto"
                                                :GENDER NTR))
  ((サ変名詞 "転送") (VP (VERBZUSATZ "weiter")(VERB "schicken")))
  ((サ変名詞 "宴会") (NOMEN "Bankett"))
  ((サ変形容名詞 "さっぱり") (ADVERB "überhaupt"))
  ((サ変形容名詞 "安心") (VP (REFLEXIVPRONOMEN "sich" :case AKK)
                              (VERB "beruhigen")))
  ((形式名詞 "もの") (PARTIKEL "etwas"))
  ((形容名詞 "楽しみ") (NOMEN "Vergnügen"))
  ((形容名詞 "好評") (VP (ADJEKTIV "berühmt")(HILFSVERB "sein")))
  ((形容詞 "熱っぽい") (ADJEKTIV "fiebrhaft"))
  ((形容詞 "恥ずかしい") (VP (REFLEXIVPRONOMEN "sich" :case AKK)
                              (VERB "schämen")))
  ((連体形容詞 "大きな") (ADJEKTIV "groß"))
  ((本動詞 "痛む") (*SUB (PERSONALPRONOMEN "es"))(VP (PARTIKEL "weh")
                                                       (VERB "tun")))
  ((本動詞 "できる") (MODALVERB "können"))
  ((本動詞 "お目にかかる") (*AKK-OBJ (PERSONALPRONOMEN "Sie" :person 2a))
                              (VERB "sehen"))
  ((感動詞 "おかげさま") (FIX-EXP "vielen" (FIX-CAP "Dank")))
  ((感動詞 "おやすみなさいませ") (NP {OHNE} (ADJEKTIV "gute")
                                             (NOMEN "Nacht" :case AKK))
  ((接続詞 "あるいは") (KOORD-KONJUNKTION "oder"))
  :
)
```

上の例で指定した生成情報のうち、AKK-OBJ, SUB, VP, NP, WHは格として認識でき、形態素の並び替えなどを行う際に利用される情報である。FRAGEADVERB, ADJEKTIV, NOMEN, PERSONALPRONOMEN, PARTIKEL,

EIGENNAME, FIX-EXP, FIX-CAP, ADVERB, KOORD-KONJUNKTION は、品詞として認識でき、変化形や派生形、冠詞などを生成する際に利用される情報である。VERB, HILFSVERB, MODALVERB, VERBZUSATZ は、格としても品詞としても認識でき、利用される情報である。{OHNE}, {UNBESTIMMT} は冠詞情報を、また、*AKK-OBJ, *SUB に付いているアスタリスク(*)は、デフォルトであることを示している。

*詳細は [8] を参照のこと。*詳細は付録 A と付録 B を参照のこと。

1.3.3 ドイツ語生成辞書

[目的]

生成処理において、文法的に正しいドイツ語を生成するために、ドイツ語形態素の属性情報を記述する辞書。

[管理者] ドイツ語生成システム担当者

[格納場所] /usr/local/TDMT/tdmt-multi-dev/g-generation/dic/tdmt-jg.*

[書式]

("GERMAN-STRING" "PHONETIC-INFO" "INTONATION" (ATTRIBUTION-INFO))

- GERMAN-STRING ... ドイツ語文字列 *動詞は語幹の表記となる
- PHONETIC-INFO ... ドイツ語音声情報
- STRESS-INFO ... ドイツ語イントネーション情報
- ATTRIBUTION-INFO ... 属性情報

[例]

```
("beweis" "b.④.-v.aI.s." "01" ((L-S "beweis" (QUOTE (200102
  (PPRF . "bewiesen"))))) (:LEXICAL) "00N0n0000;")
("abendessen" "a:.-b.④.n.t.-E.-s.④.n." "1000" ((L-S "abendessen" 130201))
  NIL NIL)
```

*詳細は [8] を参照のこと。

1.4 解析知識

解析処理では、後述の変換知識による処理がしやすいように、入力文の原言語形態素列を加工する。解析知識には、replace-word, lexical-transformation, local-transformation, total-transformation の 4 種類が存在し、実際の処理は必要に応じて下記の順で行なわれる。

replace-word → lexical-transformation → local-transformation(→ total-transformation)

各種解析知識の詳細は、以下の章ごとに説明する。

1.4.1 replace-word

〔目的〕 タギングデータおよび形態素解析結果の表記の揺れを統一し、変換知識のパタン数および用例数を削減するための正規形置換情報。日本語を原言語とする翻訳処理に共通の知識であるため、追加や修正を行なう際は、関係者で協議の上、決定する。

〔管理者〕 正規形置換情報管理者

〔格納場所〕

/usr/local/TDMT/tdmt-multi-dev/j-morph/dic/replace-word-info.lisp

〔書式〕 ((POS REG-EXP) (REVISED-REG-EXP))

- POS ... 日本語品詞シンボル
- REG-EXP ... 日本語正規形
- REVISED-REG-EXP ... 置換後の正規形

〔例〕

```
((サ変名詞 "素泊り") ("素泊まり"))
((サ変名詞 "手続") ("手続き"))
((普通名詞 "サンドウィッチ") ("サンドイッチ"))
((普通名詞 "シャンペン") ("シャンパン"))
((本動詞 "くりかえす") ("繰り返す"))
((本動詞 "したがう") ("従う"))
((補助動詞 "出来る") ("できる"))
((補助動詞 "下さる") ("くださる"))
```

1.4.2 lexical-transformation

〔目的〕 変換処理で形態素を扱いやすくするために、形態素情報の修正(正規形や品詞の変更および形態素の合成・分割など)を行なう知識。

〔管理者〕 解析変換知識作成者

〔格納場所〕

/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/a-data/lexical-*.lisp

〔書式〕

以下のように lexical-transformation は、矢印(=>)の左辺が形態素パタンの条件式、右辺が処理の定義式という構成になっており、それらは1つのルールの中でリスト構造をとる。

```
(define-lexical-transformation RULE-NAME :j-g PRIORITY
  (
    ((MORPH-INFO-LIST-1) (MORPH-INFO-LIST-2) ... (MORPH-INFO-LIST-N))
    =>
    (lex ((ARGUMENT) REVISED-MORPH-INFO-LIST))
  )
  :
)
```

- `define-lexical-transformation` .. 解析知識の定義を宣言
- `RULE-NAME` .. 任意のルール名。ただし、解析知識内でユニークとする。
- `:j-g` .. 翻訳の方向。この場合は日独翻訳を示す。
- `PRIORITY` .. 処理の実行優先番号
- `MORPH-INFO-LIST-1`, `MORPH-INFO-LIST-2`, .. `MORPH-INFO-LIST-N`
.. 形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :pron(読み), :start(発声の開始), :end(発声の終了))
- `lex` ... 形態素情報修正の宣言
- `ARGUMENT`(引数) ... 左辺の形態素列の何番目かを示す番号と任意の文字列の並び。
- `REVISED-MORPH-INFO-LIST` ... 修正後の形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :verb(動作性の単語の動作主))。ただし、
:word と :reg-exp の値は文字列やシンボルで指定するか、引数の形態素列の番号で指定する。
:pos と :conj-form の値はシンボルで指定するか、引数の形態素列の番号で指定する。引数の番号で指定した場合は、該当する形態素が持つ値をコピーする。:all-copyは該当する形態素が持つ:pos, :conj-form, :pronの値をコピーする。:all-copy と :pos, :conj-form の両方を指定した場合は、後の指定が優先される。後述する terminal rule用に合成する場合は、:compoundの指定をこの revised-morph-info-list に入れる。

右辺の処理定義式は、普通は `lex` で宣言するが、左辺で指定した形態素列に修正を加えない形態素が含まれる場合や、形態素の分割のように新しい形態素を加える場合などは、以下の形態素定義式 `define-morph` で形態素情報を定義して呼び出す。また、引数の形態素列の番号を指定する際に、異なるフィールドの情報を用いる場合には、以下の文字列定義式 `define-string` で文字列としての定義を行ってから呼び出す。

```
(define-morph @RULE-NAME :j-g (ARGUMENT)
  REVISED-MORPH-INFO-LIST)
```

```
(define-string $RULE-NAME :j-g (ARGUMENT)
  STRING-INFO)
```

- `define-morph` ... 形態素の定義を宣言
- `define-string` ... 文字列の定義を宣言
- `@RULE-NAME`, `$RULE-NAME` ... 任意の定義式名。形態素の定義と文字列の定義を区別するため、先頭に @ か \$ が必要。解析知識内でユニークとする。
- `:j-g` ... 翻訳の方向。この場合は日独翻訳を示す。
- `ARGUMENT`(引数) ... 左辺の形態素列の何番目かを示す番号と任意の文字列の並び。

- REVISSED-MORPH-INFO-LIST ... 修正後の形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :verb(動作性の単語の動作主))。記述方法は上に同じ。
- STRING-INFO ... 定義に用いる文字列情報 (記述要素 =:word(表層形), :reg-exp(正規形), :verb(動作性の単語の動作主))。記述方法は上に同じ。

[例]

1. 一般的な形態素の合成

```
(define-lexical-transformation gousei-teikokudori :j-g 3
  (
    (((:word . "定刻") (:word . "通り")))
    =>
    (lex ((1 2) (:pos . 普通名詞) (:reg-exp . "定刻通り")))
  )
)
```

分割されている日本語形態素を合成して、一形態素にするルール。例の「定刻通り」は、「通り」の対訳 "Straße" の意味を持たず、二語を合成して初めて "gemäß dem Zeitplan" の対訳を決定することができる。このルールでは、入力文の形態素列を部分的に照合し、1 番めの形態素の表層形が「定刻」、2 番めの形態素の表層形が「通り」であることを形態素パタンの条件としている。パタンの条件が合えば、二語は「普通名詞」として合成され、正規形は「定刻通り」となる。

2. 読み情報を利用した形態素の合成

```
(define-lexical-transformation gousei-tsuitachi :j-g 5
  (
    (((:word . "1") (:word . "日") (:pron . "タチ")))
    =>
    (lex ((1 2) (:pos . 普通名詞) (:reg-exp . "一日<ツイタチ>")))
  )
)
```

日本語形態素の読み情報を参照して、形態素を合成するルール。例の「1日」には、月日 ("der erste") と期間 ("ein Tag") の2つの意味があるため、単に合成すると変換処理で曖昧性が出る。そのため、合成時に読み情報を参照して、区別する。このルールでは、1 番めの形態素の表層形が「1」、2 番めの形態素の表層形が「日」であり、読みが「タチ」の場合を条件としている。条件が合えば、二語は「普通名詞」として合成され、正規形は「一日<ツイタチ>」となる。

現在、読み情報を指定できるのは、以下の13件である。

普通名詞	「方<ハウ>」「方<カタ>」「日<タチ>」「一<ヒト>」 「二<フタ>」「三<ミ>」「風<フウ>」
本動詞	「入れる<ハイレル>」「入れ<ハイレ>」「入れれ<ハイレレ>」 「入れる<イレル>」「入れ<イレ>」「入れれ<イレレ>」

3. 動作主を指定する形態素の合成

```
(define-lexical-transformation gousei-onnisuru :j-g 2
  (
    (((:word . "オン")) ((:word . "に")) ((:reg-exp . "する")))
    =>
    (lex ((1 2 3) (:pos . 本動詞) (:reg-exp . "オンにする")
           (:conj-form . 3) (:verb . 3)))
  )
)
```

主語補完モジュール (1.5.9. 「主語補完」参照) を起動している場合は、動作性の単語には、動作主の情報が付与される。この例であげた、サ変名詞+格助詞+本動詞のように、動作性の単語を2語以上含む形態素列を合成する場合は、それぞれが動作主の情報を保持している。例えば、「オン」が<AGEN G>、「する」が<AGEN S>の情報を保持しているとすると、通常は、最初に出現した動作性の単語の値を自動的にコピーすることになっているので、合成後の本動詞の動作主は、<AGEN G>となる。しかし、この場合は後の単語「する」の動作主情報<AGEN S>を使いたいため、:verbの値で明示的な指定を行なっている。

4. 形態素定義式を用いた形態素の合成

```
(define-lexical-transformation gousei-kotohaaru :j-g 3
  (
    (((:pos . 本動詞)) ((:reg-exp . "こと")) ((:pos . 係助詞))
     (:reg-exp . "ある"))
  )
  =>
  (1 @kotohaaru-1)
)
)
(define-morph @kotohaaru-1 :j-g (2 3 4)
  (:pos . 助動詞) (:conj-form . 4) (:reg-exp . "ことはある"))
```

日本語形態素を合成する際に、別の形態素定義式を呼び出すルール。例の「ことはある」という表現は、変換の都合で合成する必要がある。このルールでは、1番めの形態素が「本動詞」、2番めの形態素の正規形が「こと」、3番めの形態素の品詞が「係助詞」(「は」と「も」が該当する)、4番めの形態素の正規形が「ある」であることを条件としている。処理としては、1番めの形態素に対しては何も行わず、2番め・3番め・4番めの形態素に対して、定義式 @kotohaaru-1 を呼び出して合成を行ない、合成後の品詞を「助動詞」、正規形を「ことはある」、活用形には引数で指定した4番めの形態素の値を用いることを示している。

5. 形態素定義式を用いた形態素の分割

```
(define-lexical-transformation revised-pos-shinaosu/auxv :j-g 1
  (
    (((:reg-exp . "しなosu") (:pos . 補助動詞)))
    =>
    (@revised-suru/auxv @revised-naosu/auxv)
  )
)
(define-morph @revised-suru/auxv :j-g (1)
  (:pos . 補助動詞) (:reg-exp . "する") (:word . "し")
  (:conj-form . 連用))
(define-morph @revised-naosu/auxv :j-g (1)
  (:pos . 助動詞) (:reg-exp . "なosu") (:word . "なosu")
  (:conj-form . 1))
```

日本語形態素を分割するルール。分割する場合は、必ず形態素定義式を呼び出す。例の補助動詞「しなosu」は変換の都合で分割する必要があり、このルールでは、形態素を1語だけ照合し、正規形が「しなosu」、品詞が「補助動詞」であることを条件としている。処理は、分割後の1番めの形態素に対して定義式 @revised-suru/auxv を呼び出し、品詞が「補助動詞」、正規形が「する」、表層形が「し」、活用形が「連用」形である形態素を定義している。2番めの形態素に対しては定義式 @revised-naosu/auxv を呼び出し、品詞が「助動詞」、正規形が「なosu」、表層形が「なosu」である形態素を定義し、活用形には引数で指定した番号の形態素(「しなosu」)の値を用いることを示している。

6. 一般的な形態素の除去

```
(define-lexical-transformation delete-ne-1 :j-g 7
  (
    (((:pos . 接続詞)) ((:pos . 終助詞) (:reg-exp . "ね")))
    =>
    (1)
  )
)
```

unnecessary Japanese morphemes to be removed. Example: as above, the connector followed by the long "ne" is easily removed. In this rule, the first morpheme is the connector, the second is the final particle "ne".

7. 発声の終了情報を利用した形態素の除去

```
(define-lexical-transformation delete-desune :j-g 7
  (
    (((:pos . 格助詞)) ((:pos . 助動詞) (:reg-exp . "ですね") (:end . nil)))
    =>
    (1)
  )
)
```

発声の終了情報を利用して、不必要な日本語形態素を除去するルール。例のように、発声の途中に出現する冗長な「ですね」は、格関係の構造を容易に導くために除去したいが、発声の終わりの「ですね」については、変換で利用したい。このような場合に、発声の終了情報を条件に入れることができる。このルールでは、1番めの形態素が格助詞、2番めの形態素が助動詞「ですね」であり、しかも発声の終了情報を保持していない場合に、2番めの「ですね」を除去している。保持する場合は (:end . t)、保持しない場合は (:end . nil) で表わせる。(:end . t) の情報は、形態素の除去を行なうと、自動的に直前の形態素へとコピーされる。また、発声の開始情報 (:start . t) (:start . nil) についても、同様の処理が行なえる。

8. 一般的な正規形の修正

```
(define-lexical-transformation revised-reg-nn :j-g 1
  (
    (((:word . "ん") (:pos . 準体助詞)))
    =>
    (lex ((1) (:all-copy . 1) (:reg-exp . "の") (:word . "の")))
  )
)
```

日本語形態素の正規形を修正するルール。変換知識でのルールや用例を集約するために、機能語はなるべく正規化するのが望ましい。このルールでは、形態素1語を照合し、表層形が「ん」、品詞が準体助詞である場合に、まず、:all-copyでその形態素情報の:posと:conj-formの値をコピーし、その後で正規形と表層形を「の」に修正している。

9. 活用形を利用した正規形の修正

```
(define-lexical-transformation revised-reg-iru-ichidan :j-g 1
  (
    (((:word . "い") (:reg-exp . "いる") (:conj-form . ない)))
    =>
    (lex ((1) (:all-copy . 1) (:reg-exp . "いるー")))
  )
)
```

活用形を参照して、日本語形態素の正規形を修正するルール。例の「いる」には、上一段活用の「居る」と五段活用の「要る・射る」などの意味があるため、区別するために正規形を修正する。このルールでは、形態素を1語だけ照合し、表層形が「い」、正規形が「いる」、活用形が「ない」形である場合に、:all-copyでその形態素情報の:posと:conj-formの値をコピーし、その後で正規形を「いるー」に修正している。

10. 形態素定義式を用いた正規形修正

```
(define-lexical-transformation sahen-verb-itasu :j-g 2
  (
    (((:pos . サ変名詞)) (:reg-exp . "いたす"))
    =>
    (@add-suru2 2)
  )
)
(define-morph @add-suru2 :j-g (1 する)
  (:all-copy . 1) (:conj-form . 基本) (:pos . 本動詞) )
```

正規形を修正する際に、別の形態素定義式を呼び出すルール。サ変名詞は変換の都合で本動詞化する必要がある。このルールでは、1番めの形態素の品詞が「サ変名詞」、2番めの形態素の正規形が「いたす」であることを条件としている。処理は、1番めの形態素に対して定義式 @add-suru2 を呼び出し、その形態素情報と文字列「する」を用いて合成を行ない、活用形を「基本」形、品詞を「本動詞」に修正している。2番めの形態素に対しては、修正を行なわない。

11. 一般的な terminal rule 用の合成

```
(define-lexical-transformation gousei-n-haku-suru :j-g 4
  (
    (((:pos . 数詞)) (:word . "泊")) (:reg-exp . "する"))
    =>
    (lex ((1 2 3) (:pos . 本動詞) (:reg-exp . "N泊する")
          (:compound 1 2 3) (:conj-form . 3)))
  )
)
```

日本語形態素を合成して、変項を持つ形態素を作成するルール。「一泊する」「二泊する」など、数詞を含む表現は、単に形態素を合成すると、無限にパターンを用意しなくてはならない。そのため、形態素の合成を行なうが、数量表現部分は内部処理として変換を行ないたい。このような場合には terminal rule と呼ばれる変換処理を用い、そのための特別な合成であることを :compound で指定する。このルールでは、1番めの形態素の品詞が「数詞」、2番めの形態素の表層形が「泊」、3番めの形態素の正規形が「する」である場合に、terminal rule 用に三語を本動詞として合成している。合成後の正規形は「N泊する」、活用形は引数で指定した3番めの形態素(「する」)の値を用いることを定義している。

12. 文字列定義式を用いた terminal rule 用合成

```
(define-lexical-transformation gousei-adj-me :j-g 5
  (
    (((:pos . 形容詞)) (:word . "め") (:pos . 接尾辞))
    =>
    (lex ((1 2) (:pos . 形容名詞) (:reg-exp $word-1 2) (:compound 1 2)))
  )
)
(define-string $word-1 :j-g (1)
  (:word . 1))
```

terminal rule 用の合成を行なう際に、別の文字列定義式を呼び出すルール。形容詞+接尾辞「め」は変換の都合で合成する必要があり、しかも派生形への変換を行ないたい。そのため、1番めの形態素の品詞が「形容詞」、2番めの形態素の表層形が「め」、品詞が「接尾辞」である場合に、terminal rule用に二語を「形容名詞」として合成:compound()している。合成後の正規形には、引数で指定した1番めの形態素の表層形の値と2番めの形態素の正規形の値を使うため、1番めの形態素の表層形を文字列として定義した\$word-1を呼び出している。

〔優先順位 (priority) 別処理一覧〕

* 優先順位は、数字の小さい方が先に処理が行われる。優先度に揺れがある場合は、括弧内に揺れの範囲を示す。

<優先順位 1 >

- 形態素解析結果の誤り修正
- 懸案事項対処のための形態素除去 (接頭辞)
- 懸案事項対処または変換の都合のための品詞変更
- 変換の都合のための活用形変更
- 数詞の terminal rule 用合成
- 一般的な正規形修正

<優先順位 2 >

<優先順位 3 >

- モダリティ表現の正規形修正 (優先順位 2 - 4)
- 一般的な形態素合成 (優先順位 2 - 4)

<優先順位 4 >

<優先順位 5 >

- 一般的な terminal rule 用合成 (優先順位 4 - 7)
- 特別な形態素合成 (優先順位 5 - 6)

<優先順位 6 >

<優先順位 7 >

- 形態素除去 (接頭辞以外)

1.4.3 local-transformation

〔目的〕 変換処理で構造を導きやすくするために、特定の形態素間の境界を表すマーカの挿入を行なう。

〔管理者〕 解析変換知識作成者

〔格納場所〕

/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/a-data/local-*.lisp

〔書式〕 以下のように local-transformation は、矢印(=>)の左辺が形態素パタンの条件式、右辺が処理の定義式という構成になっており、それらは1つのルールの中でリスト構造をとる。

```
(define-local-transformation RULE-NAME :j-g PRIORITY
  (
    ((MORPH-INFO-LIST-1) (MORPH-INFO-LIST-2) ... (MORPH-INFO-LIST-N))
    =>
    (ARGUMENT & MARKER)
  )
  :
)
```

- define-local-transformation .. 解析知識のパターン定義の宣言
- RULE-NAME .. 任意のルール名。ただし、解析知識内でユニークとする。
- :j-g .. 翻訳の方向。この場合は日独翻訳を示す。
- PRIORITY ... 処理の実行優先番号
- MORPH-INFO-LIST-1, MORPH-INFO-LIST-2, .. MORPH-INFO-LIST-N .. 形態素情報のリスト (記述要素 =:word(表層形), :reg-exp(正規形), :pos(品詞), :conj-form(活用形), :pron(読み), :start(発声の開始), :end(発声の終了))
- ARGUMENT & MARKER(引数とマーカ) ... マーカの挿入位置を示した、左辺の形態素列。

〔例〕

1. 前後に隣接する品詞情報を示すマーカ

```
(define-local-transformation cn-v :j-g 3
  (
    (((:pos . 普通名詞)) ((:pos . 本動詞)))
    =>
    (1 <cn-v> 2)
  )
)
```

内容語である「普通名詞」と「本動詞」が連接する場合には、変換処理を行なう際に境界を示すマーカが必要になる。このルールでは、1番めの形態素の品詞が「普通名詞」、2番めの形態素の品詞が「本動詞」である場合、二語の形態素の間にマーカ <cn-v> を挿入している。

同種のマーカは、<adv-adj>, <adv-adjnoun>, <adv-adv>, <adv-cn>, <adv-pron>, <adv-sn>, <adv-v>, <cn-adj>, <cn-adjnoun>, <cn-adv>, <cn-cn>,

<cn-conj>, <cn-conjadv>, <cn-exp>, <cn-num>, <cn-pron>, <cn-rentai>, <cn-rentaiadj>, <cn-roman>, <cn-sadjnoun>, <cn-sn>, <cn-v>, <fn-adv>, <fn-cn>, <fn-exp>, <fn-pron>, <fn-sn>, <fn-v>, <haku-nichi>, <hi-you>, <juntaip-adv>, <na-adj>, <na-cn>, <na-fn>, <na-fn-auxv>, <na-sn>, <num-cn>, <num-num>, <num-pron>, <num-youbi>, <personpnom-adv>, <personpnom-cn>, <personpnom-exp>, <personpnom-sn>, <personpnom-v>, <pron-adv>, <pron-cn>, <pron-exp>, <pron-rentai>, <pron-sn>, <pron-v>, <rentaiadj-adjnoun>, <rentaiadj-cn>, <roman-roman>, <sn-adjnoun>, <sn-cn>, <sn-exp>, <sn-fn>, <sn-num>, <sn-pron>, <sn-sn>, <sn-v>, <subpatch> の 62 種類がある (平成 11 年 8 月 31 日現在)。

2. 統一的なマーカ

```
(define-local-transformation interj-sahennoun :j-g 3
  (
    (((:pos . 感動詞)) ((:pos . サ変名詞)))
    =>
    (1 <exp-> 2)
  )
)
```

後続する品詞を問わずに、特定の品詞＋後続表現という形の統一的なマーカを挿入するルール。このルールでは、1 番めの形態素の品詞が「感動詞」、2 番めの形態素の品詞が「サ変名詞」である場合にマーカを挿入しているが、後接する形態素の品詞を特定しなくても原言語構文解析に曖昧性が出ない場合は、後接する形態素の情報を省略したマーカ <exp-> を用いることができる。このマーカを利用することにより、変換知識のパターンや用例の集約化がはかれる。

同種のマーカは、<subp->, <sadjnoun->, <roman->, <exp->, <endp->, <conjpp->, <conjadv->, <auxv->, <adjnoun->, <adv-> の 10 種類がある (平成 11 年 8 月 31 日現在)。

3. 個別な情報を示すマーカ

```
(define-local-transformation /kihon-cn :j-g 2
  (
    (((:conj-form . 基本)) ((:pos . 普通名詞)))
    =>
    (1 </kihon-cn> 2)
  )
)
```

個別な情報を示すマーカを挿入するルール。連体修飾のように、品詞情報だけで構造を作れない表現については、活用形やその他の決めてとなる情報をマーカとして挿入する。このルールでは、1 番めの形態素の活用形が「基本」形、2 番めの形態素の品詞が「普通名詞」である場合に、マーカ </kihon-cn> を挿入している。

これらのマーカのうち、「基本」形を示すマーカは、</kihon-adj>, </kihon-adjnoun>, </kihon-cn>, </kihon-fn-auxv>, </kihon-fn>, </kihon-pron>, </kihon-rentai>, </kihon-sn>, </kihon-v>, <adj/kihon-cn>, <adj/kihon-fn-auxv>, <adj/kihon-fn>, <adj/kihon-sn>, <adj/kihon-v>

の14種類、「連用」形を示すマークは、<adj/renyou-adjnoun>、<adj/renyou-adv>、<adj/renyou-cn>、<adj/renyou-exp>、<adj/renyou-sn>、<adj/renyou-v>、<v/renyou-adjnoun>、<v/renyou-exp>、<v/renyou-fn>、<v/renyou-sn>、<auxv/renyou->、の11種類、その他、『連体助詞「な」+体言』の<na-cn>、<na-fn-auxv>、<na-fn>、<na-sn> 助動詞「そう(様態)」と準体助動詞「そう(伝聞)」を区別する<sou/auxv>、<sou/jauxv>、
「予定」処理用の<sn-yotei-pred>、<no-yotei-pred>、</kihon-yotei-auxv> 文末情報を利用する<END> 依頼文を作るための「命令」の活用形を情報とする<v/imp>の36種類がある(平成11年8月31日現在)。

〔マーク挿入の基本方針〕

マーク挿入の基本方針は、内容語か機能語かによって異なる。

内容語とは、単独で意味を持つ語で、通常は意味コード辞書に意味コードが登録されている。変換知識においては、用例として扱われることが多い。一方、機能語は、単独では意味を持たない語であり、通常は意味コードがふられていない。以下に内容語と機能語の品詞シンボルを示す。

<内容語>

代名詞／普通名詞／サ変名詞／サ変形容名詞／形式名詞／形容名詞／形容詞
／連体形容詞／本動詞／感動詞／ローマ字／副詞／接続副詞／数詞／準数詞

<機能語>

接頭辞／接続詞／連体詞／副助詞／接尾辞／人称接尾辞／助動詞／準体助動詞
／係助詞／格助詞／連体助詞／接続助詞／並立助詞／準体助詞／終助詞／
判定詞／補助動詞

内容語同士が接続する場合は、形態素間に品詞情報を示すマークを必ず挿入する。ただし、ただし、本動詞・形容詞が基本形または連用形の活用をする場合は、活用形情報を示すマークの挿入処理を先行する。また、『基本形+「予定」』『サ変名詞+「予定」』が述語となる表現については、個別なマークを挿入する。

機能語の場合は、品詞の組合せや条件によってマークを挿入するかどうか異なる。

接頭辞／接続詞／連体詞

* 内容語が前接する場合にマークを挿入する

副助詞／接尾辞／人称接尾辞

* 内容語または接頭辞、接続詞、連体詞が後接する場合にマークを挿入する

助動詞／補助動詞／準体助動詞

* 基本形および連用形の活用をする場合に、活用形情報を示すマークを挿入する

判定詞／終助詞／助動詞／補助動詞／準体助動詞

* 文末に位置し、かつ後続する文がある場合に、後続する文の先頭との間にマークを挿入する

助動詞「そう(様態)」／準体助動詞「そう(伝聞)」／『形容名詞＋連体助詞「な」』の表現／準体助詞＋「予定」』が述語となる表現
* 個別なマーカを入れる

係助詞／格助詞／連体助詞／接続助詞／並立助詞／準体助詞
* 個別なマーカを挿入する例を除いては、前後がいかなる品詞であってもマーカを挿入しない

<優先順位 1> • 形容詞の活用形を示すマーカの挿入

<優先順位 2>

- 基本形を示すマーカの挿入(ただし、「基本形＋サ変名詞」の場合を除く)
- 「基本形＋サ変名詞」が述語となる場合の個別なマーカの挿入
- 『基本形＋「予定」』『サ変名詞＋「予定」』が述語となる場合の個別なマーカの挿入
- 形容名詞＋「な」と後続する形態素の間の個別なマーカの挿入

<優先順位 3>

- 一般的な品詞情報を示すマーカの挿入
- 連用形の情報を示すマーカの挿入
- 「基本形＋サ変名詞」を示すマーカの挿入

<優先順位 4>

- 助動詞類と後続する文の先頭との間に品詞情報を示すマーカの挿入

<優先順位 5>

- 『準体助詞＋「予定」』が述語となる場合の個別なマーカの挿入

1.4.4 total-transformation

〔目的〕 1個以上の形態素列を表わす変数を用いて、局所的な形態素の接続と言語表現全体との共起関係を条件として、形態素の合成やマーカの挿入処理を行なう知識。平成11年8月31日時点での訓練単位のデフォルトは、文単位から発声単位に変わっているので、使用にあたっては、注意が必要である。現在、日独翻訳では使用していない。

理由：原言語構文解析をプライオリティで制御していた頃、構造の曖昧性を減らすために、この解析ルールを用いて文全体をパターンマッチングし、形態素の移動などを行なっていたが、現在は原言語構文解析をカテゴリで制御し、下部構造を制限することで同等の目的を達成できるようになったため、使用をやめた。また、文の先頭を示すマーカの挿入を行なうために用いることも可能であるが、その情報を変換処理で利用するような事例がない。(使用例については「TDMT 解析変換知識作成の手引(日英版)」参照)。

1.5 変換知識

1.5.1 概要

〔目的〕 この翻訳システムの中心モジュールである変換処理を行なうために参照される。解析処理の済んだ入力文は、変換パタンの用例との意味距離を構成要素ごとに計算され、最小距離のボタンを用いて目的言語への構造変換が行なわれる。変換の際には、生成処理用の情報が付加され、出力データは生成処理へと手渡される。

〔管理者〕 解析変換知識作成者

〔格納場所〕 /usr/local/TDMT/tdmt-multi-dev/transfer/j-g/p-data/*.lisp

〔書式〕

```
(define-pattern RULE-NAME :j-g CATEGORY
  (SOURCE-PATTERN) (:head N)[(CATEGORY-CONTROL)]
=>
  (((TARGET-PATTERN) ([GENERATION-INFO]))
   (
     (EXAMPLES-LIST)
   )
   [LOCAL-DIC-LIST]
  )
)
```

- `define-pattern ...` 変換パターンを定義するための宣言。他に、`define-regular-pattern`がある。
- `RULE-NAME ...` 任意のルール名。ただし、変換知識内でユニークとする。
- `:j-g ...` 翻訳の方向。この場合は日独翻訳を示す。
- `CATEGORY ...` カテゴリ (原言語構造の分類名)
- `SOURCE-PATTERN ...` 原言語パターン (記述要素：変項・マーカ・日本語形態素)
- `:head N ...` ヘッドの指定 (記述要素：数値)
- `CATEGORY-CONTROL ...` 下部構造の制限 (記述要素：変項・カテゴリ名・品詞・構造分類用の集合体名)。省略可。
- `TARGET-PATTERN ...` 目的言語パターン (記述要素：変項・生成マーカ・独語形態素)
- `GENERATION-INFO ...` 生成情報 (記述要素：生成マーカと数値)。省略可。
- `EXAMPLES-LIST ...` 用例 (記述要素：日本語形態素)
- `LOCAL-DIC-LIST ...` ローカル辞書 (記述要素：変項・品詞・生成マーカ・日独形態素)。省略可。

1.5.2 原言語パターン／目的言語パターン／用例

変換知識は、原言語表現から目的言語表現への変換をパターン形式で記述する。下の変換パターンの例では、(?x "に" ?y) が原言語パターン("に"が固定項、?x と?y が変項)であり、(!y "in" !x) や (!y "zu" !x)... は目的言語パターンである。原言語パターンと目的言語パターンの対応関係は、必ずしも 1 対 1 ではなく、原言語パターン 1 つに対して、複数の目的言語パターンが考えられることが多い。そのパターンを選択する際の鍵となるのが用例である。入力文は、意味コード辞書(「1.3.1 意味コード辞書」参照)を用いて用例との類似度が計算され、意味的に最も近い用例を持つ目的言語パターンが対訳として選ばれる。

```
(define-pattern kakujo-ni-np :j-g np
  (?x "に" ?y) (:head 3) (:x group-n)
  =>
  (((!y "in" !x)(PLACE (PRAEPOSITION 2) 3 :case DAT))
   (
     (("東京ホテル") ("滞在する"))
     (("町なか") ("在る"))
   )
  )
  (((!y "zu" !x)(PLACE (PRAEPOSITION 2) 3 :case DAT))
   (
     (("ホテル") ("お持ちする"))
     (("ソウル駅") ("行く"))
   )
  )
)
```

例えば、ここで、「旅館に泊まる」という入力文があったとする。変項?x を「旅館」、?y を「泊まる」として、各用例との意味距離を計算する。その結果、入力文「旅館に泊まる」は(("東京ホテル") ("滞在する")) の用例と距離が近いということが導き出されたとすると、目的言語パターン(!y "in" !x) が選ばれる。(!y "in" !x) は、原言語の ?x の部分と ?y の部分を入れ替え、固定項の"に"を"in"に変換することを意味する。「旅館」「泊まる」という形態素単位の変換は、日独変換辞書(「1.3.2 変換辞書」参照)から対訳を得て行なわれる。(!y "in" !x) の横に記述されている(PLACE (PRAEPOSITION 2) 3 :case DAT) は、場所格を表す生成情報である(「1.5.8 生成情報」参照)。

このような具体的な内容を伴う用例に対して、いかなる入力文にもマッチする空用例((""))がある。以下の例のように、原言語パターンと目的言語パターンが 1 対 1 で対応するなど、構造変換に大きな影響が出ないと思われる場合は、この空用例を用いることができる。ただし、空用例と具体的な用例では、具体的な用例のほうが優先され、正確な意味距離計算が行なえるため、正しい対訳を導き出すことができる。例にある 2 ヘッドおよび 3 ヘッド用の空用例については、1.5.6 「ヘッド」で説明する。

```
(define-pattern conjunction-shikashi :j-g is
  ("しかし" ?x) (:head 2)
  =>
  ((("aber" !x)(INTRO (KOORD-KONJUNKTION 1)))
   (
    ((""))
    (("" "")) ← 2ヘッド用の空用例
    (("" "" "")) ← 3ヘッド用の空用例 (注: 9908 現在3ヘッドの使用例なし)
   )
  )
)
```

1.5.3 パタンの種類

変換パタンの定義には2種類あり、原言語側の形態素やマーカをすべて表層形のままでマッチングする場合は `define-pattern`、すべて正規形でマッチングする場合は `define-regular-pattern` を用いる。以下の例のように、`define-regular-pattern` で定義すると、正規形が「ていただく」である活用形(「ていただく」「ていただき」..)のすべてに対応できる。また、解析知識で正規形を「ていただく」に修正している形態素があれば、それについても同様の処理が行なえる。

```
(define-regular-pattern modal-teitadaku-reg-neu :j-g pm
  (?x "ていただく") (:head 1) ← 「ていただく」「ていただき」.. とマッチングする
  =>
  (((!x))
   (
    (("教える"))
    (("支払う"))
   )
  )
)
```

また、マーカについても正規形 <> が用意されており、前後の形態素の種類に関わらず、共通に用いることができる。

```
(define-regular-pattern setsubi-tsuki-reg :j-g nd
  (?x <> "付") (:head 3) ← マーカの正規形 <>
  =>
  ((("mit" !x)(PP (PRAEPOSITION 1) {OHNE} 2 :case DAT))
   (
    (("バス")) ← 普通名詞
    (("食事")) ← サ変名詞
   )
  )
)
```

1.5.4 カテゴリ

各パタンの定義で指定するカテゴリは、原言語構文解析を行なう際の構造分類名であり、以下の18種類がある。日英・日独・日韓など、同じ原言語を扱う翻訳処理においては共通とする。

`is`(接続詞・感動詞・呼びかけなど、文全体を修飾する語のパタン)

`ss`(重文 / target 側で接続詞を使う複文のパタン)

se(文末の接続助詞・終助詞のパタン)

sm(文を受けるモダリティ表現／文末表現のパタン)

np(体言を格要素とする格関係のパタン)

ap(副詞・副詞句を格要素とする格関係のパタン)

sp(用言を格要素とする格関係のパタン)

aa(副詞句のパタン)

pm(用言を受けるモダリティ表現のパタン)

nm(体言を受けるモダリティ表現のパタン)

s+m(連体修飾用言を受けるモダリティ表現のパタン)

s+n(連体修飾用言を受ける体言のパタン)

n+n(並立助詞・名詞連続・名詞句のパタン)

dn(接頭辞・連体詞／名詞を修飾する副詞句のパタン)

nd(接尾辞・副助詞／名詞に後接する語のパタン)

n(複合名詞のパタン)

p(合成用言のパタン)

terminal(数詞・接頭辞を伴う類出表現／派生形のパタン)

注) カテゴリ terminalのパタンは、解析処理において:compoundの指定をして合成された語句のみに適用される。既に一つの形態素として合成されているため、構造を作る際には形態素としての扱いを受ける。

<原言語構文解析の例>

以下のように原言語構文解析が行なわれ、適用されたパタンの木構造が得られる。

"はい三宮駅から南に出てください"

```
TOP [(?X <EXP-> ?Y) --- IS]
  |--?X [(はい)]
  |
  |--?Y [(?X てください) --- SM]
    |--?X [(?X から ?Y) --- NP]
      |--?X [(三宮駅)]
      |
      |--?Y [(?X に ?Y) --- NP]
        |--?X [(南)]
        |
        |--?Y [(出)]
```

"眺めのいい部屋がよろしいですか"

```
TOP [(?X です か) --- SM]
  |--?X [(?X が ?Y) --- NP]
    |--?X [(?X <ADJ/KIHON-CN> ?Y) --- S+N]
      | |--?X [(眺めのいい)]
      | |
      | |--?Y [(部屋)]
      |
      |--?Y [(よろしい)]
```

"電話番号は零三四四三一七零零です"

```
TOP [(?X です) --- SM]
  |--?X [(?X は ?Y) --- NP]
    |--?X [(電話番号)]
    |
    |--?Y [(0344431700) --- compound]
```

* 1.5.9 にて「カテゴリ別変換パターン定義の基本方針とパターン例」を述べる

1.5.5 下部構造の制限

原言語構造を決定する際、ありえない構造を排除して構造の曖昧性の数を抑制するため、パターンごとに下部構造を制限することができる。下部構造の制限は、カテゴリや個々の形態素の品詞名で指定するが、それらを新たにグループ化した集合体(以下の7種類)を用いると簡潔に記述できる。

category-all(全形態素および全構造)

word-all(全形態素)

group-n(体言性の形態素および構造)

word-n(体言性の形態素)

group-p(用言性の形態素および構造)

word-p(用言性の形態素)

group-s(文として認知可能な形態素および構造)

この集合体のメンバは /usr/local/TDMT/tdmt-multi-dev/transfer/j-g/dic/category-jg.lispにて定義されている。これについても、同じ原言語を扱う翻訳処理においては共通である。

[書式]

```
(define-category-set GROUP-NAME :j-g  
  MEMBER)
```

- define-category-set ... 集合体を定義する
- GROUP-NAME ... 集合体の名前
- :j-g ... 翻訳の方向を示す。この場合は日独翻訳。
- MEMBER ... 集合体のメンバ(記述要素: カテゴリ・品詞シンボル)

[例]

```
(define-category-set group-n :j-g  
  s+n n+n dn nd n 普通名詞 固有名詞 代名詞 サ変名詞 形容名詞 サ変  
  形容名詞 形式名詞  
  数詞 準数詞 ローマ字 副詞 接続副詞)
```

各カテゴリが取り得る下部構造は、同じく category-jg.lispに以下のように定義されている。

[書式]

```
(define-category CATEGORY-NAME :j-g  
  MEMBER)
```

- define-category ... カテゴリの定義を宣言する
- CATEGORY-NAME ... カテゴリの名前

- :j-g ... 翻訳の方向を示す。この場合は日独翻訳。
- MEMBER ... カテゴリのメンバ(記述要素：カテゴリ・品詞シンボル・集合体の名前)

[例]

```
(define-category np :j-g
  np sp ap pm nm s+m p word-p group-n 副詞 接続副詞)
```

例に上げたカテゴリ np は、下部構造として np, sp, ap, ... の構造の他、 word-p, group-n で指定された構造や語句、さらに副詞・接続副詞を取り得る。変換パターンでは、これに対して制限を加え、構造の曖昧性を排除する。以下のパターンでは、(:x group-n) の指定を入れることにより、変項 x が取り得る構造や語句を group-n のメンバに制限している。

```
(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y !x)(AKK-OBJ 2))
   (
    (("手紙") ("受け取る"))
    (("財布") ("なくす"))
   )
  )
)
```

1.5.6 ヘッド

原言語構造を決定する際、各構造において中心となる構成要素をヘッドと言い、その構造を上部の構造へ伝える重要な働きをする。指定するヘッドは、変項であっても固定項であっても構わないが、機能語単独ではヘッドになり得ない。ただし、上部の構造の目的言語パターンを決定するのに必要であれば、機能語が内容語を伴ってヘッドとなることがある。以下のパターンでは、(:head 1 2) を指定することにより、変項?x の内容語と機能語「れる」の2つがヘッドとして上部構造に伝搬される。

```
(define-regular-pattern modal-reru-reg :j-g pm
  (?x "れる") (:head 1 2)
  =>
  (((!x)(ATTR 1 :voice PASSIV))
   (
    ("盗む")
   )
  )
)
```

ここで、「財布を盗まれました」という入力文があり、以下のように原言語構造解析が行なわれたとする。

```

TOP [(?X ました) --- SM]
  |--(?X [(?X を ?Y) --- NP]
    |--(?X [(財布)]
      |
      |--(?Y [(?X れる) --- PM]
        |--(?X [(盗ま)]

```

この文の場合は、(?x "れる") の上部構造 (?x "を" ?y) において、機能語「れる」の情報が必要になるため、2ヘッドの指定を行っている。以下のパタンのように、「財布を盗む (einen Geldbeutel stehlen)」の場合「財布」は「盗む」の4格目的語となるが、「財布を盗まれる (ein Geldbeutel wird gestohlen)」の場合は、「財布」を主語とする受動態を生成しなくてはいけないからである。

```

(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!x !y)(SUB 1)) ;MODAL の rule で passiv を定義
  (
    (("財布") ("盗む" "れる"))
    :
  )
  )
  )
  (((!y !x)(AKK-OBJ 2))
  (
    (("財布") ("盗む"))
    :
  )
  )
  )
  )

```

このように、(?x "れる") で指定された2ヘッド用例は、すぐ上部の構造 (?x "を" ?y) に伝搬されたあと、さらにそのまた上部の (?x "まし" "た") へ、2ヘッドのまま伝搬されるため、以下のように記述する。また、空用例 (「1.5.2 原言語パターン/目的言語パターン/用例」参照) の場合にも、Nヘッド用のものが用意されている。

```

(define-pattern sen-mashita :j-g sm
  (?x "まし" "た") (:head 1)
  =>
  (((!x)(ATTR 1 :tense PERFEKT))
  (
    (("盗む" "れる"))
    :
  )
  )
  )
  )
  )

```

上記の事例は、機能語部分が格を決定するために必要になる場合であるが、その他、使役の表現で訳し分けのために必要になる場合もある。例えば、以下の (?X "を" ?Y) のパターンにおいて、「何う」の対訳 "sich erkundigen" は機能語の「せる」があるから、ローカル辞書 (「1.5.7 ローカル辞書」参照) によって "schicken" に変更することができるのである。

```
(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y !x)(AKK-OBJ 2))
   (
    (("者") ("何う" "せる"))
   )
   (
    (y
     ((本動詞 "何う")(VERB "schicken"))
    )
   )
  )
)
```

以上の2つの事例のように、Nヘッ드의指定をする機能語部分は、目的言語パタンの格決定や対訳を記述する際に必要になる情報なのである。

1.5.7 ローカル辞書

「1.3.2 日独変換辞書」で定義された対訳はデフォルト値であり、変換知識のパタン中でローカルに別の対訳を与えることができる。例えば、変換辞書の中で本動詞「利用する」の対訳は”verwenden”と定義しているが、「送迎バスを利用する」などの場合には、”nehmen”を用いたい。このような場合は、以下のようにパタンごとの訳し分けを行なう。このローカル辞書は、上部構造で記述されるものほど優先される。

【書式】

```
((VARIABLE ((POS REG-EXP) TARGET)))
```

- VARIABLE ... 変数
- POS ... 日本語品詞シンボル
- REG-EXP ... 日本語正規形
- TARGET ... 生成情報と対訳のリスト

【例】

```
(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y !x)(AKK-OBJ 2))
   (
    (("送迎バス") ("利用する"))
    (("飛行機") ("利用する" "れる"))
    (("タクシー") ("利用する" "ていただける"))
    (("急行") ("利用する" "ていただける"))
    (("外回り") ("利用する"))
   )
   (
    (y
     ((本動詞 "利用する")(VERB "nehmen"))
    )
   )
  )
)
```

1.5.8 生成情報

各構造ごとに目的言語パターンが選ばれ、形態素レベルでの変換処理が済んだ入力文は、生成処理に回されるが、その際、必要に応じて生成情報が付与される。生成処理では、その情報を用いて、文法的に正しく、かつ、より自然なドイツ語を出力するため、語の並べ替え・活用形の生成・冠詞の挿入などを行なう。例えば、「日曜日は開いていますか」という入力文は、変換処理の後、以下のような形態素列データとして生成処理へ渡され、最終的に "Ist es sonntags geöffnet?" という翻訳結果が得られる。

*** Generation ***

```
((FRAGE (*SUB (PERSONALPRONOMEN "er/sie/es")) (ATTR (ADVERB "sonntags")
(|<AGEN Y>| (VERB "öffnen")) :VOICE ZS-PASSIV)))
=>
Ist es sonntags geöffnet?
```

形態素列データの中、対訳の文字列以外 (FRAGE, *SUB, PERSONALPRONOMEN, ATTR, ADVERB, <AGEN Y>, VERB, VOICE ZS-PASSIV) は、すべて生成情報である。先頭にある FRAGE は文の法性を決定する「文ケース」であり、この場合には、疑問文としての語順生成が行なわれる。*SUB は、格を決定する「ケース」であり、PERSONALPRONOMEN, ADVERB, VERB は、品詞として認識される「ケース」である。*SUB に付いているアスタリスク (*) は、その格がデフォルトであることを示す。<AGEN Y> は主語補完モジュールが起動している際に付与される動作主である (「1.5.9 主語補完」参照)。また、(ATTR... :VOICE ZS-PASSIV) は態を生成するための「生成マーカ」であり、状態受動や受動態を生成する。これらの生成情報のうち、FRAGE, *SUB, PERSONALPRONOMEN, (ATTR ... :voice ZS-PASSIV), ADVERB は、以下の変換パターン、その付加情報およびローカル辞書として記述されている。また、VERB は、変換辞書の対訳に付加された情報である。

<変換パターン>

```
(define-regular-pattern teimasuka-reg :j-g sm
  (?x "ています" "か") (:head 1)
  =>
  (((("er/sie/es" !x)(FRAGE (*SUB (PERSONALPRONOMEN 1))(ATTR 2 :voice ZS-PASSIV)))
    (
      ("開く")
      :
    )
  )
)
```

```

(define-pattern kakarijo-ha-np :j-g np
  (?x "は" ?y) (:head 3) (:x group-n 副詞)
  =>
  (((!x !y))
   (
    (("日曜日 ") ("開く "))
    :
    )
   (
    (x
     ((普通名詞 "日曜日")(ADVERB "sonntags"))
     :
    )
   )
  )
)

```

<変換辞書>

((本動詞 "開く ") (VERB "öffnen"))

必要に応じて、生成情報を変換パターンおよび変換辞書・ローカル辞書に記述するが、ドイツ語の文法の性質上、細かな記述が求められることが多い。

*詳細は [8] を参照のこと。*詳細は付録 B を参照のこと。

1.5.9 主語補完

日本語からドイツ語に変換する場合には、省略されている主語を補う必要がある。局所的なルールで補完することも可能であるが、精度が良くないため、別途、決定木を用いて言語現象と省略要素との関係を学習させた補完知識から、省略要素（ここでは主語）を統計的に判断し補完する、主語補完モジュールを利用する。

主語補完モジュールを起動している場合は、動作性の単語（本動詞・サ変名詞・サ変形容名詞）に対して動作主の情報が付与される。その動作主の情報は、目的言語パタンのデフォルト主語、*sub に上書きされるため、補完主語の誤り頻度が高い表現の場合は、*sub を用いて、補完主語を上書きするデフォルト主語を指定することができる。

動作主の情報には7種類のタグがあり、それぞれ、以下のように生成される。

- <AGEN S> ... 一人称単数 ("ich")
- <AGEN K> ... 一人称複数 ("wir")
- <AGEN H> ... 二人称敬称単数 ("Sie")
- <AGEN Y> ... 二人称敬称複数 ("Sie")
- <AGEN G> ... 一般 ("man")
- <AGEN F> ... 無生物 ("es")
- <AGEN X> ... 照応 ("er/sie/es")

ただし、<AGEN X> の照応は、同一発声中に先行詞と思える語が見出せる場合は、それを代名詞化して生成し、見出せない場合は、"es"を生成するものとする。

また、解析知識で、動作性の単語を2語以上含む形態素列を合成する場合は、通常、最初に出現した動作性の単語の動作主の情報が優先されるが、後の語の動作主の情報を優先させたい場合、あるいは全く利用せず、機能語扱いにしたい場合は、:verb属性で個別に指定することができる。

(詳細は、「1.4.2.lexical-transformation」を参照のこと。)

[例]

```
(define-regular-pattern sen-masu-reg :j-g sm
  (?x "ます") (:head 1)
  =>
  (((!x))
   (
    (("おっしゃる")
     ("痛む"))
   )
  )
  (((("ich" !x)(*SUB (PERSONALPRONOMEN 1)))
   (
    ("言う")
    ("調べる")
   )
  )
  )
  )
  )
```

上の例では、「おっしゃる」と「痛む」に対しては、主語補完モジュールの補完主語を利用し、「言う」と「調べる」については、ルールのデフォルト主語を指定している。

1.5.10 カテゴリ別変換パターン定義の基本方針とパターン例

*各パターン例の目的言語パターンおよび用例は、格納場所を参照のこと。

1. カテゴリ is(接続詞・感動詞・呼びかけなど、文全体を修飾する語のパターン)
パターン種類：通常 define-pattern
ヘッド：述語の項
変項制限：接続詞などが固定項の場合はなし。変項の場合は、通常何らかの制限をつける。

```
(define-pattern conj-shikashi-is :j-g is
  ("しかし" ?x) (:head 2)
  ...)
```

格納場所： /usr/local/TDMT/tdmt-multi-dev/transfer/j-g/p-data/conj-s.lisp
訓練文例："しかし念のために調べてさしあげましょう"

```
(define-pattern marker-conjadv-is :j-g is
  (?x <conjadv-> ?y) (:head 3) (:x 接続副詞)
  ...)
```


格納場所： /p-data/adv.lisp(p-data までのパスは上に同じ。以下同)
訓練文例： ”またもし奈良などで乗り捨てても三千元ぐらいの追加ですみませね”

```
(define-pattern marker-exp-is :j-g is  
(?x <exp-> ?y) (:head 3) (:x 感動詞 ap np sp)  
...)
```

格納場所： /p-data/marker-exp-x.lisp
訓練文例： ”ありがとうございます箱根リゾートでございます”

```
(define-pattern marker-personpnom-cn-is :j-g is  
(?x <personpnom-cn> ?y) (:head 3) (:x group-n)  
...)
```

格納場所： /p-data/marker-personpnom-x.lisp
訓練文例： ”ブリギッテヴェッカー様シングルで二日間のご利用でしたね”

2. カテゴリ ss(重文 / target 側で接続詞を使う複文のパタン)

パタン種類：通常 define-pattern

ヘッド：後の文の述語の項

変項制限：ドイツ語生成の都合で、通常 (:x np se sm sp ap pm nm s+m p word-p)

```
(define-regular-pattern causal-node-ss :j-g ss  
(?x ”ので” ?y) (:head 3)(:x np se sm sp ap pm nm s+m p word-p)  
...)
```

格納場所： /p-data/causal.lisp
訓練文例： ”最終電車がまだあるかもしれないので調べてみます”

```
(define-pattern condition-ba-ss :j-g ss  
(?x ”ば” ?y) (:head 3)(:x np se sm sp ap pm nm s+m p word-p)  
...)
```

格納場所： /p-data/conditional.lisp
訓練文例： ”もしあればトリプルルームでお願いしたいんですけど”

```
(define-pattern marker-endp-ss :j-g ss  
(?x <endp-> ?y) (:head 3)(:x np se sm np sp ap pm nm s+m p word-p)  
...)
```

格納場所： /p-data/marker-endp-x.lisp
訓練文例： ”金閣寺ですかお客様大阪から京都へはジェイアールでお越しでしょうか”

```
(define-pattern marker-/kihon-cn-ss :j-g ss  
(?x </kihon-cn> ?y) (:head 3) (:x np se sm sp ap pm nm s+m p word-p)  
...)
```

格納場所： /p-data/marker-kihon-cn.lisp
訓練文例： ”十月八日の明後日一名で宿泊予定にしていた佐藤です”

3. カテゴリ se(文末の接続助詞・終助詞のパタン)

パタン種類：通常 define-pattern

ヘッド：述語の項

変項制限：通常 (:x group-s)

```
(define-regular-pattern ending-keredomo-se :j-g se
```

```
(?x "けれども") (:head 1) (:x group-s)
```

```
...)
```

格納場所：/p-data/conj-k.lisp

訓練文例："午後八時ぐらいになってしまうと思うんですけれども"

```
(define-pattern sen-yone :j-g se
```

```
(?x "よね") (:head 1) (:x group-s)
```

```
...)
```

格納場所：/p-data/sen-etc.lisp

訓練文例："料金もシングル料金に変更してもらえますよね"

4. カテゴリ sm(文を受けるモダリティ表現/文末表現のパタン)

パタン種類：define-pattern または define-regular-pattern

ヘッド：述語の項

変項制限：通常なし

```
(define-pattern sen-desuka-sm :j-g sm
```

```
(?x "です" "か") (:head 1)
```

```
...)
```

格納場所：/p-data/sen-question.lisp

訓練文例："何時ごろご到着の予定ですか"

```
(define-regular-pattern teimasu-reg :j-g sm
```

```
(?x "ています") (:head 1)
```

```
...)
```

格納場所：/p-data/modal-te.lisp

訓練文例："向こうのお店で売ってます"

```
(define-pattern ending-kara :j-g sm
```

```
(?x "から") (:head 1)(:x group-s) ...)
```

格納場所：/p-data/ending.lisp

訓練文例："といたしますのも当社の操縦士は大変カメラマンの方々をたくさん乗せて操縦することが多いからです"

```
(define-regular-pattern imp-modal-tekudasai-reg-neu :j-g sm
```

```
(?x "てください") (:head 1) ...)
```

格納場所：/p-data/imperative.lisp

訓練文例："わかりにくいのでタクシーで行ってください"

5. カテゴリ np(体言を格要素とする格関係のパタン)

パタン種類：通常 define-pattern

ヘッド：述語の項

変項制限：通常 (:x group-n)

```
(define-pattern kakujo-ga-np :j-g np
(?x "が" ?y) (:head 3) (:x group-n)
...)
```

格納場所：/p-data/kakujo-ga.lisp

訓練文例："そのホテルの近くには何かありますか"

```
(define-regular-pattern kakujo-kara-np :j-g np
(?x "から" ?y) (:head 3)(:x group-n)
...)
```

格納場所：/p-data/kakujo-kara.lisp

訓練文例："はい学園前からはタクシーをご利用いただけますでしょうか"

```
(define-pattern marker-cn-cn-np :j-g np
(?x <cn-cn> ?y) (:head 3) (:x group-n)
...)
```

格納場所：/p-data/marker-cn-cn.lisp

訓練文例："現在ニューヨークシティホテルにお泊まりですね"

```
(define-pattern kakujo-no-np :j-g np
(?x "の" ?y) (:head 3) (:x group-n)(:y ap pm nm word-p p サ変形容名詞
副詞 サ変名詞)
...)
```

格納場所：/p-data/kakujo-no.lisp

訓練文例："夜景の見える席は空いていますか"

```
(define-pattern conj-desuga-np :j-g np
(?x "です" "が" ?y) (:head 4) (:x group-n)
...)
```

格納場所：/p-data/conj-ga.lisp

訓練文例："それから場所ですが地下鉄四条駅から近くで烏丸通りに面したところですよ"

```
(define-pattern conj-ninotte-np :j-g np
(?x "に" "乗っ" "て" ?y) (:head 5)(:x group-n)
...)
```

格納場所：/p-data/conj-te.lisp

訓練文例："こちら二階建てバスに乗って回っていただくコースになっておりまして金閣寺銀閣寺清水寺をご覧になっていただきます")

6. カテゴリ ap(副詞・副詞句を格要素とする格関係のパターン)

パターン種類：通常 define-pattern

ヘッド：被修飾語の項

変項制限：副詞句が固定項の場合はなし。変項の場合は、通常何らかの制限をつける。

```
(define-pattern adverb-dekireba :j-g ap
  ("できれ" "ば" ?x) (:head 3)
```

...)

格納場所：/p-data/marker-adv.lisp

訓練文例："できればシングルルームをお願いします"

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 aa)
```

...)

格納場所：/p-data/marker-adv.lisp

訓練文例："あいにくこのフライトの席は満席です"

```
(define-regular-pattern kakujo-ni-ap :j-g ap
  (?x "に" ?y) (:head 3) (:x 形容名詞 サ変形容名詞 サ変名詞 副詞 普通名詞
  aa)
```

...)

格納場所：/p-data/kakujo-ni.lisp

訓練文例："できたら早めにチェックインしたいんですが"

7. カテゴリ sp(用言を格要素とする格関係のパターン)

パターン種類：通常 define-pattern

ヘッド：係り先の述語の項

変項制限：通常 (:x group-p)。変項 x にカテゴリ sm の構造がくる場合は (:x group-s)。

```
(define-regular-pattern modal-zuni :j-g sp
  (?x "ずに" <> ?y) (:head 4) (:x group-p)
```

...)

格納場所：/p-data/negation.lisp

訓練文例："そのままきらずにお待ちください"

```
(define-pattern sen-kakadouka-2 :j-g sp
  (?x "かどうか" <endp-> ?y) (:head 4) (:x group-p)
```

...)

格納場所：/p-data/sen-question.lisp

訓練文例："後ろの建物が入ってるかどうか確かめてください"

```
(define-pattern condition-ba-sp :j-g sp
  (?x "ば" ?y) (:head 3) (:x word-p np se sm sp aa ap pm nm s+m p)
```

...)

格納場所： /p-data/conditional.lisp
訓練文例： ”どこに行けばそれはもらえますか”

```
(define-pattern kakujo-to-sp :j-g sp
(?x ”と” ?y) (:head 3)(:x word-p np se sm sp aa ap pm nm s+m p)(:y sp
np ap pm nm s+m p word-p group-n 副詞 接続副詞)
...)
```

格納場所： /p-data/kakujo-to.lisp
訓練文例： ”二十ドルくらいかかると思います”

8. カテゴリ aa(副詞句のパタン)

パタン種類：通常 define-pattern
ヘッド：係り先の項
変項制限：x, y ともに特別な制限が必要。

```
(define-pattern marker-adv-cn-aa :j-g aa
(?x <adv-cn> ?y) (:head 3) (:x 副詞 aa)(:y 形容詞 連体形容詞 形容名詞
...)
```

変形容名詞 副詞 普通名詞
格納場所： /p-data/adv.lisp
訓練文例： ”もう少し北に行ったところにあります”

```
(define-pattern marker-cn-adv-aa :j-g aa
(?x <cn-adv> ?y) (:head 3)(:x aa 普通名詞)(:y 副詞) ...)
```

格納場所： /p-data/marker-cn-x.lisp
訓練文例： ”今すぐ欲しいんですけど”

9. カテゴリ pm(用言を受けるモダリティ表現のパタン)

パタン種類：define-pattern または define-regular-pattern
ヘッド：述語の項。ただし、構文や格の決定に必要な場合は、述語の項と助動詞類の2ヘッド
変項制限：通常なし

```
(define-pattern fukujo-tari-pm :j-g pm
(?x ”たり”) (:head 1)
...)
```

格納場所： /p-data/fukujo.lisp
訓練文例： ”ごめんなさいね怒ったりして”

```
(define-regular-pattern modal-temi-reg :j-g pm
(?x ”てみる”) (:head 1)
...)
```

格納場所： /p-data/modal-te.lisp
訓練文例： ”さがしてみます”

```
(define-regular-pattern modal-seru-reg :j-g pm
(?x "せる") (:head 1 2)
...)
格納場所： /p-data/modal-s.lisp
訓練文例： "お客様のご予算をお聞かせ願えますか"
```

10. カテゴリ nm(体言を受けるモダリティ表現のパタン)
パタン種類： define-pattern または define-regular-pattern
ヘッド： 体現の項。
変項制限： 通常なし

```
(define-pattern modal-datta :j-g nm
(?x "だった") (:head 1)
...)
格納場所： /p-data/sen-kako.lisp
訓練文例： "確か四千五百円ぐらいだったと思います"
```

```
(define-pattern sen-na :j-g nm
(?x "な") (:head 1)
...)
格納場所： /p-data/modal-n.lisp
訓練文例： "安くてきれいなところならいいんですけど"
```

```
(define-regular-pattern modal-ninaru-nm-reg-neu :j-g nm
(?x "になる") (:head 1)
...)
格納場所： /p-data/modal-ninaru
訓練文例： "三百ドルになります"
```

11. カテゴリ s+m(連体修飾用言を受けるモダリティ表現のパタン)
パタン種類： define-pattern または define-regular-pattern
ヘッド： 連体修飾用言の項。変項制限： 通常なし

```
(define-regular-pattern modal-kotoninaru-reg :j-g s+m
(?x "ことになる") (:head 1)
...)
格納場所： /p-data/modal-necessity.lisp
訓練文例： "そうですかそれではキャンセル料をいただくことになりましたがよろしいですか"
```

12. カテゴリ s+n(連体修飾用言を受ける体言のパタン)
パタン種類： 通常 define-pattern
ヘッド： 通常は体言
変項制限： 通常 (:x group-s)(:y group-n)

```
(define-pattern rentaijo-no-s+n :j-g s+n
(?x "の" ?y) (:head 3)(:x group-s)(:y group-n)
...)
```

格納場所： /p-data/rentaijo.lisp

訓練文例： "プラザホテル八百九号室に御滞在のジョンフィリップス様でございますね"

```
(define-pattern v/renyou-fn-shidai :j-g s+n
(?x <v/renyou-fn> "次第") (:head 3)(:x group-s)
...)
```

格納場所： /p-data/marker-fn-x.lisp

訓練文例： "どうぞ手続き終わり次第乗って下さい"

```
(define-pattern marker-na-cn :j-g s+n
(?x <na-cn> ?y) (:head 3) (:x nm ap aa dn) (:y group-n)
...)
```

格納場所： /p-data/marker-na-x.lisp

訓練文例： "有名なデパートが一軒あります"

```
(define-pattern marker-adjnp/kihon-cn :j-g s+n
(?x <adj/kihon-cn> ?y) (:head 3) (:x np) (:y group-n)
...)
```

格納場所： /p-data/marker-adj-x.lisp

訓練文例： "インド人にも定評のあるレストランなんです"

13. カテゴリ n+n(並立助詞・名詞連続・名詞句のパタン)

パタン種類：通常 define-pattern

ヘッド：係り先の項

変項制限：通常なし

```
(define-pattern heiyo-to :j-g n+n
(?x "と" ?y) (:head 3)
...)
```

格納場所： /p-data/heiyo.lisp

訓練文例： "住所とお名前をここに書いてください"

```
(define-pattern marker-cn-cn-n+n :j-g n+n
(?x <cn-cn> ?y) (:head 3)
...)
```

格納場所： /p-data/marker-cn-cn.lisp

訓練文例： "洋服靴アクセサリといったところかな"

```
(define-pattern rentaijo-nosobano-n+n :j-g n+n
(?x "のそばの" ?y) (:head 3)
...)
```

格納場所： /p-data/rentaijo.lisp
訓練文例： ”では下鴨神社のそばの下鴨茶寮がお薦めですね”

```
(define-pattern to-no-z :j-g n+n  
(?x ”と” ?y ”の” ?z) (:head 5)(:x group-n)(:y group-n)  
...)
```

格納場所： /p-data/heiho.lisp
訓練文例： ”そうですかそしたら特急と普通車の値段の違いはいくらですか”

14. カテゴリ dn(接頭辞・連体詞／名詞を修飾する副詞句のパタン)

パタン種類：通常 define-pattern

ヘッド：係り先の項

変項制限：通常なし

```
(define-pattern settou-yaku-dn :j-g dn  
(”約” ?x) (:head 2)  
...)
```

格納場所： /p-data/settou.lisp
訓練文例： ”約二十分で竜安寺前に着きます”

```
(define-pattern rentai-sonoyouna :j-g dn  
(”そのような” ?x) (:head 2)  
...)
```

格納場所： /p-data/rentaishi.lisp
訓練文例： ”そのようなサービスはしていません”

```
(define-pattern adverb-aruite-dn :j-g dn  
(”歩い” ”て” ?x) (:head 3)  
...)
```

格納場所： /p-data/adv.lisp
訓練文例： ”竜安寺から歩いて十五分くらいのところに仁和寺というお寺が在ります”

```
(define-pattern adverb-dn :j-g dn  
(?x <adv-> ?y) (:head 3) (:x 副詞)(:y group-n)  
...)
```

格納場所： /p-data/adv.lisp
訓練文例： ”予約は必ず必要です”

15. カテゴリ nd(接尾辞・副助詞／名詞に後接する語のパタン)

パタン種類：通常 define-pattern

ヘッド：内容語または係り先の項(例外が他数あり)

変項制限：通常なし


```
(define-regular-pattern setsubi-sama :j-g nd
(?x "様") (:head 1)
...)
格納場所： /p-data/setsubi-sama.lisp
訓練文例： "はいフィッシャー様お伺いいたします"
```

```
(define-regular-pattern setsubi-kurai :j-g nd
(?x "くらい") (:head 1)
...)
格納場所： /p-data/setsubi.lisp
訓練文例： "二十ドルくらいかかると思います"
```

```
(define-regular-pattern setsubi-tsuki-reg :j-g nd
(?x <> "付") (:head 3)(:x group-n)
...)
格納場所： /p-data/setsubi.lisp
訓練文例： "バス付きシングルはいくらになりますか"
```

```
(define-regular-pattern setsubi-zengo-reg :j-g nd
(?x <> "前後") (:head 1)(:x group-n)
...)
格納場所： /p-data/setsubi.lisp
訓練文例： "ちょうど二万円前後で考えてたもんですから"
```

16. カテゴリ n(複合名詞のパターン)

パターン種類：通常 define-pattern
ヘッド：係り先の項(例外あり)
変項制限：特別に付ける場合がある。

```
(define-pattern marker-cn-cn-n :j-g n
(?x <cn-cn> ?y) (:head 3)
...)
格納場所： /p-data/marker-cn-cn.lisp
訓練文例： "鈴木直子です"
```

```
(define-pattern marker-cn-cn-n2 :j-g n
(?x <cn-cn> ?y) (:head 1)
...)
格納場所： /p-data/marker-cn-cn.lisp
訓練文例： "はい名前はジョンフィリップスです"
*性別を区別するため、ファーストネームをヘッドとする。
```

```
(define-pattern kakujo-no-n :j-g n
(?x "の" ?y) (:head 3)(:x 数詞 普通名詞 n)(:y 数詞 普通名詞 サ変名詞 n)
...)
```

格納場所： /p-data/rentaijo.lisp
訓練文例： ”電話番号は零三の三二二三一二三四です”

```
(define-pattern marker-roman-roman :j-g n  
(?x <roman-roman> ?y) (:head 3) (:x ローマ字)  
...)
```

格納場所： /p-data/marker-roman-x.lisp
訓練文例： ”はいジーエーケーユーイーエヌエムエーイーです”

17. カテゴリ **p**(合成用言のパタン)

パタン種類：通常 define-pattern
ヘッド：述語の項
変項制限：通常、何らかの制限をつける。

```
(define-pattern marker-cn-v-p :j-g p  
(?x <cn-v> ?y) (:head 1) (:x group-n)  
...)
```

格納場所： /p-data/marker-cn-x.lisp
訓練文例： ”この写真を焼き増ししてください”

18. カテゴリ **terminal**(数詞・接頭辞を伴う頻出表現／派生形のパタン)

*解析で terminal rule 用の形態素合成を行なうこと
パタン種類：通常 define-pattern
ヘッド：通常なし
変項制限：数量表現の場合は通常 (:x 数詞)

```
(define-pattern num-ko :j-g terminal  
(?x ”個”) (:x 数詞)  
...)
```

格納場所： /p-data/terminal-word.lisp
訓練文例： ”別にカートン四個があります”

```
(define-pattern tsuki-hi :j-g terminal  
(?x <tsuki-hi> ?y ”日”)  
...)
```

格納場所： /p-data/terminal-woed.lisp
訓練文例： ”九月十一日の便ではどうですか”

```
(define-pattern settou-o-terminal :j-g terminal  
(”お” ?x)  
...)
```

格納場所： /p-data/terminal-word.lisp
訓練文例： ”お電話番号が零三三二四五二二三三ですね”

```
(define-pattern cn-teki ;j-g terminal
(?x ”的”) (:x 普通名詞 サ変名詞 形容名詞)
...)
格納場所： /p-data/terminal-word.lisp
訓練文例： ”一般的に市内バスをよくつかいますね”
```

1.6 各種知識の統計データ

*平成 11 年 8 月 31 日現在

辞書

1. 日本語意味コード辞書 (jma-atr-sem-code.text)
 - 単語数 ... 20,944
 - バイト数 ... 787,602
2. 日独変換辞書 (japanese-to-german.lisp)
 - 単語数 ... 15,638
 - バイト数 ... 1,075,801
3. ドイツ語生成辞書 (tdmt_jg.*)
 - 単語数 ... 12,933
 - バイト数 ... 1,209,653

解析知識

1. replace-word-info.lisp
 - 単語数 ... 1,612
 - バイト数 ... 100,807
2. lexical-transformation
 - ルール数 ... 571
 - パタン数 ... 948
 - バイト数 ... 232,915

目的別内訳

項目	ルール数	パタン数
形態素の合成	376	537
形態素の分割	6	6
形態素の除去	7	25
品詞の変更	6	9
正規形の修正	77	170
表層形の修正	6	26
活用形の修正	1	1
誤りの修正	2	2
terminal rule	90	172
計	571	948

- ルール数 ... 217
- パタン数 ... 258
- バイト数 ... 40,254

3. total-transformation

- ルール数 ... 0
- パタン数 ... 0
- バイト数 ... 0

変換知識

- ルール数 ... 857
- バイト数 ... 834,513

1. パタンの種類別内訳

パタンの種類	個数
define-pattern	609
define-regular-pattern	248
計	857

2. カテゴリ別内訳

カテゴリ	個数
is	51
ss	44
se	14
sm	72
np	156
ap	37
sp	59
aa	5
pm	101
nm	15
s+m	2
s+n	58
n+n	68
dn	22
nd	51
n	10
p	1
terminal	91
計	857

2 解析変換知識作成手順

以下の章では、解析変換知識の作成手順を

- 意味距離計算を使わずに訓練する場合
- 意味距離計算を使って訓練する場合

のそれぞれの場合で例文を用いて解説する。

システムのデフォルトは、平成 11 年 3 月 31 日時点で、以下のような値になっている。

- 訓練の単位 ... 発声単位
- 意味距離計算 ... ON
- 新部分翻訳 ... ON
- 主語補完 ... ON

新部分翻訳が ON になっているため、翻訳訓練に際しては、空用例以外の exact match 用例（入力文と完全に一致する用例）をすべて追加し、望ましい翻訳結果が得られた時点で、

- 文分割(CONN) が起きていない場合は、距離がほぼ 0 になっていること
- 文分割 (CONN) が起きている場合は、分割位置の妥当性に問題がなく、距離がほぼ 5 の倍数分（文分割の数 x 5）離れていること

を訓練終了の条件として確認するものとする。

*アウトプットに現れるマーカ、CONN _ は構造を作るのに失敗した場合、訳出可能個所どうしを連結するマーカで、... は閾値以上の意味距離になった個所が落とされていることをあらわす。*詳細は付録 C を参照のこと。

作業にあたっては TDMT 翻訳知識作成ツールを利用するため、前もって、河井淳「TDMT 翻訳知識作成ツールの利用方法」を参照のこと。

2.1 意味距離計算を使わないで訓練する場合

2.1.1 解析変換知識作成手順

ここでは、意味距離計算を行わない場合の解析変換知識作成手順を説明する。作業は以下の流れで行い、既に訓練済みの知識を利用できる場合は、省略して次のステップに進む。

1. 前準備
↓
2. 翻訳実行
↓

3. 形態素情報の確認
↓
4. lexical-transformation の作成
↓
5. local-transformation の作成
↓
6. 日独変換辞書の作成
↓
7. 変換パタンの新規作成
↓
8. 目的言語パターンと用例の追加
↓
9. ドイツ語生成辞書の追加
↓
10. 訓練終了

2.1.2 例1 ”所要時間は六時間三十分になります”

1. 前準備

- (a) allegro common lisp の起動
- (b) システムの起動ならびにルールとタギングデータのロード。
setup-jg-tag.lispをロードすることにより、まとめて処理。
- (c) パッケージを翻訳モードに切替え
- (d) 距離計算を OFF に切替え。
- (e) TDMT 翻訳知識作成ツールの起動

2. 翻訳実行

TDMT ツールのメニューで翻訳を実行する。
翻訳知識が不足しているため、出力は得られない。

```
*** Input ***
  所要時間は六時間三十分になります
*** Output ***
  NIL
```

3. 形態素情報の確認

TDMT ツールのメニューで、Morph Analysis Only を選択すると以下のよう
にタギングデータが表示される。

sentence: 所要時間は六時間三十分になります

WORD	REG-EXP	POS	CONJ-FORM	SEM-CODES
所要	所要	普通名詞	---	174
時間	時間	普通名詞	---	828 152 150
は	は	係助詞	---	---
6	0 0	数詞	---	120
時間	時間	普通名詞	---	828 152 150
30	0 0	数詞	---	120
分	分	普通名詞	---	828 150 681 449 379 190 127 121
に	に	格助詞	---	---
なり	なる	本動詞	連用	283 273 260 250
ます	ます	助動詞	基本	926 62 994

次に morph analysis を選択して比較し、適用されている解析知識を確認する。

sentence: 所要時間は六時間三十分になります

WORD	REG-EXP	POS	CONJ-FORM	SEM-CODES
所要時間	所要時間	普通名詞	---	150
は	は	係助詞	---	---
6 時間	N 時間	普通名詞	---	828 153
30 分	N 分	普通名詞	---	828 153 152
になる	になる	助動詞	連用	---
ます	ます	助動詞	基本	---

ここでは、「所要時間」が既に合成され、「N時間」と「N分」も terminal rule 用に合成されていることがわかる。「六時間半」というような言い方もあるので、方針としては、さらに「N時間N分」を terminal rule 用に合成することにする。

4. lexical-transformationの作成

方針に従って、以下のルールを作成する。

```
(define-lexical-transformation num-jikan-num-fun :j-g 5
  (
    (((:pos . 数詞)) ((:word . "時間")) ((:pos . 数詞)) ((:word . "分")))
    =>
    (lex ((1 2 3 4) (:pos . 普通名詞) (:reg-exp . "N時間N分")
              (:compound 1 2 3 4)))
  )
)
```

作成したルールをコンパイルし、再度翻訳を実行して形態素情報を確認すると、以下のように表示される。

sentence: 所要時間は六時間三十分になります

WORD	REG-EXP	POS	CONJ-FORM	SEM-CODES
所要時間	所要時間	普通名詞	---	150
は	は	係助詞	---	---
6 時間 30 分	N 時間 N 分	普通名詞	---	828 153
になる	になる	助動詞	連用	---
ます	ます	助動詞	基本	---

5. local-transformationの作成

この文の場合は、係助詞と助動詞が内容語の境界を示すので、マーカの挿入は行なわない。

6. 日独変換辞書の作成

この文で対訳が必要な語は、「所要時間」と「6時間30分」。部分的に翻訳を行なうと、

```
*** Input ***
(所要時間)
*** Output ***
Laut Fahrplan. (0.0)

*** Input ***
(6時間30分)
*** Output ***
(0.0)
```

となり、「6時間30分」の対訳が不足していることがわかるが、この語は terminal rule 用に合成しているなので、変換パターンを作成することにする。

7. 変換パタンの新規作成

不足している「6時間30分」の変換パターンを以下のように作成する。

```
(define-pattern n-jikan-n-fun :j-g terminal
  (?x "時間" ?y "分")( :x 数詞)( :y 数詞)
=>
  (((!x "Stunde" !y "Minuten")(NP (NP 1 (NOMEN 2))(NP 3 (NOMEN 4))))
  (
    ("00 ") ("00 ")
  )
  )
  )
  )
```

再度部分的に翻訳すると、以下のように対訳が得られる。

```
*** Input ***
(6時間30分)
*** Output ***
Sechs Stunden dreißig Minuten. (0.0)
```

8. 目的言語パターンと用例の追加

ここで不足している用例を調べるために、部分的に翻訳を行なう。

```
*** Input ***
(6時間30分)(になる)
*** Output ***
NIL
```

(?x "になる")のパターンに用例が不足していることがわかるので、メニューの Search Rule Name でパターンを探して以下のように追加する。


```

(define-regular-pattern modal-ninaru-nm-reg-neu :j-g nm
  (?x "になる") (:head 1)
  =>
  (((("dauer" !x)(*VERB 1))
    (
      (("N分"))
      (("N時間N分")) ←用例を追加
    )
  )
  )
  )
  )
  )
  )

```

翻訳結果が以下のように得られる。

```

*** Input ***
(6時間30分) (になる)
*** Output ***
Es dauert sechs Stunden dreißig Minuten. (0.0)

```

同じように部分的に翻訳を行ない、(?x "ます")と(?x "は" ?y)にも用例を追加する。また、(?x "は" ?y)のパターンに用例を追加する際には、ローカル辞書の登録を同時に行なう。

```

(define-regular-pattern sen-masu-reg :j-g sm
  (?x "ます") (:head 1)
  =>
  :
  ((((!x))
    (
      (("N時間N分")) ←用例の追加
      (("N時" "になる"))
    )
  )
  )
  )
  )
  )

```

```

(define-pattern kakarijo-ha-np :j-g np
  (?x "は" ?y) (:head 3) (:x group-n 副詞)
  =>
  :
  ((((!x "dauer" !y)(VERB 2))
    (
      (("所要時間") ("N時間N分")) ←用例の追加
      (("所要時間") ("N分"))
    )
  )
  :
  (
    (x
      ((普通名詞 "所要時間")(SUB (PERSONALPRONOMEN "er/sie/es"))) ←ローカル辞書の追加
    )
  )
  )
  )
  )

```

すべての不足用例を補充すると、以下のように全文の翻訳結果が得られる。

```

*** Input ***
  所要時間は六時間三十分になります
*** Output ***
  Es dauert sechs Stunden dreißig Minuten. (0.0)

```

9. ドイツ語生成辞書の作成
ここでは生成辞書の追加は必要ないので、行なわない。

10. 訓練終了
TDMT ツールの画面では、上の Output を得るために、以下の変換パターンが適用されたことが表示される。

```

*** Distance calculation in Transfer ***
0.000000 :: (SEN-MASU-REG) SM :(?X ます) => (!X) .... ((N時間N分))
0.000000 :: (KAKARIJO-HA-NP) NP :(?X は ?Y)
              => (!X dauert !Y) .... ((所要時間) (N時間N分))
0.000000 :: (~LOCAL) : (普通名詞 所要時間)
              => ((SUB (PERSONALPRONOMEN er/sie/es)))
0.000000 :: (MODAL-NINARU-NM-REG) NM :(?X になる)
              => (dauern !X) .... ((N時間N分))
0.000000 :: (N-JIKAN-N-FUN) TERMINAL :(?X 時間 ?Y 分)
              => (!X stunde !Y minuten) .... ((0 0) (0 0))
0.000000 :: (~DIC) : (数詞 0 0) => ((KARDINALZAHL 6))
0.000000 :: (~DIC) : (数詞 0 0) => ((KARDINALZAHL 30))
TOTAL DISTANCE = 0.000000

```

2.1.3 例2 ”インド料理かタイ料理の店があれば”

1. 前準備
例1と同じ。
2. 翻訳実行
TDMT ツールのメニューで翻訳を実行する。翻訳知識が不足しているため、出力は得られない。

```

*** Input ***
  インド料理かタイ料理の店があれば
*** Output ***
  NIL

```

3. 形態素情報の確認
TDMT ツールのメニューで、Morph Analysis Only を選択すると以下のよう
にタギングデータが表示され、ここでは形態素情報の修正が行なわれてい
ないため、Morph Analysis でも同じ表示となる。

sentence: インド料理かタイ料理の店があれば

WORD	REG-EXP	POS	CONJ-FORM	PRON	SEM-CODES
インド	インド	普通名詞	---	---	709 719
料理	料理	サ変名詞	---	---	355 382 923
か	か	副助詞	---	---	---
タイ	タイ	普通名詞	---	---	709 719
料理	料理	サ変名詞	---	---	355 382 923
の	の	連体助詞	---	---	---
店	店	普通名詞	---	---	729
が	が	格助詞	---	---	---
あれ	ある	本動詞	ば	---	125 379
ば	ば	接続助詞	---	---	---

4. lexical-transformationの作成

形態素情報の修正は必要がないので、行なわない。

5. local-transformationの作成

ここでは、既に以下のように形態素の境界を示すマークが挿入されていることがわかる。

```

*** Process ***
>>> morphological analysis
(インド料理かタイ料理の店があれば)
>>> analysis
LOCAL-TRANSFORMATION ...
(インド<CN-SN>料理か<SUBP->タイ<CN-SN>料理の店があれば)
>>> transfer
(インド<CN-SN>料理か<SUBP->タイ<CN-SN>料理の店があれば)

```

マークに不足はないので、追加は行なわない。

6. 日独変換辞書の作成

この文で対訳が必要な語は、「インド」「料理」「タイ」「店」「ある」。
部分的に翻訳を行なうと、

*** Input *** (インド)	*** Input *** (店)
*** Output *** NIL (0.0)	*** Output *** Geschäft. (0.0)
*** Input *** (料理)	*** Input *** (あれ)
*** Output *** Speise. (0.0)	*** Output *** Es gibt. (0.0)
*** Input *** (タイ)	
*** Output *** Krawatte. (0.0)	

「インド」の対訳が不足していることがわかるので、日独変換辞書japanese-to-german.lispに以下のように追加する。

```
((普通名詞 "インド") (NP {OHNE} (NOMEN "Indien"))))
```

ここでの「タイ」は国の名前なので、「タイ料理」のところでローカル辞書を使って国名の「タイ」の形容詞 "thailändisch" の訳にに変更する。（*普通名詞「タイ」の生成辞書に形容詞の情報が付与されれば、ローカル辞書を使わずに生成マーカ A+ で処理することになる。）「インド料理」の "indisch" も同様。また、「料理」は料理法の "Küche" に、「店」は「料理の店」なので "Restaurant" の対訳に変更したい。これらも変換パターンにローカル辞書で追加することにする。

7. 変換パタンの新規作成

メニューの Search Rule Name でパターンを探して確認すると、(?x <cn-sn> ?y) (?x か <subp-> ?y) (?x の ?y) (?x が ?y) は、既に存在するが、(?x ば) のパターンが不足していることがわかる。したがって、類似パタンのカテゴリや下部構造の制限などを考慮しながら、以下のように新規パターンを作成する。

```
(define-pattern ending-ba :j-g se
  (?x "ば") (:head 1) (:x group-s)
  =>
  (((("wenn" !x)(SATZGEFUEGE (KOND-S (INTRO (SUBORD-KONJUNKTION 1)) 2)))
   (
    .(("ある "))
   )
  )
)
```

8. 目的言語パターンと用例の追加

2.1.1 と同じ要領で、部分的な翻訳を行ない、以下のように不足用例を追加する。

```
(define-pattern marker-cn-sn-n+n :j-g n+n
  (?x <cn-sn> ?y) (:head 3)
  =>
  :
  ((((!x !y)(NP 1 2)) ; 形容詞 + 名詞
   (
    ("日本") ("料理")
    ("インド") ("料理") ←用例の追加
    ("タイ") ("料理") ←用例の追加
   )
  )
  (
  (x
   ((普通名詞 "日本")(ADJEKTIV "japanisch"))
   ((普通名詞 "インド")(ADJEKTIV "indisch"))
   ((普通名詞 "タイ")(ADJEKTIV "thailändisch")) ←ローカル辞書の追加
  )
  (y
   ((サ変名詞 "料理")(NOMEN "Küche"))
  )
)
```

```

)
)
)

(define-pattern marker-ka/subp-n+n :j-g n+n
  (?x "か" <subp-> ?y) (:head 4)(:x group-n)
=>
  (((!x "oder" !y)(NP 1 (KOORD-KONJUNKTION 2) 3))
  (
    (("料理 ") ("料理 ")) ←用例の追加
    (("付 ") ("なし "))
    :
  )
)
)
)

(define-pattern rentaijo-no-n+n :j-g n+n
  (?x "の" ?y) (:head 3)
=>
  :
  (((!y "für" !x)(NP 1 (PP (PRAEPOSITION 2) OHNE 3 :case AKK)))
  (
    (("料理 ") ("店 "))
  )
  (
    (y
      ((普通名詞 "店 ") (NOMEN "Restaurant")) ←ローカル辞書の追加
    )
  )
)
)
)

(define-pattern kakujo-ga-np :j-g np
  (?x "が" ?y) (:head 3) (:x group-n)
=>
  :
  (((!x !y)(AKK-OBJ UNBESTIMMT 1))
  (
    (("専門店 ") ("ございます "))
    (("店 ") ("ございます "))
    (("店 ") ("ある ")) ←用例の追加
  )
  :
)
)
)
)

```

すべての不足用例を補充すると、以下のように全文の翻訳結果が得られる。

```

インド料理かタイ料理の店があれば
*** Output ***
Wenn es ein Restaurant für indische Küche oder Küche gibt. (0.0)

```

9. ドイツ語生成辞書の作成

"thailändisch"が生成されていないので、生成辞書に登録の有無を確かめ、未登録であれば以下の手順で追加登録する。

(a) 生成辞書登録のチェック

allegro package: transfer で生成のチェックをする。

(check '([part-of-speech] "[string]"))

単語だけのチェックであれば、(check "[string]")

例 1):

```
TRANSFER(30): (check "lang")
```

```
Generate: "lang"
```

```
Analysis: SATZ
```

```
SATZ (:SECOND)
```

```
<F-middle> (((VERBZUSATZ "lang") (ADVERB "lang" (:COMPARATION  
(POS KOM SUP))) (ADJEKTIV "lang" (:COMPARATION (POS KOM SUP) :ARTICLE  
(OHNE BESTIMMT UNBESTIMMT) :GENDER (NTR MAS FEM) :NUMBER (SG PL) :CASE  
(NOM GEN DAT AKK) :ATTRIBUTIVE-USED-P (NIL T PRAEDIKATIV-GEBRAUCHT)  
:CAP (NIL T))))))
```

```
"Lang."
```

← NIL と出たら生成辞書に未登録

例 2):

```
TRANSFER(51): (check '(ADVERB "lang"))
```

```
Generate: (ADVERB "lang")
```

```
Analysis: SATZ
```

```
SATZ (:SECOND)
```

```
<F-middle> (((ADVERB "lang" (:COMPARATION (POS KOM SUP))))))
```

```
"Lang."
```

*ここで与えられる品詞情報は、その時点で生成辞書に登録されている品詞名のみであり、生成の必要に応じ今後追加・変更の可能性がある。

(b) 当該単語の情報をデータベースから検索

ドイツ語のデータベース： ~paul/Work/LDC/DB/german/gendic*

(c) 生成辞書に追記

生成辞書の追加用ファイル：

~paul/Work/TDMT/g-generation/dic/tdmt_jg.add

ドイツ語のデータベースから検索した情報をコピーし、ロードする。

```
:  
("steig" "S.t.aI.g." "1" ((L-S "steig" 110407)) NIL NIL)  
("hain" "h.aI.n." "1" ((L-S "hain" 110207)) NIL NIL)  
("thaila~ndisch" "t.aI.-l.E.n.-d.I.S." "100" ((L-S "thaila~ndisch" 310001))  
NIL NIL) ←生成辞書の追加
```

翻訳結果は以下のようになる。

*** Input ***

インド料理かタイ料理の店があれば

*** Output ***

Wenn es ein Restaurant für indische Küche oder thailändische Küche gibt. (0.0)

10. 訓練終了

TDMT ツールの画面では、上の Output を得るために、以下の変換パターンが適用されたことが表示される。

```
*** Distance calculation in Transfer ***
0.000000 :: (ENDING-BA) SE :(?X ば) => (wenn !X) .... ((ある))
0.000000 :: (KAKUJO-GA-NP) NP :(?X が ?Y) => (!X !Y) .... ((店) (ある))
0.000000 :: (RENTAIJO-NO-N+N) N+N :(?X の ?Y)
=> (!Y für !X) .... ((料理) (店))
0.000000 :: (MARKER-KA/SUBP-N+N) N+N :(?X か <SUBP-> ?Y)
=> (!X oder !Y) .... ((料理) (料理))
0.000000 :: (MARKER-CN-SN-N+N) N+N :(?X <CN-SN> ?Y)
=> (!X !Y) .... ((インド) (料理))
0.000000 :: (~LOCAL) : (普通名詞 インド) => ((ADJEKTIV indisch))
0.000000 :: (~LOCAL) : (サ変名詞 料理) => ((NOMEN küche))
0.000000 :: (MARKER-CN-SN-N+N) N+N :(?X <CN-SN> ?Y)
=> (!X !Y) .... ((タイ) (料理))
0.000000 :: (~LOCAL) : (普通名詞 タイ) => ((ADJEKTIV thailändisch))
0.000000 :: (~LOCAL) : (サ変名詞 料理) => ((NOMEN küche))
0.000000 :: (~LOCAL) : (普通名詞 店) => ((NOMEN restaurant))
0.000000 :: (~LOCAL) : (本動詞 ある) => ((SUB (PERSONALPRONOMEN es))
(*AKK-OBJ (PERSONALPRONOMEN er/sie/es)) (VERB gibt))
TOTAL DISTANCE = 0.000000
```

2.2 意味距離計算を使って訓練する場合

2.2.1 解析変換知識作成手順

ここでは、意味距離計算を行なう場合の解析変換知識作成手順を説明する。作業は以下の流れで行い、それぞれのステップで確認事項に問題がある場合は、2.1 「意味距離計算を使わずに訓練する場合」の手順に従って対処する。

1. 前準備
↓
2. 翻訳実行
↓
3. 形態素情報の確認
↓
4. 構文解析の確認
↓
5. 翻訳結果の確認
↓
6. exact match 用例の追加
↓
7. 訓練終了

2.2.2 例3 ”わたくしは実は野球のことは本当によくわからないんですね”

1. 前準備

- (a) allegro common lisp の起動
- (b) システムの起動ならびにルールとタギングデータのロード。
setup-jg-tag.lisp をロードすることにより、まとめて処理できる。
- (c) パッケージを翻訳モードに切替え
- (d) TDMT 翻訳知識作成ツールの起動

2. 翻訳実行

TDMT ツールのメニューで翻訳を実行する。意味距離計算を ON にしているため、出力は得られるが、望ましい翻訳ではない。

*** Input ***

わたくしは実は野球のことは本当によくわからないんですね

*** Output ***

Eigentlich bin ich Baseball? _ er weiß die Sache wirklich nicht gut. (8.644449)

CONN マーカ "_"は、翻訳知識が不足している箇所を示す。

*詳細は付録Cを参照のこと。

3. 形態素情報の確認

TDMT ツールの Morph Analysis を選択する。

sentence: わたくしは実は野球のことは本当によくわからないんですね

WORD	REG-EXP	POS	CONJ-FORM	PRON	SEM-CODES
わたくし	私	代名詞	---	---	501
は	は	係助詞	---	---	---
実は	実は	接続詞	---	---	---
野球	野球	サ変名詞	---	---	898
の	の	連体助詞	---	---	---
こと	こと	形式名詞	---	---	811
は	は	係助詞	---	---	---
本当	本当	普通名詞	---	---	175
に	に	格助詞	---	---	---
よく	良い	形容詞	連用	---	172 173
わから	分かる	本動詞	ない	---	413
ない	ない	助動詞	基本	---	---
のですね	のですね	助動詞	---	---	---

ここでは、

- (a) タギングデータに誤りがないか
→あれば担当者に報告するが、すぐには修正できないのでパッチを当てる
- (b) 意味コードの登録に誤りや不足がないか
→あれば担当者に報告し、修正あるいは追加してもらう。
- (c) replace-word の登録に誤りや不足がないか
→あれば担当者に報告し、必要に応じて日本語を原言語とする解析変換知識作成者で協議する

- (d) lexical-transformation に誤りや不足がないか
→あれば該当知識の修正や追加を行なう

を確認する。この文については問題ない。

4. 構文解析の確認

原言語構文解析結果を確認し、不足している知識を検討する。

```
TOP [(?X ?Y) --- :CONN]
|--?X [(?X の) --- SE]
|  |--?X [(?X は ?Y) --- NP]
|  |  |--?X [(わたくし)]
|  |  |
|  |  |--?Y [(実は ?X) --- AP]
|  |  |  |--?X [(野球)]
|  |  |
|  |--?Y [(?X ですね) --- SM]
|  |  |--?X [(?X は ?Y) --- NP]
|  |  |  |--?X [(こと)]
|  |  |  |
|  |  |  |--?Y [(?X に ?Y) --- AP]
|  |  |  |  |--?X [(本当)]
|  |  |  |  |
|  |  |  |  |--?Y [(?X <ADJ/RENYOU-V> ?Y ない) --- AP]
|  |  |  |  |  |--?X [(よく)]
|  |  |  |  |  |
|  |  |  |  |  |--?Y [(わから)]
```

ここでは、まず連体助詞「の」が(?X の こと)のパタンを選べず、係り受けをつくることに失敗している。理由は、すでに記述されている用例とサ変名詞「野球」の意味距離が近くないため、ここに用例を追加する。

```
(define-pattern setsubi-no-koto-nd :j-g nd
 (?x "の" "こと") (:head 1)
 =>
 (((!x))
 (
 ("バスツアー ")
 ("インサドン ")
 ("市電 ")
 :
 ("野球 ") ←用例の追加
 )
 )
 )
```

```
TOP [(?X ?Y) --- :CONN]
|--?X [(?X は) --- SE]
|  |--?X [(わたくし)]
|  |
|  |--?Y [(実は ?X) --- IS]
|  |  |--?X [(?X ですね) --- SM]
|  |  |  |--?X [(?X は ?Y) --- NP]
|  |  |  |  |--?X [(?X の こと) --- ND]
|  |  |  |  |  |--?X [(野球)]
```

```

|
|--?Y [(?X に ?Y) --- AP]
|   |--?X [(本当)]
|   |
|   |--?Y [(?X <ADJ/RENYOU-V> ?Y ない) --- AP]
|       |--?X [(よく)]
|       |
|       |--?Y [(わから)]

```

ここでは、係助詞「は」において、(?x は ?y)のパタンを選べず、「わたくし」と下部構造の関係を作ること失敗していることがわかる。理由は、カテゴリ NP の (?x は ?y) が取り得ないカテゴリ IS が下部構造にあるからである。したがって、カテゴリ AP の (実は ?x) に用例を追加することにする。

```

(define-pattern conj-jitsuha-ap :j-g ap
  ("実は" ?x) (:head 2)
  =>
  (((("eigentlich" !x)(SATZ_verb-first (INTRO (ADJEKTIV 1)) 2))
   (
    ("日本人")
    ("分かる") ←用例の追加
   )
  )
)
)

```

再度翻訳すると、以下のように正しい構文解析が得られる。

```

TOP [(?X のですね) --- SM]
|--?X [(?X は ?Y) --- NP]
|   |--?X [(わたくし)]
|   |
|   |--?Y [(実は ?X) --- AP]
|       |--?X [(?X は ?Y) --- NP]
|           |--?X [(?X の こと) --- ND]
|               | |--?X [(野球)]
|               |
|               |--?Y [(?X に ?Y) --- AP]
|                   |--?X [(本当)]
|                   |
|                   |--?Y [(?X <ADJ/RENYOU-V> ?Y ない) --- AP]
|                       |--?X [(よく)]
|                       |
|                       |--?Y [(わから)]

```

5. 翻訳結果の確認

ここで翻訳結果は以下ようになる。

```

*** Input ***
わたくしは実は野球のことは本当によくわからないんですね
*** Output ***
Eigentlich weiß ich Baseball wirklich nicht gut. (2.6444445)

```

本動詞「分かる」は"verstehen" に対訳を変えたいので、(?X は ?Y)のパタンに用例とともにローカル辞書を記述する。

```

(define-pattern kakarijo-ha-np :j-g np
  (?x "は " ?y) (:head 3) (:x group-n 副詞)
  =>
  (((!y !x) (AKK-OBJ 2)) ;:extrawort definition
   (
    ("地図") ("用意する")
    ("銀閣寺") ("は入れる")
    ("野球") ("分かる") ←用例の追加
   )
   (
    (y
     ((本動詞 "分かる")(VERB "verstehen")) ←ローカル辞書の追加
    )
   )
  )
)

```

以下のような翻訳結果が得られるので、これを最終結果とする。

*** Input ***

わたくしは実は野球のことは本当によくわからないんですね

*** Output ***

Eigentlich verstehe ich Baseball wirklich nicht gut. (1.977778)

ただし、(?x "のですね")で適用された目的言語パターン(!x)は、この訓練文については問題ないが、自由入力文では疑問文の可能性もある。主語決定や疑問文か肯定文かの決定の精度を上げるためには、文脈情報や音声情報を利用する必要がある。

6. exact match 用例の追加

翻訳結果が決まったので、以下の適用パターン一覧を見て exact match 用例の不足を補う。

*** Distance calculation in Transfer ***

```

0.533333 :: (SEN-NODESUNE) SM :(?X のですね) => (!X) .... ((思う))
0.777778 :: (KAKARIJO-HA-NP) NP :(?X は ?Y)
=> (!X !Y) .... ((私) (見る))
0.000000 :: (~DIC) : (代名詞 私) => ((PERSONALPRONOMEN ich))
0.000000 :: (CONJ-JITSUHA-AP) AP : (実は ?X)
=> (eigentlich !X) .... ((分かる))
0.000000 :: (KAKARIJO-HA-NP) NP :(?X は ?Y)
=> (!Y !X) .... ((野球) (分かる))
0.000000 :: (SETSUBI-NO-KOTO-ND) ND :(?X の こと) => (!X) .... ((野球))
0.000000 :: (~DIC) : (サ変名詞 野球) => ((NP {OHNE} (NOMEN baseball)))
0.666667 :: (KAKUJO-NI-AP) AP :(?X に ?Y)
=> (!X !Y) .... ((本当) (申し訳ございません))
0.000000 :: (~DIC) : (普通名詞 本当) => ((ADJEKTIV wirklich))
0.000000 :: (MARKER-ADJRENYOU-V-HITEI-AP) AP :(?X <ADJ/RENYOU-V> ?Y ない)
=> (nicht !X !Y) .... ((良い) (分かる))
0.000000 :: (~DIC) : (形容詞 良い) => ((ADJEKTIV gut))
0.000000 :: (~LOCAL) : (本動詞 分かる) => ((VERB verstehen))
TOTAL DISTANCE = 1.977778

```

用例が不足しているのは、距離が"0.000000"でない箇所である。(?X のですね)、(?X は ?Y)と(?X に ?Y)のパターンに用例を追加する。

7. 訓練終了

念のため、意味距離計算を OFF にして、翻訳結果を出力する。結果が出なければ、どこかのパターンに exact match 用例が不足していることになるので、追加する。

2.3 解析変換知識作成時に使用するツール類

解析変換知識を作成するにあたって各種ツールまたは機能を使ってチェック等を効果的に行うこと目的とする。

[グローバル変数]

1. 距離計算

```
OFF : (setq transfer::*disable-sem-code* t)
ON  : (setq transfer::*disable-sem-code* nil)
```

2. 分割翻訳

```
OFF : (transfer::translate-method nil)
ON  : (transfer::translate-method :separate)
```

3. 発声単位

```
OFF : (load "/usr/local/TDMT/tdmt-multi-dev/build/text-definitions.lisp")
ON  : (load "/usr/local/TDMT/tdmt-multi-dev/build/text-definitions-uttr.lisp")
```

[ツール]

1. 全パターンの抽出

• find-patterns-with-lex

```
例 : (transfer::find-patterns-with-lex :all)
出力 : (KAKUJO-NI-AA) AA (?X "に" ?Y) head:(3)
```

• get-all-rules

```
例 : (load "/usr/local/TDMT/tdmt-multi-dev/misc/get-all-rules")
      (public::get-all-rules
        "/tmp/output"
        "/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/p-data/*.lisp")
→ 変項制限も出る
```

```
出力 : adv.lisp marker-conjadv-is (?x <conjadv-> ?y) is (:head 3)
      (:x 接続副詞)
      => ((((!x !y) (pp 1 2 :case dat)) ((("また") ("その時"))))
        ((x ((接続副詞 "また") (preposition "bis"))))
        (y ((副詞 "その時") (adverb "dann")))) ...
```

2. 任意の用例が使われているトレーニング文を探す

```
例 : (load "/usr/local/TDMT/tdmt-multi-dev/misc/search-sentence.lisp")
      (search-sentence
        'kakujo-ga-np
        '(("利用")("ある"))
        '(("JG-closed")("JG-training")("CstarII")("WEARABLE"))
      )
```

```
出力 : ("CstarII" "JJ" "1" "TRS80161") ::
      "はいそうでしたら九月十三日の浪速のれんを一等席九千四百五十円
      が二枚でよろしいですか" ... 0.0
```

3. コードまたは辞書のアップデート

格納場所："/usr/local/TDMT/tdmt-multi-dev/g-generation/code/utils.lisp"

- 変換知識
変換辞書 : (transdic)
変換コード : (transcode)
- 生成知識
生成辞書 : (ggdic) (gg) : (ggdic) + (ggcode)
生成コード : (ggcode)

4. 訓練文数のカウント

格納場所："/usr/local/TDMT/g-generation/code/utils.lisp"

例：(public::count-sen :menu '(("CstarII")))

5. 独自変換辞書のチェック

格納場所："/usr/local/TDMT/g-generation/code/utils.lisp"

例：(test-TransDic
:file "/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/dic/japanese-to-german.lisp"
:log :ERROR)
(test-TransDic
:file "/usr/local/TDMT/tdmt-multi-dev/transfer/j-g/dic/japanese-to-german.lisp"
:log :ERROR
:PartOfSp '(EIGENNAME EIGENNAME+))

-->"ERROR" を探す。

6. 形態素解析知識のスイッチ (タギングデータ vs. 形態素解析)

格納場所："/usr/local/TDMT/g-generation/code/utils.lisp"

タギングデータ利用 : (tag-init)
形態素解析利用 : (tag-exit)

7. 主語補完モードのスイッチ

格納場所："/usr/local/TDMT/g-generation/code/utils.lisp"

ON : (hokan-init)
(hokan-init :verbose NIL) *補完情報表示しない
OFF : (hokan-exit)

2.4 精度を確認するためのツール類

〔目的〕 新規の訓練を続けていく間に、訓練済みの言語表現に対して副作用を起こしてしまうことがあるため、200 文程度の訓練が終了した時点を目安に、定期的に各種のツールを用いて解析変換知識の精度を確認する作業を行なうのが望ましい。

[管理者] システム管理者

[格納場所]

```
/usr/local/TDMT/tdmt-multi-dev/transfer/code/compare-results.lisp  
/usr/local/TDMT/tdmt-multi-dev/misc/*.lisp
```

* misc に格納されているプログラムは実行前にロードすること。

[書式および例]

1. 翻訳結果の作成

素読み用ファイル、また差分確認用のマスタファイルおよび比較ファイルを作成するのに用いる。指定する対象テキストは、text-definitions.lisp 及び text-definitions-uttr.lisp の階層構造に従うものとする。

書式：

```
(make-result <翻訳結果出力ファイル> '((<対象テキスト>)))
```

例：

```
(make-result "~/RESULT/JG-closed.990831"  
             '("JG-closed"))  
:firstp nil)
```

2. 翻訳結果の差分の作成

1 で作成したマスタファイルと比較ファイルとの差分を作成するのに用いる。

書式：

```
(compare-results <マスタファイル> <比較ファイル> <差分ファイル>)
```

例：

```
(compare-results "~/RESULT/JG-closed.master"  
                "~/RESULT/JG-closed.990831"  
                "~/compare/JG-closed.comp-990831")
```

3. 重複用例の確認

同一原言語ボタン内で、重複して出現する用例を出力するのに用いる。

書式：

```
(find-same-examples <p-data の場所> <出力ファイル>)
```

あるいは、

```
(check-example-duplicates <p-data の場所> <出力ファイル>)
```

例：

```
(find-same-examples  
  "~/TDMT/transfer/j-g/p-data/*.lisp"  
  "~/test/same-examples")
```

あるいは、

```
(check-example-duplicates
  "~/TDMT/transfer/j-g/p-data/*.lisp"
  "~/test/same-examples")
```

4. 過剰訓練の確認

訓練文に使用していない、過剰な解析知識、および変換知識の原言語パターン・目的言語パターン・用例を出力するのに用いる。指定する対象テキストは、text-definitions.lisp 及び text-definitions-uttr.lisp の階層構造に従うものとする。

書式：

```
(count-unused-patterns '((<<対照テキスト>>)) <出力ファイル>)
```

例：

```
(count-unused-patterns '("JG-closed"))
  "~/test/closed-unused-patterns")
```

99年8月同ツール実行後、消去せずに残した知識は以下の通り。(データに“but unused ...”を付与)

- keyboard 用パターン、用例。
 - 例： KIHON-HOUNO-S+N: (?X <> 方 の ?Y)
 - CONJ-DESHITARA-NP: (?X "でし" "たら" ?Y)...(("温泉") ("入れる"))
- 一般的な terminal word。
 - 例： NUM-CHOUME: (?X 丁目)
 - N-BURROKU: (?X ブロック)
 - NUM-DAI: (?X 台)
- a-data での加工をしない場合を見込んだもの。
 - 例： ADVERB-AP: (?X <ADV-> ?Y)...(("どうか") ("する")) ← 合成
 - MARKER-AUXV-: (?X <AUXV-> ?Y)...(("ちょうど") ("ご覧する"))
 - "ですね" deletion
- 2ヘッド用例に対応するためのもの。
 - 例： JUNTAIP-NO-S+N: (?X "の")...((" " " ")) ← ("行く" "れる")
- utterance changed による係り受け変更で unused。用例が少ないので残したもの。
 - 例： BEGINNING-JA: ("じゃ" ?X) ...(("ツォー駅"))
- パターン変更により unused。一般的な使用の見込まれるもの。
 - 例： FUKUJO-DEMO-NP: (?X "でも" <SUBP-> ?Y) ...(("海外") ("有名"))
 - KAKARIJO-HA-NP: (?X "は" ?Y) ...(("これ") ("ホテル"))
- 第一候補以外で使われるもの。
 - 例： KAKARIJO-HA-NP: (?X "は" ?Y) ...(("こちら") ("食事する"))
- 名前。
 - 例： SETSUBI-SAMA: (?X "様") ...(("herr-frau" !X)((" ")))
 - ("herr" !X)(("元")) ("寿幸")

5. 辞書ツール

すべての形態素に対する意味距離辞書及び変換辞書のエントリー不足を抽出するツール。仕様については「<http://www.itl.atr.co.jp/syamada/link-files/design.txt>」を参照のこと。

格納場所："/home/as44/doi/work/NEW-DIC-TOOL/code/dic-tool"

書式（日独の場合）：

```
(in-package :DIC-TOOL)
(make-all-set :JG :corpus :REMAKE)
(compare :JG)
(make-template :JG :trans-hint *JE-TRANSFER-DIC-FILE*)
```

3 懸案事項

トレーニングに関する現在（99年8月）の懸案事項は以下の通り。

《解析変換知識関連》

[文意をくんだ前置詞の選択]

例1："もう一つですねたぶん京都の有名なお見せたいところでしょう"

Noch eins. Ich zeige es wahrscheinlich von einem bekannten Ort in Kyoto.

(KAKUJO-DE-REG) NP : (?X で ?Y) => (!Y von !X) ((ところ) (見せる))

例2："これは立派な壺ですね日本に持って帰ったらみんな感心しますよいろいろありがとうございました"

Das ist ein prächtiger Topf. Jeder bewundert ihn, wenn man ihn nach Japan mitnimmt. Vielen Dank für alles.

(MODAL-TEKAERU-REG) PM : (?X てかえる) => (man !X) ((持つ))
(KAKUJO-NI-NP) NP : (?X に ?Y) => (!Y nach !X) ((日本) (持つ))

どちらも場所をあらわす "in" ではなく、「～の中から」「～へ」の文意をくんで "von" "nach" を使う。

例3："ということは日本と韓国は時差がないんですね"

0.000000 :: (HEIJO-TO-HA) NP : (?X と ?Y は ?Z)
=> (!Z zwischen !X und !Y) ... ((日本) (韓国) (無い))

「と」が並列を意味する場合との区別ができない。
「日本と韓国は8月に祭日がないのですね。」

[不自然な構造]

例："せっかく本場にはオペラを見に行きますんで多少お金は高くてもできるだけよい席でオペラ見たいなというふうに考えてるんですけども"


```

|--?X [(?X ます) --- SM]
|   |--?X [(?X <ADV-> ?Y) --- AP]
|       |--?X [(せっかく)]
|           |
|           |--?Y [(?X に ?Y) --- SP]
|               |--?X [(?X を ?Y) --- NP]
|                   |--?X [(?X に ?Y) --- N+N]
|                       |--?X [(本場)]
|                           |
|                           |--?Y [(オペラ)]
|                               |
|                               |--?Y [(見)]
|                                   |
|                                   |--?Y [(行き)]

```

本来なら (本場) に (行く) np を構造としたいところだが、「本場のオペラ」の訳出を得るために (?X に ?Y)n+n とする。

[無理のあるローカル]

例： "宮廷料理ちょっとわからないんですどなんが出るんでしょう"
 Ich kenne keine königliche Küche. Was wird serviert?

(LOCAL) : (本動詞 出る) => ((VERB servieren VOICE PASSIV))

「料理」と「出る」の直接係受けがないため、servieren のローカル挿入はやや強引。

[文中の "ですね" の消去]

例 1 : "その時期ですね当南座では昼の部浪速のれん夜の部八月納涼歌舞伎っていうのを上演しておりますが"

Wir führen ' Naniwa Noren ' in der Matinee und ' Sommer-Kabuki im August ' in der Abendaufführung zu dieser Zeit in unsrem Minamiza Theater auf.

例 2 : "分かりましたちょうどですね今この時期でしたら京都で祇園祭というのがありますので

その写真を今お出ししますのでご覧ください"
 Okay. Das Gion Fest findet gerade zu dieser Zeit in Kyoto statt.
 Ich bringe gleich das Foto. Bitte sehen Sie es.

「その時期」も「ちょうど」も次の文の動詞を修飾する副詞句として扱いたいので、a-data において形態素の削除を行っているが、「～ですね」を削除せず独立した文を作りたい場合不都合となる。

[xyz パタンのトレーニング不足]

例 : "はい九月十三日から十五日までフランクフルトに滞在しますそれで九月十三日か十四日どちらかで見たいんですが"

```

*** Input ***
十月三日か四日どちらかで見たいんですが
*** Output ***
Ich möchte ihn gerne entweder am dritten Oktober oder
dem vierten sehen. (5.0e-6)

(define-pattern marker-ka-de-np :j-g np ;;Cstar
  (?x <cn-pron> ?y "か" "で" ?z) (:head 6) (:x group-n)
=>
  (((!y "an" !x !z)(AP 1 (TIME (PRAEPOSITION 2) 3 :case DAT)))
   (
    (("N日") ("どちら") ("見る"))
    ;;(("N日") ("どちら") (""))
   )
   (
    (y
     ((代名詞 "どちら")(ADVERB "entweder"))
     )
     ;; N日かN日どちらか だと、"an" が規定できない
   )
  )
)

```

全体的に xyz パタンの用例が不足しているため、かなり近いテキストにおいても未訓練であれば、別の構造を選んでしまうことが多い。(→ デフォルトとして空用例を準備する?)

```

*** Input ***
十月三日か四日どちらかを取りたいのです
*** Output ***
Wo möchte ich gerne der dritte Oktober oder vierter nehmen? (0.6111295)
TOP [(?X のです) --- SM]
|--?X [(?X で ?Y) --- NP]
  |--?X [(?X <CN-PRON> ?Y) --- N+N]
  |   |--?X [(?X か <SUBP-> ?Y) --- N+N]
  |   |   |--?X [(10月3日) --- compound]
  |   |   |
  |   |   |--?Y [(4日) --- compound]
  |   |   |
  |   |--?Y [(?X か) --- ND]
  |   |   |--?X [(どちら)]
  |   |
  |--?Y [(?X たい) --- PM]
  |   |--?X [(取り)]

```

[時制の違い]

例1: "どうされましたか"
Was haben Sie gemacht?

例2: "今日は特別無料なんですよかったですね"
Es ist heute ausnahmsweise kostenlos __ ... __ es ist gut.

本来なら、訳出としては過去時制を出したくないが、:(?X た)での過去時制指定は避けられない。

[Ja/Nein の区別]

例: "はい持っていません"

Ja. Ich habe es nicht dabei.

本来ここでは「はい」を "nein" としたい。

[ローカル辞書]

1. 複数のローカル辞書のエントリーがあった場合

例: "もしもしわたくし鈴木和子と申しますちょっとお尋ねしたいんですがわたくし
今度ドイツのミュンヘンに観光旅行に行くんですがよく知りませんので
名所などを紹介していただけたらと思ってお電話いたしました"

(ミュンヘン)に(行く)、(観光旅行)に(行く)の二つの「行く」
に対するローカルがある場合、上部の(ミュンヘン)に(行く)が優先
される。→ ここでは「ミュンヘンに行く」で "gehen" を "fahren" に変
えるローカルを廃止し、「観光旅行に行く」で挿入されるローカル
"machen" が生きるようにする。

2. ローカル辞書の有無によるパタンの区別

ローカルの有無によってターゲットパターンを区別する基本方針があるが、
オープン文において、選ばれたターゲットパターン内に記述されたローカ
ル辞書がたまたまあたって、正解となることもあるし、またその逆もあ
りうるため、同一ターゲットパタンのマージあるいは区別には注意が必
要である。

[単数/複数]

例: "京都の金閣寺銀閣寺奈良の東大寺それに東京周辺では富士山と歌舞伎公演の写真です"
Das Foto des Kinkakuji Tempels und des Ginkakuji Tempels in Kyoto, des
Todaiji Tempels in Nara und des Fujisan und des Kabuki-Dramas in der
Umgebung von Tokyo.

修飾句から判断して、本来なら「写真」を複数で扱いたい。

[補完モジュール]

主語補完のモジュールがアップデートされた場合、p-data に記述された主語
が相応かどうかをチェックする必要がある。優先度の切り替え(補完主語>
デフォルト主語) → 翻訳結果の比較 → p-data 調節

上記の調節を行なうための、(補完主語>デフォルト主語)を実行するツ
ールが必要。

[話者情報] J E、E J に導入されている機能に音声認識情報があるが、J G に
は現在導入されていない。これは音声認識装置と接続している場合に、発話
者の役割情報、発話者の性別情報をルールで参照し、訳し分けなどに利用す
るものであるが、ドイツ語の名詞の性との調整が必要となる。

例: "Herr"/"Frau", "Japaner"/"Japanerin", "Beamter"/"Beamtin"

《生成関連》

[CONCAT]

訓練済の名詞句についてのみ CONCAT をみとめる。連結部分の "s" の有無の区別。

[構造修正]

NOM-OBJ とマークされていない名詞節の組み合わせ。他動詞の四格目的語挿入と、前置詞の組み合わせ。節の境界の判断。並列文と従属文。

- 文タイプの選択

例：(FORCE-S)"動かないでそのままですべてください"

Bitte bewegen Sie sich nicht. Und bleiben Sie dabei. (0.0)

*** Generation ***

```
((IMP-S-I (FIX-INTRO "bitte") (*SUB (PERSONALPRONOMEN "sie"
:PERSON 2A)) (|{<AGEN F>}| (VP (REFLEXIVPRONOMEN "sich"
:CASE AKK) (VERB "bewegen")))) {NOT} (FORCE-S IMP-S-I
(INTRO "und") (|{<AGEN Y>}| (VERB "dabeibleiben")))))
```

=>

Bitte bewegen Sie sich nicht. Und bleiben Sie dabei.

「～してください」を「動かない」「そのままですべて」の両方にかけるために生成マーカ

(FORCE-S IMP-S-I ...) を (?X で ?Y) のパターンで付与しているが、ここでは命令の情報がないため、不都合が生じる可能性がある。（"私は動かないでそのままですべてしました"）

- ふさわしい主語の選択

[先行詞選択誤り]

例：はい祇園祭は京都三大祭の一つで最も人気のあるお祭です

Das Gion Fest ist eines der drei größten Feste von Kyoto und sie sind die beliebtesten Feste.

本来なら das Gion Fest を先行詞とし、es ist die beliebteste... を導きたい。

[翻訳の精度を高める機能]

- 照応解析…照応解析による、ローカル辞書等でのよりふさわしい対訳選択。
- フォーカス…強調される文要素の認識。

《Cstar のデモ用特別ルール》

〔候補数〕

```
("DemoMay99_3" "TOS50002")
"はい後程何かご確認したいことなどございましたらシースタートラベルの予約係松
田の方までご連絡いただければ対応させていただきます"
```

通常、副助詞＋普通名詞で挿入されるマークは汎用性を高めるために後続の品詞を問わない <subp-> にしているが、候補数を絞る目的で、特殊なマークを挿入した。

```
(define-local-transformation subp-patch :j-g 2 ;;demoMay99_2
(
  (((:pos . 副助詞)) ((:pos . 本動詞)(:reg-exp . "確認する")))
  =>
  (1 <subp-patch> 2)
)
)

(define-pattern marker-ka/subp-patch-n+n :j-g n+n
(?x "か" <subp-patch> ?y) (:head 4)(:x group-n);(:y group-n 副詞)
=>
(((!x !y)(NP 1 2))
(
  (("何") ("こと")) ;;demoMAY99
)
(
  (x
    ((普通名詞 "何")(ADVERB "irgend"))
  )
  (y
    ((形式名詞 "こと")(PARTIKEL "etwas"))
  )))
))
```

〔性別〕

```
("DemoMay99_3" "TOS50002")
"はいホワン様 "
"はいバク様 "
```

いずれも性別が判断できないが、デモ用に男性の敬称を付与した。

```
((("Herr" !x)(NP HERR OHNE (NOMEN 1) 2))
(
  (("ホワン")) ;;patch demoMay_3
  :
))
```

References

- [1] 河井淳：TDMT 翻訳知識作成ツールの利用方法 TR-IT-0088(1994)
- [2] 言語データベースの概要 (1999、TR:1995)
- [3] 日本語意味コード辞書の作成の手引き (1998)
- [4] TDMT 用日本語形態素仕様明細 (1999)
- [5] replace-word の基本原則 (1999)
- [6] 山本：主語補完の概要 TR-IT-???? (1999) (to appear)
- [7] 脇田 意味的類似性を用いた音声認識正解部分の特定法 自然言語処理 V.5-4, p.111-125 (1998)
- [8] Paul：TDMT's German Generation Module TR-IT-???? (1999) (to appear)

A 生成モジュール

Table 1: German Part-of-Speeches

category	part-of-speech (品詞)	description
verb	VERB HILFSVERB MODALVERB VERBZUSATZ	verb (本動詞) auxiliary verb (助動詞) modal verb (話法の助動詞) verb prefix (動詞の前綴)
noun	NOMEN INITIAL BUCHSTABE UNIT	common noun (普通名詞) initial (かしら文字) letter (文字) unit (単位)
adjective	ADJEKTIV KARDINALZAHL ORDINALZAHL	adjective (形容詞) cardinal number (基数) ordinal number (序数)
pronoun	POSSESSIVPRONOMEN PERSONALPRONOMEN REFLEXIVPRONOMEN DETERMINATIV DETERMINATIV-INDEF RELATIVPRONOMEN INTERROGATIVPRONOMEN	possessive pronoun (所有代名詞) personal pronoun (人称代名詞) reflexive pronoun (再帰代名詞) determiner (<指示>代名詞) indefinite determiner (不定代名詞) relative pronoun (関係代名詞) interrogative pronoun (疑問代名詞)
adverb	ADVERB FRAGEADVERB	adverb (副詞) interrogative adverb (疑問副詞)
particle	PARTIKEL	particle (不変化詞)
preposition	PRAEPOSITION	preposition (前置詞)
conjunction	KOORD-KONJUNKTION SUBORD-KONJUNKTION	coord. conjunction (並列接続詞) subord. conjunction (従属接続詞)
others	INTERPUNKTION STOP-WORD	punctuation (記号) unknown word (未知語)

*パウル研究員の資料に拠るが、文法用語等については日本で通りのよいと思われる名称に変更してあるものがある。また単語に付与される品詞は、生成の都合上、実際の品詞とは違うことがある。ex."jeder" は不定代名詞だが、"DETERMINATIV-INDEF" ではなく"DETERMINATIV" とされている。

Table 2: German Part-of-Sentences

usage	part-of-sentence	description
grammatical	SUB	subject (force nominative case)
	NOM-OBJ	nominative object (force nominative case)
	GEN-OBJ	genitive object (force genitive case)
	DAT-OBJ	dative object (force dative case)
	AKK-OBJ	accusative object (force accusative case)

grammatical	VP	verb phrase
	NP	noun phrase
	PP	prepositional phrase (force successor case of preposition)
	AP	adverbial phrase
	WH	interrogative phrase
	PLACE	locative expression
	TIME	temporal expression
derivative	ORDINAL	transform enclosed cardinal number into an ordinal one
	ORDINAL+	extract/transform enclosed cardinal number to an ordinal one
	KARDINAL+	extract enclosed cardinal number
	N+	nominalization of verbs/adjectives
	V+	verbalization of nouns/adjectives
	A+/A+pprf/A+pres	adjectivalization of nouns/verbs (pprf/pres participle)
	FIX-EXP	fixed expression, no analysis on word-level
	MAL	adverb derivation from cardinal with stem form "...mal"
MALIG	adjective derivation from cardinal with stem form "...malig"	
functional	SUPERLATIV	superlativ form ("am X-ten" vs. "X-te <i>Nomen</i> ")
	CONCAT	concatenae nouns to a compound noun using boundary "." ("Hotel" (<i>hotel</i>), "Zimmer" (<i>room</i>) ⇒ "Hotel.zimmer" (<i>hotel room</i>))
	DIGIT	defining enclosed cardinal number as a chain of single numbers
	AM	transform time expression into "a.m.", i.e. add daytime affix
	PM	transform time expression into "p.m.", i.e. add daytime affix
	YEAR	transform a number ($1100 \leq n \leq 2000$) into its year-expression ("1997" = "19" * "100" + "97" ⇒ "19"."Hundert"."97")
	ETAGE	transform Japanese floor expression to the German one
	FLIGHT	specification for a noun phrase concerning flight information ("Flug X" vs. "X Flüge")
	FIX-CAP	capitalized fixed expression, no analysis on word-level
	FIX-UP	upper-cased fixed expression, no analysis on word-level
	NP-END	additional phrase after noun without any article insertion
	DUMMY	suppress output of word (due to redundance in compound NP)
inflectional	ATTR	specifying inflection attributes for enclosed parts
	EIGENNAME	proper noun phrase; unknown words are treated as FIX-CAP
	EIGENNAME+	proper noun phrase without any article insertion
topological	INTRO	leading expression, used for conjunction, interjection, etc.
	FIX-INTRO	leading expression, no analysis on word-level
	END	ending expression; used for heavy NP-shift, subord.sentences
	FIX-END	ending expression, no analysis on word-level
	AP+	adverbial phrase with special positioning
	VERB-ADD	verb phrase addition (equivalent to non-verbal parts of VP)

Table 3: Generation Markers

category	generation marker	description
negation	{NOT}	sentence negation
sentence type	{COORD}	sentence coordination
article	{OHNE} {BESTIMMT} {UNBESTIMMT}	no insertion of an article insert definite article insert indefinite article
pronoun	{POSS} {MEIN} / {IHR}	possessive pronoun (subject agrrement) 1.person / address form
address form	{HERR} {FRAU} {VORNAME}	male addressee female addressee first name addressee
subject (automatically inserted by HOKAN module)	{<AGEN S>} {<AGEN K>} {<AGEN H>} {<AGEN Y>} {<AGEN G>} {<AGEN X>} {<AGEN F>}	1.person, singular ("ich") 1.person, plural ("wir") 2.person, address form / singular ("Sie"/"du") 2.person, address form / plural ("Sie"/"ihr") 3.person, general person ("man") 3.person, singular ("er/sie/es") 3.person, singular, neutral ("es")

Table 4: German Sentence Types

main clauses					
type	finite verb	tense	mood	person	end
SATZREIHE	—	coordination of main clauses			
SATZGEFUEGE	—	coordination of at least one subordinated clause			
SATZ	:second	—	INDIKATIV KONJUNKTIV	—	"."
SATZ_verb-first	:first	statement preceded by subordinated sentence			"."
FRAGE	—	WH part-of-sentence is included → yes (WH-Q), no (YN-Q)			
WH-Q	:second	—	INDIKATIV KONJUNKTIV	—	"?"
YN-Q	: first	—	INDIKATIV KONJUNKTIV	—	"?"
FRAGE_verb-first	:first	question preceded by subordinated sentence			"?"
IMP-S-I	:first	PRAESENS	IMPERATIV	2 2a	"!"
IMP-S-II	:first	PRAESENS	IMPERATIV	2 2a	"!"
IMP-S_verb-first	:first	imperative preceded by subordinated sentence			"!"
DES-S-I	:second	—	KONJUNKTIV	3	"."
DES-S-II	:first	—	KONJUNKTIV	3	"!"

subordinated clauses (finite verb = :final)						
type	intro	end	type	intro	end	tense
REL-S	der/die/das, welcher/e/es	,	INH-S-ANF	-	,	INFINITIV+ZU
KAU-S	da, weil, weshalb, zumal		INH-S-ZU	-		
KONS-S	dass, so dass, um zu, als daß		INH-S-DASS	daß		
KONZ-S	obgleich, obwohl, wenn auch		INH-S-ALS	als		
TEM-S	sobald, nachdem, wenn, bevor		INH-S-WENN	wenn		
KOND-S	wenn, falls, sofern		INH-S-ALSOB	als ob		
MOD-S	indem, dardurch daß, so daß		INH-S-OB	ob		
FIN-S	damit, daß, auf daß, um zu		INH-S-WIE	wie		
AUS-P	wobei, was, nur daß, insofern		INH-S-WH	WH phrase		

Table 5: Inflection Attributes

category	attribute values	description
person (人称)	1 2 2a 3	first person (一人称) second person (二人称) form of address (二人称敬称) third person (三人称)
case (格)	NOM GEN DAT AKK	nominative case (一格) genitive case (二格) dative case (三格) accusative case (四格)
number (数)	SG PL	singular (单数) plural (複数)
gender (性别)	MAS FEM NTR	masculine (男性) feminine (女性) neuter (中性)
comparison (階級)	POS KOM SUP	positive (原級) comparative (比較級) superlative (最高級)
tense (時相)	PRAESENS IMPERFEKT PERFEKT PLUSQUAMPERFEKT FUTUR-1 FUTUR-2	present (現在) imperfect (過去) perfect (現在完了) past perfect (過去完了) future 1 (未来) future 2 (未来完了)
	INFINITV INFINITIV+ZU PARTIZIP-PRAESENS PARTIZIP-PERFEKT KONJUNKTIV-1 KONJUNKTIV-2	infinitive (不定詞) infinitive (不定詞)+”zu” present participle (現在分詞) perfect participle (過去分詞) subjunctive (接統法-1) subjunctive (接統法-2)
mood (ムード)	INDIKATIV KONJUNKTIV IMPERATIV	indicative (直說法) subjunctive (接統法) imperative (命令法)
voice (態)	AKTIV PASSIV ZS-PASSIV	active (能動態) progressive passive (受動態) adjectival passive (狀態受動)

Table 6: Part-of-Speech Inflection

part-of-speech	attribute key	attribute values
VERB HILFSVERB MODALVERB	:TENSE : :NUMBER :PERSON :MOOD :VOICE	PRAESENS IMPERFEKT PERFEKT PLUSQUAMPERFEKT FUTUR-1 FUTUR-2 KONJUNKTIV-1 KONJUNKTIV-2 INFINITIV INFINITIV+ZU PARTIZIP-PRAESENS PARTIZIP-PERFEKT SG PL 1 2 2a 3 INDIKATIV KONJUNKTIV IMPERATIV AKTIV PASSIV ZS-PASSIV
NOMEN	:GENDER :NUMBER :CASE	MAS FEM NTR SG PL NOM GEN DAT AKK
ADJEKTIV	:COMPARATION :ARTICLE :GENDER :NUMBER :CASE :ATTRIBUTIVE-USED-P	POS KOM SUP OHNE BESTIMMT UNBESTIMMT MAS FEM NTR SG PL NOM GEN DAT AKK NIL T
POSSESSIV- PRONOMEN	:TYPE :GENDER :NUMBER :CASE	ARTIKELWORT SUBSTANTIVWORT MAS FEM NTR SG PL NOM GEN DAT AKK
PERSONAL- PRONOMEN	:PERSON :NUMBER :CASE :GENDER	1 2 2a 3 SG PL NOM GEN DAT AKK MAS FEM NTR
REFLEXIV- PRONOMEN	:PERSON :NUMBER :CASE	1 2 2a 3 SG PL DAT AKK
ARTIKEL, RELATIV- PRONOMEN	:GENDER :NUMBER :CASE :DET	MAS FEM NTR SG PL NOM GEN DAT AKK DET T
DETERMINATIV, DETERMINATIV- INDEF	:GENDER :NUMBER :CASE	MAS FEM NTR SG PL NOM GEN DAT AKK
ORDINALZAHL	:ARTICLE :GENDER :NUMBER :CASE :ATTRIBUTIVE-USED-P	OHNE BESTIMMT UNBESTIMMT MAS FEM NTR SG PL NOM GEN DAT AKK NIL T
no inflection		ADVERB, BUCHSTABE, FRAGEADVERB, INTERROGATIV- PRONOMEN, INITIAL, KARDINALZAHL, KOORD-KONJUNKTION, PRAEPOSITION, PARTIKEL, SUBORD-KONJUNKTION, UNIT VERBZUSATZ, STOP-WORD

B 生成マーカ使用例

*** Part-of-Sentence (文成分)

《grammatical usage》

【SUB】 主語

例: "荷物はどこでチェックインすればいいですか"
"Wo soll ich das Gepäck einchecken lassen?"

```
(define-pattern sen-s-desuka-sm :j-g sm
  (?x "です" "か") (:head 1)(:x group-s)
=>
  (((("ich" !x) (FRAGE (SUB (PERSONALPRONOMEN 1)) 2))
    (
      ( ("チェックインする" "ばよい")
        )
      )
    )
  )
)
```

【NOM-OBJ】 同格の主語

例: "子供さんは男の子ですか"
"Ist Ihr Kind ein Junge?"

```
(define-pattern kakarijo-ha-np :j-g np
  (?x "は" ?y) (:head 3)
=>
  ((((!x "sein" !y)(SUB 1)(*HILFSVERB 2)(NOM-OBJ 3))
    (
      ( ("子供") ("男の子")
        )
      )
    )
  )
)
```

【GEN-OBJ】 2格目的語

例: "er bedarf meiner Hilfe"
"sie erinnert sich ihrer Studienzeit"

★現在(99/8)使用例なし。

【DAT-OBJ】 3格目的語

例: "そのバスのドライバーに契約書を見せて下さい"
"Bitte zeigen Sie dem Busfahrer den Vertrag."

```
(define-pattern kakujo-ni-np :j-g np
  (?x "に" ?y) (:head 3)
=>
  ((((!x !y)(DAT-OBJ 1))
    (
      ( ("ドライバー") ("見せる")
        )
      )
    )
  )
)
```

【AKK-OBJ】 4 格目的語

例: "今晚のホテルを予約したいのですが"
"Wir möchten gerne ein Hotel für heute abend reservieren."

```
(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y !x)(AKK-OBJ 2))
   (
    (("ホテル") ("予約する"))
   )
  )
)
```

【VP】 動詞句

例: "めまいがします"
"Ich bin schwindelig."
変換辞書: ((本動詞 "めまいがする") (VP (ADJEKTIV "schwindelig")(HILFSVERB "sein")))

【NP】 名詞句

例: "日本語のメニューはありますか"
"Gibt es eine japanische Speisekarte?"

```
(define-pattern rentaijo-no-n+n :j-g n+n
  (?x "の" ?y) (:head 3)
  =>
  (((!x !y)(NP 1 2))
   (
    (("日本語")("メニュー"))
   )
  )
)
```

【PP】 前置詞句

例: "あれガイドブックで見た料金と違うようですが"
"Bitte? Es scheint, die Gebühr, die ich im Reiseführer gesehen habe, ist anders."

```
(define-regular-pattern kakujo-de-reg :j-g np
  (?x "で" ?y) (:head 3) (:x group-n)
  =>
  (((!y "in" !x)(PP (PRAEPOSITION 2) 3 :case DAT))
   (
    (("ガイドブック") ("見る"))
   )
  )
)
```

* 格が限定できないパターンでは :case (DAT AKK) と記す。

```
(define-regular-pattern mae-reg :j-g nd
  (?x <> "前") (:head 3)
  =>
  (((!y "vor" !x)(PP (PRAEPOSITION 1) 2 :case (DAT AKK)))
   (
    (("ターミナル")
     ("烏丸口")
     ("奈良駅"))
   )
  )
)
```

【AP】 副詞句

例： "平安神宮ですとすぐ近くですが"
"Der Heianjingu Schrein ist ganz nahe."

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 ap)
=>
  (((!x !y)(AP 1 2))
   (
    ("すぐ") ("近く")
   )
  )
)
```

【WH】 疑問詞を含む節

例： "今出川停留所にはいくつ目にとまりますか"
"An der wievielten hält er an der Haltestelle Imadegawa?"

変換辞書: ((代名詞 "いくつ目") (WH (DETERMINATIV "der/die/das")
(ADJEKTIV "wievielte" :ARTICLE BESTIMMT
:ATTRIBUTIVE-USED-P T)))

マーカ WH は疑問副詞 ("wo", "wann") あるいは疑問代名詞 ("welcher") を含む節をし、自動的に翻訳処理の加工前段階に挿入される。

【PLACE】 場所

例： "ソウル市内のサボイホテルに宿泊しています"
"Ich übernachtete im Hotel Savoy in der Innenstadt von Seoul."

```
(define-pattern kakujo-ni-np :j-g np
  (?x "に" ?y) (:head 3)
=>
  (((!y "in" !x)(PLACE (PRAEPOSITION 2) 3 :case DAT))
   (
    ("サボイホテル") ("宿泊する")
   )
  )
)
```

【TIME】 時間、時刻

例： "十一時三十二分発の新幹線ですか"
"Shinkansen um elf Uhr zweiunddreißig?"

```
(define-regular-pattern setsubi-hatsu-reg :j-g nd
  (?x <> "発") (:head 3)
=>
  (((!um" !x)(TIME (PRAEPOSITION 1) 2 :case AKK))
   (
    ("N時N分")
   )
  )
)
```

《derivative usage》

【ORDINAL】 序数

例: "七つ目の駅に鶴橋という駅がございます"

"Es gibt einen Bahnhof mit Namen Tsuruhashi an der siebten Station."

```
(define-pattern setsubi-me :j-g nd
  (?x <cn-cn> "目") (:head 1)
  =>
  (((!x)(ORDINAL 1))
   (
    ((("七つ"))
     )
    )
  )
)
```

基数を序数に変更する。

【ORDINAL+】 序数のみの訳出

例: "はい十四日でしたらエービーシー共に空席がございますが"

"Ja. Es gibt für den vierzehnten einen freien Platz in A B C."

```
(define-pattern n-nichi :j-g terminal
  (?x "日") (:x 数詞 普通名詞)
  =>
  (((!x)(ORDINAL+ 1))
   (
    ((("00"))
     )
    )
  )
)
```

目下、(NOMEN "Tag")と(UNIT "Tag")のみを消去し、他の名詞性語句は落とさない。

例: (ORDINAL+ (KARDINALZAHL "14") (NOMEN "Tag")) --> "der vierzehnte"

(ORDINAL+ (KARDINALZAHL "14") (NOMEN "Juli")) --> "der vierzehnte Juli"

【KARDINAL+】 基数のみの訳出

例: "二時間から三時間ほどあれば往復できると思いますが"

"Ich glaube, man kann in etwa zwei bis drei Stunden hin und zurück kommen."

```
(define-pattern kakujo-kara-n+n1 :j-g n+n
  (?x "から" ?y) (:head 1 3)
  =>
  (((!x "bis" !y)(NP {OHNE} (KARDINAL+ 1) (PRAEPOSITION 2) {OHNE} 3))
   (
    ((("N時間") ("N時間"))
     )
    )
  )
)
```

目下、(NOMEN "Stunde")、(NOMEN "Person")と(NOMEN "Minute")のみを消去し、他の名詞性語句は落とさない。

例: (KARDINAL+ (KARDINALZAHL "2") (NOMEN "Minuten")) --> "zwei"

(KARDINAL+ (KARDINALZAHL "2") (NOMEN "Zimmer")) --> "zwei Zimmer"

【N+】 形容詞、動詞の名詞化

例: "まず重さを計らなければなりません"
"Man muß zunächst das Gewicht wiegen."

```
(define-pattern adj-sa :j-g terminal
  (?x "さ") (:x 形容詞)
  =>
  (((!x)(N+ 1))
   (
    ("重い")
    ("大きい")
    ("広い")
   )
  )
)
```

名詞化の優先順位:

- (1) 生成辞書に記述されている、形容詞または動詞の名詞形
 - (2) 生成辞書に記述されている、形容詞または動詞から派生した動詞または形容詞の名詞形 (【V+】、【A+】を参照のこと)
 - (3a) 現在分詞と過去分詞で形容詞的に用いられる動詞の基本形の名詞化 (【A+pprf】、【A+pres】参照のこと)
 - (3b) 動詞の基本型の名詞化: 性は中性
- * (1) から (3) の処理において符合する名詞が見つからない場合は作用しない

【V+】 名詞、形容詞の動詞化

例: "この写真を焼き増ししてください"
"Bitte ziehen Sie dieses Foto ab."

```
(define-pattern marker-cn-v-p :j-g p
  (?x <cn-v> ?y) (:head 1) (:x group-n)
  =>
  (((!x)(V+ 1))
   (
    ("焼き増し")("する")
   )
  )
)
```

動詞化の優先順位:

- (1) 生成辞書に記述されている、形容詞または名詞の動詞形
 - (2) 生成辞書に記述されている、形容詞または名詞から派生した名詞または形容詞の動詞形 (【N+】、【A+】を参照のこと)
 - (3) 現在分詞と過去分詞で形容詞的に用いられる動詞の基本形 (【A+pprf】、【A+pres】参照のこと)
- * (1) から (3) の処理において符合する動詞が見つからない場合は作用しない

【A+】 名詞の形容詞化

例: "はい当日の演題は勸進帳といまして歌舞伎の代表的な出し物になっております"
"Der Titel am betreffenden Tag heißt Kanjincho und er ist ein repräsentatives
Programm für das Kabuki Drama."

```
(define-pattern cn-teki :j-g terminal
  (?x "的") (:x 普通名詞 寸変名詞)
  =>
  (((!x)(A+ 1))
   (
    ("代表")
   )
  )
)
```


形容詞生成の優先順位：

- (1) 生成辞書に記述されている、動詞または名詞の形容詞形
 - (2) 生成辞書に記述されている、動詞または名詞から派生した名詞または動詞の形容詞形 (【N+】、【V+】を参照のこと)
 - (3a) 動詞 (派生形含む) の過去分詞の形容詞的用法 (【A+pprf】参照のこと)
 - (3b) 動詞 (派生形含む) の現在分詞の形容詞的用法 (【A+pres】参照のこと)
- * (1) から (3) の処理において符合する形容詞が見つからない場合は作用しない

【A+pprf】 過去分詞の形容詞的用法

例： "ご希望の車のクラスをおっしゃってください"
"Bitte sagen Sie mir die gewünschte Klasse vom Auto."

```
(define-pattern rentaijo-no-n+n :j-g n+n
  (?x "の" ?y) (:head 3)
=>
  (((!x !y) (NP 1 2))
   (
    (("希望") ("クラス"))
   )
  )
  (
   (x
    ((サ変名詞 "希望") (A+pprf (VERB "wünschen"))))
  )
 )
 )
 )
```

【A+pres】 現在分詞の形容詞的用法

例： "その部屋で十分ですが料金は追加になりますか"
"Das Zimmer ist ausreichend. Aber erhöht sich die Gebühr?"

```
(define-regular-pattern kakujo-de-reg :j-g np
  (?x "で" ?y) (:head 3) (:x group-n)
=>
  (((!x !y) (SUB 1))
   (
    (("部屋") ("十分"))
   )
  )
  (
   (y
    ((形容名詞 "十分") (VP (A+pres (VERB "ausreichen")) (HILFSVERB "sein"))))
  )
 )
 )
 )
```

【FIX-EXP】 慣用表現

例： 変換辞書： ((感動詞 "ありがとうございました") (FIX-EXP "vielen" (FIX-CAP "Dank")))
((普通名詞 "八月納涼歌舞伎") (EIGENNAME+ "' Sommer-Kabuki " (FIX-EXP "im")
"August "' :GENDER NTR))

機能：品詞とともに明示的にマークされた単語だけが語形的に解析される。その他は影響を受けない。
(関連して、FIX-CAP, FIX-UP, FIX-INTRO, FIX-END 参照のこと)

注:

FIX-EXP は、他の文要素によって修飾されないイディオム的な表現と定義されるので、使用に際しては注意が必要である。

例: ((感動詞 "申し訳ございません")(FIX-EXP ". es tut mir leid ."))
★問題★ "本当に申し訳ございません" -?-> "Wirklich. Es tut mir leid."
 =!=> "Es tut mir wirklich leid."

EIGENNAME 中の小文字で始まる単語には、FIX-EXP の記述が必要になる。(上述の例を参照のこと)

名詞の直前に挿入されて修飾する前置詞句の場合。

例: "ここから一番近いスーパーマーケットはどこにありますか"
 "Wo ist der von hier aus nächste Supermarkt?"

```
(define-regular-pattern kakujo-kara-np :j-g np
  (?x "から" ?y) (:head 3) (:x group-n)
  =>
  (((("von" !x "aus" !y)(FIX-EXP 1 2 3)) ;; PP modifier in front of noun
   (
    (("ここ") ("近い"))
    (("ミュンヘン") ("近い"))
    (("ミュンヘン") ("行く"))
   )
  )
)
```

【MAL】 「数字・mal」の副詞化

例: "二等席を二名ですぬ演目の方はご存じでしょうか"
 "Zweimal in der zweiten Klasse, nicht wahr? Kennen Sie das Programm?"

```
(define-pattern kakujo-wo-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y "in" !x)(PP (PRAEPOSITION 2) 3 :case DAT))
   (
    (("N等席")("N名"))
   )
   (
    (y
     ((普通名詞 "N名")(n-mei-reg (MAL !x)))
    )
   )
  )
)
```

【MALIG】 「数字・malig」の形容詞化

★現在(99/8) 使用例なし。

《functional usage》

【SUPERLATIV】 最高級 % 最高級名詞

例: "一番近いところで釜山港ですぬ"
 "Der nächste Ort ist der Hafen Pusan."

```
(define-pattern marker-cn-adj-ap :j-g ap
  (?x <cn-adj> ?y) (:head 3) (:x 普通名詞 aa ap dn)
=>
  (((!x !y)(AP 1 2))
   (
    (("一番")("近い"))
    )
   (
    (x
     ((普通名詞 "一番")(SUPERLATIV (DUMMY "meisten")))
    )
   )
  )
)
```

SUPERLATIV のマーカが付与された文要素に名詞が後続する場合、その文要素の最高級だけが伝授されて名詞を修飾する (例上記)。それ以外は 最高級の副詞節、‘am...(s)ten’ を生成する (例下記)。

例: "神戸の夜景でしたら山上から見るのが一番美しいと思いますが"
 "Ich glaube, der Wachtanblick von Kobe ist am schönsten, wenn man ihn vom Berggipfel aus betrachtet."

```
(define-pattern marker-cn-adj-ap :j-g ap
  (?x <cn-adj> ?y) (:head 3) (:x 普通名詞 aa ap dn)
=>
  (((!x !y)(AP 1 2))
   (
    (("一番")("美しい"))
    )
   (
    (x
     ((普通名詞 "一番")(SUPERLATIV (DUMMY "meisten")))
    )
   )
  )
)
```

【CONCAT】 複合名詞

例: "空港バスの運行時間はもう終わりました"
 "Die Betriebszeit des Flughafenbusses ist schon beendet."

```
(define-pattern marker-cn-cn-n :j-g n
  (?x <cn-cn> ?y) (:head 3)(:x group-n) (:y group-n)
=>
  (((!x !y)(CONCAT 1 2))
   (
    (("空港")("バス"));CONCAT
    )
  )
)
```

【DIGIT】 数字の羅列

例: "お電話番号が零三三二四五二二三三ですね"
 "Die Telefonnummer ist null drei drei zwei vier fünf zwei zwei drei drei."

```
(define-pattern kakujo-ga-np :j-g np
  (?x "が" ?y) (:head 3)
=>
  (((!x "sein" !y)(SUB 1)(*HILFSVERB 2)(DIGIT 3))
   (
    (("電話番号") ("00列"))
    )
  )
)
```

[AM] 午前

例: "平日が午前八時三十分から午後九時までです"
"Zwischen acht Uhr dreißig morgens und neun Uhr abends am Werktag."

```
(define-pattern gozen-n-ji-fun :j-g terminal
  ("午前" ?x "時" ?y "分") (:x 数詞) (:y 数詞)
  =>
  (((!x "Uhr" !y) (AM 1 (UNIT 2) 3))
   (
    (("00") ("00"))
   )
  )
)
```

[PM] 午後

例: "午後七時に二名予約できますか"
"Kann man für sieben Uhr abends für zwei Personen reservieren?"

```
(define-pattern gogo-n-ji :j-g terminal
  ("午後" ?x "時") (:x 数詞)
  =>
  (((!x "Uhr") (PM 1 (UNIT 2)))
   (
    (("00"))
   )
  )
)
```

[YEAR] 年号

例: "1999年" --> (YEAR "1999") --> "1999"
"昭和43年" --> (YEAR "43" :tenno :shouwa) --> "1968"

数詞から年号への変換。

1100以上2000未満の数字: "1999" → "neunzehnhundertneunundneunzig"

それ以外: 通常の数詞 "2000" → "zweitausend"

★現在(99/8)使用例なし。

[ETAGE] 階数

例: "分かりました一階の売店ですね"
"Okay. Verkaufsstand in der ersten Etage, oder?"

```
(define-pattern n-etage :j-g terminal
  (?x "階") (:x 数詞)
  =>
  (((!x) (ETAGE 1))
   (
    (("00"))
   )
  )
)
```

[FLIGHT] 便

例: (1) "九月十日出発東京行きケーイー二百七十便を一席予約してください"
"Bitte reservieren Sie einen Platz für den Flug K E zweihundertsiebzig,
Abfahrt am zehnten September nach Tokyo."

(2) "三便ありますが天候によって欠航になることもあります"
"Es gibt drei Flüge. Aber es kann aufgrund des Wetters ausfallen."

```
(define-pattern n-bin :j-g terminal
  (?x "便") (:head 1)(:x 数詞)
  =>
  (((!x "Flug")(FLIGHT 1 (NOMEN 2)))
   (
    ("00")
    ("00列")
   )
  )
)
```

飛行機の便に関する生成マーク:

(1) "Flug ...": ある特定の便を指す場合。エアライン名、文字、閾値 (現在は 10) 以上の基数詞ともにあらわされる。

例: (NP (BUCHSTABE "k") (BUCHSTABE "e") (FLIGHT (KARDINALZAHL "270") (NOMEN "flug")))
 --> "Flug KE270"

(2) "... Flüge": 便数を指す場合。序数詞、閾値 (現在は 10) 以下の基数詞ともにあらわされる。

[FIX-CAP] 最初の文字の大文字書き。

例: 変換辞書: ((感動詞 "お疲れ様です") (FIX-EXP "vielen" (FIX-CAP "Dank") "für"
 (FIX-CAP "Ihre Mühe")))

機能: 品詞とともに明示的にマークされた単語だけが語形的に解析される。その他は全て、当生成処理 (最初の文字を大文字書きする) 影響を受けない。(FIX-EXP, FIX-UP, FIX-INTRO, FIX-END 参照のこと)

注:

FIX-CAP は、他の文要素によって修飾されないイディオム的な表現と定義される。

FIX-EXP 中の大文字書きに必要。(上記例を参照のこと)

EIGENNAME においてはマークされていない全ての単語が自動的に FIX-CAP として扱われる。

例: (EIGENNAME "Lotteworld" (NOMEN "Hotel") :GENDER NTR)
 = (EIGENNAME (FIX-CAP "Lotteworld") (NOMEN "Hotel") :GENDER NTR)

[FIX-UP] 全ての文字の大文字書き。

例: ((普通名詞 "オービービール") (EIGENNAME (FIX-UP "OB") (NOMEN "Bier") :gender NTR))

機能: 品詞とともに明示的にマークされた単語だけが語形的に解析される。その他は全て、当生成処理 (全文字を大文字書きする) 影響を受けない。(FIX-EXP, FIX-CAP, FIX-INTRO, FIX-END 参照のこと)

注:

EIGENNAME 中のイニシャルの大文字書きに必要。(上述の例を参照のこと)

[NP-END] 接尾辞 - NP において名詞に続いて無冠詞で記述される。

例: "市バス五十九番に乗っていただきます"
 "Sie nehmen den städtischen Bus Nummer neunundfünfzig."

```
(define-pattern n-ji-hun :j-g terminal
  (?x "時" ?y "分") (:x 数詞)(:y 数詞)
  =>
  (((!x "Uhr" !y)(TIME 1 (UNIT 2 :gender NTR) (NP-END 3)))
   (
    ("00")("00")
   )
  )
)
```

[DUMMY] ダミー

訳出されないが、単なる deletion とは違いその下部構造を上部に伝達する。

例: "いえその右下の部分です"
"Nein. Da rechts unten."

```
(define-pattern rentaijo-no-n+n :j-g n+n
  (?x "の" ?y) (:head 3)
  =>
  (((!x !y)(NP 1 {OHNE} 2))
   (
    ("下") ("部分")
   )
  )
  (
   (y
    ((普通名詞 "部分") (DUMMY "Teil"))
   )
  )
 )
 )
```

普通名詞 "部分" = "Teil" は訳出されないが、"部分"に係る (その) (右) (下) が落ちていない。

《inflectional usage》

[ATTR] 属性の記述

例: "ここに鞆がありませんでしたか"
"Gab es hier keine Tasche?"

```
(define-pattern sen-deshita :j-g sm
  (?x "でした") (:head 1)
  =>
  (((!x)(ATTR 1 :tense IMPERFEKT)) →属性:過去時制
   (
    ("ある")
   )
  )
 )
```

[EIGENNAME] 固有名詞

例: ((普通名詞 "オズボーンカメラショップ") (EIGENNAME (NOMEN "Fotogeschäft")
"Osborne" :gender NTR))

名詞句中の語形変化する文成分は確実にマークされなければならない。

例: ((普通名詞 "アルテピナコテーク") (EIGENNAME (ADJEKTIV "Alte")
(NOMEN "Pinakothek") :GENDER FEM))
--> "die Alte Pinakothek"
(ATTR (EIGENNAME (NOMEN "Fotogeschäft") "Osborne" :gender NTR) :case GEN)
--> "des Fotogeschäftes Osborne"

マークされていない単語は全て自動的に FIX-CAP として扱われる。

イニシャルは FIX-UP を用いて大文字書きされる。

例: ((普通名詞 "エービーエム日本社") (EIGENNAME (FIX-UP "ABM") "Japan"))
--> "ABM Japan"

機能語は FIX-EXP を用いて小文字書きされる。

例：((普通名詞 "エーアンドビー出版") (EIGENNAME (NOMEN "Verlag") "A"
(FIX-EXP "und") "B" :GENDER MAS))
--> "Verlag A und B"

注：料理、動物の名前などは複数の語句から成立しない限り、普通の名詞とする。

例：((普通名詞 "お好み焼") (NOMEN "Okonomiyaki"))

[EIGENNAME+] 無冠詞の固有名詞

人名

例：((普通名詞 "ズザンネ") (EIGENNAME+ (FIX-CAP "Susanne") :gender FEM))

演題、書名など

ミュージカルの演題など (" " でマークされる)

例：((普通名詞 "不思議の国のアリス") (EIGENNAME+ "' Alice" (FIX-EXP "im") "Wunderland "'))
--> "' Alice im Wunderland '"

複数の語句からなる都市、国、料理、動物の名前

例：((普通名詞 "ロサンゼルス") (EIGENNAME+ "Los Angeles" :GENDER NTR))

注：それ以外の都市、国名は普通名詞とする ({OHNE} か {BESTIMMT} を付与)

例：((普通名詞 "ワシントン") (NP OHNE (NOMEN "Washington")))
((普通名詞 "スイス") (NP BESTIMMT (NOMEN "Schweiz")))

《topological usage》

[INTRO] 文要素を文頭に配置。

例："それと劇場の場所を教えてください"
"Und bitte sagen Sie mir den Ort des Theaters."

```
(define-pattern beginning-soreto :j-g is
  ("それと" ?x) (:head 2)
  =>
  (((("und" !x)(INTRO (KOORD-KONJUNKTION 1)))
   (
    ("教える" "てください")
   )
  )
 )
 )
```

[FIX-INTRO] 句などを文頭に配置。

例："エアコンの調子が悪いみたいですよ"
"Es scheint, die Klimaanlage funktioniert nicht richtig."

```
(define-pattern juntaijo-mitai-pm :j-g pm
  (?x "みたい") (:head 1)
  =>
  (((("es scheint ," !x)(FIX-INTRO 1))
   (
    ( ("調子が悪い")
      )
    )
  )
)
```

注：INTRO と違って、文要素の解析を行なわない。

【END】 文要素を文末に配置。特に、関係文による主文の中断を避けたいとき有効。

例："割引してくれる宿があるときいたのですがどこにありますか"
 "Ich habe gehört, daß es ein Gasthaus gibt, das eine Ermäßigung anbietet.
 Wo ist es?"

```
(define-pattern marker-/kihon-cn-s+n :j-g s+n
  (?x </kihon-cn> ?y) (:head 3)
  =>
  ((((!y "der/die/das" !x)(END (SATZGEFUEGE (REL-S (SUB (RELATIVPRONOMEN 2)) 3))))
   (
    ( ("割引する" "てくれる")("宿")
      )
    )
  )
)
```

【FIX-END】 句などを文末に配置。

例："終日コースの方が絶対お得ですよね"
 "Die Tagestour ist sicherlich vorteilhafter, oder?"

```
(define-pattern sen-yone :j-g se
  (?x "よね") (:head 1)(:x group-s)
  =>
  ((((!x ", oder ?")(FIX-END 2))
   (
    ( ("得")
      )
    )
  )
)
```

注：END と違って、文要素の解析を行なわない。

【AP+】 副詞句を2位の動詞直後に配置。

例："六時からどうしてもサッカーの放送を見たいんです"
 "Ich möchte unbedingt gerne das Fußballprogramm ab sechs Uhr sehen."

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 aa)
  =>
  ((((!x !y)(AP+ 1))
   (
    ( ("どうしても") ("見る")
      )
    )
  )
)
```


☆ 上記 AP で書くと:

↓
<<!x !y>>(AP 1)

↓
"Ich möchte gerne das Fußballprogramm ab sechs Uhr unbedingt sehen."

【VERB-ADD】 動詞の付加語

例: "右の足首がずきずき痛みます"
"Das rechte Fußgelenk tut bohrend weh."

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 aa)
=>
  (<<!x !y>>(VERB-ADD 1))
  (
    (("ずきずき") ("痛む"))
  )
)
```

*** Generation Marker (生成マーカー)

<否定>

{NOT}: 否定

例: "いなくて失礼いたしました"
"Es tut mir leid, daß ich nicht da war."

```
(define-regular-pattern negation-nai-reg :j-g pm
  (?x "ない") (:head 1);(:x group-s)
=>
  (<<"ich" {NOT} !x>>(*SUB (PERSONALPRONOMEN 1)))
  (
    (("いる-"))
    (("分かる"))
  )
)
```

{NOT} が不定冠詞をともなった名詞とともに使われると、否定の冠詞 "kein" が生成される。

例: "バスはもうないと思います"
"Ich glaube, es gibt keinen Bus mehr."

変換辞書: (<<形容詞 "無い">>(*AKK-OBJ (PERSONALPRONOMEN "es"))
(SUB (PERSONALPRONOMEN "es"))
(VP NOT (VERB "geben")))

```
(define-pattern kakarijo-ha-np :j-g np
  (?x "は" ?y) (:head 3) (:x group-n 副詞)
=>
  (<<!y !x>> (AKK-OBJ 2))
  (
    (("バス") ("ある"))
  )
)
```

《文の並列》

{COORD} : 句読点の無い主文及び従属節の並列

例: "はいかしこまりました京都といえますと寺社仏閣がやはり有名なんですその中でも特に金閣銀閣がよろしいかと思えます"

"Ja. Alles klar. In Kyoto sind Tempel und Schreine doch bekannt und ich glaube, der Kinkaku und der Ginkaku sind davon besonders schön."

```
(define-pattern conj-ga-ss :j-g ss
  (?x "が" ?y) (:head 3) (:x np se sm np sp ap aa pm nm p word-p)
  =>
  (((!x "und" !y) (SATZREIHE {COORD} (INTRO (KOORD-KONJUNKTION 2)) 3))
   (
    ("有名") ("思う")
   )
  )
)
```

《冠詞》

{OHNE} : 無冠詞

例: ((普通名詞 "徒歩") (PP (PRAEPOSITION "zu") (NP OHNE (NOMEN "Fuß"))) :case DAT))
--> "zu Fuß"

{BESTIMMT} : 定冠詞挿入

例: ((普通名詞 "最終便") (NP {BESTIMMT} (ADJEKTIV "letzter") (NOMEN "Flug")))
--> "der letzte Flug"

{UNBESTIMMT} : 不定冠詞挿入

例: ((副詞 "しばらく") (AP {UNBESTIMMT} (NOMEN "Moment") :gender MAS :case AKK))
--> "einen Moment"

{POSS} : 人称の所有代名詞

例: "子供さんのパスポートを見せてください"
"Bitte zeigen Sie mir den Reisepaß von Ihrem Kind."

{MEIN} : 一人称の所有代名詞

例: ((普通名詞 "妻") (NP {MEIN} (NOMEN "Frau")))
--> "meine Frau"

{IHR} : 二人称敬称の所有代名詞

例: ((普通名詞 "御社") (NP {IHR} (NOMEN "Firma")))
--> "Ihre Firma"

《敬称ボタン》

{HERR} : 男性 --> "Herr ..."
{FRAU} : 女性 --> "Frau ..."
{VORNAME} : 名前 --> "..." (敬称ボタン無し)

〈主語補完〉

* 研究の詳細は [6] を参照のこと。

|<AGEN S>| : 一人称、単数 --> (PERSONALPRONOMEN "ich")
|<AGEN K>| : 一人称、複数 --> (PERSONALPRONOMEN "wir")
|<AGEN H>| : 二人称、丁寧 / 単数 --> (PERSONALPRONOMEN "Sie"/"du")
|<AGEN Y>| : 二人称、丁寧 / 複数 --> (PERSONALPRONOMEN "Sie"/"ihr")
|<AGEN G>| : 三人称、一般 --> (PARTIKEL "man")
|<AGEN X>| : 三人称、単数 --> (PERSONALPRONOMEN "er/sie/es")
|<AGEN F>| : 三人称、中性 --> (PERSONALPRONOMEN "es")

*** Special Generation Marker Characters (特殊生成マーカー)

[*] デフォルトの文成分。主語、目的語など。(*SUB, *AKK-OBJ)
default でない文成分によって上書きされなければ訳出される。

例: "予約をキャンセルしたい"
"Ich möchte die Reservierung rückgängig machen."

*** Generation ***
((*SUB (PERSONALPRONOMEN "ich")) (MODALVERB "möchte" :TENSE KONJUNKTIV-2)
(*AKK-OBJ (PERSONALPRONOMEN "er/sie/es"))
(VP (ADJEKTIV "rückgängig") (VERB "machen"))
(AKK-OBJ BESTIMMT (NOMEN "reservierung")))

(*SUB (PERSONALPRONOMEN "ich")) はデフォルトでない主語が他にないから訳出され、
「キャンセルする」の目的語 (*AKK-OBJ (PERSONALPRONOMEN "er/sie/es")) は
(AKK-OBJ BESTIMMT (NOMEN "reservierung")) に上書きされる。

[!] 主語補完の特殊マーカー (!SUB)

主語補完マーカー (上記参照のこと) の自動的な書き換え。

優先順位: SUB > *SUB > !SUB

*SUB の用法について:

(1) 補完モジュールによる主語の選択誤りを修正する。

(2) 補完モジュールが主語を補完できないとき (動作性の単語を含まない文章など) に主語を挿入する。

*** default article for part of sentences (生成マーカーのデフォルト冠詞)

definite {BESTIMMT} → SUB, DAT-OBJ, GEN-OBJ, PP, EIGENNAME, ORDINAL
定冠詞 主語、3格目的語、2格目的語、前置詞句、固有名詞、序数。

indefinite {UNBESTIMMT} → AKK-OBJ, NOM-OBJ
不定冠詞 4格目的語、同格の主語。

force none {OHNE} → EIGENNAME+, NP-END
無冠詞 固有名詞+, 接尾辞。

デフォルトの冠詞は冠詞マーカー {BESTIMMT}, {UNBESTIMMT}, {OHNE} によって上書きされる。

 *** sentence clauses (文ケース)

[SATZREIHE] 《主文の並列 = 重文》

例: "チェックアウトしますので会計お願いします"
 "Ich reise ab. Die Rechnung, bitte."

```
(define-pattern causal-node-ss :j-g ss
  (?x "ので" ?y) (:head 3)
=>
  (((!x !y)(SATZREIHE 2))
   (
    ("チェックアウトする") ("おねがいます")
   )
  )
)
```

[SATZGEFUEGE] 《主文+副文あるいは句 = 複合文》

副文はその細分化された種類の記述が必要となる。(副文の種類については後述→subordinate clauses)

例: "レンタカーは便利というご意見が多い"
 "Man sagt oft, daß der Mietwagen praktisch ist."

```
(define-pattern rentaijo-toiu-s+n :j-g s+n
  (?x "という" ?y) (:head 3)
=>
  (((!y !x)(SATZGEFUEGE (INH-S-DASS 2)))
   (
    ("便利") ("意見")
   )
  )
)
```

[SATZ] Aussagesatz 《平叙文》

+ V-pos: verb-second (定動詞2位)
 + V-mood: INDIKATIV / KONJUNKTIV (直接法/接続法)
 (never: IMPERATIV !) (命令法にはならない)
 + punctuation: "." (ピリオド)

[SATZ_verb-first] Aussagesatz with preceding subord-clause 《先行する従属節を伴う平叙文》

+ V-pos: verb-first (定動詞1位)
 + V-mood: INDIKATIV / KONJUNKTIV (直接法/接続法)
 (never: IMPERATIV !) (命令法にはならない)
 + punctuation: "."
 + others: - preceding subord before main-clause
 (主節に先行して従属節がある)

例: "もちろん三人様でお泊まりいただけます"
 "Natürlich können drei Personen übernachten."

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 ap)
=>
  (((!x !y)(SATZ_verb-first (INTRO 1) 2))
   (
    ("もちろん") ("泊まる") ("いただける")
   )
  )
)
```

[FRAGE] question 《疑問文》

- + WH phrase included ? --> **[WH-Q]**
- + otherwise --> **[YN-Q]**

- + punctuation: "?" (疑問符)

[WH-Q] Ergaenzungsfrage 《補足疑問》

- + V-pos: verb-second; sometimes: verb-final (定動詞2位: 時により文末→副文で用いられる時)
(never: verb-first !) (定動詞1位にならない)
- + V-mood: INDIKATIV / KONJUNKTIV (直接法/接続法)
(never: IMPERATIV !) (命令法にはならない)
- + punctuation: "?" (疑問符)

例: "どうやって払えばいいですか"
"Wie soll ich bezahlen?"

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 ap)
=>
  (((!x !y)(WH 1))
   (
    (("どうやって") ("払う" "ばよい"))
   )
  )
)
```

[YN-Q] Entscheidungsfragesatz 《決定疑問》

- + V-pos: verb-first (定動詞1位)
- + V-mood: INDIKATIV / KONJUNKTIV (直接法/接続法)
(never: IMPERATIV !) (命令法にはならない)
- + punctuation: "?" (疑問符)

例: "片道ですか"
"Einfach?"

```
(define-pattern sen-n-desuka-sm :j-g sm
  (?x "です" "か") (:head 1) (:x group-n)
=>
  (((!x) (FRAGE 1))
   (
    ("片道")
   )
  )
)
```

[FRAGE_verb-first] - question with preceding subord-clause 《先行する従属節を伴う疑問文》

★(99/8) 使用例文無し

[IMP-S-I] Aufforderungssatz 《命令文》

- + V-pos: verb-first (定動詞1位)
- + V-mood: IMPERATIV (命令法)
- + punctuation: "." (ピリオド)

例: "ガソリンの方は満タンにしてご返却をお願いいたします"
 "Tanken Sie Benzin voll. Und geben Sie zurück, bitte."

```
(define-pattern imp-kakujo-woonegaishimasu-np :j-g np
  (?x "を" ?y) (:head 3) (:x group-n)
  =>
  (((!y "Sie" !x)(IMP-S-I (PERSONALPRONOMEN 2 :person 2a) 3))
   (
    (("返却")("おねがいたします"))
   )
  )
)
```

[IMP-S-II] Aufforderungssatz 《命令文》

+ V-pos: verb-first (定動詞1位)
 + V-mood: IMPERATIV (命令法)
 + punctuation: "!" (感嘆符)

★現在 (99/8) 使用例なし。

[DES-S-I] Wunschsatz Typ I 《願望文》

+ V-pos: verb-second (person: 3) (定動詞2位)
 + V-mood: KONJUNKTIV I (接続法 I 式)
 + punctuation: "." / "!" (ピリオド/感嘆符)

★現在 (99/8) 使用例なし。

[DES-S-II] - Wunschsatz Typ II 《願望文》要求語法

+ V-pos: verb-first / verb-final (定動詞1位/後置)
 + V-mood: KONJUNKTIV II (接続法 II 式)
 + punctuation: "!" (感嘆符)
 + others: - characteristic particle [bloss,nur,doch]

(願望文の類は :tense KONJUNKTIV-2 でカバーできる)

参考例: "教えてほしい"

"Ich möchte wissen."

```
(define-regular-pattern conj-tehoshii-reg-sp :j-g sp
  (?x "て" "欲しい") (:head 3)
  =>
  (((("ich" "möchte" !x)(*SUB (PERSONALPRONOMEN 1))
      (MODALVERB 2 :tense KONJUNKTIV-2))
   (
    ("教える")
   )
  )
  (
    (x
     ((本動詞 "教える")(VERB "wissen"))
    )
  )
)
```

★現在 (99/8) 使用例なし。

;----- subordinate clauses (副文) -----;

[REL-S] Relativsatz (関係文)

- + intro: - :RELATIV- / :INTERROGATIVPRONOMEN (イントロ: 関係代名詞/疑問代名詞)
[der, die, das, welcher, welche, welches]
- :ADVERB (Pronominaladverb) (イントロ: 副詞 [代名詞的副詞 = da, hier, wo と前置詞の融合形])
[wo, wie, wohin, wodurch]
- + pronom-pos: - first: [<REL-PRON> ...] (代名詞の位置: 1位 [<代名詞> ...])
- second: [<PREP> <REL-PRON> ...] (代名詞の位置: 2位 [<前置詞><代名詞> ...])
- + others: - near to reference word (先行詞の近く)

例: "日帰りで行ける所を教えてください"

"Bitte sagen Sie mir einen Ort, der für die Tagesreise geeignet ist."

```
(define-pattern marker-/kihon-fn-s+n :j-g s+n
  (?x </kihon-fn> ?y) (:head 3)(:x group-s) (:y group-n)
  =>
  (((!y "der/die/das" !x)(SATZGEFUEGE (REL-S (INTRO (RELATIVPRONOMEN 2)
                                                    :case NOM) 3)))
   (
    (("行ける")("所"))
    (("分かる")("物"))
    (("お持ちする")("もの"))
    (("言う" "れる")("ところ"))
   )
  )
)
```

[KAU-S] Kausalsatz (因由文)

- + intro: [da, weil; weswegen, weshalb; zumal(da); dafuer_, dass]

例: "歩いたことがないのでわかりません"

"Da ich noch nie zu Fuß gegangen bin, kenne ich mich nicht aus."

```
(define-pattern causal-node-ss :j-g ss
  (?x "ので" ?y) (:head 3)
  =>
  (((("da" !x !y)(SATZGEFUEGE (KAU-S (INTRO (SUBORD-KONJUNKTION 1) 2)))
   (
    (("無い")("分かる"))
   )
   (
    (y
     ((本動詞 "分かる")(VP (REFLEXIVPRONOMEN "sich") (VERB "auskennen") :case AKK))
    )
   )
  )
)
```

[KONS-S] Konsekutivsatz (結果文)

- + intro: [dass, so dass, so...dass, um...zu, zu, als dass]

★現在 (99/8) 使用例なし。

[KONZ-S] Konzessivsatz (認容文)

- + intro: [obgleich, obwohl, obschon, wenn auch, wengleich, wenschon]

例: "設定をいじっても涼しくならないんですよ"
 "Obwohl ich die Regulierung ändere, wird es nicht kühl."

```
(define-pattern conj-temo-ss :j-g ss
  (?x "でも" ?y) (:head 3)
  =>
  (((("obwohl" "ich" !x !y)(SATZGEFUEGE (KONZ-S (INTRO (SUBORD-KONJUNKTION 1))
                                                (*SUB (PERSONALPRONOMEN 2)) 3)))
   (
    (("オンする") ("なる"))
    (("いじる") ("なる"))
   )
  )
)
```

[TEM-S] Temporalsatz 《時の副文》

+ intro: * simultan: [während, indem, indes, indessen, solange, sobald, sowie, sooft, als, wie, wenn]
 * previous: [nachdem, als, seit, seitdem, sobald, sowie, wenn]
 * posterior: [bis, bevor, ehe]

例: "待合室でアナウンスがあるまでお待ちください"
 "Bitte warten Sie im Warteraum, bis eine Durchsage erfolgt."

```
(define-pattern temp-made-sp :j-g sp
  (?x "まで" ?y) (:head 3)
  =>
  ((((!y "bis" !x)(SATZGEFUEGE (TEM-S (INTRO (SUBORD-KONJUNKTION 2)) 3)))
   (
    (("ある") ("待つ"))
   )
  )
)
```

[KOND-S] Konditionalsatz 《条件文》

+ intro: [wenn, wofern, sofern, falls]

例: "万が一配達できなければ手元に戻って来ますから"
 "Es kommt zu Ihnen zurück, falls es nicht zugestellt wird."

```
(define-pattern adverb-ap :j-g ap
  (?x <adv-> ?y) (:head 3) (:x 副詞 ap)
  =>
  ((((!x !y)(SATZGEFUEGE (KOND-S (INTRO 1) 2)))
   (
    (("万が一") ("配達する"))
   )
   (
    (x
     ((副詞 "万が一")(SUBORD-KONJUNKTION "falls")))
   )
  )
)
```

[FIN-S] Finalsatz 《目的文》

+ intro: [damit, dass, auf dass, um...zu]

例: "無くさないようご注意ください"
 "Bitte passen Sie auf, damit es nicht verloren geht."


```

(define-pattern modal-you-sp :j-g sp
  (?x "よう" <auxv-> ?y) (:head 4)
  =>
  (((!y "damit" "er/sie/es" !x)(SATZGEFUEGE (FIN-S (INTRO (SUBORD-KONJUNKTION 2))
    (*SUB (PERSONALPRONOMEN 3) 4)))
  (
    (("無くす")("注意する"))
  )
  (
    (x
      ((本動詞 "無くす")(VP (ADJEKTIV "verloren")(VERB "gehen"))))
  )
  )
  )

```

[MOD-S] Modalsatz 《状況文》

+ intro: [indem, dardurch,..,dass, so...dass,]

例: "じゃあホテルで何か買い物でもして時間をつぶそうかしら"

"Also, kann ich die Zeit verbringen, indem ich einige Einkäufe im Hotel mache?"

```

(define-pattern conj-te-ss :j-g ss
  (?x "て" ?y) (:head 3) (:x group-s)(:y group-s)
  =>
  (((!y "indem" !x)(SATZGEFUEGE MOD-S (INTRO (SUBORD-KONJUNKTION 2)) 3))
  (
    (("する") ("つぶす"))
  )
  )
  )

```

[AUS-P] Aussagepräzisierung 《》

+ intro: [außer daß, nur daß; was; insofern; soviel, soweit; wobei]

例: "はい環状線は同じ駅を回っている線なんですけれども外回りと内回りと逆方向に回っている形になっております"

"Die Loop Line ist die Linie, die an den gleichen Bahnhöfen vorbeifährt, wobei die Sotomawari und die Uchimawari in die entgegengesetzte Richtung fahren."

```

(define-regular-pattern conj-keredomo-ss :j-g ss
  (?x "けれども" ?y) (:head 3)(:x np se sm sp ap aa pm nm s+m p word-p)
  =>
  (((!x "wobei" !y)(SATZGEFUEGE AUS-P (INTRO (ADVERB 2)) 3))
  (
    (("線")("回る"))
  )
  )
  )

```

[IND-R] Indirekte Rede 《間接話法》

+ V-mood: KONJUNKTIV 接続法 (I式)

★現在 (99/8) 使用例なし。

《INH-S》 Inhaltssatz 《内容文》

[INH-S-DASS]

- intro: "daß"
- V-pos: verb-final

例: "こんなに早く時間がなくなるのが残念です"
"Schade, daß die Zeit so schnell vorbeigeht."

```
(define-pattern kakujo-ga-np :j-g np
  (?x "が" ?y) (:head 3)
  =>
  (((!y !x)(SATZGEFUEGE INH-S-DASS 2))
   (
    (("無い" "なる") ("残念"))
   )
  )
)
```

[INH-S-ZU]

- intro: "--"
- V-pos: INFINITIV with/without "zu" + verb-final

例: "途中で乗換は必要ですか"
"Ist es notwendig, unterwegs umzusteigen?"

```
(define-pattern kakarijo-ha-np :j-g np
  (?x "は" ?y) (:head 3)
  =>
  (((!y !x)(SATZGEFUEGE (INH-S-ZU 2)))
   (
    (("乗換") ("必要"))
   )
  )
)
```

[INH-S-WENN]

- intro: "wenn" (not konditional / temporal)
- V-pos: verb-final

例: "ご予約なさるのがよろしいかと思えます電話番号を申し上げます"
"Ich glaube, es ist in Ordnung, wenn Sie reservieren.
Ich sage Ihnen die Telefonnummer."

```
(define-pattern kakujo-ga-np :j-g np
  (?x "が" ?y) (:head 3) (:x group-n)
  =>
  (((!y !x)(SATZGEFUEGE INH-S-WENN 2))
   (
    (("見る") ("美しい"))
    (("予約する") ("よろしい"))
   )
  )
)
```

[INH-S-ALS]

- intro: "als" (not temporal, no compare, not "als ob")
- V-pos: verb-final

★現在(99/8) 使用例なし。[als er ploetzlich auftauchte, freute mich das]

[INH-S-ALSOB]

- intro: "als" / "als ob"
- V-pos: verb-second / verb-final

★現在 (99/8) 使用例なし。 [mir schien, als wuesste er nicht weiter]
[mir schien, als ob er nicht weiter wuesste]

[INH-S-WIE]

- intro: "wie"
- V-pos: verb-final

★現在 (99/8) 使用例なし。 [ich merkte, wie mein Kraefte nachliessen]

[INH-S-ANF]

- intro: -* (reported speech)
- V-pos: verb-second

例: "ホテルなんですがミッドタウンのデイズホテルに泊まっています"
"Bezüglich des Hotels, übernachtete ich im Day's Hotel in Midtown."

```
(define-pattern conj-nanodesuga :j-g np
  (?x "なのです" "が" ?y) (:head 4)(:x group-n)
=>
  (((("bezüglich" !x !y)(INTRO (SATZGEFUEGE INH-S-ANF
                                (PP (PRAEPOSITION 1) 2 :case GEN))))
   (
    ((("ホテル")("泊まる"))
     ((("値段")("N円"))
      ((("日")("おねがいしたい"))
       )
      )
    )
  )
)
```

[INH-S-OB]

- intro: "ob"
- V-pos: verb-final

例: "近くにバス停があったら教えて下さい"
" Bitte sagen Sie mir, ob es in der Nähe eine Bushaltestelle gibt."

```
(define-pattern condition-tara-ss :j-g ss
  (?x "たら" ?y) (:head 3)
=>
  ((((!y !x)(AKK-OBJ (SATZGEFUEGE (INH-S-OB 2))))
   (
    ((("ある") ("教える"))
     )
    )
  )
)
```

[INH-S-WH]

- intro: WH conjunction (reported speech)
- V-pos: verb-final

例: "だいたいどのあたりでなくしたか見当はつきますか"
"Wissen Sie vielleicht, ungefähr wo es verlorengegangen ist?"

```

(define-pattern marker-ka+endp-sp :j-g sp
  (?x "か" <endp-> ?y) (:head 4)
  =>
  (((!y !x)(SATZGEFUEGE (INH-S-WH 2)))
   (
    (("なくす")("見当はつく "))
    (("なくす")("分かる "))
   )
  )
)

```

《補足》

上記の副文のうち、KAU-S, KONS-S, KONZ-S, TEM-S, KOND-S, FIN-S, MOD-S に必要な情報は定動詞後置だけということになるので、その種類分けは現在、なんら訳出に違いをもたらさない。

例えば "じゃあホテルで何か買い物でもして時間をつぶそうかしら"

"Also, kann ich die Zeit verbringen, indem ich einige Einkäufe im Hotel mache?"

ここで ((!y "indem" !x)(SATZGEFUEGE MOD-S (INTRO (SUBORD-KONJUNKTION 2)) 3)) の MOD-S を KAU-S, KONS-S, KONZ-S, TEM-S, KOND-S, FIN-S, としても同じ訳出が得られる。

C 分割 + 部分翻訳方式

*詳細は [7] を参照のこと。

方針は以下の通り。

<方針 1>

分割翻訳で、N ベストの解析木を得る
各解析木の CONN を含まない各部分木に対して、
従来の部分翻訳の条件（形態素数，意味距離）に基づいて
○×をつける。
生成では○のついた部分木のみ翻訳

<方針 2>

意味距離の条件を満たさない部分構造は、下位構造が条件を満たさないか
逐次的にチェックしていく。
具体的処理手順としては、意味距離の条件を満たさない部分構造は、トップ
ノードの係受けを CONN ルールに置き換え（下位の部分構造への分割，用いら
れる CONN ルールの数は置き換え前のルールの変項の数などによる）、新たに
CONN でつながれたそれぞれの部分構造に関して再帰的に<方針 1>を適用する。

<方針 3>

方針 1、2 に基づいて○×付け、再生成のなされた N ベストの解析木において、
その意味距離が最も小さいものから翻訳を生成する。
意味距離は、CONN のペナルティー値を含む解析木全体の意味距離を用いる。

方針 1、2 に基づいた処理手順を模式的に示す。

```
for 各解析木の CONN を含まない各部分木
  if (形態素数 < 最小形態素数) /* 形態素数が小さいものは捨てる */
    部分木に×をつける；
  else
    if (意味距離 > 最大意味距離) /* 意味距離の大きいものは方針 2 適用 */
      トップノードの係受けを CONN ルールに置き換え；
      分割された各下位部分構造に対して最初の for からの処理にもどる；
    else /* 条件を満たしている */
      部分木に○をつける；
```

現在の設定：部分翻訳（形態素数 2，意味距離 0.5） + （新）平均意味距離

Index

- :all-copy, 8, 12
 - :compound, 8, 13, 14, 22
 - :conj-form, 8, 9, 15
 - :end, 8, 12, 15
 - :pos, 8, 9, 15
 - :pron, 8, 9, 15
 - :reg-exp, 8, 9, 15
 - :start, 8, 12, 15
 - :verb, 9, 10, 30
 - :word, 8, 9, 15
- C
- category-jg.lisp, 24
 - check-example-duplicates, 61
 - compare-results, 60
 - count-sen, 59
 - count-unused-patterns, 61
 - CSTAR, 2
- D
- define-morph, 8, 10, 11, 13
 - define-pattern, 19-21
 - define-regular-pattern, 19, 21
 - define-string, 8, 14
 - dic-tool, 62
- E
- exact match 用例, 43
- F
- find-patterns-with-lex, 58
 - find-same-examples, 61
- G
- get-all-rules, 58
 - gg, 59
 - ggcode, 59
 - ggdic, 59
- H
- hokan-exit, 59
 - hokan-init, 59
- L
- lexical-transformation, 6, 7, 9-14, 45, 49
 - local-transformation, 6, 15-18, 46, 49
- M
- make-result, 60
- R
- replace-word, 6, 7
- S
- search-sentence, 58
 - setup-jg.lisp, 44
- T
- terminal rule, 8, 13, 14, 22
 - terminal rule 用の合成, 13, 14
 - test-TransDic, 59
 - text-definitions-uttr.lisp, 1, 3
 - text-definitions.lisp, 1, 3
 - total-transformation, 6, 18
 - transcode, 59
 - transdic, 59
- い
- 意味距離計算, 4, 20, 43, 53
 - 意味コード, 4, 20
 - (日本語) 意味コード辞書, 4
- か
- 解析処理, 6
 - 解析知識, 6
 - 解析変換知識の作成手順, 43
 - 過剰訓練, 61
 - 活用形, 12, 16
 - カテゴリ, 21, 23-25, 30
 - 下部構造, 23-25, 56
 - 下部構造の制限, 23, 25
 - 空用例, 43
- き
- 機能語, 17, 25-27
 - 基本表現集, 2
- く
- 訓練終了の条件, 43
 - 訓練の単位, 43
- け
- ケース, 28
 - 形態素, 3
 - 形態素情報, 7-9, 15, 44, 48, 54

- 形態素定義式, 8, 10, 11, 13
- 形態素の合成, 9, 10
- 形態素の除去, 11
- 形態素の分割, 11
- 決定木, 29
- 懸案事項, 62
- こ
- コーパス, 1
- 構文解析, 21, 23, 55
- 固定項, 20, 25
- さ
- 差分確認, 60
- し
- システムのデフォルト, 43
- 主語補完モジュール, 10, 29, 43
- 照応, 30
- 省略要素, 29
- 書式に関する注意事項, 1
- 新部分翻訳, 43
- じ
- 辞書ツール, 62
- せ
- 正規形でマッチングするパターン, 21
- 正規形の修正, 12, 13
- 生成処理, 6, 28
- (ドイツ語)生成辞書, 6, 48, 52
- 生成情報, 5, 28
- 生成マーカ, 28
- た
- タギングデータ, 3, 44, 48
- ち
- 重複用例, 60
- つ
- ツール, 43, 58, 59
- て
- テキスト, 1, 3, 60
- と
- 統計データ, 41
- ど
- 動作主, 8-10, 29
- 動作主のタグ, 29
- 動作性の単語, 10, 29
- な
- 内容語, 17, 25
- は
- 発声, 43
- 発声の開始情報, 8, 12, 15
- 発声の終了情報, 8, 11, 12, 15
- ば
- バイリンガル会話, 1
- ぱ
- パターン
- 形態素パターン, 7, 9, 15
- 原言語パターン, 20
- パタンの種類, 21
- 変換パターン, 19-21, 30, 46, 50
- 目的言語パターン, 20, 46, 50
- ひ
- 表層形でマッチングするパターン, 21
- 品詞, 17
- ふ
- 副作用, 59
- ぶ
- 文ケース, 28
- 文分割, 43
- へ
- ヘッド, 20, 25-27
- 変換処理, 19
- (日独)変換辞書, 4, 20, 27, 28, 46, 49, 50
- 変換知識, 19
- 変項, 20, 25
- ほ
- 補完主語, 29
- 翻訳訓練対象テキスト, 1
- ま
- マーカ, 15-18
- マーカの正規形, 21
- も
- 文字列定義式, 8, 13, 14
- よ
- 用例, 4, 19, 20, 46, 50
- exact match 用例, 57, 58
- Nヘッド用例, 25, 26
- 空用例, 20
- 読み情報, 9
- ろ
- ローカル辞書, 27, 28, 47
- わ
- 話者情報, 4
- グ
- グローバル変数, 58