TR-I-0329

日英変換規則解説書

The Japanese-English Transfer Rules for ASURA

鈴木 雅実　　関 倫彦*
Masami SUZUKI　Michihiko SEKI

1993年3月

概要

音声言語翻訳実験システムASURAで、日英の言語変換処理に用いられる規則について、例を含めた説明を行なうとともに、規則体系としての現状を解説する。

ATR自動翻訳電話研究所
ATR Interpretin Telephony Research Laboratories
* 日本アイアール株式会社
JIRCO

はじめに

本解説書は、音声言語翻訳実験システムASURAで、日英の言語変換処理に用いられる規則について、例を含めた説明を行なうとともに、規則体系としての現状を述べることを目的としている。

ATR自動翻訳電話研究所の7年間のプロジェクトの終結に際し、これまでの言語変換規則の開発の経緯をここで簡単に振り返る。ASURAシステムの前身であったSL-TRANSの最初の版が動作し始めたのは1989年のことで、当時は「国際会議の参加に関する問い合わせ」をタスクとしたモデル会話の最初の3ファイル(A, B, 1)程度が処理対象であった。日英の言語変換処理については、言語処理研究室の長谷川敏郎 研究員の手によって、「素性構造書き換えシステム」を処理系として用いる手法が確立しつつあった。また、同研究室の久米 雅子 研究員らの研究により、発話意図のタイプの抽出とその翻訳への利用が提案されており、その他の言語学的知見等も合わせて取り込んだ、言語変換規則の最初の小規模な体系が作成された。

その後、研究員の出向元への復帰・交替等を経て、本解説書の筆者らが日英変換規則の量的・質的な拡大を行ない、現在(1993年3月)に至っている。最終的には、上記のモデル会話(A, B, 1～10)全文(m-set)および、国際会議に関する話題を前提とする機能試験文 約600文(e-set)の大多数を扱える規則体系を作り上げた。この過程で、変換処理系である「素性構造書き換えシステム」自体にも種々の機能拡張や性能向上のための改良を実施した。このため、規則の定義用の関数の仕様も初期とは異なっている他、記述面での柔軟性が増している。これらの詳細は同時発行予定の「言語変換処理系解説書」(ATR Technical Report TR-I-0330)を参照されたい。また、本報告は m-set と e-set を包含する、日英変換規則体系"Grammar10"(1993年3月版)に基づいており、次の位置にファイル群が存在する。

as26:/usr/project/asura/develop/translation/transfer/rules/grammar-eset/

本解説書の内容として随所に述べられている通り、現状の言語変換処理の問題点は多い。その大部分は、変換処理単独の問題というよりも、翻訳の手法全体に関わるものである。文脈情報や領域知識等の外部知識の利用が翻訳結果の質的向上に欠かせないであることは、談話処理に関する考察や小規模な実験を通して検証されつつあるが、本格的な導入への手順はまだ明らかになったとは言えない状態である。さらに、言語体系の相違を吸収し、かつ可能な限り自然な生成文を作り出すために必要な翻訳手法については、言語変換処理部と密接な関連をもつ言語解析・生成処理部も合わせて、今後探求を続ける必要がある。

話し言葉の翻訳処理の達成度という観点では、一通り多くの現象に遭遇して基盤を固めた段階と言えるだろう。ただし、従来から言われている通り、翻訳のための規則化手法は一意に定まるものではなく、何通りものパラダイムが成立可能である。その意味で、本解説書に示されている日英変換規則は、「素性構造書き換えシステム」を利用して、現在得られる日本語解析結果に基づく、英語素性構造の作成例に過ぎない。とはいえ、各項目毎に述べられている規則作成の背景を読み取ることにより、さらなる研究の契機や題材として本書が役立つことを願うものである。なお、規則ファイル中の個々の規則にはコメントや対象文が付与されているので、本解説書の説明の至らない点をある程度補えるものと期待している。

目次

1　はじめに

本解説書では、素性構造書き換えシステムを用いる日英変換規則について、概要を述べる。
ここでいう日英変換規則とは、音声言語翻訳実験システムＡＳＵＲＡ(SL-TRANS)における日本語依存意味構造から英語依存意味構造への言語変換過程で用いられるものである。

本報告では、モデル会話文A,B,1,2,3,4,5,6,7,8,9,10 の２６２文、追加例文１８文、機能試験文６００文、追加例文３８文の計９１８文のうち、解析部を通った文の出力文のうち、正解構造とされるものについて、変換処理を行なうものとして作成したルールについて、書き換え環境ごとに概説する。
また、あわせて現在の問題点、今後の拡張・整備の方向について述べる。

音声言語翻訳実験システムＡＳＵＲＡ(SL-TRANS)日英変換処理部は、以下のような前提のもとに開発が進められた。

- 解析処理部の出力である日本語依存意味構造は、素性の値に日本語の語彙を用いて表現され、変換の出力である英語依存意味構造は、素性の値に英語の語彙を用いて表わされ、生成処理部へ出力される。

- 変換処理は、基本的に、意味構造から意味構造への変換と考えられ、複数解はありえないとして、出力は１構造としている。
  （概念表記の素性が、日本語語彙から英語語彙によるものとなる）

- 解析処理部の出力である日本語依存意味構造は、表層の語彙表記を深層格による格構造で表現したものであるため、変換処理部(JJ変換)部で、日本語構造から日本語構造へと変換する場合がある。

- 変換処理部で書き換えられた英語意味構造の語彙表記は、概念表記であるが、処理の対象としている領域が比較的狭いことから、語の多義性については限られたものとなっており、一部を除いて、英語依存意味構造の英語語彙を用いた概念表記の素性値を、英語単語として生成している。
  （動詞の訳し分けは、格要素により生成処理部で行なう。）
  （名詞の訳し分けは、一部、変換処理部で行なっている。）

- 生成処理部とのインタフェースを考慮し、単一化ＰＤルールによる生成処理のために、変換処理部において適当な規則により英語依存意味構造を出力する。

- 日本語依存意味構造内の語彙表記である日本語関係名は、語義として、解析処理部で決定しているものとする。

- 対象となるタスク領域を、『国際会議の電話による問い合わせ』というコーパス内の設定とする。

- 解析処理で決定された１文単位の日英変換処理の精度向上、および、コーパスに特有な規則の拡張を主とし、文脈処理による談話構造処理は現在、行なっていない。

2　対象例文

対象例文は、以下のセットである。

1. モデル会話	モデル会話文(MSET)	262文	(解析後 258文)
モデル会話追加文	(デモ用)	18文	(解析後 18文)
2. 標準日本語表現	機能試験文 (ESET)	600文	(解析後 579文)
3. 標準日本語表現	追加例文	38文	(解析後 38文)

計９１８文のうち、解析に成功した例文８９３文について、変換処理を行なった。

1. モデル会話文は、『電話による国際会議の問い合わせ』というテーマで、事務局側(secretariat)と問い合わせ側(applicant)との対話形式となっている。対話形式となっているのは、このモデル会話文だけである。

2. 機能試験文は、いろいろなヴァリエーションの短い一般文と、『電話による国際会議の問い合わせ』という設定で、用例を１文単位で、集めたものである。

3. 追加文は、日本語文法項目で、さらに必要と思われるものについて、簡単な一般文で、実験したものである。

3. の追加文には、以下の日本語文法項目が含まれている。

しなくてはいけない／～が（接続助詞）／のみ（副助詞）／しかも／
何ら＋もの（物に関する形式名詞）／上（形式名詞）／者（形式名詞）／
ただし／することがある／類別詞後置形（原稿用紙三枚）／～たり～たり／
に（格助詞；出所）／～ておる（継続）／～しようとしている／～するところだ／
～は～でない／～というのは～だ／～を～にする／～が一番～／
～したことがある／～したものだ／～すべきだ／～するといい／～そうだ／
～ことにする／かしら／～ていただきたい／～ようにお願いします／～なら／
～である／または／つまり／どうも／に（格助詞；資格）／～のようだ／～につき／
～にかけて／うえ（事に関する形式名詞）／はなし（事に関する形式名詞）

現行の日英変換部では、解析出力で分割された文単位での書き換え変換を行なっているため、文脈処理や文の統合などで今後考慮する点がある。

モデル会話文では、通常１文として（連続して）発話される文が、２文に分割されている場合がある。

Ex.　はい、そうです → はい。　そうです。
　　いいえ、まだです → いいえ。　まだです。

また、その発話がどちら側（事務局／問い合わせ側）の発話かを特定する情報は現在与えられていない。

以下の２文セットも、本来は、１文として（連続して）発話される可能性がある。

(db-5) まず登録用紙で手続きをしていただかなくてはなりませんが。
(db-6) もう登録用紙はお持ちでしょうか。

(d1-6) 会議に申し込みたいのですが。
(d1-7) どのような手続きをすればよろしいのでしょうか。

(d2-3) 会議の参加料について教えていただきたいのですが。
(d2-4) 今会議に申し込めば参加料はいくらですか？

(d2-9) わたしは情報処理学会の会員なのですが。
(d2-10) 参加料の割引はないのですか？

(d3-3) 会議に論文を発表したいと思っているのですが。
(d3-4) 会議の内容について教えて下さい。

(d5-14) 案内書にも書いていますが。
(d5-15) 九月二十七日以後の取り消しに対する払い戻しはできません。

(d6-3) 会議の間に市内観光があるそうですが。
(d6-4) まだ参加できますか。

(d7-6) 機械翻訳という話題が案内書に載っていますが。
(d7-7) 具体的にこれはどういう内容のものなんですか。

(d8-4) わたしは今度の会議に発表したいと思っているんですが。
(d8-5) どのような手続きをすればよろしいでしょうか。

(d10-4) 会議の宿泊施設についてお尋ねしたいのですが。
(d10-5) そちらでどこか紹介していただけますか。

このような日本語における１文から２文への分割処理や、逆に２文から１文への統合処理など、発話者の特定、前文情報、文脈考慮などによりシステム中ではどこで行なうかが問題となる。
また、変換処理においても、日本文の１文が、英語文の２文以上に相当する場合や日本文の２文以上が、英語文の１文に相当する場合も考慮する必要がある。

例）
p-288 人工知能研究所の渡辺ですけれどまだ登録用紙は届いていませんか
"This is Watanabe of the Artificial Intelligence Research Center, haven't you receive the registration form yet?"

以下の区別は、前文の保持などの局所的文脈理解により処理できるが、現在の変換ルールでは、取り扱っていない。

・あいづちの「はい」と、承諾・肯定の「はい」の区別。
・あいづちの「そうですか」と、文字どおりの疑問としての「そうですか」の区別。
・「よろしくお願いいたします」と感謝の「ありがとうございます」の区別。
（どちらも変換後は、"Thank you" としている。）

また、機能試験文においては、前文情報などがあってはじめて、文の意味が明確となるであろう例も見受けられた。

3 規則数

書き換え変換規則数は以下の通りである。それぞれの規則は、書き換え環境ごとのルール群として、ファイル別に分けられている。

・MAIN.SCHEMA	(main.schema)	1 rule
・ELLIPSES.SCHEMA	(ellipses.euc)	75 rules
・IFT.SCHEMA	(ift.euc)	69 rules
・PRAG.SCHEMA	(prag.euc)	24 rules
・JAPANESE.SCHEMA	(japanese.euc)	143 rules
・IDIOM.SCHEMA	(idiom.euc)	135 rules
・GENERAL.SCHEMA	(general.euc)	309 rules
・DEFVERB.SCHEMA	(defverb.euc)	520 rules
・DEFNOUN.SCHEMA	(defnoun.euc)	739 rules
・MISC.SCHEMA	(misc.euc)	21 rules
・ASPECT.SCHEMA	(aspect.euc)	27 rules
・TYPE.SCHEMA	(type.euc)	154 rules

total 2216 rules

★ 書き換え環境別による規則数

:PHASE :ELLIPSIS-INIT	1 rule
:MOOD :DECLARATIVE *	30 rules
:MOOD :INTERROGATIVE *	12 rules
:PHASE :ELLIPSIS-RESOLUTION-SEM	
:PHASE :ELLIPSIS-RESOLUTION-PRAG	32 rules
:IF :REDUCE :TYPE :DEFAULT	7 rules
:IF :REDUCE :TYPE :GENERAL	51 rules
:IF :REDUCE :TYPE :INTENTION	10 rules
:PHASE :PRAG-INIT	5 rules
:PHASE :PRAG-MODIFICATION	15 rules
:PHASE :PRAG-DEFAULT-1	1 rule
:PHASE :PRAG-DEFAULT-2	1 rule
:PHASE :PRAG-DEFAULT-3	1 rule
:PHASE :PRAG-DEFAULT-4	1 rule
:PHASE :JAPANESE	104 rules
:PHASE :JAPANESE :EVENT-OR-OBJECT :EVENT*	34 rules
:PHASE :JAPANESE :EVENT-OR-OBJECT :OBJECT*	2 rules
:PHASE :JAPANESE :IFT :REQUEST	2 rules
:PHASE :J-E :TYPE :IDIOM	132 rules
:PHASE :J-E :TYPE :IDIOM :IFT :INFORM	1 rule
:PHASE :J-E :TYPE :IDIOM :IFT :REQUEST	1 rule
:PHASE :J-E :TYPE :IDIOM :IFT :RESPONSE	1 rule
:PHASE :J-E :TYPE :IDIOM :IFT :PHATIC	1 rule
:PHASE :J-E :TYPE :PRE-GENERAL	3 rules
:PHASE :J-E :TYPE :GENERAL	306 rules
:PHASE :J-E	2 rules
:PHASE :J-E :IFT :QUESTION	1 rule
:PHASE :J-E :TYPE :DEFAULT	1257 rules
:PHASE :ASPECT-INIT	3 rules
:PHASE :ASPECT-CHANGE	11 rules
:PHASE :ASPECT	11 rules
:PHASE :ENGLISH	21 rules
:PHASE :INDEX-REMOVED	2 rules
:PHASE :TYPES :TYPE :DEFAULT	154 rules

total 2216 rules

* は、書き換えコールによって呼び出されるローカルな書き換え環境である。

4　書き換え環境の設定

MAIN.SCHEMA （トップレベルの変換規則）では、書き換え環境の設定を行なう前に、日本語意味表現中に、発話タイプの素性を導入する。
ここでは、発話タイプを、デフォルト値としてUNKNOWN-IFTに設定し、日本語意味表現の.SEM素性を命題内容として、無条件にOBJE素性に置き換える。

UNKNOWN-IFT　：　（未定義の発話タイプ）

また、PRAG素性の SPEAKER/HEARER に対応させて、SEM素性中に、発話の話し手(AGEN)、聞き手(RECP)として素性を導入する。

```
set ?SP to input.prag.speaker
set ?HR to input.prag.hearer
   in=     [[sem ?sem]
            ?rest]
   input = [[sem [[reln UNKNOWN-IFT]
                 [agen ?sp]
                 [recp ?hr]
                 [obje ?sem]]]
            ?rest]
```

ここで、導入される素性は、

reln素性(relation)	発話タイプ	
agen素性(agent)	話し手	
recp素性(recipient)	聞き手	
obje素性(object)	命題内容	を表わす。

命題内容である object素性は、この時点では、入力構造のままである。
この日本語解析結果である入力構造は、日本語依存の構造であり、基本的には、格構造で表現される意味構造である。

4.1 書き換え環境設定

変換過程は７つのサブ・プロセスにより構成され、すべての入力素性構造に対して、常に一定の順序で処理される。これは、main.schema内で設定された書き換え呼び出し（rewrite命令）によって制御される。

書き換え環境設定の入力構造指定は、
a). SEM素性を変換する呼び出しルールと、
b). PRAG素性を対象とする呼び出しルール
c). SEM素性とPRAG素性の両方を適用するもの に分類される。

c).のSEM素性とPRAG素性の両方を適用するものについては、一部使用しているが、SEM素性とPRAG素性間にタグで関係付けられている場合などの変換規則の拡張が、今後必要となる。

書き換え環境によるサブ・プロセス化の理由としては、
・規則の競合を少なくして計算量の軽減を図る
・段階的に規則を適用することにより、規則の増加を抑える
・ある部分素性構造に適用可能な規則が、複数存在する場合に、規則をIDIOM,GENERAL,DEFAULT に分類して、規則の適用に順番を与えて、優先順位を持たせる
などがあげられるが、
イディオム変換以降の日英変換は、さらにサブ・プロセス化することによって見通しのよいものとする必要がある。

変換過程の７つの基本的な流れ（サブ・プロセス）は、以下の順である。
1)の省略補完、2)の発話タイプ指定、4)の日本語変換までは、日本語解析の後処理として考えることもできる。

1) 省略補完
```
:PHASE :ELLIPSIS-INIT
:PHASE :ELLIPSIS-RESOLUTION-SEM
:PHASE :ELLIPSIS-RESOLUTION-PRAG
```
↓
2) 発話タイプ変換
```
:IF :REDUCE :TYPE :GENERAL
:IF :REDUCE :TYPE :DEFAULT
```
↓
3) PRAG素性変換
```
:PHASE :ELLIPSIS-RESOLUTION
:PHASE :PRAG-MODIFICATION
```
↓
4) 日本語構造変換
```
:PHASE :JAPANESE :PRSP :INIT
:PHASE :JAPANESE
```
↓
5) 日英変換
```
:PHASE :J-E :TYPE :IDIOM
:PHASE :J-E :TYPE :GENERAL
:PHASE :J-E :TYPE :DEFAULT
```
↓
6) 英英変換処理
```
:PHASE :ENGLISH
```
↓
7) ASPECT処理
```
:PHASE :ASPECT-INIT
:PHASE :ASPECT-CHANGE
:PHASE :ASPECT
```

４.２ 規則適用の制御記述（CONTROL）

規則の適用は、書き換え呼び出し（rewrite命令）の際に記述されるCONTROLにより、素性の辿りかたや規則適用の終了条件を指定することができる。これにより、特定の書き換え環境の規則群の性質に従って、並列に規則適用を行なうか、すべての部分構造にたいして規則適用を行なうか、などを書き換え環境ごとに指定できる。

:LOOP
- 適用できる規則がなくなるまで、繰り返して規則の検索と適用を行なう
- LOOPが指定されない場合、規則の検索と適用は、一度しか行なわれない

:RECURSIVE
- すべての部分素性構造に対して規則の適用を行なう
- RECURSIVEが指定されない場合、素性構造のルートのみに対して、規則の検索と適用を行なう

:ONCE
- 規則の適用が、一度だけ行なわれる。

:LOOP を指定しない場合の検索の順番は、規則の並び順である。
規則適用の優先順位を以下に示す。
- サブプロセスの書き換え呼び出しの順番
- 書き換え環境による制限
- LOOP指定の有無による制限
- 素性PATHによる入力素性構造(FS)の条件指定
- 入力素性構造(FS)パターンのマッチング

N.B. control部が :LOOPの場合、同じ書き換え環境内に適用可能な規則を複数記述してしまうと、on memoryと二次記憶化の処理で結果が異なる。

書き換え呼び出し例（rewrite命令）

main.schemaで指定される書き換え呼び出し（rewrite命令）と
CONTROL部の指定を示す。

```
・MAIN
      (rewrite INPUT with :MAIN)

1)・省略補完 初期化
      (rewrite SEM  with :PHASE :ELLIPSIS-INIT by :RECURSIVE)
  ・SEM省略補完
      (rewrite SEM with :PHASE :ELLIPSIS-RESOLUTION-SEM by :RECURSIVE)
  ・PRAG省略補完
      (rewrite PRAG with :PHASE :ELLIPSIS-RESOLUTION-PRAG by :RECURSIVE)
  ・TYPE 付与
      (rewrite SEM with :PHASE :TYPES  :TYPE :DEFAULT by :RECURSIVE)

2)・発話タイプの指定
      (rewrite SEM with  :IF :REDUCE :TYPE :GENERAL by :LOOP :RECURSIVE)
  ・発話タイプ デフォルト
      (rewrite INPUT with  :IF :REDUCE :TYPE :DEFAULT by :RECURSIVE)
  ・ INTENTION FEATURES 追加
      (rewrite INPUT with  :IF :REDUCE :TYPE :INTENTION by :RECURSIVE)

3)・prag素性の尊敬・丁寧・謙遜など
      (rewrite PRAG with :PHASE :PRAG-INIT by :LOOP :RECURSIVE)
  ・prag素性の縮退
      (rewrite PRAG with :PHASE :PRAG-MODIFICATION by :RECURSIVE)
  ・prag素性のDEFAULT追加
      (rewrite PRAG with :PHASE :PRAG-DEFAULT-1 by :ONCE)
  ・prag素性のDEFAULT追加
      (rewrite PRAG with :PHASE :PRAG-DEFAULT-2 by :ONCE)
  ・prag素性のDEFAULT追加
      (rewrite PRAG with :PHASE :PRAG-DEFAULT-3 by :ONCE)
  ・prag素性のPOLITENESS DEFAULT追加
      (rewrite PRAG with :PHASE :PRAG-DEFAULT-4 by :ONCE)
4)・J-J 日本語変換
      (rewrite SEM with  :PHASE :JAPANESE by :LOOP :RECURSIVE)

5)・J-E変換 (idiom)
      (rewrite SEM with  :PHASE :J-E :TYPE :IDIOM by :LOOP :RECURSIVE)
  ・J-E変換 (general)
      (rewrite SEM with  :PHASE :J-E :TYPE :PRE-GENERAL by :LOOP :RECURSIVE)
  ・J-E変換 (general)
      (rewrite SEM with  :PHASE :J-E :TYPE :GENERAL by :LOOP :RECURSIVE)
  ・J-E変換 (default)
      (rewrite SEM with  :PHASE :J-E :TYPE :DEFAULT-1 by :LOOP :RECURSIVE)
      (rewrite SEM with  :PHASE :J-E :TYPE :DEFAULT-2 by :LOOP :RECURSIVE)
      (rewrite SEM with  :PHASE :J-E :TYPE :DEFAULT-3 by :LOOP :RECURSIVE)

6)・英英変換
      (rewrite SEM with  :PHASE :ENGLISH by :RECURSIVE)

7)・英語アスペクト初期化
      (rewrite SEM with  :PHASE :ASPECT-INIT by :RECURSIVE)
  ・英語アスペクト変更
      (rewrite SEM with  :PHASE :ASPECT-CHANGE by :RECURSIVE)
  ・英語アスペクト指定
      (rewrite SEM with  :PHASE :ASPECT by :RECURSIVE)
```

5 省略補完

日本語対話文に顕著に見られる主語などのゼロ代名詞の補完を行なう。
前身のシステムにおいては、省略補完処理は、解析処理部の後処理として
独立したプログラムによって行なわれていたが、等価な処理を書き換え規則を
用いて実現することにより、規則の保守更新が容易となっている。

ここでの省略補完は、文間にまたがらない「一文内の省略補完」を、日本語素性
構造内に表わされる文末表現による人称制限を主に利用して、行なっている。
概要を以下に示す。

1) 情報のなわばりに関する制約（疑問文では制約が反転する）
   ★「～たい」の主語
      a. 肯定文の場合 (speaker)
      b. 疑問文の場合(hearer)
   ★「～てほしい」の主語と間接目的語
      a. 肯定文の場合：主語(speaker)、間接目的語(hearer)
      b. 疑問文の場合：主語(hearer)、間接目的語(speaker)
   ★状態動詞の現在形の主語 （「持つ」のみ）
      a. 肯定文の場合(speaker)
      b. 疑問文の場合(hearer)
   ★動作動詞の現在形の主語(speaker) （「送る」の肯定文のみ）
2) 敬語表現・受給表現に関する制約
   ★動詞句の尊敬形(respect)では、話し手は、主語を非主語より
    上位に待遇する。
      (主語(hearer))
   ★名詞の尊敬形(respect)では、話し手は、名詞の所有者を
    話し手より上位に待遇する。
      (名詞の所有者(hearer))
   ★動詞句の謙譲形(condescend)では、話し手は、主語を非主語より
    下位に待遇する。
      (主語(speaker))
   ★「てもらう」では、話し手は、間接目的語に対して主語よりの
    視点をとる。
3) 約束表現(commisive)に関する制約
   ★「させてもらう」の主語(speaker)
   ★「させて頂きます」の主語(speaker) ．．．追加パタン
4) 依頼表現(directive)に関する制約
   ★「お願いします」の主語(speaker)と間接目的語(hearer)
   ★「下さい」の主語(hearer)
   ★「して頂く」の主語(speaker)、間接目的語(hearer)
5) 移動行為に関する制約
   ★主語と間接目的語は排他的である（ことが多い）。

省略補完を変換のサブ・プロセスのどのタイミングで扱うかについては、
省略補完の規則と発話タイプの決定の規則が、互いに対応がとれることから
発話タイプの決定後に、補完を行なうことも考えられる。

また省略を与えられる情報に基づいてどこまで補完できるかは、自ずと限界が
ある。現在の解析処理部や変換処理部で省略補完ができない部分があるが、
変換で補完を行なわずに、生成処理部へ送ったとすると、生成ルール（ＰＤ）
では、前後構造などの要素を見ずに、動詞のAGEN格なら SPEAKERを、RECP格なら
HEARER を埋めてしまう。

N.B. 但し、構造内の素性値が空〔〕かどうかを見て、生成処理部でその格を
表層構造に表わさない英文構造で生成することができる場合がある。

従って、変換処理部では、上下の語彙構造などの要素をある程度、考慮して、
多少、ヒューリスティックでも、補完できるようルールを作成している。

また、日英変換などの本来の変換処理内部で、補完する場合もある。

発話参加者以外の補完は、文脈処理が必要なため、現在は行なっていない。

5.1 省略補完初期動作

sem部の入力構造に対して、疑問文か疑問文以外かの２通りに分類し、
それぞれに応じた個々の省略補完規則を呼び出して、適用する。

```
(rws:defrwschema2 ELLIP001 ELL INIT
"on <reln> UNKNOWN-IFT in :PHASE :ELLIPSIS-INIT
     if input =! [[RELN UNKNOWN-IFT]
                 [OBJE [[RELN S-REQUEST]
                        ?rest]]
                 ?rest2]
       then
     => input with  :MOOD :DECLARATIVE   ;*<======(a)
      endif

     if input =? [[RELN UNKNOWN-IFT]
                 [OBJE [[RELN S-REQUEST]
                        ?rest]]
                 ?rest2]
       then
     => input with  :MOOD :INTERROGATIVE ;*<======(b)
      endif

     return input
end")
```

(a) 疑問文以外の場合の書き換え呼び出し
(b) 疑問文の場合の書き換え呼び出し

・「疑問文」と「疑問文以外」という２通りの分類から、さらに細かく分ける。

・呼び出し先のルールにおいて、トップからinputを指定しなくてもいいように
動詞の変数でコールするようにする。

・NEGATE（否定）やPASSIVE（受動態）の標識が存在する場合、素性構造内で
個々の動詞構造のレベルが一段深くなるので、それも考慮に入れたルールを
作成する必要がある。

5.2 平叙文の主語補完

能動態で平叙文の主語省略は、動詞を指定の上、SPEAKERを補完する。
電話会話においては、ほとんど問題なく補完可能と思われる。
Ex. （私は）指定の金額を送ります

但し、RECPに一律に HEARERを補完するのは、電話会話であっても、
文脈次第では、無理な場合が出てくる可能性がある。
Ex. （私は）（事務局に）指定の金額を送ります

```
(rws:defrwschema2 ELLIP044 V RE
"on <obje reln> 送る-1 IN :MOOD :DECLARATIVE
¥"動作動詞の現在形の平叙文の主語 --> 一人称¥"
   in= [[reln UNKNOWN-IFT]
        [agen ?agen]
        [recp ?recp]
        [obje [[reln 送る-1]
               [aspt unrl]
               ?rest]]]
   if input.obje.agen =? []   then
      input.obje.agen = input.agen   endif ;*<======(a) AGEN補完
   if input.obje.recp =? []   then
      input.obje.recp = input.recp   endif ;*<======(b) RECP補完
      RETURN INPUT
   end")
```

>P-255 指定の金額を送ります

```
[[SEM !X28[[RELN  送る-1]
          [AGEN !X87[]]   ;*<-------------------------(a)
          [RECP !X18[]]   ;*<-------------------------(b)
          [OBJE !X32[[PARM !X06[[PARM !X05[]]
                                [RESTR [[RELN  金額-1]
                                        [ENTITY !X05]]]]]
                    [RESTR [[RELN  の-連体修飾]
                            [OBJE !X06]
                            [IDEN !X65[[PARM !X07[]]
                                      [RESTR [[RELN  指定-1]
                                              [ENTITY !X07]]]]]]]]]
          [ASPT  UNRL]]]]

;;;============ Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X6[[LABEL *SPEAKER*]]]
        [RECP !X5[[LABEL *HEARER*]]]
        [OBJE [[RELN SEND-V-1]
               [ASPT UNRL]
               [TENSE FUTURE]
               [AGEN !X6]  ;*<-----------------------------------(a)'
               [RECP !X5]  ;*<-----------------------------------(b)'
               [OBJE [[PARM !X4[[PARM !X2[[PARM !X1[]]
                                          [RESTR [[RELN AMOUNT-N-1]
                                                  [ENTITY !X1]]]]]
                                [RESTR [[RELN SPECIFIED-ADJ-1]
                                        [OBJE !X2]]]]]
                     [RESTR [[RELN MODIFY]
                             [ARG1 [[PARM !X3[]]
                                    [RESTR [[RELN MONEY-N-1]
                                            [ENTITY !X3]]]]]
                             [ARG2 !X4]]]]]
               [SEM-ASPE UNREAL]]]]]
```

5.3 受動態の主語補完

受動態(PASSIVE VOICE)内の能動態主語の省略は、一律に〔LABEL *OTHER*〕を補完する。受動態内の能動態主語の省略理由は、以下のような場合である。

A)能動態主語が不明・簡単に言えない・重要でない
　殺人犯が昨日逮捕された。The murderer was caught yesterday (by ～).
　医者が呼ばれた　　The doctor was sent for (by ～).
B)能動態主語が文脈から明らか
　前発話などから明確に判別される

```
(rws:defrwschema2 ELLIP007 V DI
"on <reln> れる-PASSIVE IN :PHASE :ELLIPSIS-RESOLUTION-SEM
  ¥" れる-PASSIVE 内の ACTION VERB--> agen:*OTHERS*¥"
  in=  ((reln れる-passive)
        (obje ((reln ?action)
               ?rest0))
        ?rest1)
 if input.obje.agen =? () then
   input.obje.agen = ((LABEL *OTHER*)) endif ;*<====== PASSIVEのAGEN補完
      RETURN INPUT
    end")
```

>P-208　今度の会議にはわたしが代表として派遣されます

```
((SEM !X10((RELN れる-PASSIVE)
           (ASPT UNRL)
           (OBJE ((RELN 派遣する-2)
                  (AGEN ()) ;*<-------------------------(x)
                  (OBJE !X2((LABEL *SPEAKER*)))
                  (ROLE ((PARM !X4())
                         (RESTR ((RELN 代表-1)
                                 (ENTITY !X4)))))))
           (LOCT !X8((PARM !X6((PARM !X5())
                               (RESTR ((RELN 会議-1)
                                       (ENTITY !X5)))))
                     (RESTR ((RELN の-連体修飾)
                             (OBJE !X6)
                             (IDEN ((PARM !X7())
                                    (RESTR ((RELN 今度-1)
                                            (ENTITY !X7)))))))))))
;;;=========== Transfer Result ============
((SEM ((RELN INFORM)
       (AGEN !X1((LABEL *SPEAKER*)))
       (RECP !X6((LABEL *HEARER*)))
       (OBJE ((RELN PASSIVE)
              (ASPT UNRL)
              (TENSE FUTURE)
              (OBJE ((RELN SEND-V-1)
                     (AGEN ((LABEL *OTHER*))) ;*<--------------------(x)'
                     (OBJE !X1)
                     (LOCT ((PARM !X4((PARM !X3())
                                      (RESTR ((RELN CONFERENCE-N-1)
                                              (ENTITY !X3)))))
                            (RESTR ((RELN MODIFY)
                                    (ARG1 ((PARM !X2())
                                           (RESTR ((RELN NEXT_TIME-ADV-1)
                                                   (ENTITY !X2)))))
                                    (ARG2 !X4)))))
                     (ROLE ((PARM !X5())
                            (RESTR ((RELN REPRESENTATIVE-N-1)
                                    (ENTITY !X5)))))))
              (SEM-ASPE UNREAL)))))
```

5.4 従属節内の主語などの省略補完

従属節内の主語は、以下のように分類されるが、決定は難しい。

a:主節の主語と一致する
b:発話者と一致する
c:文脈情報で決定できる
d:文脈情報で決定できない
e:I／ｙｏｕどちらでもよい（通念）

N.B. 現在の時点では、一部をヒューリスティックに補完している。

Ex. 従属節(TLOC)内の主語省略は、主節の主語を補完する。
但し、下の規則では、補文標識を指定することにより、特定化している。

```
(rws:defrwschema2 ELLIP012 V DI
"on <tloc restr reln> 際-1 IN :PHASE :ELLIPSIS-RESOLUTION-SEM
   in= [[TLOC [[PARM !X[]]
               [RESTR [[RELN 際-1]
                       [iden ?iden]
                       [ENTITY !X]]]]]
        ?rest]
 if input.tloc.restr.iden.agen =? [] then
   input.tloc.restr.iden.agen = input.agen endif :*<======主節のAGEN補完
      RETURN INPUT
   end")
```

>P-142　学会に出席される際お立ち寄りください

```
[[SEM !X09[[RELN  下さい-REQUEST]
           [AGEN !X10[[LABEL  *SPEAKER*]]]
           [RECP !X03[[LABEL  *HEARER*]]]
           [ASPT  UNRL]
           [OBJE !X21[[RELN  立ち寄る-1]
                      [AGEN !X03] ;*<=============主節のAGEN
                      [TLOC [[PARM !X02[]]
                             [RESTR [[RELN  際-1]
                                     [IDEN !X22[[RELN  出席する-1]
                                                [AGEN !X18[]] ;*<-----従属節のAGEN(x)
                                                [LOCT !X51[[PARM !X01[]]
                                                           [RESTR [[RELN  学会-1]
                                                                   [ENTITY !X01]]]]]
                                                [ASPT  UNRL]]]
                                     [ENTITY !X02]]]]]
                      [LOCT !X04[]]]]]]]
;;;============ Transfer Result ============
 [[SEM [[RELN REQUEST]
        [AGEN !X3[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [OBJE [[RELN STOP_BY-V-1]
               [ASPT UNRL]
               [TENSE FUTURE]
               [AGEN !X2]  ;*<=============主節のAGEN
               [TLOC [[RELN WHEN-CONJ-1]
                      [OBJE [[RELN ATTEND-V-1]
                             [ASPT UNRL]
                             [TENSE FUTURE]
                             [AGEN !X2]  ;*<-----------------------(x)'
                             [LOCT [[PARM !X1[]]
                                    [RESTR [[RELN CONFERENCE-N-1]
                                            [ENTITY !X1]]]]]
                             [SEM-ASPE UNREAL]]]]]
               [LOCT []]
               [SEM-ASPE UNREAL]]]
        [MANNER DIRECT]
        [ATTD DECLARATIVE]]]]
```

5.5 その他の省略補完

その他の省略については、指定した補助動詞などの特徴に応じて、補完を行なう。

Ex.『ましょう-OFFER』の空の主語格（AGEN）は、発話者を補完する。

```
(rws:defrwschema2 ELLIP020 V TE
"on <reln> ましょう-OFFER in  :PHASE :ELLIPSIS-RESOLUTION-SEM
   in= [[RELN ましょう-OFFER]
        [obje [[reln ?action]
               [agen ?agen]
               ?rest1]]
        ?rest2]

   if ?agen =? [] then
      ?agen = root.prag.speaker   endif  ;*<======発話者SPEAKERを補完

      out= [[RELN ましょう-OFFER]
            [obje [[reln ?action]
                   [agen ?agen]
                   ?rest1]]
            ?rest2]
 end")
```

>P-175 結果は書面で確認しましょう

```
[[SEM !X12[[RELN  ましょう-OFFER]
          [ASPT  UNRL]
          [OBJE !X05[[RELN  確認する-1]
                     [AGEN !X52[]] ;*<-----------------(x)
                     [INST !X72[[PARM !X02[]]
                                [RESTR [[RELN  書面-1]
                                        [ENTITY !X02]]]]]
                     [OBJE !X06[[PARM !X03[]]
                                [RESTR [[RELN  結果-1]
                                        [ENTITY !X03]]]]]]]]]

;;;=========== Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X3[[LABEL *SPEAKER*]]] ;*<======発話者SPEAKER
        [RECP !X4[[LABEL *HEARER*]]]
        [OBJE [[RELN INVITED-LETS]
               [ASPT UNRL]
               [TENSE FUTURE]
               [OBJE [[RELN CONFIRM-V-1]
                      [AGEN !X3]  ;*<-----------------(x)'
                      [OBJE [[PARM !X2[]]
                             [RESTR [[RELN RESULT-N-1]
                                     [ENTITY !X2]]]]]
                      [INST [[PARM !X1[]]
                             [RESTR [[RELN ON_PAPER-ADV-1]
                                     [ENTITY !X1]]]]]]
               [SEM-ASPE UNREAL]]]]]
  [PRAG [[BENEFIT NONE]
         [HEARER !X4]
         [INTENTION INFORM]
         [POLITENESS [[DEGREE 1]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X3]
         [TOPIC [[TOPIC-MOD WA]]]]]]
```

『～して下さい』などの要求の場合、AGENに対話者、RECPに発話者を補完する。

```
(rws:defrwschema2 ELLIP022 V DI
"on <reln> 下さい-REQUEST IN :PHASE :ELLIPSIS-RESOLUTION-SEM
   in= [[reln 下さい-REQUEST]
        [agen ?agen]
        [recp ?recp]
        [obje ?obje]
        ?rest]
if input.obje.agen =? [] then
   input.obje.agen = root.prag.hearer  endif  ;*<====(a)対話者を補完
if input.obje.recp =? [] then
   input.obje.recp = root.prag.speaker  endif ;*<====(b)発話者を補完
   RETURN INPUT
end")
```

>95 会議に出席できるかどうか至急お知らせください

```
[[SEM !X121[[RELN  下さい-REQUEST]
           [AGEN !X04[[LABEL  *SPEAKER*]]]
           [RECP !X03[[LABEL  *HEARER*]]]
           [OBJE !X48[[RELN  知らせる-2]
                      [AGEN !X03]  ;*<------------------(a)解析で補完済み
                      [RECP !X45[]] ;*<-----------------(b)
                      [MANN [[PARM !X46[]]
                             [RESTR [[RELN  至急-1]
                                     [ENTITY !X46]]]]]
                      [OBJE !X44[[RELN  かどうか-INFORMIF]
                                 [OBJE !X25[[RELN  できる-POSSIBLE]
                                            [EXPR !X21[]] ;*<=====従属節補完
                                            [ASPT  UNRL]
                                            [OBJE !X84[[RELN  出席する-1]
                                                       [AGEN !X21] ;*<=====従属節補完
                                                       [LOCT !X18[[PARM !X13[]]
                                                                  [RESTR [[RELN  会議-1]
                                                                          [ENTITY !X13]]
                                                                   ]]]]]]]]]]]
           [ASPT  UNRL]]]]
;;;=========== Transfer Result ============
 [[SEM [[RELN REQUEST]
        [AGEN !X4[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [OBJE [[RELN INFORM-V-1]
               [ASPT UNRL]
               [TENSE FUTURE]
               [AGEN !X2]
               [RECP !X4] ;*<----------------------(B)対話者HEARERを補完
               [OBJE [[RELN WHETHER-CONJ-1]
                      [OBJE [[RELN POSSIBILITY]
                             [ASPT STAT]
                             [TENSE PRESENT]
                             [EXPR !X2] ;*<============従属節補完 (heuristicに補完)
                             [OBJE [[RELN ATTEND-V-1]
                                    [AGEN !X2];*<======従属節補完 (heuristicに補完)
                                    [LOCT [[PARM !X3[]]
                                           [RESTR [[RELN CONFERENCE-N-1]
                                                   [ENTITY !X3]]]]]]]
                             [SEM-ASPE STATIVE]]]]]
               [MANN [[PARM !X1[]]
                      [RESTR [[RELN IMMEDIATELY-ADV-1]
                              [ENTITY !X1]]]]]
               [SEM-ASPE UNREAL]]]
        [MANNER DIRECT]
        [ATTD DECLARATIVE]]]]
```

5.6 PRAGの省略補完

SEM部で意味構造の面から省略補完を行なった後で、PRAG部の補完に入るが、
PRAG部では、値がタグで示される AGENとRECPのセットに注目して補完する。

例えば、話し言葉では尊敬表現は常に発話者から対話者に向かうものとし、
politeの関係にある AGEN,RECPには、SPEAKER,HEARERを補完する。

```
(rws:defrwschema2 ELLIP070 V PO
"on <reln> POLITE  IN :PHASE :ELLIPSIS-RESOLUTION-PRAG
   in= [[reln POLITE]
        [AGEN ?agen]
        [RECP ?recp]
        ?rest]
   if ?agen =? [] then
     ?agen = root.prag.speaker ;*<====== 発話者 SPEAKERを補完
   endif
   if ?recp =? [] then
     ?recp = root.prag.hearer  ;*<====== 対話者 HEARERを補完
   endif
   RETURN INPUT
end")
```

但し、AGEN-RECP-METHOD と云われる 関係名を参照せずに、AGEN,RECPの
どちらかが、SPEAKERかHEARERで埋まっている場合に
　　AGENとRECP は一致しない、という前提で、
片方がSPEAKERで片方が[]なら、[]にHEARERを埋める(vice versa)
　　というルールは、
AGEN とRECP が一致する場合がありうる、
AGEN がSPEAKERであるからといって、[]のRECP格にHEARERを一律に埋めるのは
無理がある、という理由からはずしてある。

```
;;; ellipsis-resolution-by-agen-recp-method

(rws:defrwschema2 ellaaa V RE
"on <agen label> *speaker*  IN :PHASE :ELLIPSIS-RESOLUTION-PRAG
¥"agent が speaker ならば、recipient は hearer¥"
   in= [[agen [[label *speaker*]]]
        [recp []]
        ?rest]
   input.recp = root.prag.hearer
   RETURN INPUT
end")

(rws:defrwschema2 'b V RE
"on <agen label> *hearer*  IN :PHASE :ELLIPSIS-RESOLUTION-PRAG
¥"agent が hearer ならば、recipient は speaker¥"
   in= [[agen [[label *hearer*]]]
        [recp []]
        ?rest]
   input.recp = root.prag.speaker
   RETURN INPUT
end")

(rws:defrwschema2 'C  V RE
"on <recp label> *speaker*  IN :PHASE :ELLIPSIS-RESOLUTION-PRAG
¥"recipient が speaker ならば、agent は hearer¥"
   in= [[agen []]
        [recp [[label *speaker*]]]
        ?rest]
   input.agen = root.prag.hearer
   RETURN INPUT
end")
```

6 タイプシステム

define.lisp （タイプシソーラスの定義ファイル）
type.schema （名詞の解析構造に対してタイプを付与する規則）

タイプは、日本語構造中の関係名（主に名詞など）に、いわゆる意味素性を与えて、有効な変換処理を行なう目的で、実験的に一部が導入されている。解析処理部が使用する意味素性semfとは、独立した意味素性である。

また、変換終了時点で、プログラムにより削除され、生成処理部には送られない。現在、名詞の日本語素性構造のみに付与している。

１つの日本語関係名に、複数のタイプを付与しようとする場合、選言が記述できる。タイプ付与は、対象領域コーパスと語義を考慮して、限定して行なわなければ、変換規則内で利用しようとする条件にならない。

Ex. 『妻-1』 :human-beings :status

タイプは明示して変換しない限り、日本語関係名に付与されたタイプは、その日本語関係名が変換されると同時に、削除される。

N.B. 変換終了部で、自動的に削除される前にタイプをINDEX素性に転換し、生成処理部で使用することも考えられる。

タイプ付与のサブ・プロセスは、日本語変換（JJ変換）終了後に、設定したほうがよいと思われる。
（現在は、発話タイプ指定のプロセスの前で行なっている。）

Ex. 『医者-1』に、:human-beings のタイプを付与するルール。

```
(rws:defrwschema2 TYPE004 N ttt
"on <restr reln> 医者-1 in :PHASE :TYPES :TYPE :default
   in= [[PARM !x[]]
        [RESTR [[RELN 医者-1]
                [ENTITY !x]]]
        ?rest]
   out= [[PARM !x[]]
         [RESTR [[RELN :human-beings 医者-1]
                 [ENTITY !x]]]
         ?rest]
end")
```

シソーラス体系について

次ページにあげたタイプシソーラスの定義ファイルは、やや普遍的過ぎ、変換規則内で条件分岐として参照するには、分類として扱いにくい。
従って、『電話による国際会議の問い合わせ』のような特定の領域に合わせた、小回りの効くシソーラス体系に縮小する必要がある。

タイプシソーラス（木構造version） ver.1

1|もの|things
　11|具体物|concrete-things
　　111|生物|living-things
　　1111|人間|human-beings
　　1112|動物|animals
　　1113|植物|plants
　　112|生物の部分|organs
　　1121|動物の部分|animal-parts
　　1122|植物の部分|plant-parts
　　113|自然物|natural-things
　　114|人工物|artificial-things
　　1141|材料|materials
　　1142|組織性|groups
　　1143|生産物|products
　　11431|飲食物|food-drink
　　11432|乗り物|transportations
　　115|組織|organizations
　　1151|地縁組織など|gemeinschafts
　　1152|組織|gesellschafts
　　1153|その他の組織|other-organizations
　12|場所|places
　　121|部分位置|inside-positions
　　122|相対位置|relative-positions
　　123|向き|directions
　　124|機能的場所|functional-places
　　125|形態,まとまりから見た場所|regions
　　126|その他の場所|abstract-places
　13|抽象物|abstract-things
　　131|知的抽象物|mental-productions
　　1311|書き物|writings
　　1312|絵|pictures
　　1313|音による知的産物|sound-productions
　　1314|映像|images
　　1315|言語|language
　　1316|思考内容|ideas
　　13161|実質的中身|substance
　　1317|表現|expressions
　　13171|タイトル|titles
　　1318|慣用句,故事成句,ことわざ|idioms
　　132|その他の抽象物|other-abstracts
　　1321|力|power
　　1322|法則|laws
　　1323|学問|studies
　　1324|習俗,宗教|customs
　　1325|構造|structures
　　1326|経過,歴史|history
　　1327|分野別|fields
　　1328|属性|attributes
　　13281|金銭的価値|monetary-attributes
　　13282|時間|time-attributes
　　1329|総括概念|notional-nouns
　14|その他|others
　　141|その他|others
　　1411|風・息|breath
　　1412|音|sound
　　1413|伝達手段|communications
　　1414|心象物|imagery
　　1415|光,光の種類|light
　15|技術|technology

2|こと|affairs
201|現象|phenomena
2011|自然現象|natural-phenomena
2012|物理現象|physical-phenomena
2013|生理現象|physiological-phenomena
2014|心理現象|psychological-phenomena
2015|社会現象|social-phenomena
202|移動|movement
2021|物理的移動|physical-movement
2022|所有的移動|giving-receiving
2023|情報移動|transmission
203|動作・活動|actions
2031|身体的活動|physical-actions
20311|娯楽|recreations
2032|感情活動|emotions
2033|思考的活動|thoughts
2034|具体物に対する活動|handling
2035|抽象物に対する活動|affecting
2036|人同士の相互行為|human-relations
2037|対人行為|treatment
2038|イベント|events
204|変化・変動|changes
2041|発生・出現|appearance
2042|消滅|disappearance
2043|性質の変化|nature-changes
2044|形の変化|form-changes
2045|量の変化|quantity-changes
2046|経過|process
2047|変動|shifts
2048|光の変化|luminosity-changes
2049|音の変化|sound-changes
205|性状・性向|value
2051|空間の属性値|space-specification
2052|時間の属性値|time-specification
2053|具体物のエネルギー的属性値|energies
2054|具体物の質的属性値|conditions
2055|数量の程度や値|quantities
2056|具体物に対する感覚的属性値|senses
2057|気象の値|weather
2058|生物の生理的属性値|biological-value
2059|人の姿,格好の形容|figures
205a|人の気持ちの様子|mental-state
205b|人の内面的属性値|personalities
205c|人の社会的属性値|status
205d|様態・状況の形容|modes
205e|物事に対する評価値|evaluation
205f|関係|comparison
205g|程度の形容|degree
205h|シーケンス|sequence
205i|名称|names
205i1|人名|person's-names
205i2|イベント名|conference-names
205i3|地名|place-names
205i4|組織名|organization-names
205i5|施設名|facilities-names
206|その他のこと概念|other-affairs
3|文|sentences

7　発話タイプなどの指定

発話意図の抽出は、IFT.SCHEMA（発話タイプ規則群）で行なわれる。
ここでは、日本語素性構造内の文末表現などから、発話意図を決定し、
MAIN.SCHEMAで、設定されたSEM部のトップにある未決定の発話タイプ
UNKNOWN-IFT を、適当な発話意図として置き換える。

変換過程における発話タイプ(Illocutinary Force Type)の分類

日本語の部分意味表現（文末構造・副詞意味構造）により、入力発話を分類する

- PHATIC　（挨拶などで用いられるイディオム的な発話）
- EXPRESSIVE　（話者の感情表現に関するイディオム的な発話）
- PROMISE　（約束を表わす発話）
  （構造）使役表現（～させる）＋受給表現（～て頂く，～てもらう）
- QUESTIONIF　（YES-NO疑問文に対応する発話）
- QUESTIONREF　（WH疑問文に対応する発話）
- QUESTIONCONF（確認を表わす疑問文〔～ね〕に対応する発話）
  （構造）～ね
- REQUEST　（話者が何らかの要求を表わす発話）
  （構造）a.要求表現（～たい，～下さい，～願う）
  b.受給表現（～て頂く，～てもらう）＋願望表現（～たい）
  c.話者の行為を表す動詞＋「できる」＋疑問表現）
- RESPONSE　（質問に対する応答や相づちなどの発話）
- INFORM　（上記以外の発話:デフォルトの発話タイプ）

（話者の願望を表す発話は間接的な要求である場合も多いがここに含める）

・『ましょう-OFFER』などIFTタイプ９種類に分類できないものがある。
（現在は、デフォルト値としてINFORMである）

・発話タイプ（IFT）により、生成で出力される英文構造がほぼ決定してしまう。
(REQUEST vs QUESTIONREF)

・『～したいと思う』の場合、『～と思う』を縮退して INFORMとしている。

・IFT.SCHEMA で、生成のために、COND格、CONJ格などを発話意図レベルに、
引き上げる処理を行なっている。

・IFTの後のINTENTION決定も同様に日本語の表層意味表現から導かれる。
（INTENTIONの拡張・指定方法などは今後の課題といえる）
INTENTIONについては、現在、PRAG素性に単に付与しているが、今後
IFT-TYPEを補完する形で、拡張する必要がある。
（生成でPRAG部を参照する:environmentの使用法も考慮に入れる）

重文・複文のように命題内容が複数存在する場合でも、
主文に対してのみの発話タイプしか与えられない

・今後の課題
文脈、対話処理の応用に合わせて、IFTタイプが変更されることがある。

『はい』などのIFTが、承諾、あいづち、確認などの場合。
前発話者の発話が、疑問文(QUESTIONIF,QUESTIONREF)である時、
『はい』は RESPONSEとなる。

『よろしくお願いします。』など、電話会話でも使われるイディオム的
慣用句の集中処理。

トップレベルに一番近い、動詞句(主節の動詞)の検索。
現在は、入力素性構造（in=）を、トップレベルのUNKNOWN-IFTから、
記述して適用を行なっているが、否定・可能・受動態などの挿入で、
参照しようとする述語構造などの素性構造レベルが一段下がった場合に
ついても処理できるようにする。

Ex. 発話タイプ PROMISE の例

```
        (rws:defrwschema2 IFT014 I PR
        "on <obje reln> あげる-GIVE_FAVOR in :IF :REDUCE :Type :General
           in= [[reln UNKNOWN-IFT] ;*<====== 未定義発話タイプ
                [agen ?sp]
                [recp ?hr]
                [obje [[reln あげる-GIVE_FAVOR]
                       [agen [[LABEL *SPEAKER*]]]
                       [recp [[LABEL *HEARER*]]]
                       [obje ?obje]
                       ?rest2]] ?rest]
                              set ?obje2 to ?obje
                              add ?rest2 to ?obje2
             out= [[reln PROMISE]    ;*<====== PROMISE 発話タイプ
                [agen ?sp]
                [recp ?hr]
                [obje ?obje2] ?rest]
        end")

            >P-432 登録用紙を送ってあげる
[[SEM !X12[[RELN あげる-GIVE_FAVOR]
          [AGEN !X04[]]
          [RECP !X05[]]
          [OBJE !X53[[RELN  送る-1]
                    [AGEN !X04]
                    [RECP !X03[]]
                    [OBJE !X50[[PARM !X02[]]
                              [RESTR [[RELN  登録用紙-1]
                                      [ENTITY !X02]]]]]]]
          [ASPT  UNRL]]]
 [PRAG !X09[[RESTR [[IN !X17[[FIRST [[RELN  EMPATHY-DEGREE]
                                     [MORE !X04]
                                     [LESS !X05]]]
                            [REST !X06[]]]]
                   [OUT !X06]]]
           [SPEAKER !X15[[LABEL  *SPEAKER*]]]
           [HEARER !X16[[LABEL  *HEARER*]]]
           [ASPE [[IN !X20[]]
                 [OUT !X21[]]]]
           [PRSP-TERMS [[IN !X19[]]
                       [OUT !X31[]]]]
           [TOPIC [[IN !X18[]]
                  [OUT !X29[]]]]]]]
;;;=========== Transfer Result ============
 [[SEM [[RELN PROMISE] ;*<----------------------送ってあげるという約束を示す
        [AGEN !X3[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [OBJE [[RELN SEND-V-1]
               [ASPT UNRL]
               [TENSE FUTURE]
               [AGEN !X3]
               [RECP !X2]
               [OBJE [[PARM !X1[]]
                      [RESTR [[RELN REGISTRATION_FORM-N-1]
                              [ENTITY !X1]]]]]
               [SEM-ASPE UNREAL]]]]
  [PRAG [[BENEFIT HEARER-SIDE] ;*<========= BENEFIT
         [HEARER !X2]
         [INTENTION OFFER]     ;*<========= INTENTION
         [POLITENESS [[DEGREE 1]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X3]
         [TOPIC [[TOPIC-MOD NULL]]]]]]
```

8 PRAG部の処理

8.1 敬語表現の標準化

PRAG.SCHEMAでは、入力構造のPRAG部に対して、処理を行なう。
PRAG部には、敬語表現や取り立て助詞などの情報が、解析部で付与されるが構造上の問題や正しく付与されていないなど、変換・生成部でアクティブに使うには、まだ固定されていない場合がある。
従って、変換部では、変換結果の見やすさ・簡潔性などを考慮し、現在は敬語表現は、一律化して、数値に置き換えておくことにし、また生成処理から見て、不必要な素性は、縮退・削除している。

PRAG内の尊敬・丁寧・謙遜は、それぞれ、RESPECT,POLITE,CONDESCENDなどで示されるが、これらを POLITENESS という素性に置き換え、一文中にトータルで、上記の敬語表現が何回使用されたかを計算する。（敬語表現の標準化）

```
RESPECT     -> POLITENESS
POLITE      -> POLITENESS
CONDESCEND -> POLITENESS
EMPATHY-DEGREE -> POLITENESS
```

N.B. 敬語表現が多く使われるほど、値は上がるが、
敬語の度合いが、必ずしもそれに比例するわけではない。

Ex. まず、PRAGの初期動作として、POLITE（丁寧）を、素性POLITENESSに転換する。（他の敬語表現も同様）

```
(rws:defrwschema2 PRAG002 N PO
"on <reln> POLITE IN :PHASE :PRAG-INIT
¥"for modification of pragmatic features.¥"
   in=  [[RELN POLITE]
         [AGEN ?agen]
         ?rest]
   out= [[RELN POLITENESS]]
END")
```

Ex. 次に、素性POLITENESS を、パターンマッチングで数値化する。
（この規則は、POLITENESSが２つの場合）

```
(rws:defrwschema2 PRAG006 N PO
"on <restr in first reln> politeness IN :PHASE :PRAG-MODIFICATION
   in= [[RESTR [[IN [[FIRST [[RELN POLITENESS]]]
                    [REST [[FIRST [[RELN POLITENESS]]]
                          [REST []]]]]]]
               [OUT ?OUT]]]
         ?REST]
   out= [[POLITENESS [[DEGREE 2]]]
         ?REST]
END")
```

Ex. 素性POLITENESS を持たない構造には、数値０を代入する。

```
(rws:defrwschema2 PRAG023 N X
"on <speaker> :unspecified IN :PHASE :PRAG-DEFAULT-4
     if input =! [[SPEAKER ?agen]
                  [HEARER ?recp]
                  [POLITENESS ?politeness]
                  ?rest]
       then
          set ?out to [[POLITENESS [[DEGREE 0]]]]
          add ?input to ?out
       out= ?out
     endif
END")
```

8.2 不要な構造の縮退・削除

・空である TOPIC、PRSP-TERMS、TOPIC、PRSP、ASPE、RESTR の削除

```
Ex. (rws:defrwschema2 PRAG015 N X
     "on <topic in> :unspecified IN :PHASE :PRAG-MODIFICATION
        in= [[TOPIC [[IN []]
                     [OUT []]]]
             ?REST]
        out= [?REST]
     END")
```

・TOPIC-MOD やPRSP-MODなどは、値のみを残して、縮退しているが、(a)
元々、TOPIC-MOD やPRSP-MODなどの構造自体がないものについても
別の書き換え環境を設定し、以下のルール (b)で DEFAULT値を与える。

(DEFAULT値を与えないと、生成PDで、単一化してしまうため)

Ex. (a) PRSP-TERMS の縮退

```
(rws:defrwschema2 PRAG013 N X
"on <prsp-terms in> :unspecified IN :PHASE :PRAG-MODIFICATION
   in= [[PRSP-TERMS [[IN [[first [[FOCUS ?focus]
                                  [SCOPE ?scope]
                                  [PRSP-MOD ?mod]]]
                          [rest []]]]
                    [out []]]]
        ?REST]
   out= [[PRSP-TERMS [[PRSP-MOD ?mod]]]
        ?REST]
END")
```

Ex. (b) PRSP-TERMS を持たない構造へのDEFAULT値の代入

```
(rws:defrwschema2 PRAG024 N X
"on <speaker> :unspecified IN :PHASE :PRAG-DEFAULT-3
     if input =! [[SPEAKER ?agen]
                  [HEARER ?recp]
                  [PRSP-TERMS [[PRSP-MOD ?mod]
                               ?rest0]]
                  ?rest]
        then
           set ?out to [[PRSP-TERMS [[PRSP-MOD NULL]]]]
           add ?input to ?out
           out= ?out
           endif
END")
```

Ex. (b)' TOPIC-MOD を持たない構造へのDEFAULT値の代入

```
(rws:defrwschema2 PRAG021 N X
"on <speaker> :unspecified IN :PHASE :PRAG-DEFAULT-1
     if input =! [[SPEAKER ?agen]
                  [HEARER ?recp]
                  [TOPIC [[TOPIC-MOD ?mod]
                          ?rest0]]
                  ?rest]
        then
           set ?out to [[TOPIC [[TOPIC-MOD NULL]]]]
           add ?input to ?out
        out= ?out
          endif
END")
```

Ex.　>P-9 あなたも会議に出席されますか

```
((SEM !X122((RELN  S-REQUEST)
            (AGEN !X04((LABEL  *SPEAKER*)))
            (RECP !X03((LABEL  *HEARER*)))
            (OBJE ((RELN  INFORMIF)
                   (AGEN !X03)
                   (RECP !X04)
                   (OBJE !X05((RELN  出席する-1)
                              (AGEN !X03)
                              (LOCT !X78((PARM !X02())
                                         (RESTR ((RELN  会議-1)
                                                 (ENTITY !X02)))))
                              (ASPT  UNRL)))))))
 (PRAG ((RESTR ((IN !X12((FIRST ((RELN  RESPECT) ;*<--------------(尊敬)
                                 (AGEN !X04)
                                 (RECP !X03)))
                          (REST !X28((FIRST ((RELN  POLITE) ;*<----(丁寧)
                                             (AGEN !X04)
                                             (RECP !X03)))
                                    (REST !X06())))))
                (OUT !X06)))
        (SPEAKER !X04)
        (HEARER !X03)
        (ASPE ((IN !X15())
               (OUT !X16())))
        (PRSP-TERMS ((IN !X14((FIRST ((SCOPE !X05)
                                      (PRSP-MOD  MO) ;*<------(取り立て助詞)
                                      (FOCUS !X03)))
                              (REST !X33())))
                     (OUT !X124())))
        (TOPIC ((IN !X13())   ;*<----------------------------(TOPIC-MOD)
                (OUT !X123()))))))

;;;============ Transfer Result ============
 ((SEM ((RELN QUESTIONIF)
        (AGEN !X3((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN ATTEND-V-1)
               (ASPT UNRL)
               (TENSE FUTURE)
               (AGEN !X2)
               (LOCT ((PARM !X1())
                      (RESTR ((RELN CONFERENCE-N-1)
                              (ENTITY !X1)))))
               (SEM-ASPE UNREAL)))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X2)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 2)))    ;*<------------(敬語表現の数値化)
         (PRSP-TERMS ((PRSP-MOD MO))) ;*<------------(取り立て助詞の縮退)
         (SPEAKER !X3)
         (TOPIC ((TOPIC-MOD NULL))))))) ;*<-----------(TOPIC-MOD DEFAULT値の代入)
```

生成結果
P-9 あなたも会議に出席されますか
"Will you attend the conference, too?"

９　日本語変換

変換処理部の入力構造である解析結果は、表層的な構造となっているため、
変換処理部では、日英変換に入る前に、日本語素性構造の変換を行なう。

日本語構造から日本語構造への変換（JJ変換）は、変換が解析処理部より
受け取る日本語意味構造をさらに中間的な（抽象的な）意味構造に
書き換えようとするものである。
日本語変換には、次のような問題点がある。
・書き換えられた意味構造が、英語意味構造に偏り過ぎる
・JJ変換ではなく、IDIOM変換として、いきなり英語意味構造に
書き換えることもできる。
（その区別の基準を明確にする必要がある）

９.１ サ変名詞からサ変動詞への変換

動詞『する』『行なう』『至る』などは、サ変名詞を目的語とする場合、
サ変動詞に変換することができる。
勉強（サ変名詞） を する → 勉強する（サ変動詞）
このＪＪ変換をしておくことで、後にかかる各種の変換規則を統一でき、
また、語義的に等しい構造を、形式的にも等しく表すことができる。

サ変名詞からサ変動詞への変換のパターンは、NPをサ変名詞とすると
以下のような場合である。

```
NPを希望する -> NPする(WANT-DESIRE)
NPが要る     -> NPする(NECESSITY)
NPの方法     -> NPする方法
NPをお願いする -> NPして下さい(REQUEST)
XXのNPをお願いする -> XXをNPして下さい(REQUEST)
NPを行なう    -> NPする
NPをする     -> NPする
NPに至る     -> NPする
NPができる   -> NPする(POSSIBILITY)
```

(A) NPをする → NPする の場合

まず、動詞『する-1』に注目し、OBJE素性に対して、書き換え環境
:PHASE :Japanese :Event-or-Object :Event を指定して、書き換え
呼び出しを行なう。

```
(rws:defrwschema2 JAP008 V SU
"on <reln> する-1 in :PHASE :Japanese
   ¥"〔サ変名詞〕 + する --> サ変動詞¥"
   in= {[reln する-1]
        [OBJE ?OBJE]
        ?rest]

   -> ?OBJE with :PHASE :Japanese :Event-or-Object :Event
                                   ;*^===== 書き換え呼び出しの書き換え環境
   if it is false then fail endif
   if it is true  then
      add ?rest to ?OBJE
      out= ?OBJE
   else
      Fail
   endif
end")
```

(B) 書き換え環境 :PHASE :Japanese :event-or-object :event を持つ一群のルールは、サ変名詞からサ変動詞への変換規則となっている。そこで、適用できるルールがある場合は、そのルールを通ったのち、元のルールに戻る。

```
(rws:defrwschema2 JAP037 N MO
"on <restr reln> 電話-1 in :PHASE :Japanese :event-or-object :event
   in= [[parm !x[]]                    ;*^===== 書き換え呼び出しの書き換え環境
        [restr [[reln 電話-1]
                [entity !x]]]]
   out= [[reln 電話する-1]
         [agen []]
         [recp []]]
   end")
```

>P-483 後ほど電話をいたします

```
[[SEM !X26[[RELN  する-1] ;*<------------------------ 書き換え呼び出しの条件となる
          [AGEN !X04[]]                                          (A)
          [TLOC [[PARM !X01[]]
                 [RESTR [[RELN  後ほど-1]
                         [ENTITY !X01]]]]]
          [OBJE !X29[[PARM !X02[]]
                     [RESTR [[RELN  電話-1] ;*<----- 書き換え呼び出し ?OBJE
                             [ENTITY !X02]]]]                    (B)
          [ASPT  UNRL]]]
 [PRAG !X12[[RESTR [[IN !X20[[FIRST [[RELN  CONDESCEND]
                                     [AGEN !X06[[LABEL  *SPEAKER*]]]
                                     [RECP !X04]]]
                              [REST !X25[[FIRST [[RELN  POLITE]
                                                 [AGEN !X06]
                                                 [RECP !X07[[LABEL  *HEARER*]]]]]
                                        [REST !X05[]]]]]]
                    [OUT !X05]]]
           [SPEAKER !X06]
           [HEARER !X07]
           [ASPE [[IN !X03[]]
                  [OUT !X03]]]
           [PRSP-TERMS [[IN !X52[]]
                        [OUT !X78[]]]]
           [TOPIC [[IN !X54[]]
                   [OUT !X77[]]]]]]]
;;;============ Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X2[[LABEL *SPEAKER*]]]
        [RECP !X3[[LABEL *HEARER*]]]
        [OBJE [[RELN CALL-V-1]   ;*<------ 書き換え呼び出しによって動詞化
               [ASPT UNRL]
               [TENSE FUTURE]
               [AGEN !X2]
               [RECP []]
               [TLOC [[PARM !X1[]]
                      [RESTR [[RELN LATER-ADV-1]
                              [ENTITY !X1]]]]]
               [SEM-ASPE UNREAL]]]]
  [PRAG [[BENEFIT NONE]
         [HEARER !X3]
         [INTENTION INFORM]
         [POLITENESS [[DEGREE 2]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X2]
         [TOPIC [[TOPIC-MOD NULL]]]]]]
```

サ変動詞への変換ルールには、以下の３つのパターンがある。
※いずれの場合でも、主語などAGEN/RECPをどう埋めるかは問題となる

1. 基本変換
```
"on <restr reln> インタビュー-1 in :Phase :Japanese :Event-or-object :Event
   in= [[parm !x[]]
        [restr [[reln インタビュー-1]
                [entity !x]]]]
   out= [[reln インタビューする-1]
         [agen []]    ;*<------ ここでは補完できない
         [recp []]]
end")
```

2.1 目的語を伴った変換（の-連体修飾 -> 目的語）
※の-連体修飾の?idenは、TYPE素性でhuman-beingsを除外する
```
"on <parm restr reln> 送付-1 in :Phase :Japanese :event-or-object :event
   in= [[PARM !x[[PARM !y[]]
                 [RESTR [[RELN 送付-1]
                         [ENTITY !y]]]]]
        [RESTR [[RELN  の-連体修飾]
                [OBJE !x]
                [IDEN ?iden]]]
         ?rest]
  out= [[RELN 送付する-1]
        [AGEN []]
        [RECP []]
        [OBJE ?iden]
         ?rest]
end")
```

2.2 目的語を伴った変換（の-連体修飾 -> 目的語）
※の-連体修飾構造が、DESTである
```
"on <parm restr reln> 連絡-1 in :Phase :Japanese :event-or-object :event
   in= [[PARM !x[[PARM !y[]]
                 [RESTR [[RELN 連絡-1]
                         [ENTITY !y]]]]]
        [RESTR [[RELN  の-連体修飾]
                [OBJE !x]
                [DEST ?arg-2]]]
         ?rest]
  out= [[RELN 連絡する-3]
        [AGEN []]
        [RECP ?arg-2]
         ?rest]
end")
```

3. 形容詞を伴った変換（形容詞 -> 任意格の副詞）
※現在は『個人的だ-1』を指定している
```
"on <parm restr reln> 相談-1 in :Phase :Japanese :event-or-object :event
   in= [[PARM !x[[PARM !y[]]
                 [RESTR [[RELN 相談-1]
                         [ENTITY !y]]]]]
        [RESTR [[RELN 個人的だ-1]
                [OBJE !x]
                [ASPT STAT]]]]
 set ?ACCM to root.prag.hearer
  out= [[RELN 相談する-2]
        [AGEN []]
        [MUTL ?ACCM]
        [MANN [[PARM !X[]]
               [RESTR [[RELN 個人的に-1]
                       [ENTITY !X]]]]]]
end")
```

9.2 1語動詞化（サ変名詞からサ変動詞への変換以外）

以下に規則例を示す。

・動詞＋名詞 → 1語動詞化
（～する）必要-2（が）ある-1 --> （～する）NECESSITY

```
(rws:defrwschema2 JAP017 V A
"on <obje restr reln> 必要-2 in :PHASE :japanese
   in= [[reln ある-1]
        [obje [[PARM !x[]]
               [RESTR [[RELN 必要-2]
                       [ENTITY !x]
                       [IDEN ?iden]]]]]
        ?rest]
  set ?expr to root.prag.hearer
   out= [[reln NECESSITY]
         [expr ?expr]
         [obje ?iden]
         ?rest]
end")
```

・受動態＋動詞 → 1語動詞化
... される予定だ --> ...するのを予定する

```
(rws:defrwschema2 JAP021 V YO
"on <obje restr reln> 予定-1 in :phase :Japanese
   in= [[RELN だ-STATEMENT]
        [OBJE [[PARM !X1[]]
               [RESTR [[RELN 予定-1]
                       [IDEN      [[RELN れる-passive]
                                   [OBJE ?OBJE]
                                   [ASPT UNRL]
                                   ?rest0]]
                       [ENTITY !X1]]]]]
        [ASPT STAT]
        ?rest]
     set  ?agen to [[label *they*]]
      input.obje.restr.iden.obje.agen = ?agen
     add ?rest0 to ?obje
   out= [[reln 予定する-1]
         [agen ?agen]
         [obje ?obje]
         [aspt prog]
         ?rest]
end")
```

置かれている --> ある

```
(rws:defrwschema2 JAP100 V S
"on <obje reln> 置く-1 in :PHASE :JAPANESE
    in= [[RELN れる-PASSIVE]
         [ASPT RSLT]
         [OBJE [[RELN 置く-1]
                [AGEN [[LABEL *OTHER*]]]
                [OBJE ?obje]
                ?rest0]]
         ?rest1]
   out= [[RELN ある-1]
         [ASPT STAT]
         [OBJE ?OBJE]
         ?REST0
         ?REST1]
end")
```

9.3 熟語の分解および語句の結合

日本語意味構造で一つの関係名であるが、英語意味構造では分割したい場合や逆に、複数の関係名を合わせて、一つの関係名としたい場合がある。

分割した関係名にしたい場合には、下の３つの方法が考えられる。

1. 解析で行なう
2. JJ変換で行なう（中間化）
3. 熟語ごと変換する(IDIOM/DEFAULT)

ここでは、2. JJ変換で行なった例をあげる。

```
(rws:defrwschema2 JAP061 N さん
"on <restr reln> 参加人数-1 in :PHASE :JAPANESE
   in= [[PARM !X[]]
        [RESTR [[RELN 参加人数-1]
                [ENTITY !X]]]]

   out= [[PARM !X[[PARM !Z[]]
                  [RESTR [[RELN 数-1]
                          [ENTITY !Z]]]]]
         [RESTR [[RELN MODIFY]
                 [ARG1 [[PARM !Y[]]
                        [RESTR [[RELN 参加者-1]
                                [ENTITY !Y]]]]]
                 [ARG2 !X]]]]
end")

(rws:defrwschema2 JAP084 V S
"on <iden restr reln> 国-1 in :PHASE :JAPANESE
   in= [[RELN の-連体修飾]
        [IDEN [[PARM !X[]]
               [RESTR [[RELN 国-1]
                       [ENTITY !X]
                       [SUFFIX 内-1]]]]]
        [OBJE ?obje]]
   out= [[RELN の-連体修飾]
         [IDEN [[PARM !X[]]
                [RESTR [[RELN 国内-1]
                        [ENTITY !X]]]]]
         [OBJE ?obje]]
end")
```

既に、日本語解析で接尾辞などで分かれている関係名も、概念として、一つの関係名の方が良い場合は、ＪＪ変換で統合する。

・名詞＋接尾辞 → 名詞化

```
(rws:defrwschema2 JAP026 N DO
"on <restr reln> どこ-1 in :PHASE :JAPANESE
    in= [[PARM !x[]]
         [RESTR [[RELN どこ-1]
                 [SUFFIX か-INDEFINITE]
                 [ENTITY !x]]]]

   out= [[PARM !x[]]
         [RESTR [[RELN どこか-1]
                 [ENTITY !x]]]]
end")
```

9.4 日本語動詞の格の変換

日本語意味表現で表された動詞や格名は、表層的な構造であって、
そのまま、日英変換して生成した場合、英語表現として適当でない
場合がある。

個々の動詞の変換を行なう前に、任意的な格を動詞を参照しながら、
JJ変換を行なう。

会議は９時から行なわれる。→ 会議は９時に始まる。

```
(rws:defrwschema2 JAP101 V S
"on <obje reln> 行なう-1 in :PHASE :JAPANESE
    in= [[RELN れる-PASSIVE]
         [ASPT ?aspt]
         [OBJE [[RELN 行なう-1]
                [AGEN [[LABEL *OTHER*]]]
                [OBJE ?obje]
                ?rest0]]
         [TDEP ?tdep]
         ?rest1]
   out= [[RELN 始まる-1]
         [ASPT ?aspt]
         [OBJE ?OBJE]
         [TLOC ?tdep]
         ?REST0
         ?REST1]
end")
```

会議を９時から行ないます。→ 会議は９時に始まる。

```
(rws:defrwschema2 JAP102 V S
"on <reln> 行なう-1 in :PHASE :JAPANESE
    in= [[RELN 行なう-1]
         [AGEN []]
         [OBJE ?obje]
         [TDEP ?tdep]
         ?rest1]
   out= [[RELN 始まる-1]
         [OBJE ?OBJE]
         [TLOC ?tdep]
         ?REST1]
end")
```

また、次のような例においても、意味的にはほとんど同じであるにも
かかわらず、助詞によって、格名が異なった構造となる。
ここでも、JJ変換を用いて、該当動詞の標準化を行なっている。

(a) >P-520 他の研究論文 から 教えられることが多い DEPT
(b) >P-521 他の研究論文 によって 教えられることが多い CAUS
(c) >P-522 他の研究論文 に 教えられることが多い SOUR

この場合は、『教える＋れるPASSIVE』を『学ぶ』という一語動詞にし、
CAUS格、SOUR格は、DEPT格に合わせる形式とした。

(d) >P-612 私はその研究論文に多くを教わりました。

『教わる』を『学ぶ』にJJ変換した。

(a),(b),(c),(d)のルール例を次にあげる

```
(a)   >P-520 他の研究論文 から 教えられることが多い        DEPT
      "on <obje reln> 教える-1 in :PHASE :JAPANESE
          in= [[RELN れる-PASSIVE]
               [OBJE [[RELN 教える-1]
                      [AGEN [[LABEL *other*]]]
                      [OBJE ?obje]
                      [DEPT ?dept] ;*<====
                      ?rest0]]
               ?rest1]
          out= [[RELN 学ぶ-1]
               [AGEN []]
               [OBJE ?OBJE]
               [DEPT ?DEPT] ;*<====
               ?REST0
               ?REST1]
      end")

(b)   >P-521 他の研究論文 によって 教えられることが多い  CAUS -> DEPT
      "on <obje reln> 教える-1 in :PHASE :JAPANESE
          in= [[RELN れる-PASSIVE]
               [OBJE [[RELN 教える-1]
                      [AGEN [[LABEL *other*]]]
                      [OBJE ?obje]
                      [CAUS ?caus] ;*<====
                      ?rest0]]
               ?rest1]
          out= [[RELN 学ぶ-1]
               [AGEN []]
               [OBJE ?OBJE]
               [DEPT ?caus] ;*<====
               ?REST0
               ?REST1]
      end")

(c)   >P-522 他の研究論文 に 教えられることが多い        SOUR -> DEPT
      "on <obje reln> 教える-1 in :PHASE :JAPANESE
          in= [[RELN れる-PASSIVE]
               [OBJE [[RELN 教える-1]
                      [AGEN [[LABEL *other*]]]
                      [OBJE ?obje]
                      [SOUR ?sour] ;*<====
                      ?rest0]]
               ?rest1]
          out= [[RELN 学ぶ-1]
               [AGEN []]
               [OBJE ?OBJE]
               [DEPT ?sour] ;*<====
               ?REST0
               ?REST1]
      end")

(d)   >P-612 私はその研究論文に多くを教わりました。
      "on <reln> 教わる-1 in :PHASE :JAPANESE
          in= [[RELN 教わる-1]
               [AGEN ?agen]
               [OBJE ?obje]
               [SOUR ?sour] ;*<====
               ?rest0]
          out= [[RELN 学ぶ-1]
               [AGEN ?agen]
               [OBJE ?OBJE]
               [DEPT ?sour] ;*<====
               ?REST0]
      end")
```

9.5 イディオム的なJJ変換

・動詞＋名詞 が 1語の形容詞に変換される場合

```
(rws:defrwschema2 JAP114 V S
"on <reln> 持つ-1 in :PHASE :JAPANESE
   in= [[RELN 持つ-1]
        [AGEN ?agen]
        [OBJE [[PARM !Y[]]
               [RESTR [[RELN 興味-1]
                       [ENTITY !Y]]]]]
        [LOCT ?loct]
        ?REST]
   out= [[RELN 興味がある-1]
         [AGEN ?agen]
         [OBJE ?loct]
         ?REST]
end")
```

・形容詞 が否定＋動詞に変換される場合

```
(rws:defrwschema2 JAP096 V S
"on <reln> 未定だ-1 in :PHASE :JAPANESE
   in= [[RELN 未定だ-1]
        [OBJE ?OBJE]
        ?REST]

  set ?output to [[RELN NEGATE]
                  [OBJE [[RELN 決まる-1]
                         [OBJE ?OBJE]
                         ?rest]]]

      if ?output.obje.aspt then
         ?output.aspt = ?output.obje.aspt
         delete aspt from ?output.obje
      endif
     out= ?output
end")
```

・述語のイディオム的な変換の場合（中間化）

（～しても）大丈夫だ-1 --> できる-POSSIBLE

```
(rws:defrwschema2 JAP015 V I
"on <obje reln> 大丈夫だ-1 in :PHASE :JAPANESE
   in=  [[OBJE [[RELN  大丈夫だ-1]
                [OBJE []]
                [ASPT ?aspt]]]
         [OPPS [[PARM !x[]]
                [RESTR [[RELN  ても-1]
                        [IDEN ?iden]
                        [ENTITY !x]]]]]
         ?rest]

        if input.opps.restr.iden.agen then
          set  ?expr to input.opps.restr.iden.agen
        endif

  out=  [[OBJE [[RELN  できる-POSSIBLE]
                [ASPT ?aspt]
                [expr ?expr]
                [OBJE ?iden]]]
         ?rest]
end")
```

9.6 縮退や削除のJJ変換

日本語表層表現の中で、単なる言い回しや形式名詞などの不要語句は、必要な動詞部分などにして、縮退または構造から削除する。

Ex. ～することにしましょう。→ ～しましょう。

◇『てみる-1』の縮退　　～してみたい -> ～したい

```
(rws:defrwschema2 JAP016 V HO
"on <obje reln> てみる-1 in :PHASE :JAPANESE
   in= [[RELN WANT-DESIRE]
        [EXPR ?expr]
        [OBJE [[RELN てみる-1]
               [AGEN ?expr]
               [OBJE ?obje]
               ?rest0]]
         ?rest]
  out= [[RELN WANT-DESIRE]
        [ASPT  STAT]
        [EXPR ?expr]
        [OBJE ?obje]
         ?rest]
end")
```

◇形式名詞の縮退（こと-FN-1）

```
>P-542 申し上げた通り全体を独占することはできません
 (rws:defrwschema2 JAP014 V I
 "on <obje obje restr reln> こと-FN-1 in :PHASE :JAPANESE
    in= [[OBJE [[RELN  できる-POSSIBLE]
                [EXPR ?expr]
                [OBJE [[PARM !x[]]
                       [RESTR [[RELN こと-FN-1]
                               [IDEN ?IDEN]
                               [ENTITY !x]]]]]
                ?REST0]]
         ?REST]
;       -------- tag linked ----------
    if ?expr =? [] and input.obje.obje.restr.iden.agen and
       input.obje.obje.restr.iden.agen =? []
            then
     set ?expr to input.obje.obje.restr.iden.agen
          endif
;       ----------------------------
       if input.obje.obje.restr.iden.agen and
          input.obje.obje.restr.iden.agen =? []
            then
      input.obje.obje.restr.iden.agen = ?expr
           endif
;       ------------ ASPT delete ----
       if input.obje.obje.restr.iden.aspt
            then
      delete aspt from input.obje.obje.restr.iden
                          endif
;       ----------------------------
  out= [[OBJE [[RELN  できる-POSSIBLE]
               [EXPR ?expr]
               [OBJE ?IDEN]
               ?REST0]]
         ?REST]
end")
```

>P-609 質問の時間を変更することがあります

```
(rws:defrwschema2 JAP086 V S
"on <obje restr reln> こと-FN-1 in :PHASE :JAPANESE
   in= [[RELN ある-1]
        [ASPT STAT]
        [OBJE [[PARM !Y[]]
              [RESTR [[RELN こと-FN-1]
                      [IDEN ?IDEN]
                      [ENTITY !Y]]]]]]
     if input.obje.restr.iden.aspt =! $past then fail endif
      input.obje.restr.iden.aspt =  $rslt
  out= ?iden
end")
```

>P-214 わたしの代理の者を出席させることになるでしょう。

```
(rws:defrwschema2 JAP087 V S
"on <resl restr reln> こと-FN-1 in :PHASE :JAPANESE
   in= [[RELN なる-4]
        [ASPT UNRL]
        [RESL [[PARM !Y[]]
              [RESTR [[RELN こと-FN-1]
                      [IDEN ?IDEN]
                      [ENTITY !Y]]]]]
        ?REST]
      add ?rest to ?iden
   out= ?iden
end")
```

>P-255 詳細は当日決めることにしましょう。

```
(rws:defrwschema2 JAP088 V I
"on <obje restr reln> こと-FN-1 in :PHASE :JAPANESE
   in= [[RELN する-3]
        [AGEN ?agen]
        [OBJE [[PARM !x[]]
              [RESTR [[RELN こと-FN-1]
                      [IDEN @iden
                            [[RELN ?action]
                             [ASPT ?aspt]
                              ?rest]]
                      [ENTITY !x]]]]]
        ?rest3]
      out= @iden
end")
```

◇形式名詞の縮退（の-FN-1）

>P-340 その時期に会議に出席する／の／は難しいです
>P-399 では会議でお目にかかる／の／を楽しみにしています
>P-473 日本へ行く／の／は初めてです

```
(rws:defrwschema2 JAP019 V SO
"on <restr reln> の-FN-1 in :phase :japanese
   in=  [[PARM !X1[]]
         [RESTR [[RELN の-FN-1]
                 [IDEN ?IDEN]
                 [ENTITY !X1]]]]
    if input.restr.iden.reln =? $いう-1 then fail endif
   out= ?IDEN
end")
```

9.7 その他

◇ 形容詞の比較 （～のほうが～だ）-> 『もっと』の付与・『ほう-1』の縮退

英語には、形容詞の比較級変化があるが、日本語では『～のほうが～だ』で表される。このため、比較の意味を持たせるために、DEGR格で『もっと』を付与しておき、MORE-ADV-1 に変換してから、生成に送っている。
（COMP格を持たない場合でも、DEGR格の『MORE-ADV-1』で処理できる）
最上級についても、同様の処理が行なわれる。

```
(rws:defrwschema2 JAP058 V SO
"on <obje restr reln> ほう-1 in :Phase :Japanese
   in= [[RELN ?adjective]
        [OBJE [[PARM !y[]]
              [RESTR [[RELN ほう-1]
                      [IDEN @iden
                            [[parm ?parm]
                             [restr ?restr]]]
                      [ENTITY !y]]]]]
        ?rest]

  out= [[RELN ?adjective]
        [OBJE @iden]
        [DEGR [[PARM !z[]]
              [RESTR [[RELN もっと-1]
                      [ENTITY !z]]]]]
        ?rest]
end")
```

◇数量後置の正規化

『原稿用紙三枚』 と 『三枚の原稿用紙』の構造が異なるため、
『三枚の原稿用紙』の構造に統一する。

『原稿用紙三枚』の場合の解析日本語構造

```
      [[PARM !X06[[PARM !X04[]]
                  [RESTR [[RELN  NUMBER]
                          [ENTITY !X04]
                          [UNIT  枚]
                          [IDEN [[PLACE-1  三-1]]]]]]]
       [RESTR [[RELN  数量詞後置]
               [OBJE !X06]
               [IDEN [[PARM !X05[]]
                      [RESTR [[RELN  原稿用紙-1]
                              [ENTITY !X05]]]]]]]]
```

>P-610 要約は原稿用紙三枚で書いてください

```
(rws:defrwschema2 JAP065 V S
"on <restr reln> 数量詞後置 in :PHASE :JAPANESE
   in=  [[PARM !Z ?number]
         [RESTR [[RELN 数量詞後置]
                 [IDEN ?iden]
                 [OBJE !Z]]]]

  out= [[PARM !Z ?iden]
        [RESTR [[RELN の-連体修飾]
                [IDEN ?number]
                [OBJE !Z]]]]
end")
```

◇数量副詞を修飾型に正規化

数量的な副詞は、主語と目的語のどちらにに係るのかは、意味的な問題で、
曖昧であり、素性構造から判断するのは難しい。

Ex. 新幹線がたくさんトンネルを通った。
→ たくさんの新幹線 がトンネルを通った。
→ 新幹線が たくさんのトンネル を通った。

そこで、数量的な副詞の素性構造中の単位名詞（UNIT）に着目し、
OBJE素性 か AGEN素性 が、その単位で用いられる場合に、付加する
ようにした。（但し、現在は、関係名指定で行なっている。）

今後は、TYPE素性か、または単位名詞群などのテーブルなども用意す
る必要がある。

>P-18 案内書を二百通送った
↓
二百通の案内書を送った

```
(rws:defrwschema2 JAP066 V S
"on <quant restr unit> 通 in :PHASE :JAPANESE
   in= [[RELN ?ACTION]
        [OBJE @OBJE<parm* restr reln> 案内書-1] ;*<--(UNITの単位を使う名詞)
        [QUANT @QUANT
               [[PARM !Y[]]
                [RESTR [[RELN NUMBER]
                        [IDEN ?IDEN]
                        [UNIT 通]  ;*<-----------------(単位名詞 UNIT)
                        [ENTITY !Y]]]]]
        ?REST]

  out= [[RELN ?ACTION]
        [OBJE [[PARM !Z @OBJE]
               [RESTR [[RELN の-連体修飾]
                       [IDEN @QUANT]
                       [OBJE !Z]]]]]
        ?REST]
end")
```

>P-163 京都の夜のツアーには現在で百名ばかり集まっておられます
数量副詞を主語に転換

```
(rws:defrwschema2 JAP073 V S
"on <reln> 集まる-1 in :PHASE :JAPANESE
   in= [[RELN 集まる-1]
        [AGEN []] ;*<--------------- AGEN は human-beings
        [QUANT @QUANT
               [[PARM !Y[]]
                [RESTR [[RELN NUMBER]
                        [IDEN ?IDEN]
                        [UNIT 名]  ;*<-----------------(UNIT)
                        [ENTITY !Y]
                        ?rest1]]]]
        ?REST]

  out= [[RELN 集まる-1]
        [AGEN @QUANT]
        ?REST]
end")
```

１０　日英変換

１０.１ イディオム変換

イディオム変換は、文字通り、日本語意味構造をイディオムとして、大きく取って、一度に対応する英語意味構造に変換するパターンである。

JJ変換を通してから、イディオム変換できる場合もあるが、基準としては、英語意味表現が完全に合致している場合や、英語表現に多様性や多義性がないことがあげられる。

逐次的な変換では、解決できない場合などに効果的であるものの、単純に変換してしまうので、多用せずに、今後、文脈情報なども取り入れることが必要である。

◇ RESPONSE の『分かりました』→ "I_SEE-IDIOM-1"

```
(rws:defrwschema2 IDIOM008 V わか
"on <reln> 分かる-1  in :PHASE :J-E :TYPE :IDIOM :IFT :RESPONSE
   in= [[RELN 分かる-1]
        [expr [[LABEL *SPEAKER*]]]
        [obje []]
        [aspt stat]]
   out= [[RELN I_SEE-IDIOM-1]
         [aspt STAT]]
end")
```

◇ 『他に何か』→ "something_else-PRON-1"

```
(rws:defrwschema2 IDIOM020 N なに
 "on <obje quant restr reln> 何か-1 in :PHASE :J-E :TYPE :IDIOM
    in=[[OBJE [[OBJE []]
               [QUANT [[PARM !Z[]]
                       [RESTR [[RELN 何か-1]
                               [ENTITY !Z]]]]]
               [LOCT [[PARM !x[]]
                      [RESTR [[RELN  他-1]
                              [ENTITY !x]]]]]
                ?rest]]
        ?rest2]

   out=[[OBJE [[OBJE [[PARM !y[]]
                      [RESTR [[RELN something_else-PRON-1]
                              [ENTITY !y]]]]]
               ?rest]]
        ?rest2]
end")
```

◇ 奥様 --> "your wife"

```
(rws:defrwschema2 IDIOM064 V おく
"on <restr reln> 奥様-1 in :PHASE :J-E :TYPE :IDIOM
   in=  [[PARM !y[]]
         [RESTR [[RELN 奥様-1]
                 [ENTITY !y]]]]
   set ?arg1 to root.prag.hearer

  out=  [[PARM !x[[PARM !y[]]
                  [RESTR [[RELN WIFE-N-1]
                          [ENTITY !y]]]]]
        [RESTR [[RELN MODIFY]
                [ARG1 ?arg1]
                [ARG2 !x]]]]
end")
```

◇『（何も～する）こと（がない）』 --> "have nothing to～"

```
(rws:defrwschema2 IDIOM021 V こと
"on <obje obje parm restr reln> こと-1 in :PHASE :J-E :type :idiom
   in= [[RELN NEGATE]
        [OBJE [[RELN  ある-1]
               [OBJE [[PARM !y[[PARM !x[]]
                               [RESTR [[RELN  こと-1]
                                       [ENTITY !x]]]]]
                      [RESTR [[RELN  ?reln]
                              [AGEN ?agen]
                              [OBJE !y]
                              ?rest1]]]]
               [DEGR [[PARM !Z[]]
                      [RESTR [[RELN 何も-1]
                              [ENTITY !Z]]]]]
               ?rest2]]
        ?rest3]

   out= [[RELN HAVE-V-1]
         [AGEN ?agen]
         [OBJE [[PARM !y[[PARM !Z[[PARM !x[]]
                                  [RESTR [[RELN nothing-PRON-1]
                                          [ENTITY !x]]]]]
                         [RESTR [[RELN MORE-ADV-1]
                                 [OBJE !Z]]]]]
                [RESTR [[RELN ?reln]
                        [AGEN ?agen]
                        [OBJE !y]
                        ?rest1]]]]
         ?rest2
         ?rest3]
end")
```

◇『～する機会がある』 --> have chance (TO DO)

```
(rws:defrwschema2 IDIOM022 V こと
"on <obje restr reln> 機会-FN-1 in :PHASE :J-E :type :idiom
   in= [[RELN  ある-1]
        [OBJE @OBJE
              [[PARM !x[]]
               [RESTR [[RELN  機会-FN-1]
                       [IDEN [[RELN ?action]
                              [AGEN ?agen]
                              [OBJE ?obje]
                              ?rest1]]
                       [ENTITY !x]]]]]
        ?rest]

   out= [[RELN HAVE-V-1]
         [AGEN ?agen]
         [OBJE [[PARM !Y[[PARM !x[]]
                         [RESTR [[RELN  CHANCE-N-1]
                                 [ENTITY !x]]]]]
                [RESTR [[RELN APPOSITION]
                        [IDEN [[RELN ?action]
                               [AGEN ?agen]
                               [OBJE ?obje]
                               ?rest1]]
                        [OBJE !Y]]]]]
         ?rest]
end")
```

◇CONTENT そちら --> THIS 電話会話の場合の相手（HEARER）

```
(rws:defrwschema2 IDIOM023 N そち
"on <obje obje restr reln> そちら-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[RELN  QUESTIONIF]
        [OBJE [[RELN  だ-IDENTICAL]
               [OBJE [[PARM !x[]]
                      [RESTR [[RELN  そちら-1]
                              [ENTITY !x]]]]]
               [IDEN [[PARM !y[]]
                      [RESTR [[RELN  NAMED]
                              [ENTITY !y]
                              [IDEN  ?iden]]]]]
                ?rest1]]
         ?rest2]
  out= [[RELN  QUESTIONIF]
        [OBJE [[RELN  だ-IDENTICAL]
               [OBJE [[PARM !x[]]
                      [RESTR [[RELN  this-PRON-1]
                              [ENTITY !x]]]]]
               [IDEN [[PARM !y[]]
                      [RESTR [[RELN  NAMED]
                              [ENTITY !y]
                              [IDEN  ?iden]]]]]
                ?rest1]]
         ?rest2]
end")
```

◇『～が集合場所だ』 --> 動詞 "meet"+LOCT

```
(rws:defrwschema2 IDIOM025 V しゅ
"on <obje restr reln> 集合場所-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[RELN だ-IDENTICAL]
        [OBJE [[PARM !X[]]
               [RESTR [[RELN 集合場所-1]
                       [ENTITY !X]]]]]
        [IDEN ?Y]
        ?REST]
  set ?agen to []
   out= [[RELN meet-V-1]
         [AGEN ?AGEN]
         [LOCT ?Y]
         ?REST]
end")
```

◇『留守にする』 --> ABSENT-ADJ-1

```
(rws:defrwschema2 IDIOM038aa V るす
"on <resl iden restr reln> 留守-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[RELN する-6]
        [OBJE ?obje]
        [RESL [[RELN だ-IDENTICAL]
               [OBJE ?obje]
               [IDEN [[PARM !X[]]
                      [RESTR [[RELN 留守-1]
                              [ENTITY !X]]]]]
               ?REST0]]
        ?REST1]

   out= [[RELN ABSENT-ADJ-1]
         [OBJE ?obje]
         ?REST0
         ?REST1]
end")
```

◇動詞などの群前置詞（前置詞マーカー）へのイディオム変換

意味格名が相応しくない場合は、格名も変換する場合がある。

◆ MANN『～に敬意を表して』 --> MANN 前置詞マーカー"IN_HONOR_OF"

```
(rws:defrwschema2 IDIOM068 V て
"on <mann restr reln> て-CIRCUMSTANCES in :PHASE :J-E :TYPE :IDIOM
   in= [[MANN [[PARM !x[]]
               [RESTR [[RELN て-CIRCUMSTANCES]
                       [IDEN [[RELN 表する-1]
                              [AGEN ?AGEN]
                              [RECP ?RECP]
                              [OBJE [[PARM !Z[]]
                                     [RESTR [[RELN 敬意-1]
                                             [ENTITY !Z]]]]]
                              ?REST1]]
                       [ENTITY !x]]]]]
        ?rest]

   out= [[MANN [[RELN IN_HONOR_OF-PREP-1]
                [OBJE ?RECP]]]
         ?REST]
end")
```

◆ MANN『～を含めて』 --> MANN 前置詞マーカー "INCLUDING"

```
(rws:defrwschema2 IDIOM069 V て
"on <mann restr reln> て-CIRCUMSTANCES in :PHASE :J-E :TYPE :IDIOM
   in= [[MANN [[PARM !x[]]
               [RESTR [[RELN て-CIRCUMSTANCES]
                       [IDEN [[RELN 含める-1]
                              [AGEN ?AGEN]
                              [LOCT []]
                              [ASPT ?ASPT]
                              [OBJE ?OBJE]
                              ?REST1]]
                       [ENTITY !x]]]]]
        ?rest]

   out= [[MANN [[RELN INCLUDING-PREP-1]
                [OBJE ?OBJE]]]
         ?REST1
         ?REST]
end")
```

◆ MANN『～に代わって』 --> ROLE 前置詞マーカー"INSTEAD_OF"

```
(rws:defrwschema2 IDIOM099 N かわ
"on <mann restr iden reln> 代わる-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[MANN [[PARM !x[]]
               [RESTR [[reln て-circumstances]
                       [IDEN [[RELN 代わる-1]
                              [agen ?agen]
                              [loct ?loct]]]
                       [entity !x]]]]]
        ?rest]

  out= [[ROLE [[RELN INSTEAD_OF-PREP-1]
               [OBJE ?loct]]]
        ?rest]
end")
```

◇ 副詞句への変換

意味格名が相応しくない場合は、格名も変換する場合がある。

◆ QUANT『一度』 --> QUANT "once"

```
(rws:defrwschema2 IDIOM073 V ど
"on <quant restr unit> 度 in :PHASE :J-E :TYPE :IDIOM
   in= [[QUANT [[PARM !x[]]
                [RESTR [[RELN NUMBER]
                        [IDEN [[PLACE-1 一-1]]]
                        [ENTITY !x]
                        [UNIT 度]]]]]
        ?rest]

   out= [[QUANT [[PARM !y[]]
                 [RESTR [[RELN once-ADV-1]
                         [ENTITY !y]]]]]
         ?REST]
end")
```

◆ QUANT 『二度』 --> QUANT "twice"

```
(rws:defrwschema2 IDIOM074 V ど
"on <quant restr unit> 度 in :PHASE :J-E :TYPE :IDIOM
   in= [[QUANT [[PARM !x[]]
                [RESTR [[RELN NUMBER]
                        [IDEN [[PLACE-1 二-1]]]
                        [ENTITY !x]
                        [UNIT 度]]]]]
        ?rest]

   out= [[QUANT [[PARM !y[]]
                 [RESTR [[RELN twice-ADV-1]
                         [ENTITY !y]]]]]
         ?REST]
end")
```

◆ QUANT『何度も』 --> QUANT "many times"

```
(rws:defrwschema2 IDIOM075 V ど
"on <quant restr unit> 度 in :PHASE :J-E :TYPE :IDIOM
   in= [[QUANT [[PARM !x[]]
                [RESTR [[RELN NUMBER]
                        [IDEN 何-1]
                        [ENTITY !x]
                        [UNIT 度]
                        [CONTR も-概数-1]]]]]
        ?rest]

   out= [[QUANT [[PARM !y[]]
                 [RESTR [[RELN MANY_TIMES-ADV-1]
                         [ENTITY !y]]]]]
         ?REST]
end")
```

◆ OPPS『遅くとも』 --> DEGR "at the latest"

```
(rws:defrwschema2 IDIOM066 V とも
"on <opps restr reln> とも-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[OPPS [[PARM !x[]]
              [RESTR [[RELN とも-1]
                      [IDEN [[RELN 遅い-1]
                             [ASPT STAT]
                             [OBJE []]]]
                      [ENTITY !x]]]]]
       ?rest]
   out= [[DEGR [[PARM !y[]]
                [RESTR [[RELN at_the_latest-ADV-1]
                        [ENTITY !y]]]]]
        ?REST]
end")
```

◆ TLOC 『近い内に』 --> TLOC "soon"

```
(rws:defrwschema2 IDIOM078 V うち
"on <tloc restr reln> 内-FN-TIME in :PHASE :J-E :TYPE :IDIOM
   in= [[TLOC [[PARM !y[]]
              [RESTR [[RELN 内-FN-TIME]
                      [IDEN [[RELN 近い-1]
                             ?rest0]]
                      [ENTITY !y]]]]]
       ?rest]
   out= [[TLOC [[PARM !y[]]
                [RESTR [[RELN SOON-ADV-1]
                        [ENTITY !y]]]]]
        ?REST]
end")
```

◆ TLOC『もう長い間』 --> TLOC "for a long time"

```
(rws:defrwschema2 IDIOM084 V あい
"on <tloc restr reln> 間-FN-TIME in :PHASE :J-E :TYPE :IDIOM
   in= [[TLOC [[PARM !Z[]]
              [RESTR [[RELN 間-FN-TIME]
                      [IDEN [[RELN 長い-1]
                             [ASPT STAT]
                             [OBJE ?OBJE]
                             [TASP @TASP<restr reln> もう-1]]]
                      [ENTITY !Z]]]]]
       ?REST]
  out= [[TLOC [[PARM !Z[]]
              [RESTR [[RELN FOR_A_LONG_TIME-ADV-1]
                      [ENTITY !Z]]]]]
       ?REST]
end")
```

◆ NEGATE+決して-1 --> 関係詞 NEGATE-NEVER

```
(rws:defrwschema2 IDIOM095 V けっ
"on <obje mann restr reln> 決して-1 in :PHASE :J-E :TYPE :IDIOM
   in= [[RELN NEGATE]
        [OBJE [[RELN ?action]
               [MANN [[PARM !Y[]]
                      [RESTR [[RELN 決して-1]
                              [ENTITY !Y]]]]]
               ?rest1]] ?REST]
  out= [[RELN NEGATE-NEVER]
        [OBJE [[RELN ?action]
               ?rest1]] ?REST]
end")
```

１０.２ 一般変換

個々の語句に対するデフォルト変換に入る前に、モーダルや格関係などの機能語周りを変換する。ここで、決定された構造がそのまま英語意味構造となる。

１０.２.１ コピュラの基本変換

```
(rws:defrwschema2 GEN001 V だ
"on <reln> だ-IDENTICAL in :PHASE :J-E :type :GENERAL
   in= ((reln だ-IDENTICAL)
        (OBJE ?OBJE)
        ?rest)
   out= ((reln BE-V-COPULA)
         (OBJE ?OBJE)
         ?rest)
end")
```

N.B.　BE動詞は、コピュラの BE-V-COPULA と 存在を示すBE-V-EXIST がある。

１０.２.２ 連体修飾

『の-連体修飾』の基本変換
の-連体修飾(IDEN,OBJE)を、MODIFY(ARG1,ARG2)に変換する。
この規則のための書き換え環境を特別にPRE-GENERALと設定する。

```
(rws:defrwschema2 GEN003 N の
"on <reln> の-連体修飾 in :phase :J-E :Type :PRE-GENERAL
   in= ((reln の-連体修飾)
        (iden ?iden)
        (obje ?obje)
        ?rest)

   out= ((RELN MODIFY)
         (ARG1 ?iden)
         (ARG2 ?obje)
         ?rest)
end")
```

『の-連体修飾』が意味格を伴う場合は、それぞれの意味格を連体修飾名に示す形で変換する。

・ LOCT格＋『の-連体修飾』の場合は、連体修飾名をMODIFY-LOCTとする

```
(rws:defrwschema2 GEN006 N の
"on <reln> の-連体修飾 in :PHASE :J-E :TYPE :GENERAL
   in= ((reln の-連体修飾)
        (LOCT ?loct)
        (OBJE ?obje))
   out= ((reln MODIFY-LOCT)
         (ARG1 ?loct)
         (ARG2 ?obje))
end")
```

・ TLOC格＋『の-連体修飾』の場合は、連体修飾名をMODIFY-TLOCとする

```
(rws:defrwschema2 GEN009 N の
"on <reln> の-連体修飾 in :PHASE :J-E :TYPE :GENERAL
   in= ((reln の-連体修飾)
        (TLOC ?tloc)
        (OBJE ?obje))
   out= ((reln MODIFY-TLOC)
         (ARG1 ?tloc)
         (ARG2 ?obje))
end")
```

・意味格に転換できる日本語構造がある場合は、さらに連体修飾名を
特定化する。
～の近くの～ -> MODIFY-LOCT-NEAR

```
(rws:defrwschema2 GEN014 N ちか
"on <arg1 parm restr reln> 近く-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[reln MODIFY]
        [ARG1  [[PARM !x[[PARM !y[]]
                        [RESTR [[RELN 近く-1]
                                [ENTITY !y]]]]]
               [RESTR [[RELN MODIFY]
                       [ARG1 ?arg1]
                       [ARG2 !x]]]]]
        [ARG2  ?arg2]]

   out= [[reln MODIFY-LOCT-NEAR]
         [arg1 ?arg1]
         [arg2 ?arg2]]
end")
```

１０.２.３　時間を表す構造

・ＰＯＤ(PART OF DAY)は、午前、午後などを表わし、特別の修飾関係名に
変換する。 (MODIFY-POD)

```
(rws:defrwschema2 GEN022 N TI
"on <arg1 restr reln> TIME in :PHASE :J-E :TYPE :GENERAL
   in= [[RELN MODIFY]  ;*<======連体修飾
        [ARG1 @ARG1
              [[PARM !X[]]
               [RESTR [[RELN TIME]
                       [POD ?POD]  ;*<======PART-OF-DAY
                       [ENTITY !X]]]]]
        [ARG2 ?arg2]]
  out= [[RELN MODIFY-POD]  ;*<======PART-OF-DAY連体修飾
        [ARG1 @ARG1]
        [ARG2 ?arg2]]
end")
```

・午前、午後Ｘ時の構造は、"o'clock"を付加する。

```
(rws:defrwschema2 GEN023 N TI
"on <restr reln> TIME in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN TIME]
                [HOUR [[RELN NUMBER]  ;*<====== HOUR素性名
                       [IDEN ?iden]
                       ?rest]]
                [POD ?POD]  ;*<======PART-OF-DAY
                [ENTITY !X]]]]
  out= [[PARM !Z[[PARM !P[[PARM !X[]]
                          [RESTR [[RELN o-clock-N-1]  ;*<===『時』に相当する
                                  [ENTITY !X]]]]]
                 [RESTR [[RELN MODIFY]
                         [ARG1[[PARM !Z2[]]
                               [RESTR [[RELN NUMBER]
                                       [IDEN ?iden]  ;*<======数値
                                       [ENTITY !Z2]]]]]
                         [ARG2 !P]]]]]
        [RESTR [[RELN MODIFY-POD]  ;*<======PART-OF-DAY連体修飾
                [ARG1[[PARM !X2[]]
                      [RESTR [[RELN TIME]
                              [POD ?POD]  ;*<====== PART-OF-DAY
                              [ENTITY !X2]]]]]
                [ARG2 !Z]]]]
end")
```

・午前、午後X時の構造は、"o'clock"を付加する。

```
(rws:defrwschema2 GEN024 N TI
"on <restr reln> TIME in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN TIME]
                [HOUR [[RELN NUMBER]
                       [IDEN ?iden]
                       ?rest]]
                [ENTITY !X]]]]
 if ?iden.place-10 and ?iden.place-10 =! $一-1    ;*<=== 1～11に限定
              then fail endif
 if ?iden =? $何-1 then fail endif  ;*<======疑問副詞の場合、排除
  out= [[PARM !Z[[PARM !X[]]
                 [RESTR [[RELN o-clock-N-1]
                         [ENTITY !X]]]]]
        [RESTR [[RELN MODIFY]
                [ARG1[[PARM !Z2[]]
                      [RESTR [[RELN NUMBER]
                              [IDEN ?iden]
                              [ENTITY !Z2]]]]]
                [ARG2 !Z]]]]
end")
```

・時間の構造『１５時２４分』などは、前置詞マーカー"AT"を付加する。

```
(rws:defrwschema2 GEN025 N TI
"on <restr reln> TIME in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN TIME]
                [MINUTE [[RELN NUMBER]
                         [IDEN ?iden-m]]]
                [HOUR [[RELN NUMBER]
                       [IDEN ?iden-h]
                       ?rest]]
                [ENTITY !X]]]]

  out= [[RELN AT-PREP-1]
        [OBJE
       [[PARM !Z[[PARM !X[]]
                 [RESTR [[RELN NUMBER]
                         [IDEN ?iden-m]
                         [ENTITY !X]]]]]
        [RESTR [[RELN MODIFY-TIME]
                [ARG1[[PARM !Z2[]]
                      [RESTR [[RELN NUMBER]
                              [IDEN ?iden-h]
                              [ENTITY !Z2]]]]]
                [ARG2 !Z]]]]]]
end")
```

・西暦の構造は、前置詞マーカー"IN"を付加し、英語の読みを
想定した２つの１０の位の構造に分割しておく。

```
(rws:defrwschema2 GEN026 N TI
"on <restr reln> TIME in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN TIME]
                [YEAR [[RELN NUMBER]
                       [IDEN [[PLACE-1 ?place-1]
                              [PLACE-10 ?place-10]
                              [PLACE-100 ?place-100]
                              [PLACE-1000 ?place-1000]]]
                       ?rest]]
                [ENTITY !X]]]]
  out= [[RELN IN-PREP-1]
        [OBJE
   [[PARM !Z[[PARM !X[]]
             [RESTR [[RELN NUMBER]
                     [IDEN [[PLACE-1 ?place-1]
                            [PLACE-10 ?place-10]]]
                     [ENTITY !X]]]]]
    [RESTR [[RELN MODIFY-TIME]
            [ARG1[[PARM !Z2[]]
                  [RESTR [[RELN NUMBER]
                          [IDEN [[PLACE-1 ?place-100]    ;*<==PLACE-1へ転換
                                 [PLACE-10 ?place-1000]]];*<==PLACE-10へ転換
                          [ENTITY !Z2]]]]
            [ARG2 !Z]]]]
        ]]
end")
```

生成結果
>P-277 次回の会議は千九百九十三年に開かれる予定です
"They're planning to hold the next conference in nineteen ninety three."

・日付（月名のみ）の場合は、DATE-MONTH連体修飾名とする。

```
(rws:defrwschema2 GEN087 N X
"on <reln> date in :PHASE :J-E :TYPE :GENERAL
    in= [[RELN date]
         [month ?month]
         [entity ?entity]]

   out= [[RELN DATE-MONTH]
         [month ?month]
         [entity ?entity]]
end")
```

・日付（月日）の場合は、DATE-MONTH-DAY連体修飾名とする。

```
(rws:defrwschema2 GEN088 N X
"on <reln> date in :PHASE :J-E :TYPE :GENERAL
    in= [[RELN date]
         [day ?day]
         [month ?month]
         [entity ?entity]]

   out= [[RELN DATE-MONTH-DAY]
         [day ?day]
         [month ?month]
         [entity ?entity]]
end")
```

１０．２．４ 補助動詞・接続詞の変換

・補助動詞の変換
補助動詞はモーダルに変換する場合と、縮退する場合がある。
縮退する場合には、表層のASPT素性を補助動詞とともに指定して、
的確なASPT素性に書き換える。

・日本語補助動詞から英語モーダルへの変換例

```
(rws:defrwschema2 GEN042 V なく
"on <reln> なくてはいけない-MUST  in :PHASE :J-E :TYPE :GENERAL
   ¥"なくてはいけない --> obliteration¥"
   in= [[reln なくてはいけない-MUST]
        [AGEN ?AGEN]
        [OBJE ?OBJE]
        ?rest]
   out= [[reln OBLIGATION-MUST]
         [AGEN ?AGEN]
         [OBJE ?OBJE]
         ?rest]
end")
```

・日本語の複層構造から英語モーダル助動詞への変換例

```
(rws:defrwschema2 GEN059 V よい
"on <reln> 良い-2  in :PHASE :J-E :TYPE :GENERAL
    in= [[RELN  良い-2]
         [OBJE [[PARM !X01[]]
                [RESTR [[RELN  ほう-1]
                        [IDEN [[RELN ?action]
                               [ASPT ?ASPT]
                               ?REST0]]
                        [ENTITY !X01]]]]]
         ?REST]
   if input.obje.restr.iden.agen    then
     set ?agen to input.obje.restr.iden.agen
   endif
   if ?agen =? [] then ?agen = root.prag.hearer
   endif
 set ?out to [[RELN HAD_BETTER-OBLIGATION]
              [AGEN ?agen]
              [OBJE [[reln ?action]
                     ?REST0]]
              ?REST]
     if ?out.infmann and
        ?out.infmann.restr.reln =? $むしろ-1 ;*<===強調の陳述副詞の縮退
         then    delete infmann from ?out
     endif
  out= ?out
end")
```

・日本語補助動詞を縮退してASPT素性を変更(1)
～しているところだ -> ～している

```
(rws:defrwschema2 GEN072 V とこ
"on <reln> ところだ-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[reln ところだ-1]
        [OBJE @OBJE
              [[RELN ?action]
               [ASPT PROG];*<=====ている
               ?rest0]]
        ?rest1]
  out=  [[RELN ?action]
         [ASPT PROG];*<=====現在進行
         ?rest0]
end")
```

・日本語補助動詞を縮退してASPT素性を変更（2）
　～したばかりだ -> ～してしまっている（現在の時点での結果）

```
(rws:defrwschema2 GEN074 V ばか
"on <reln> ばかりだ-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[reln ばかりだ-1]
        [OBJE @OBJE
              [[RELN ?action]
               [ASPT PAST] ;*<=====タ形
               ?rest0]]
  out=  [[RELN ?action]
         [ASPT RSLT] ;*<=====現在の時点での結果
         ?rest0]
end")
```

・接続詞の変換

接続詞や接続詞相当語句については、通常の副詞と同様にそのまま変換が行なわれ、接続詞マーカーを意味格中に示す。但し、出力英文で文頭に置かれて、先行する文（発話者が同じとは限らない）とのつながりを示すことから、談話処理では、前文の参照が必要となる場合がある。

N.B. 日本語の逆説の接続詞が、英語表現にすると強調され過ぎる場合がある。

・接続詞一語の変換例

```
"on <opps restr reln> けれど-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[OPPS [[PARM !X[]]
              [RESTR [[reln けれど-1];*<======逆説の表現
                      [IDEN ?IDEN]
                      [ENTITY !X]]]]] ?REST]
  out= [[OPPS [[RELN ALTHOUGH-CONJ-1];*<======逆説の接続詞
              [OBJE ?IDEN]]] ?REST]
```

・陳述副詞を伴った場合の接続詞への変換例

```
"on <cond restr reln> たら-CONDITIONAL in :PHASE :J-E :TYPE :GENERAL
   in= [[COND [[PARM !x[]]
              [RESTR [[RELN たら-CONDITIONAL];*<======条件の表現
                      [IDEN ?IDEN]
                      [ENTITY !x]]]
              [INFMANN @INFMANN<restr reln> もし-2]]];*<======仮定を強調する
        ?REST]                                                    陳述副詞
  out= [[COND [[RELN if-CONJ-1];*<======
              [OBJE ?IDEN]]]
        ?REST]
```

N.B. 次の文のように、電話会話では逆説の意味ではない OPPSについては英英変換で構造処理を行なっている。

>P-288 人工知能研究所の渡辺ですけれどまだ登録用紙は届いていませんか

```
"on <opps reln> ALTHOUGH-CONJ-1 in :PHASE :ENGLISH
  in= [[OPPS [[RELN ALTHOUGH-CONJ-1]
             [OBJE @OBJE<iden parm restr reln> name]]];*<======名前の提示
       ?REST]
    if input.opps.obje.obje =? []              then
     input.opps.obje.obje = [[parm !x[]]
                             [restr [[reln this-pron-1]
                                     [entity !x]]]]          endif
 out= [[CONJ [[RELN FIRST-SENT] ;*<======重文の最初の文というマーカー
             [OBJE @OBJE]]]
       ?REST]
```

生成結果　"This is Watanabe of the Artificial Intelligence Research Center, haven't you receive the registration form yet?"

１０.２.５ 名詞構造内部の素性

取り立て助詞(CONTR)や接頭辞(PREFIX)、接尾辞(SUFFIX)は、日本語構造で通常の名詞構造内部に表層に従って任意に現われる。こうした任意の素性は生成処理では規則化することが難しいため、名詞構造の外へ変換する必要がある。

・取り立て助詞(CONTR)の名詞構造外への変換

```
(rws:defrwschema2 GEN154 N だけ
"on <restr contr> だけ-P-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN ?reln]
                [CONTR だけ-P-1];*<======取り立て助詞
                [entity !X]
                ?REST]]
        ?REST0]

   out= [[PARM !Y[[PARM !X[]]
                  [RESTR [[RELN ?reln]
                          [entity !X]
                          ?REST]]
                  ?REST0]]
         [RESTR [[RELN ONLY-ADV-1];*<======名詞を修飾する副詞
                 [OBJE !Y]]]]
end")
```

・接尾辞(SUFFIX)の名詞構造外への変換

```
(rws:defrwschema2 GEN163 N め
"on <restr suffix> 目-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER]
                [SUFFIX 目-1];*<======接尾辞
                [UNIT 番];*<======単位を表わす
                [entity !X]
                ?REST]]
        ?REST0]
  out= [[PARM !X[]]
        [RESTR [[RELN ORDINARY-NUMBER];*<======序数を示す関係名
                [entity !X]
                ?REST]]
        ?REST0]
end")

(rws:defrwschema2 GEN167 N いじ
"on <restr suffix> 以上-1 in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER]
                [SUFFIX 以上-1];*<======接尾辞
                [entity !X]
                ?REST]]
        ?REST0]
   out= [[PARM !Y[[PARM !X[]]
                  [RESTR [[RELN NUMBER]
                          [entity !X]
                          ?REST]]
                  ?REST0]]
         [RESTR [[RELN MORE_THAN-ADV-1];*<======名詞修飾の副詞
                 [OBJE !Y]]]]
end")
```

・接頭辞(PREFIX)の名詞構造外への変換

```
(rws:defrwschema2 GEN168 N やく
"on <restr prefix> 約 in :PHASE :J-E :TYPE :PRE-GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER]
                [PREFIX 約];*<======接頭辞
                [entity !X]
                ?REST]]
        ?REST0]
   out= [[PARM !Y[[PARM !X[]]
                  [RESTR [[RELN NUMBER]
                          [entity !X]
                          ?REST]]
                  ?REST0]]
         [RESTR [[RELN ABOUT-ADV-1];*<======名詞修飾の副詞
                 [OBJE !Y]]]]
end")

(rws:defrwschema2 GEN170 N だい
"on <restr prefix> 第 in :PHASE :J-E :TYPE :GENERAL
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER];*<======数を表わす関係名
                [PREFIX 第] ;*<======接頭辞
                [entity !X]
                ?REST]] ?REST0]
  out= [[PARM !X[]]
        [RESTR [[RELN ORDINARY-NUMBER] ;*<======序数を示す関係名
                [entity !X]
                ?REST]] ?REST0]
end")
```

１０.２.６ 前置詞構造への変換

意味格は、そのまま生成処理に送られ、終端語に応じた前置詞が決定されるが、日本語の形式名詞などでさらに特定化されている場合は、前置詞マーカーに変換する。（等位構造や単数・複数も指定する必要がある）

N.B. 前置詞マーカーは概念化していない

```
(rws:defrwschema2 GEN177 N あい
"on <LOCT restr reln> 間-FN-LOCATION in :PHASE :J-E :TYPE :GENERAL
   in= [[LOCT [[PARM !x[]]
              [RESTR [[reln 間-FN-LOCATION];*<======形式名詞
                      [IDEN ?iden] ;*<==== 等位構造、または名詞複数
                      [entity !x]]]]]
        ?rest]
  out= [[LOCT [[RELN AMONG-PREP-1];*<======前置詞マーカー
              [OBJE ?iden]]]    ;*<====== ?iden 内の名詞構造は複数となる
        ?rest]
end")

(rws:defrwschema2 GEN174 N うえ
"on <loct parm restr reln> 上-LOCATION in :PHASE :J-E :TYPE :GENERAL
   in= [[LOCT [[PARM !Z[[PARM !x[]]
                        [RESTR [[reln 上-LOCATION] ;*<===場所格の特定
                                [entity !x]]]]]
              [RESTR [[RELN MODIFY]
                      [arg1 ?iden]
                      [arg2 !Z]]]]]
        ?rest]
  out= [[LOCT [[RELN ON-PREP-1] ;*<====== 前置詞マーカー
              [OBJE ?iden]]]
        ?rest]
end")
```

１０.２.７ RESPONSEの変換

『はい・いいえ』について
解析処理からの入力構造において、『はい』『いいえ』は、それぞれ「はい-AFFIRMATIVE」「いいえ-NEGATIVE」と肯定・否定の標識が既に付与されているが、談話処理（前文参照）や次文との結合処理を変換処理前で行なうとすると、前文の発話が QUESTIONIF（yes/no疑問文）で、否定疑問文か肯定疑問文かにかかわらず、英語表層では、次文が意味的に肯定ならyes,否定ならnoと、日本語表層と逆転する場合を考慮する必要がある。

・「はい-AFFIRMATIVE」が、次文と結合した場合の変換例

```
(rws:defrwschema2 GEN206 V はい
"on <conj reln> はい-AFFIRMATIVE  in :PHASE :J-E :TYPE :GENERAL
   in=[[RELN INFORM]
       [OBJE ?OBJE]  ;*<================ (a)
       [CONJ [[reln はい-AFFIRMATIVE]
              [AGEN ?AGEN]
              [RECP ?RECP]
              [ASPT ?aspt]
              ?rest]]
       ?rest1]

  out=[[RELN INFORM]
       [OBJE ?OBJE]
       [CONJ [[reln YES-AFFIRMATIVE]
              [AGEN ?AGEN]
              [RECP ?RECP]
              [ASPT ?aspt]
              ?rest]]
       ?rest1]
end")
```

(a)の ?obje は、『はい』につづく次文の内容を示す。
?obje 内の意味構造が、否定か肯定かを判断する必要がある。

・RESPONSEの単純な変換例

```
(rws:defrwschema2 GEN207 V はい
"on <obje reln> はい-AFFIRMATIVE  in :PHASE :J-E :TYPE :GENERAL
   in=[[RELN RESPONSE]
       [OBJE [[reln はい-AFFIRMATIVE]
              [AGEN ?AGEN]
              [RECP ?RECP]
              [ASPT ?aspt]
              ?rest]]
       ?rest1]

  out=[[RELN RESPONSE]
       [OBJE [[reln YES-AFFIRMATIVE]
              [AGEN ?AGEN]
              [RECP ?RECP]
              [ASPT ?aspt]
              ?rest]]
       ?rest1]
end")
```

１０.２.８ 意志未来

未来は、以下のように説明される。
・単純未来（無意志未来） Non-volitional Future
・意志未来 Volitional Future

『つもりだ-INTENTION』『よう-INTENTION』で表される意志は、
主語(EXPR)の意志を示し、
1. 主語(EXPR)が、１人称([LABEL *SPEAKER*])の場合は、
助動詞 WILL、"intend to" "be determined to"で示され、
2. 主語(EXPR)が、２人称、３人称の場合は、
"intend to" などで示される。

主語(EXPR)の意志は、共通して "intend to"で表すことができるが、
主語(EXPR)が、１人称の場合は、英語側で、意志未来とし、
助動詞 WILL を使えるようにする。

Ex. 私は～するつもりだ。 I will go.
I intend to go.
I am determined to go.
君は～するつもりだ。 You intend to go.
彼は～するつもりだ。 He intends to go.

```
▽ 解析==『つもりだ-INTENTION』    『よう-INTENTION』
          〔EXPR ?expr〕            〔EXPR ?expr〕
                          ↓
▽ 変換==      〔reln SUBJECT-VOLITION〕
              〔EXPR ?expr〕
                          ↓
▽ 生成==   ?expr = 〔〔LABEL *SPEAKER*〕〕   → (I) will ...
           ?expr ≠ 〔〔LABEL *SPEAKER*〕〕   → (?expr) intend to ...
```

N.B.解析で、『つもりだ-INTENTION』に先行する日本語動詞の ASPT は、
PASTであってはならない。
「～したつもりだ」---->意志を表さない。

```
if input.obje.aspt and input.obje.aspt =? $past
      then fail
```

N.B. （～しているつもりだ）についても、考慮する必要がある。
『彼は本を読んでいるつもりだ』 He thinks he is reading a book.
N.B.生成で、will に続く英語動詞は、意志動詞(Voltional Verb)で
なければならない。

```
(rws:defrwschema2 GEN059 V つも
"on <reln> つもりだ-INTENTION in :PHASE :J-E :TYPE :GENERAL
   in= 〔〔reln つもりだ-INTENTION〕
        〔EXPR ?expr〕
        〔OBJE ?obje〕
        〔ASPT ?aspt〕
        ?rest〕

    if input.obje.aspt and input.obje.aspt =? $past
          then fail
     else delete aspt from ?obje
    endif

   out= 〔〔reln SUBJECT-VOLITION〕
        〔EXPR ?expr〕
        〔OBJE ?obje〕
        〔ASPT ?aspt〕
        ?rest〕
end")
```

『つもりだ-INTENTION』 から、『SUBJECT-VOLITION』への変換

```
        >P-243 参加するつもりです

[[SEM !X05[[RELN  つもりだ-INTENTION] ;*<======
           [EXPR !X03[[LABEL  *SPEAKER*]]]
           [OBJE !X14[[RELN  参加する-1]
                      [AGEN !X03]
                      [LOCT !X15[]]
                      [ASPT  UNRL]]]
           [ASPT  STAT]]]
 [PRAG !X08[[RESTR [[IN !X20[[FIRST [[RELN  POLITE]
                                     [AGEN !X03]
                                     [RECP !X02[[LABEL  *HEARER*]]]]]
                              [REST !X01[]]]]
                    [OUT !X01]]]
            [SPEAKER !X03]
            [HEARER !X02]
            [ASPE [[IN !X23[]]
                  [OUT !X52[]]]]
            [PRSP-TERMS [[IN !X22[]]
                         [OUT !X51[]]]]
            [TOPIC [[IN !X21[]]
                    [OUT !X50[]]]]]]]

;;;=========== Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X1[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [OBJE [[RELN SUBJECT-VOLITION] ;*<======
               [ASPT STAT]
               [TENSE PRESENT]
               [EXPR !X1]
               [OBJE [[RELN ATTEND-V-1]
                      [AGEN !X1]
                      [LOCT []]]]
               [SEM-ASPE STATIVE]]]]
  [PRAG [[BENEFIT NONE]
         [HEARER !X2]
         [INTENTION INFORM]
         [POLITENESS [[DEGREE 1]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X1]
         [TOPIC [[TOPIC-MOD NULL]]]]]]
```

生成結果

```
        >P-243 参加するつもりです
              "I'll attend."
```

１０.３ デフォルト変換

デフォルト変換は、書き換え環境を３つに分け、まず動詞・形容詞などの
述語変換を行ない、次に普通名詞・固有名詞・副詞などの変換に移る。
固有名詞のうち、姓名の名前の変換は、後に述べる理由で、最後に独立した
書き換え環境で変換される。

１０.３.１ 動詞のデフォルト変換

・格要素を変更しない場合

```
(rws:defrwschema2 DEFV060 V おと
"on <reln> 訪れる-1  in :PHASE :J-E :TYPE :default-1
   in= [[reln 訪れる-1]
        [AGEN ?AGEN]
        [OBJE ?OBJE]
        ?rest]
   out= [[reln visit-V-1]
         [AGEN ?AGEN]
         [OBJE ?OBJE]
         ?rest]
end")
```

・空の格要素を縮退した場合

```
(rws:defrwschema2 DEFV037 V いら
"on <reln> 依頼する-1 in :PHASE :J-E :TYPE :default-1
   in= [[reln 依頼する-1]
        [AGEN ?AGEN]
        [RECP ?RECP]
        [OBJE ?OBJE]
        ?rest]
  switch ?recp
   case []
   out= [[reln ASK-V-5]
         [AGEN ?AGEN]
         [OBJE ?OBJE]
         ?rest]
   default
   out= [[reln ASK-V-6]
         [AGEN ?AGEN]
         [RECP ?RECP]
         [OBJE ?OBJE]
         ?rest]
   endswitch
end")
```

・必須格と任意格について

(a) 解析部で動詞が必須格と共に定義されるが、変換部では、どの格が
　　必須格かが任意格と区別がつかない。
(b) 複数の同じ格要素は持てない。
(c) 意味的には同じ文であるが、助詞の違いでまったく別の意味格で
　　解析されている。

```
>db-3 会議に参加する
       [[RELN  参加する-1]
        [AGEN !X01]
        [LOCT !X70[[PARM !X02[]]
                  [RESTR [[RELN  会議-1]
                          [ENTITY !X02]]]]]]
```

例えば、『参加する-1』は、主語のAGENと、「に格」のLOCTを必須格として
認定されるが、この時、既に任意格の「で格」のLOCTを受け入れる余地はない。
従って、 「私は パリで 会議に 参加する」は、現在、解析できない。(b)

```
 AGEN  LOCT?  LOCT
```

『参加する-1』のデフォルトルールは以下の通りである。

```
(rws:defrwschema2 DEFV128 V さん
"on <reln> 参加する-1  in :PHASE :J-E :TYPE :default-1
   in= [[reln 参加する-1]
        [agen ?AGEN]  ;*<----- 必須格 AGEN
        [LOCT ?LOCT]  ;*<----- 必須格 LOCT
        ?rest]        ;*<===== その他、任意格 (a)
   out= [[reln ATTEND-V-1]
         [agen ?agen]
         [LOCT ?LOCT]
         ?rest]
end")
```

変換後の構造の構造においても、LOCTがATTEND-V-1の必須格としている。

```
          [[RELN ATTEND-V-1]
           [AGEN !X1]
           [LOCT [[PARM !X2[]]
                 [RESTR [[RELN CONFERENCE-N-1]
                         [ENTITY !X2]]]]]]
```

解析レベルで、必須格と任意格の格名を、動詞によらず、特定できるようにする(?)。
『参加する-1』の日本語構造において、LOCTが必須格であることは、動詞である『参加する-1』が規定しているわけであるから、必須格の特定は、特に必要ないように思われる。
しかし、変換部では、動詞を参照せずに、任意格のみを処理するような場合がある。(a)

変換部で、必須格・任意格を特定する。
以下のルールのように、変換と同時に必須格を認定する。

```
"on <reln> 参加する-1  in :PHASE :J-E :TYPE :default-1
   in= [[reln 参加する-1]
        [AGEN ?AGEN]
        [LOCT ?LOCT]
        ?rest]
```

A.

```
  out= [[reln ATTEND-V-1]
        [AGEN-FIX ?AGEN]
        [LOCT-FIX ?LOCT]
        ?rest]
```

B.

```
  out= [[reln ATTEND-V-1]
        [subject ?AGEN]
        [object ?LOCT]
        ?rest]
```

この場合、英語側の動詞 ATTEND-V-1の必須格は、この時点で決まってしまう。
現在の英語構造の表記は、そのまま英単語として生成するので、これでもよいが、実際は、ATTEND-V-1 は概念表記であるので、ATTEND-V-1 と同じ語義で、別の動詞を生成しようとする場合、その動詞の必須格が、"attend"の同じ必須格とは限らない。
英語の概念表記としては、解析の格認定を引きずらない B. の方がよいかもしれない。

日本語構造の格認定で、動詞『構う-1』の必須格に、AGEN,CONDをあてているが、任意格の COND格と区別するために、動詞のRELNも参照しなければならない。(a)

```
>P-171 ;  (AGEN[])   |銀行振込でも (COND) |  構いません
>P-270 ; 招待者への連絡 (AGEN) は|  明日でも (COND) |  構いません
>P-477 ; 値段は (AGEN) |  いくらでも (COND) |  構いません
```

(c)の問題については、JJ変換部のところで、述べたが、
変換部のルールで、ある程度、相応しい意味格へ変更している。
(但し、変更先の新しい意味格を既に構造が持っているかどうかはルール内で、確認する必要がある。(b)のため)

１０.３.２ 形容詞・限定詞などのデフォルト変換

・形容詞の基本デフォルト変換

```
(rws:defrwschema2 DEFV381 V あた
"on <reln> 新しい-1  in :PHASE :J-E :TYPE :default-1
   in= [[reln 新しい-1]
        [OBJE ?OBJE]
        ?rest]
   out= [[reln NEW-ADJ-1]
         [OBJE ?OBJE]
         ?rest]
end")
```

・名詞＋MODIFYの基本デフォルト変換

```
(rws:defrwschema2 DEFV475 ADJ つう
"on <arg1 restr reln> 通常-1  in :PHASE :J-E :TYPE :default-1
    in= [[RELN MODIFY]
         [ARG1 [[PARM !x[]]
                [restr [[reln 通常-1]
                        [entity !x]]]]]
         [ARG2 ?arg-2]]
   out= [[RELN USUAL-ADJ-1]
         [OBJE ?arg-2]]
end")
```

・限定詞の基本デフォルト変換

```
(rws:defrwschema2 DEFV428 ADJ こん
"on <reln> こんな-1  in :PHASE :J-E :TYPE :default-1
   in= [[reln こんな-1]
        [OBJE ?OBJE]
        ?rest]
   out= [[reln such-DET-1]
         [OBJE ?OBJE]
         ?rest]
end")
```

・限定詞(敬語表現)のデフォルト変換

```
(rws:defrwschema2 DEFN418 N きょ
"on <restr honorific-title> 教授-2 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN NAME]
                [FAMILY-NAME ?name]
                [ENTITY !X]
                [HONORIFIC-TITLE 教授-2]
                ?REST]]
        ?REST-INDEX]

  out= [[PARM !Z[[PARM !X[]]
                 [RESTR [[RELN NAME]
                         [FAMILY-NAME ?name]
                         [ENTITY !X]
                         ?REST]]
                 ?REST-INDEX]]
        [RESTR [[RELN PROFESSOR-DET-1]
                [OBJE !Z]]]]
end")
```

１０.３.３ 名詞などのデフォルト変換

まず、一般的な普通名詞の例と固有名詞の変換例をあげる。いずれも、デフォルト変換で、INDEX素性を付与することにする。

・普通名詞のデフォルト変換

```
(rws:defrwschema2 DEFN024 N えき
"on <restr reln> 駅-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN 駅-1]
                [ENTITY !X]]]]
   out= [[PARM !X[]]
         [RESTR [[RELN station-N-1]
                 [ENTITY !X]]]
         [INDEX [[NUMBER SING-PL]
                 [GENDER NONE]
                 [DEFINT INDEF]
                 [PERSON THIRD]]]]
end")
```

・固有名詞のデフォルト変換

```
(rws:defrwschema2 DEFN484 N ろい
"on <restr iden> ロイヤルホテル-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN NAMED]
                [IDEN ロイヤルホテル-1]
                [ENTITY !X]]]]
   out= [[PARM !X[]]
         [RESTR [[RELN NAMED]
                 [IDEN Royal_Hotel-PROPN]
                 [ENTITY !X]]]
         [INDEX [[NUMBER SING]
                 [GENDER NONE]
                 [DEFINT DEFN]
                 [PERSON THIRD]]]]
end")
```

固有名詞の中でも地名を表す入力構造は、２種類あるので、デフォルト変換規則も、住所用の地名の変換ルールと固有名詞タイプの地名の変換ルールの２通りを作成している。

(a) 固有名詞タイプの地名の入力構造（通常の固有名詞として動作する）

Ex. 京都へ行く

```
[[RELN  行く-1]
 [AGEN []]
 [DEST [[PARM !X[]]
        [RESTR [[RELN  NAMED]
                [IDEN  京都]
                [ENTITY !X]]]]]]]]]
```

(b) 住所用の地名の入力構造
（解析で決定された素性名に従ったルールが必要となる）

Ex. 京都府精華町光台

```
[[PARM !X[]]
 [RESTR [[RELN  NAMED]
         [IDEN [[RELN  ADDRESS]
                [AZA1  光台]
                [PREF  京都-1]
                [TOWN  精華町]]]
         [ENTITY !X]]]]
```

(a)の入力構造に対しては、通常の固有名詞と同じルールがかかる。

```
(rws:defrwschema2 DEFN465 N きょ
"on <restr iden> 京都 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN NAMED]
                [IDEN 京都]
                [ENTITY !X]]]]
   out= [[PARM !X[]]
         [RESTR [[RELN NAMED]
                 [IDEN Kyoto-PROPN]
                 [ENTITY !X]]]
         [INDEX [[NUMBER SING]
                 [GENDER NONE]
                 [DEFINT DEFN]
                 [PERSON THIRD]]]]
end")
```

(b)の入力構造に対しては、ADDRESSのレベルにあるPREF,CITY,TOWN,
WARDなどの住所単位ごとのルールが必要となる。

```
(rws:defrwschema2 DEFN447 N きょ
"on <restr iden pref> 京都-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN NAMED]
                [IDEN [[RELN ADDRESS]
                       [PREF 京都-1]
                       ?rest]]
                [ENTITY !X]]]]

   out= [[PARM !X[]]
         [RESTR [[RELN NAMED]
                 [IDEN [[RELN ADDRESS]
                        [PREF KYOTO-PROPN]
                        ?rest]]
                 [ENTITY !X]]]]
end")
```

１０.３.４ INDEX素性について

INDEX素性は、本来は文脈情報などを参照する書き換え環境で、
付与すべきであるが、とりあえず名詞のデフォルト変換部で付ける。

このINDEX素性は,国際会議に関する電話での問い合わせというコーパスを
前提として用いられる.

1. INDEX素性のある項目
   普通名詞
   固有名詞(ファミリーネーム,ファーストネーム)
   固有名詞(地名)

以下の項目にはINDEX素性は与えていない.
敬称
住所用地名
数詞(日付)
名詞(月)
単位名詞
数詞
POD

2 素性名とその値
素性はNUMBER,GENDER,DEFINT,PERSONの4種類がある.
素性...NUMBER
値...SING　単数形しかありえないもの,または数えられない名詞
PL　複数形しかありえないもの
SING-PL　単数形と複数形の両方がありうるもの

素性...GENDER
値...NEUT　男性の場合も女性の場合もある
(代名詞で受ける場合HEまたはSHE)
MALE　男性の場合しかない(代名詞で受ける場合HE)
FEMA　女性の場合しかない(代名詞で受ける場合SHE)
NONE　男性でも女性でもない(代名詞で受ける場合IT)

・普通名詞ではMALEはなく,FEMAは女性-1,妻-1の2つだけ.
・ファミリーネームはすべてNEUT.
・ファーストネームはMALEとFEMAに分かれる.

素性...DEFINT
値...DEFN　固有名詞,またはこのコーパス内で特定的なもの
INDEF　一般的なもの

・普通名詞でDEFNのもの...アブストラクト-1,英語-1,オープニングセッション-1,
開会式-1,会議場-1,会場-1,学会-1,歓迎会-1,基調講演-1,記念式典-1,
言語学研究室-1,国際会議-1,コングレスキット-1,招待講演-1,セッション-1,
組織委員長-1,組織委員会-1,日本語-1,フェアウェルパーティー-1,
プログラム委員会-1,プロシーディングス類-1,閉会式-1,予稿集-1(23個)

素性...PERSON
値...FIRST　1人称
SECOND　2人称
THIRD　3人称

生成部において、英文出力の際に、決定できないものに、数と冠詞の問題がある。

例えば、『講演者』の英訳には、単数・複数の両方が有り得る。

また、『講演者』の対訳である"speaker"を出す生成ルール(PD)には、
単数・複数の両方を想定して、〔number (:SET sing pl)〕と書かれてある。

従って、ほとんどの名詞については、数を決定できず、"sing" "pl"の
両方の値を持ったまま、生成が行なわれるが、両方の可能性がある場合は、
プログラムで、最初の値のみを選択するようになっている。

次に、冠詞の場合であるが、"speaker"を出す生成ルール(PD)に、
不定冠詞・定冠詞の両方を想定して、〔arti (:SET a the)〕と書かれてある。
ただ、冠詞の場合は、"a" "the" を出すルールが異なるため、
両方の場合の英文が生成される。
(但し、生成で冠詞を絞り込んでいる場合がある)
(疑問文・THERE構文など。関係節は絞り込んでいない)

p-96 『会議にはいろいろな国から講演者がいらっしゃいます』の結果は
以下のようになる。

"A speaker will come to the conference from various countries."
"A speaker will come to the conference from the various countries."
"A speaker will come to a conference from various countries."
"A speaker will come to a conference from the various countries."
"The speaker will come to the conference from various countries."
"The speaker will come to the conference from the various countries."
"The speaker will come to a conference from various countries."
"The speaker will come to a conference from the various countries."

しかし、『会議にはいろいろな国から講演者がいらっしゃいます』の
日本語文章では、『いろいろな国から』というフレーズにより、
主語の数が複数であることがわかる。

同様の例としては、
p-108 『会議の冒頭で首相のメッセージが読まれます。』が、
『会議の冒頭で（各国の）首相のメッセージが読まれます。』のように
（各国の）があった場合は、"message" は複数となる。

これは、文脈、発話状況から判断しなければならないので、
個々の単語単位で見ても、しかたがない。

そこで、変換部で、文脈を参照することを想定し、
変換部で、数や定・不定を決定し、INDEX素性として、生成部に送る。
（但し、変換部でも今のところ、単語単位で付加している）

N.B.
主語が特定された（前文脈に既出）場合
『（その）講演者は会議にはいろいろな国からいらっしゃいます』は、
ある講演者が、世界各地を飛回っていて、会議たびにという読みから、
『いろいろな国から』というフレーズがあるが、単数である。

問題点
・数や定・不定が、変換部で事実上決定されることになれば、
生成部での生成英文に合わせた自由度が狭くなる。

・姓名などで、姓と名のINDEX素性が、重複する。

『鈴木真弓』は以下の構造で、表される。

```
〔〔PARM !X〕〕
 〔RESTR 〔〔RELN  NAME〕
          〔ENTITY !X〕
          〔FIRST-NAME  真弓-1〕
          〔FAMILY-NAME  鈴木-1〕〕〕〕
```

この構造に対して、以下のA.B.の変換ルールがかかるが、
INDEX素性が同じ位置に重複してしまうかたちとなり、
INDEX素性は、後にかかったルールが残ることになる。
（この場合、GENDERの値が異なる）

これに対処するために、姓と名で、姓を先にかかるように、
書き換え環境を設定しているが、FAMILT-NAMEとFIRST-NAMEの構造を
前もって分割し、別々のINDEX素性を付加できるようにするなどが
考えられる。

A.『鈴木-1』のdefault変換ルール

```
"on <restr family-name> 鈴木-1 in :PHASE :J-E :TYPE :default-2
   in=  ([PARM !X[]]
         [RESTR ([RELN NAME]
                 [FAMILY-NAME 鈴木-1]
                 [ENTITY !X]
                 ?REST])
          ?REST-INDEX]

   out=  ([PARM !X[]]
          [RESTR ([RELN NAME]
                  [FAMILY-NAME SUZUKI-PROPN]
                  [ENTITY !X]
                  ?REST])
          [INDEX ([NUMBER SING]
                  [GENDER NEUT]
                  [DEFINT DEFN]
                  [PERSON THIRD])]
         ?REST-INDEX]
```

B.『真弓-1』のdefault変換ルール

```
"on <restr first-name> 真弓-1 in :PHASE :J-E :TYPE :default-3
   in= ([PARM !X[]]
        [RESTR ([RELN name]
                [FIRST-NAME 真弓-1]
                [ENTITY !X]
                ?REST])
        ?REST-INDEX]

   out= ([PARM !X[]]
         [RESTR ([RELN name]
                 [FIRST-NAME MAYUMI-PROPN]
                 [ENTITY !X]
                 ?REST])
         [INDEX ([NUMBER SING]
                 [GENDER FEMA]
                 [DEFINT DEFN]
                 [PERSON THIRD])]
         ?REST-INDEX]
```

１０.３.５ その他の名詞および副詞のデフォルト変換

・日付のデフォルト変換

```
(rws:defrwschema2 DEFN485 N つい
"on <restr day> 一日-1 in :PHASE :J-E :TYPE :default-2
   in=  ([PARM !X[]]
         [RESTR ([reln ?DATE]
                 [DAY 一日-1]
                 [ENTITY !X]
                 ?REST])
         ?REST-INDEX]

   out=  ([PARM !X[]]
          [RESTR ([reln ?DATE]
                  [DAY 1st-DATE]
                  [ENTITY !X]
                  ?REST])
          ?REST-INDEX]
end")
```

・月名のデフォルト変換

```
(rws:defrwschema2 DEFN516 N いち
"on <restr month> 一月-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[reln ?DATE]
                [MONTH 一月-1]
                [ENTITY !X]
                ?REST]]]

   out= [[PARM !X[]]
         [RESTR [[reln ?DATE]
                 [MONTH JANUARY-N-1]
                 [ENTITY !X]
                 ?REST]]]
end")
```

・午前・午後などPODのデフォルト変換

```
(rws:defrwschema2 DEFN569 N ごぜ
"on <restr pod> 午前-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !x[]]
        [RESTR [[RELN TIME]
                [POD 午前-1]
                [ENTITY !x]]]]

  out= [[PARM !x[]]
        [RESTR [[RELN morning-N-1]
                [ENTITY !x]]]]
end")
```

・単位名詞のデフォルト変換

```
(rws:defrwschema2 DEFN528 N えん
"on <restr unit> 円 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER]
                [IDEN ?iden]
                [UNIT 円]
                [ENTITY !X]
                ?rest]]]

   out= [[PARM !X[[PARM !Y[]]
                  [RESTR [[RELN yen-N-1]
                          [ENTITY !Y]]]]]
         [RESTR [[RELN MODIFY]
                 [ARG1 [[PARM !Z[]]
                        [RESTR [[RELN NUMBER]
                                [IDEN ?iden]
                                [ENTITY !Z]
                                ?rest]]]]
                 [ARG2 !X]]]]
end")
```

・一般副詞のデフォルト変換

```
(rws:defrwschema2 DEFN652 ADV すで
"on <restr reln> 既に-1 in :PHASE :J-E :TYPE :default-2
   in= [[PARM !X[]]
        [RESTR [[reln 既に-1]
                [ENTITY !X]]]]
   out= [[PARM !X[]]
         [RESTR [[reln already-ADV-1]
                 [ENTITY !X]]]]
end")
```

１０.３.６ 数詞の変換

・数詞
入力構造では、数の処理は、PLACE-XXX の素性名で、桁ごとの漢数字を用いた構造となっている。変換では、そのままアラビア数字に置き換え、空の値には、0-NUMBER を代入する。

```
>P-23 委員会には十一人のメンバーがいる

   ([PARM !X03([PARM !X01()]
               [RESTR ([RELN  メンバー-1]
                       [ENTITY !X01])])]
    [RESTR ([RELN  の-連体修飾]
            [IDEN !X31([PARM !X02()]
                       [RESTR ([RELN  NUMBER]
                               [UNIT  人]
                               [IDEN !X58([PLACE-1  一-1]   ;*<-----(a)
                                          [PLACE-10  一-1])]  ;*<-----(b)
                               [ENTITY !X02])])])
            [OBJE !X03])])]
;;;=========== Transfer Result ============
   ([PARM !X3([PARM !X2()]
              [RESTR ([RELN MEMBER-N-1]
                      [ENTITY !X2])])]
    [RESTR ([RELN MODIFY]
            [ARG1 ([PARM !X1()]
                   [RESTR ([RELN NUMBER]
                           [IDEN ([COLUMN-1 1-NUMBER] ;*<-----(a)
                                  [COLUMN-10 1-NUMBER])] ;*<-----(b)
                           [ENTITY !X1]
                           [UNIT PERSON])])])
            [ARG2 !X3])])]
```

規則は、他に効率的な方法があるかもしれないが、現在は、switch文の入れ子で作成している。（序数も同様に行なう）

```
(rws:defrwschema2 DEFN555 N 1
"on <restr reln> NUMBER in :PHASE :J-E :TYPE :default-2
in=  ([PARM !X()]
      [RESTR ([reln NUMBER]
              [IDEN ?IDEN]
              [ENTITY !X]
              ?REST)])

if ?iden has PLACE-100 then
switch input.restr.iden.place-100
 case $一-1
input.restr.iden.column-100 = $1-NUMBER
delete place-100 from input.restr.iden
       …
 case $九-1
input.restr.iden.column-100 = $9-NUMBER
delete place-100 from input.restr.iden

default input.restr.iden.column-100 = $0-NUMBER
delete place-100 from input.restr.iden
endswitch
       …
       …
       …
endif
endif
  RETURN INPUT
end")
```

・棒読み型の数詞

```
(rws:defrwschema2 DEFN557 N 1
"on <restr reln> 棒読み型の数詞 in :PHASE :J-E :TYPE :default-2
   in=  [[PARM !X[]]
         [RESTR [[RELN 棒読み型の数詞]
                 [DIGITS [[IN ?IN]
                          [OUT ?OUT]]]
                 [ENTITY !X]]]]
  out=  [[PARM !X[]]
         [RESTR [[RELN NUMBER-SEQ]
                 [DIGITS [[IN ?IN]
                          [OUT ?OUT]]]
                 [ENTITY !X]]]]
end")
```

規則は、〇から九まで、すべて作成する。

```
(rws:defrwschema2 DEFN558 N 1
"on <first> 〇-1 in :PHASE :J-E :TYPE :default-2
   in= [[FIRST 〇-1]
        ?REST]
  out= [[FIRST zero-digit-1]
        ?REST]
end")

(rws:defrwschema2 DEFN558 N 1
"on <first> 一-1 in :PHASE :J-E :TYPE :default-2
   in= [[FIRST 一-1]
        ?REST]
  out= [[FIRST one-digit-1]
        ?REST]
end")
```

この時点では、まだ、FIRST-RESTの入れ子構造であり、英英変換の
次のルールで、生成処理部が扱いやすい適当な構造に変換している。

```
(rws:defrwschema2 MISC006 N B
"on <restr reln> NUMBER-SEQ in :PHASE :ENGLISH
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER-SEQ]
                [DIGITS [[IN [[FIRST ?first]
                              [REST [[FIRST ?second]
                                     [REST [[FIRST ?third]
                                            [REST [[FIRST ?last]
                                                   [REST !Z[]]]]]]]]]]
                         [OUT !Z]]]
                [ENTITY !X]]]]

  out= [[PARM !X[]]
        [RESTR [[RELN NUMBER-SEQ]
                [DIGITS [[FIRST ?first]
                         [SECOND ?second]
                         [THIRD ?third]
                         [LAST ?last]]]
                [ENTITY !X]]]]

end")
```

１０.３.７ 多義性（訳し分け）

多義性の問題は、解析レベルで、語義番号により解決されているものとするが、難しいものについては、述語との共起で、訳し分けを行なっている。現在、形容詞と被修飾語、副詞と述語、名詞と述語などの共起のルールがあるが、タイプを使うことにより、単語単位の指定から、意味素性を使った共起に今後は移行する。

Ex. 動詞との共起による名詞の訳し分け

学会-1 → 通常 conference-N-1 （学会に出席する） 会議の意味
institute-N-1 （学会に問い合わせる） 団体の意味

```
(rws:defrwschema2 DEFN048 N がっ
"on <restr reln> 学会-1 in :PHASE :J-E :TYPE :default
   in= ([PARM !X()]
        [RESTR ([RELN 学会-1]
                [ENTITY !X])])
   out= ([PARM !X()]
         [RESTR ([RELN conference-N-1]
                 [ENTITY !X])]
         [INDEX ([NUMBER SING-PL]
                 [GENDER NONE]
                 [DEFINT DEFN]
                 [PERSON THIRD])])
end")

(rws:defrwschema2 GEN299 V とい
"on <goal restr reln> 学会-1 in :PHASE :J-E :Type :GENERAL
   in=        ([reln 問い合わせる-1]
               [AGEN ?AGEN]
               [GOAL ([PARM !X()]
                      [RESTR ([RELN 学会-1]
                              [ENTITY !X])])]
               ?rest)
  out=        ([reln 問い合わせる-1]
               [AGEN ?AGEN]
               [GOAL ([PARM !X()]
                      [RESTR ([RELN INSTITUTE-N-1]
                              [ENTITY !X])])]
               ?rest)
end")
```

同じ語義であっても、格要素の違いや必須格が空であるかなどで、生成部で、訳し分けを行なうことができる。

Ex. 必須格が空の場合の動詞の訳し分け（生成）

```
CONTACT-V-1   AGEN(I)  RECP(YOU)  ->  "I contact you."
CONTACT-V-1   AGEN(I)  RECP(   )  ->  "I make a contact."
```

N.B. 動詞の英語語義番号は同じである。

１１ 英英変換

デフォルトの変換部を経て、英英変換に入る素性構造は、既に英語の素性構造となっているため、ここで行なわれる変換処理を英英変換としている。
（但し、逐次的な変換の結果、英語構造からみて不適当な構造に対して、構造変換を行なう場合もある -> 生成のため）

・序数詞の関係名の指定

```
(rws:defrwschema2 MISC005 N B
"on <restr unit> ORDINARY-NUMBER in :PHASE :ENGLISH
   in= [[PARM !X[]]
        [RESTR [[RELN NUMBER]
                [IDEN ?IDEN]
                [UNIT ORDINARY-NUMBER]
                [ENTITY !X]
                ?rest]]]

  out= [[PARM !X[]]
        [RESTR [[RELN ORDINARY-NUMBER]
                [IDEN ?IDEN]
                [ENTITY !X]
                ?rest]]]
end")
```

・NEGATEレベルにある任意格などの動詞レベルへの移動

```
(rws:defrwschema2 MISC016 N xx
"on <reln> negate in :PHASE :ENGLISH
   in= [[RELN NEGATE]
        [ASPT ?aspt]
        [OBJE ?obje]
        ?rest]
   add ?rest to ?obje
  out= [[RELN NEGATE]
        [ASPT ?aspt]
        [OBJE ?obje]]
end")
```

◇ 係り受けの構造変換

・連体修飾の名詞句全体を修飾する形容詞の構造変換

```
(rws:defrwschema2 MISC008 N xx
"on <parm restr arg1 restr reln> money-n-1 in :PHASE :ENGLISH
   in= [[PARM !X[[PARM !Y[[PARM !Z[]]
                          [RESTR [[RELN ?reln]
                                  [ENTITY !Z]]]
                          ?rest]]
                 [RESTR [[RELN ?MODIFY]
                         [ARG1 ?arg1]
                         [ARG2 !Y]]]]]
        [RESTR [[RELN ?adjective]
                [OBJE !X]
                ?rest-adj]]]
  out= [[PARM !X[[PARM !Y[[PARM !Z[]]
                          [RESTR [[RELN ?reln]
                                  [ENTITY !Z]]]
                          ?rest]]
                 [RESTR [[RELN ?adjective]
                         [OBJE !Y]
                         ?rest-adj]]]
        [RESTR [[RELN ?MODIFY]
                [ARG1 ?arg1]
                [ARG2 !X]]]]
end")
```

・『三枚の原稿用紙』の生成出力のための構造指定

```
(rws:defrwschema2 MISC013 N number
"on <restr reln> modify in :PHASE :ENGLISH
   in= [[PARM !Y @NOUN
              [[PARM !Z[]]
               [RESTR [[RELN ?reln]
                       [ENTITY !Z]]]
               ?rest]]
        [RESTR [[RELN MODIFY]
                [ARG1 @ARG1<parm restr reln> piece-N-1]
                [ARG2 !Y]]]]

  out= [[PARM !Y @NOUN]
        [RESTR [[RELN MODIFY-UNIT]
                [ARG1 @ARG1]
                [ARG2 !Y]]]]
end")
```

・NAME が、姓名の両方を持つことを示す関係名の指定（生成用）

```
(rws:defrwschema2 MISC020 N X
"on <reln> name in :PHASE :INDEX-REMOVED
    in= [[RELN NAME]
         [FAMILY-NAME ?family]
         [FIRST-NAME  ?first]
         [entity ?entity]]
   out= [[RELN NAME-FULL]
         [FAMILY-NAME ?family]
         [FIRST-NAME  ?first]
         [entity ?entity]]
end")
```

１２ テンス・アスペクト

テンスとアスペクトの処理は、基本的には、入力構造にある ASPTの値から、
処理テーブルに従って、TENSEとSEM-ASPEの素性を導入する。
ASPTは、その時点で不要となるが、削除せずに生成へ送っている。

・英語アスペクト初期化　(ASPTの値を述語参照で変更)
　　(SEM with :PHASE :ASPECT-INIT by :RECURSIVE)
・英語アスペクト変更　(従属節内のASPTの値を主節に対応させる)
　　(SEM with :PHASE :ASPECT-CHANGE by :RECURSIVE)
・英語アスペクト指定　(ASPT変換テーブルの規則化)
　　(SEM with :PHASE :ASPECT by :RECURSIVE)

```
(rws:defrwschema2 ASPECT017 aspt pstat
"on <aspt> pstat in :PHASE :ASPECT
     in= [(aspt pstat)
          ?rest)
   input.tense = $past
   input.sem-aspe = $STATIVE
     return input
end")
```

テンス・アスペクトの処理テーブル

ASPT	TENSE	SEM-ASPE
UNRL	FUTURE	UNREAL
EXPR	PRESENT	PERF-EXPR
STAT PSTAT	PRESENT PAST	STATIVE STATIVE
PROG	PRESENT	CONTINUOUS
	+だろう FUTURE	CONTINUOUS
	+ASPECT副詞 (以来、ずっと) PRESENT	PERF-PROG
PPROG	PAST	CONTINUOUS
	+ASPECT副詞 (以来、ずっと) PAST	PERF-PROG
PAST	PAST	(SIMPLE)
	+ASPECT副詞 (今、ちょうど) PRESENT	PERF
RSLT	PRESENT	PERF
PRSLT	PAST	(SIMPLE)
ITER	PRESENT	CONTINUOUS
-	PRESENT	(SIMPLE)

・解析で指定されるアスペクト(ASPT)の種類と意味
（日本語の終止形、連体形、命令形に対して付与される）

〔〔aspt stat〕〕　　現在の状態を表す　stat-chng+
・このセッションは四時までです
〔〔aspt expr〕〕　　（知覚動詞の）現在　mome-chng-
・1983年に私はアメリカに行っている
〔〔aspt rslt〕〕　　現在の時点での結果を表す　mome-chng+
・まだすべての論文が集まっていません
〔〔aspt prog〕〕　　現在進行
・いま北大路駅でバスを待っています
〔〔aspt unrl〕〕　　未然、未完了　stat-chng+
・大勢の研究者がアメリカから参加します
〔〔aspt iter〕〕　　継続　mome-chng-
・既に登録料の八万五千円を振り込まれておられますね
〔〔aspt past〕〕　　過去　stat-chng-
・会議に参加した
〔〔aspt pstat〕〕　　過去の状態を表す　stat-chng+
・論文の内容が人に知られてしまった
〔〔aspt pprog〕〕　　過去進行　mome-chng-
・きのう北大路駅でバスを待っていました
〔〔aspt prslt〕〕　　過去の時点での結果を表す　chng+
・雨から雪になった
〔〔aspt -〕〕　　アスペクトがないもの（変換で指定）
・はい

N.B.解析結果では、ASPTの値の違いだけで、複数の結果が出ている場合が多い。
（素性計算による）

N.B.変換部でも、ASPTと述語の素性計算で処理していたが、動詞語義が
増えるに従って、正しい結果を出力させるのは困難になる。

解析では、ASPT が 終止形・連体形の語尾で決定されるため、
従属節で未然形の場合など、ASPTが入っていない場合がある。
これは、生成では、原形で出力されてしまうので、主節のASPTを参照して、
ASPTを補完する。
また、時制の一致などの問題に対応するためにも、主節のASPTを参照して、
従属節の ASPTを書き換える場合がある。

・ASPECT-INIT での処理

```
 * aspt が UNRL の特定の述語に対して、STATに書き換える
(rws:defrwschema2 ASPECT001 aspt unrl
"on <aspt> unrl in :PHASE :ASPECT-INIT
     in= [[reln ?action]
          [aspt unrl]
          ?rest]
 switch ?action
 case  $THINK-V-1
    out= [[reln ?action]
          [aspt stat]
          ?rest]
     case  $COST-V-1
    out= [[reln ?action]
          [aspt stat]
          ?rest]
         ...
       default
    out= [[reln ?action]
          [aspt unrl]
          ?rest]
    endswitch
end")
```

・ASPECT-CHANGE での処理(1)

```
  * 主節のASPTを参照して、従属節の ASPTを書き換える
  (rws:defrwschema2 ASPECT007 aspt unrl
  "on <tloc reln> AFTER-CONJ-1 in :PHASE :ASPECT-CHANGE
       in= [[aspt unrl]
            [tloc [[reln AFTER-CONJ-1]
                   [obje [[aspt past]
                          ?rest0]] ?rest1]] ?rest]
      out= [[aspt UNRL]
            [tloc [[reln AFTER-CONJ-1]
                   [obje [[aspt stat]
                          ?rest0]] ?rest1]] ?rest]
  end")
```

>P-560 ;夕食を取った後に京都の町に出かけます

```
[[SEM !X15[[RELN  出かける-1]
          [AGEN !X23[]]
          [TLOC [[PARM !X02[]]
                 [RESTR [[RELN  後-FN-TIME]
                        [IDEN !X130[[RELN  取る-1]
                                    [AGEN !X09[]]
                                    [ASPT  PAST] ;*<----------従属節ASPT=PAST
                                    [OBJE !X147[[PARM !X01[]]
                                                [RESTR [[RELN  夕食-1]
                                                        [ENTITY !X01]]]]]]]
                        [ENTITY !X02]]]]]
          [ASPT  UNRL] ;*<-----------------------------------主節ASPT=UNRL
          [DEST !X24[[PARM !X05[[PARM !X03[]]
                                [RESTR [[RELN  町-1]
                                        [ENTITY !X03]]]]]
                    [RESTR [[RELN  の-連体修飾]
                            [IDEN !X61[[PARM !X04[]]
                                       [RESTR [[RELN  NAMED]
                                               [IDEN  京都]
                                               [ENTITY !X04]]]]]
                           [OBJE !X05]]]]]]
;;;=========== Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X5[[LABEL *SPEAKER*]]]
        [RECP !X6[[LABEL *HEARER*]]]
        [OBJE [[RELN GO-V-1]
               [ASPT UNRL];*<-----------------------------------主節ASPT=UNRL
               [TENSE FUTURE]
               [AGEN []]
               [TLOC [[RELN AFTER-CONJ-1]
                      [OBJE [[RELN HAVE-V-1]
                             [ASPT STAT];*<----------従属節ASPT=STAT
                             [TENSE PRESENT]
                             [AGEN []]
                             [OBJE [[PARM !X1[]]
                                    [RESTR [[RELN DINNER-N-1]
                                            [ENTITY !X1]]]]]
                             [SEM-ASPE STATIVE]]]]]
               [DEST [[PARM !X4[[PARM !X3[]]
                               [RESTR [[RELN DOWNTOWN-N-1]
                                       [ENTITY !X3]]]]]
                     [RESTR [[RELN MODIFY]
                             [ARG1 [[PARM !X2[]]
                                    [RESTR [[RELN NAMED]
                                            [IDEN KYOTO-PROPN]
                                            [ENTITY !X2]]]]]
                            [ARG2 !X4]]]]]
               [SEM-ASPE UNREAL]]]]]
```

・ASPECT-CHANGE での処理(2)

```
    * 主節のASPTを参照して、従属節の ASPTを書き換える
    (rws:defrwschema2 ASPECT009 aspt past
    "on <obje cont aspt> stat in :PHASE :ASPECT-CHANGE
        in= [[reln negate]
             [aspt past]
             [obje [[cont [[reln ?reln]
                           [aspt stat]
                           ?rest0]] ?rest1]] ?rest]
        out= [[reln negate]
              [aspt past]
              [obje [[cont [[reln ?reln]
                           [aspt past]
                           ?rest0]] ?rest1]] ?rest]
    end")
```

>P-362 それほど大規模な会議だとは知りませんでした

```
[[SEM !X10[[RELN  NEGATE]
          [OBJE !X54[[RELN  知る-4]
                    [AGEN !X47[]]
                    [CONT !X48[[RELN  だ-IDENTICAL]
                              [OBJE !X70[]]
                              [IDEN !X69[[PARM !X06[[PARM !X05[]]
                                                   [RESTR [[RELN  会議-1]
                                                           [ENTITY !X05]]]]]
                                        [RESTR !X87[[RELN  大規模だ-1]
                                                   [ASPT  STAT]
                                                   [DEGR [[PARM !X04[]]
                                                          [RESTR [[RELN  それほど-1]
                                                                  [ENTITY !X04]]]]]
                                                   [OBJE !X06]]]]]
                              [ASPT  STAT]]]]];*<----------------従属節ASPT=STAT
          [ASPT  PAST]]] ;*<-----------------------------------主節ASPT=PAST

;;;=========== Transfer Result ===========
 [[SEM [[RELN INFORM]
        [AGEN !X4[[LABEL *SPEAKER*]]]
        [RECP !X5[[LABEL *HEARER*]]]
        [OBJE [[RELN NEGATE]
               [ASPT PAST]  ;*<----------------従属節ASPT=PAST
               [TENSE PAST]
               [OBJE [[RELN KNOW-V-4]
                      [AGEN !X4]
                      [CONT [[RELN BE-V-COPULA]
                             [ASPT PAST] ;*<----------------従属節ASPT=PAST
                             [TENSE PAST]
                             [OBJE []]
                             [IDEN [[PARM !X3[[PARM !X2[[PARM !X1[]]
                                                        [RESTR [[RELN CONFERENCE-N-1]
                                                                [ENTITY !X1]]]]]
                                                 [RESTR [[RELN LARGE-ADJ-1]
                                                         [ASPT STAT]
                                                         [TENSE PRESENT]
                                                         [OBJE !X2]
                                                         [SEM-ASPE STATIVE]]]]]
                                           [RESTR [[RELN SUCH-DET-1]
                                                   [ENTITY !X3]]]]]
                             [SEM-ASPE SIMPLE]]]]]
               [SEM-ASPE SIMPLE]]]]]
```

１３ 付記

変換処理部の素性構造書き換えシステムは、解析部・生成部の単一化方法と比較して、自由度が高いため、解析の後処理と生成の前処理などのインターフェースにかなり効果がある。

機能試験文６００文の変換実験が終了したことにより、大体の日本語文法項目について、処理できることがわかった。しかし、会話形式ではないので発話意図や前文参照などの文脈情報を使っての規則拡張が、必要である。

説明書で述べてきた問題点については、解決方法が見えているものもあるが、変換部だけで解決できる問題ではない部分については、解析部・生成部とかかわる形での処理が必要と思われる。

解析・変換・生成と関係名の接続の問題は重要である。語義番号が１つ変更したり、必須格の改訂・追加などがあれば、それ以降のルールはすべて影響を受ける。（変換では、生成基本辞書に合わせる。）
格要素の解析でのFIXが、最重要となる。

解析結果の問題点のうち、係り受けや構造的に正解とは考えにくいが、一応‘決め’の問題であるとして、正解構造とされたものについては、変換でその構造を正しいものに変更した場合がある。（意味格名についても同様）

また、それぞれのインターフェースで、タグなどが正しくなく送られているにもかかわらず、正しいとした英文が出力されてしまう例があった。
それを排除する規則などの作成の必要性を検討する。

変換規則で用いる手法で、解決されていないものがある。(upward修飾子)

ＪＪ変換後の出力構造の検討、英語構造内の関係名の概念化などは、さらに整理する必要がある。

参考文献

長谷川敏郎：素性構造書き換えシステムのSL-TRANS変換過程への適用
人工知能学会全国大会 1990.

長谷川敏郎：SL-TRANSにおける変換過程の処理内容について
ATR Technical Report,TR-I-0188.

長谷川敏郎：素性構造書き換えシステムマニュアル（改訂版）
ATR Technical Report,TR-I-0187.

古崎博久・鈴木雅実：素性構造書き換えシステムを利用した日英変換処理の高速化
情報処理学会第46回全国大会5B-6,1993.

鈴木雅実 ：対話文の日英構造変換への関連知識の利用 ～基礎検討～
電子情報通信学会技術研究報告、NLC92-13,1992

M.Suzuki : A Method of Utilizing Domain and Language Specific Constraints
in Dialogue Trnaslation, Proceedings of COLING '92,pp.756-762,1992.

鈴木雅実：対話翻訳における領域知識による補完手法の検討
情報処理学会 第45回全国大会2E-1,1992.

鈴木雅実・菊井玄一郎 他：日独音声言語翻訳実験システム
情報処理学会 第46回全国大会6B-6,1993.

鈴木雅実・古崎博久：言語変換処理系解説書－素性構造書き換えシステム改良版－
ATR Technical Report,TR-I-0330,1993.

鈴木雅実・古崎博久・関倫彦：音声言語翻訳のための言語変換処理の現状と課題
ATR Technical Report,TR-I-0331,1993.

鈴木雅実・関倫彦：ATR音声言語翻訳実験システムASURAにおける日英変換処理の
現状と課題
電子情報通信学会技術研究報告、NLC92-58,1993.

竹澤寿幸 他：ATR音声言語翻訳実験システムASURA
情報処理学会 第46回全国大会,6B-5,1993.

浦谷則好 他：話し言葉の日英翻訳システムの評価法
情報処理学会第46回全国大会,6B-4,1993.

APPENDIX 1

解析処理部から出力される格名一覧

・accm　随伴者　家内と参加します
・caus　原因・理由　学会に出席するために三日ほど留守にします
・comp　比較される対象　ロイヤルホテルに比べればプリンスホテルは安いです
・cont　認識・思考・判断・発言などの内容　会議に論文を発表したいと思っているのですが
・dept　起点　そこから会議場へ行くバｓが利用できます
・dest　終点　会議場までバスでいくらかかりますか
・dest　終点　明日そちらに伺います
・goal　行為が向かうもの　ツアーには現在で百名ばかり集まっておられます
・loct　場所　アカンパニーパーソンの欄に妻の名前を記入した
・obje　変化・移動・行為の対象　会議に申し込みました
・obje　変化・移動・行為の対象　会議を開く
・purp　目的　発表にスライドを使いたいのですが
・recp　所有移動における対象物の受け手　事務局に登録用紙を送る
・resl　結果　参加者は五百人以上になると思います
・role　役割　代理に助手を行かせましょう
・rout　移動における通過経路　会議中に京都を回る
・sour　受動態における動作主語　他の研究論文に教えられることが多い
・tdep　時間的起点　シンポジウムは午前十時より開かれます
・tdep　時間的起点　会議は八月二十二日から開催されます
・tdes-1　時間的終点（格マーカが「まで」）　八日の朝までお願いします
・tdes-2　時間的終点（格マーカが「までに」）一週間前までには論文集を作る予定です
・tloc　時点　オープニングセッションは九時に始まります
・agen　有性・無性の行為主体　彼が京都へ行く
・caus　原因・理由　会議で京都へお越しならばお寄りください
・comp　比較される対象　タクシーより地下鉄を使うことを薦めます
・comp　比較される対象　ロイヤルホテルと比べればプリンスホテルは安いです
・cond　条件　銀行振込でも構いません
・dept　起点　北大路駅よりバスがあります
・dest　終点　会議への参加を申し込みたいのですが
・expr　精神的事象の体験者　わたしは（が）驚いた
・inst　手段・方法　銀行振込か郵便為替でお支払いください
・loct　場所　会議は京都国際会議場で開催される
・mann　様態　個人でホテルの予約をしてください
・meth　手段・方法　お名前はローマ字で書いてください
・mutl　相互動詞の相手　担当者と相談します
・obje　状態動詞の主語　登録料の割引がある
・resl　結果　シンポジウムは混乱となった
・tloc　時点　ツアーには現在で百名ばかり集まっておられます

APPENDIX 2

P-490 : 発表なさるテーマについて要点をまとめていただきます。

```
;;; ------------ Transfer Input ------------
((SEM ((RELN てもらう-RECEIVE_FAVOR)
       (ASPT UNRL)
       (AGEN !X4())
       (RECP !X5())
       (OBJE ((RELN まとめる-1)
              (AGEN !X5)
              (OBJE ((PARM !X6())
                     (RESTR ((RELN 要点-1)
                             (ENTITY !X6)))))
              (RANG ((PARM !X9((PARM !X7())
                               (RESTR ((RELN テーマ-1)
                                       (ENTITY !X7)))))
                     (RESTR ((RELN 発表する-1)
                             (ASPT UNRL)
                             (AGEN !X8())
                             (OBJE !X9)))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X1((LABEL *SPEAKER*)))
                              (RECP !X8)))
                     (REST ((FIRST ((RELN POLITE)
                                    (AGEN !X1)
                                    (RECP !X2((LABEL *HEARER*)))))
                            (REST ((FIRST ((RELN POLITE)
                                           (AGEN !X1)
                                           (RECP !X2)))
                                   (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                  (LESS !X5)
                                                  (MORE !X4)))
                                          (REST ((FIRST ((RELN POLITE)
                                                         (AGEN !X1)
                                                         (RECP !X2)))
                                                 (REST !X3())))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X2)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X1)
         (TOPIC ((IN ())
                 (OUT ())))))))
;;;=========== Transfer Result ============
((SEM ((RELN INFORM)
       (AGEN !X4((LABEL *SPEAKER*)))
       (RECP !X2((LABEL *HEARER*)))
       (OBJE ((RELN WANT-DESIRE)
              (ASPT STAT)
              (TENSE PRESENT)
              (EXPR !X4)
              (OBJE ((RELN SUMMARIZE-V-1)
                     (AGEN !X2)
                     (OBJE ((PARM !X3((PARM !X1())
                                      (RESTR ((RELN SUBJECT-N-1)
                                              (ENTITY !X1)))
                                      (INDEX ((DEFINT INDEF)
                                              (GENDER NONE)
                                              (NUMBER SING-PL)
                                              (PERSON THIRD)))))
                            (RESTR ((RELN PRESENT-V-2)
                                    (ASPT UNRL)
                                    (TENSE FUTURE)
                                    (AGEN !X2)
                                    (OBJE !X3)
                                    (SEM-ASPE UNREAL)))))))
              (SEM-ASPE STATIVE)))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X2)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
;;;   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
;===================================================== 以下は、trace mode
;;; ------------ Transfer Input ------------
((SEM ((RELN てもらう-RECEIVE_FAVOR)
       (ASPT UNRL)
       (AGEN !X4())
       (RECP !X5())
       (OBJE ((RELN まとめる-1)
              (AGEN !X5)
              (OBJE ((PARM !X6())
                     (RESTR ((RELN 要点-1)
                             (ENTITY !X6)))))
              (RANG ((PARM !X9((PARM !X7())
                               (RESTR ((RELN テーマ-1)
                                       (ENTITY !X7)))))
                     (RESTR ((RELN 発表する-1)
                             (ASPT UNRL)
```

```
                              (AGEN !X8())
                              (OBJE !X9))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X1((LABEL *SPEAKER*)))
                              (RECP !X8)))
                     (REST ((FIRST ((RELN POLITE)
                                    (AGEN !X1)
                                    (RECP !X2((LABEL *HEARER*)))))
                            (REST ((FIRST ((RELN POLITE)
                                           (AGEN !X1)
                                           (RECP !X2)))
                                   (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                  (LESS !X5)
                                                  (MORE !X4)))
                                          (REST ((FIRST ((RELN POLITE)
                                                         (AGEN !X1)
                                                         (RECP !X2)))
                                                 (REST !X3()))))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X2)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X1)
         (TOPIC ((IN ())
                 (OUT ()))))))
;;; | 1(1):> MAINRULE
Parameter                    Value
---------                    -----
:PHASE                       :ELLIPSIS-INIT
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X4())
               (RECP !X5())
               (OBJE ((RELN まとめる-1)
                      (AGEN !X5)
                      (OBJE ((PARM !X6())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X6)))))
                      (RANG ((PARM !X9((PARM !X7())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X7)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X8())
                                     (OBJE !X9)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X1)
                              (RECP !X8)))
                     (REST ((FIRST ((RELN POLITE)
                                    (AGEN !X1)
                                    (RECP !X2)))
                            (REST ((FIRST ((RELN POLITE)
                                           (AGEN !X1)
                                           (RECP !X2)))
                                   (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                  (LESS !X5)
                                                  (MORE !X4)))
                                          (REST ((FIRST ((RELN POLITE)
                                                         (AGEN !X1)
                                                         (RECP !X2)))
                                                 (REST !X3()))))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X2)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X1)
         (TOPIC ((IN ())
                 (OUT ()))))))
;;; |  | 2(1):> ELLIP001

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 S-REQUEST and  てもらう-RECEIVE_FAVOR
;; Set local parameter.
Parameter                    Value
---------                    -----
:MOOD                        :DECLARATIVE
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X4())
               (RECP !X5())
               (OBJE ((RELN まとめる-1)
                      (AGEN !X5)
                      (OBJE ((PARM !X6())
```

```
                                    (RESTR ((RELN 要点-1)
                                            (ENTITY !X6)))))
                                (RANG ((PARM !X9((PARM !X7())
                                                 (RESTR ((RELN テーマ-1)
                                                         (ENTITY !X7)))))
                                    (RESTR ((RELN 発表する-1)
                                            (ASPT UNRL)
                                            (AGEN !X8())
                                            (OBJE !X9)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X1)
                              (RECP !X8)))
                     (REST ((FIRST ((RELN POLITE)
                                    (AGEN !X1)
                                    (RECP !X2)))
                            (REST ((FIRST ((RELN POLITE)
                                           (AGEN !X1)
                                           (RECP !X2)))
                                   (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                  (LESS !X5)
                                                  (MORE !X4)))
                                          (REST ((FIRST ((RELN POLITE)
                                                         (AGEN !X1)
                                                         (RECP !X2)))
                                                 (REST !X3()))))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X2)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X1)
         (TOPIC ((IN ())
                 (OUT ()))))))
;; Reset local parameter.
Parameter                    Value
---------                    -----
:PHASE                       :ELLIPSIS-INIT
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X4())
               (RECP !X5())
               (OBJE ((RELN まとめる-1)
                      (AGEN !X5)
                      (OBJE ((PARM !X6())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X6)))))
                      (RANG ((PARM !X9((PARM !X7())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X7)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X8())
                                     (OBJE !X9)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X1)
                              (RECP !X8)))
                     (REST ((FIRST ((RELN POLITE)
                                    (AGEN !X1)
                                    (RECP !X2)))
                            (REST ((FIRST ((RELN POLITE)
                                           (AGEN !X1)
                                           (RECP !X2)))
                                   (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                  (LESS !X5)
                                                  (MORE !X4)))
                                          (REST ((FIRST ((RELN POLITE)
                                                         (AGEN !X1)
                                                         (RECP !X2)))
                                                 (REST !X3()))))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X2)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X1)
         (TOPIC ((IN ())
                 (OUT ()))))))

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 S-REQUEST and  てもらう-RECEIVE_FAVOR
;;; |  | 2(1):< ELLIP001 Success
Parameter                    Value
---------                    -----
:PHASE                       :ELLIPSIS-RESOLUTION-SEM
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
```

```
                (ASPT UNRL)
                (AGEN !X4())
                (RECP !X5())
                (OBJE ((RELN まとめる-1)
                       (AGEN !X5)
                       (OBJE ((PARM !X6())
                              (RESTR ((RELN 要点-1)
                                      (ENTITY !X6)))))
                       (RANG ((PARM !X9((PARM !X7())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X7)))))
                              (RESTR ((RELN 発表する-1)
                                      (ASPT UNRL)
                                      (AGEN !X8())
                                      (OBJE !X9)))))))))))
    (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                                (AGEN !X1)
                                (RECP !X8)))
                        (REST ((FIRST ((RELN POLITE)
                                       (AGEN !X1)
                                       (RECP !X2)))
                               (REST ((FIRST ((RELN POLITE)
                                              (AGEN !X1)
                                              (RECP !X2)))
                                      (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                     (LESS !X5)
                                                     (MORE !X4)))
                                             (REST ((FIRST ((RELN POLITE)
                                                            (AGEN !X1)
                                                            (RECP !X2)))
                                                    (REST !X3())))))))))))
                   (OUT !X3)))
           (ASPE ((IN ())
                  (OUT ())))
           (HEARER !X2)
           (PRSP-TERMS ((IN ())
                        (OUT ())))
           (SPEAKER !X1)
           (TOPIC ((IN ())
                   (OUT ()))))))
;;; |  | 2(1):> ELLIP051

;;; --- PATTERN MATCHING FAIL. ---
;;; Defferent feature set (OBJE RECP AGEN RELN) and (RELN AGEN RECP OBJE ASPT)
;;; |  | 2(1):< ELLIP051 Fail
;;; |  | 2(1):> ELLIP052

;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (RECP AGEN RELN) and (RELN AGEN OBJE RANG)
;;; |  | 2(1):< ELLIP052 Fail
Parameter                       Value
---------                       -----
:PHASE                          :ELLIPSIS-RESOLUTION-PRAG
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X4())
               (RECP !X5())
               (OBJE ((RELN まとめる-1)
                      (AGEN !X5)
                      (OBJE ((PARM !X6())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X6)))))
                      (RANG ((PARM !X9((PARM !X7())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X7)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X8())
                                     (OBJE !X9)))))))))))
    (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                                (AGEN !X1)
                                (RECP !X8)))
                        (REST ((FIRST ((RELN POLITE)
                                       (AGEN !X1)
                                       (RECP !X2)))
                               (REST ((FIRST ((RELN POLITE)
                                              (AGEN !X1)
                                              (RECP !X2)))
                                      (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                     (LESS !X5)
                                                     (MORE !X4)))
                                             (REST ((FIRST ((RELN POLITE)
                                                            (AGEN !X1)
                                                            (RECP !X2)))
                                                    (REST !X3())))))))))))
                   (OUT !X3)))
           (ASPE ((IN ())
                  (OUT ())))
           (HEARER !X2)
           (PRSP-TERMS ((IN ())
                        (OUT ()))))
```

```
        (SPEAKER !X1)
        (TOPIC ((IN ())
                (OUT ())))))
;;; |  | 2(1):> ELLIP072

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *SPEAKER*)) and
 ()
;;; |  | 2(1):< ELLIP072 Success
;;; |  | 2(1):> ELLIP071

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *SPEAKER*)) and
 ()

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *HEARER*)) and
 ()
;;; |  | 2(1):< ELLIP071 Success
;;; |  | 2(1):> ELLIP071

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *SPEAKER*)) and
 ()

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *HEARER*)) and
 ()
;;; |  | 2(1):< ELLIP071 Success
;;; |  | 2(1):> ELLIP074
;;; |  | 2(1):< ELLIP074 Success
;;; |  | 2(1):> ELLIP071

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *SPEAKER*)) and
 ()

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((LABEL *HEARER*)) and
 ()
;;; |  | 2(1):< ELLIP071 Success
Parameter                       Value
---------                       -----
:PHASE                          :TYPES
:TYPE                           :DEFAULT
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X1)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X6)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X2)
                              (RECP !X1)))
                      (REST ((FIRST ((RELN POLITE)
                                     (AGEN !X2)
                                     (RECP !X1)))
                             (REST ((FIRST ((RELN POLITE)
                                            (AGEN !X2)
                                            (RECP !X1)))
                                    (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                   (LESS !X1)
                                                   (MORE !X2)))
                                           (REST ((FIRST ((RELN POLITE)
                                                          (AGEN !X2)
                                                          (RECP !X1)))
                                                  (REST !X3())))))))))
                 (OUT !X3)))
        (ASPE ((IN ())
               (OUT ())))
        (HEARER !X1)
        (PRSP-TERMS ((IN ())
                     (OUT ())))
        (SPEAKER !X2)
```

```
        (TOPIC ((IN ())
                (OUT ())))))
Parameter                       Value
---------                       -----
:IF                             :REDUCE
:TYPE                           :GENERAL
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X1)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X6)))))))))))
   (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                               (AGEN !X2)
                               (RECP !X1)))
                       (REST ((FIRST ((RELN POLITE)
                                      (AGEN !X2)
                                      (RECP !X1)))
                              (REST ((FIRST ((RELN POLITE)
                                             (AGEN !X2)
                                             (RECP !X1)))
                                     (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                    (LESS !X1)
                                                    (MORE !X2)))
                                            (REST ((FIRST ((RELN POLITE)
                                                           (AGEN !X2)
                                                           (RECP !X1)))
                                                   (REST !X3())))))))))))
                  (OUT !X3)))
          (ASPE ((IN ())
                 (OUT ())))
          (HEARER !X1)
          (PRSP-TERMS ((IN ())
                       (OUT ())))
          (SPEAKER !X2)
          (TOPIC ((IN ())
                  (OUT ()))))))
;;; |  | 2(1):> IFT012

;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (OBJE RECP AGEN RELN) and (RELN AGEN OBJE RANG)
;;; |  | 2(1):< IFT012 Fail
;;; |  | 2(1):> IFT013

;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (OBJE RECP AGEN RELN) and (RELN AGEN OBJE RANG)
;;; |  | 2(1):< IFT013 Fail
Parameter                       Value
---------                       -----
:IF                             :REDUCE
:TYPE                           :DEFAULT
 ((SEM ((RELN UNKNOWN-IFT)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X1)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X6)))))))))))
   (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                               (AGEN !X2)
                               (RECP !X1)))
                       (REST ((FIRST ((RELN POLITE)
                                      (AGEN !X2)
                                      (RECP !X1)))
                              (REST ((FIRST ((RELN POLITE)
                                             (AGEN !X2)
                                             (RECP !X1)))
                                     (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                    (LESS !X1)
```

```
                                                    (MORE !X2)))
                                                (REST ((FIRST ((RELN POLITE)
                                                               (AGEN !X2)
                                                               (RECP !X1)))
                                                      (REST !X3()))))))))))
                    (OUT !X3)))
            (ASPE ((IN ())
                   (OUT ())))
            (HEARER !X1)
            (PRSP-TERMS ((IN ())
                         (OUT ())))
            (SPEAKER !X2)
            (TOPIC ((IN ())
                    (OUT ()))))))
;;; |  | 2(1):> IFT044

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 てもらう-RECEIVE_FAVOR and  う-GUESS
;;; |  | 2(1):< IFT044 Success
Parameter                       Value
---------                       -----
:IF                             :REDUCE
:TYPE                           :INTENTION
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X1)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X6)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X2)
                              (RECP !X1)))
                      (REST ((FIRST ((RELN POLITE)
                                     (AGEN !X2)
                                     (RECP !X1)))
                             (REST ((FIRST ((RELN POLITE)
                                            (AGEN !X2)
                                            (RECP !X1)))
                                    (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                   (LESS !X1)
                                                   (MORE !X2)))
                                           (REST ((FIRST ((RELN POLITE)
                                                          (AGEN !X2)
                                                          (RECP !X1)))
                                                  (REST !X3()))))))))))))
                    (OUT !X3)))
            (ASPE ((IN ())
                   (OUT ())))
            (HEARER !X1)
            (PRSP-TERMS ((IN ())
                         (OUT ())))
            (SPEAKER !X2)
            (TOPIC ((IN ())
                    (OUT ()))))))
;;; |  | 2(1):> IFT045

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 INFORM and  PHATIC
;;; |  | 2(1):< IFT045 Fail
;;; |  | 2(1):> IFT046
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 INFORM and  PROMISE
;;; |  | 2(1):< IFT046 Fail
;;; |  | 2(1):> IFT047
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 INFORM and  REQUEST
;;; |  | 2(1):< IFT047 Fail
;;; |  | 2(1):> IFT048
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 てもらう-RECEIVE_FAVOR and  まとめる-1
;;; |  | 2(1):< IFT048 Fail
;;; |  | 2(1):> IFT049
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 送る-1 and  てもらう-RECEIVE_FAVOR
```

```
;;; |  | 2(1):< IFT049 Fail
;;; |  | 2(1):> IFT050
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 できる-POSSIBLE and  まとめる-1
;;; |  | 2(1):< IFT050 Fail
;;; |  | 2(1):> IFT051

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 どういたしまして-YOUR-WELCOME and  てもらう-RECEIVE_FAVOR
;;; |  | 2(1):< IFT051 Fail
Parameter                       Value
---------                       -----
:PHASE                          :PRAG-INIT
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X1)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X6)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN RESPECT)
                              (AGEN !X2)
                              (RECP !X1)))
                      (REST ((FIRST ((RELN POLITE)
                                     (AGEN !X2)
                                     (RECP !X1)))
                             (REST ((FIRST ((RELN POLITE)
                                            (AGEN !X2)
                                            (RECP !X1)))
                                    (REST ((FIRST ((RELN EMPATHY-DEGREE)
                                                   (LESS !X1)
                                                   (MORE !X2)))
                                           (REST ((FIRST ((RELN POLITE)
                                                          (AGEN !X2)
                                                          (RECP !X1)))
                                                  (REST !X3())))))))))))
                 (OUT !X3)))
         (ASPE ((IN ())
                (OUT ())))
         (HEARER !X1)
         (PRSP-TERMS ((IN ())
                      (OUT ())))
         (SPEAKER !X2)
         (TOPIC ((IN ())
                 (OUT ()))))))
;;; |  | 2(1):> PRAG001
;;; |  | 2(1):< PRAG001 Success
;;; |  | 2(1):> PRAG002
;;; |  | 2(1):< PRAG002 Success
;;; |  | 2(1):> PRAG002
;;; |  | 2(1):< PRAG002 Success
;;; |  | 2(1):> PRAG004
;;; |  | 2(1):< PRAG004 Success
;;; |  | 2(1):> PRAG002
;;; |  | 2(1):< PRAG002 Success
Parameter                       Value
---------                       -----
:PHASE                          :PRAG-MODIFICATION
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X3((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X2)
               (RECP !X3)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X3)
                      (OBJE ((PARM !X4())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X4)))))
                      (RANG ((PARM !X6((PARM !X5())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X5)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X3)
                                     (OBJE !X6)))))))))))
  (PRAG ((RESTR ((IN ((FIRST ((RELN POLITENESS)))
```

```
                                (REST ((FIRST ((RELN POLITENESS)))
                                       (REST ((FIRST ((RELN POLITENESS)))
                                              (REST ((FIRST ((RELN POLITENESS)))
                                                     (REST ((FIRST ((RELN POLITENESS)))
                                                            (REST !X1())))))))))))
                        (OUT !X1)))
                (ASPE ((IN ())
                       (OUT ())))
                (HEARER !X3)
                (PRSP-TERMS ((IN ())
                             (OUT ())))
                (SPEAKER !X2)
                (TOPIC ((IN ())
                        (OUT ()))))))
;;; |  | 2(1):> PRAG005
;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 () and
 ((FIRST ((RELN POLITE)
          (AGEN !X2((LABEL *SPEAKER*)))
          (RECP !X1((LABEL *HEARER*)))))
  (REST ((FIRST ((RELN POLITE)
                 (AGEN !X2)
                 (RECP !X1)))
         (REST ((FIRST ((RELN EMPATHY-DEGREE)
                        (LESS ())
                        (MORE ())))
                (REST ((FIRST ((RELN POLITE)
                               (AGEN !X2)
                               (RECP !X1)))
                       (REST ())))))))))
;;; |  | 2(1):< PRAG005 Fail
;;; |  | 2(1):> PRAG006

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 () and
 ((FIRST ((RELN POLITE)
          (AGEN !X2((LABEL *SPEAKER*)))
          (RECP !X1((LABEL *HEARER*)))))
  (REST ((FIRST ((RELN EMPATHY-DEGREE)
                 (LESS ())
                 (MORE ())))
         (REST ((FIRST ((RELN POLITE)
                        (AGEN !X2)
                        (RECP !X1)))
                (REST ())))))))
;;; |  | 2(1):< PRAG006 Fail
;;; |  | 2(1):> PRAG007

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 () and
 ((FIRST ((RELN EMPATHY-DEGREE)
          (LESS ())
          (MORE ())))
  (REST ((FIRST ((RELN POLITE)
                 (AGEN ((LABEL *SPEAKER*)))
                 (RECP ((LABEL *HEARER*)))))
         (REST ())))))
;;; |  | 2(1):< PRAG007 Fail
;;; |  | 2(1):> PRAG008

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 () and
 ((FIRST ((RELN POLITE)
          (AGEN ((LABEL *SPEAKER*)))
          (RECP ((LABEL *HEARER*)))))
  (REST ()))
;;; |  | 2(1):< PRAG008 Fail
;;; |  | 2(1):> PRAG009
;;; |  | 2(1):< PRAG009 Success
;;; |  | 2(1):> PRAG010

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((FIRST ((RELN POLITENESS)))
  (REST ())) and
 ()
;;; |  | 2(1):< PRAG010 Fail
;;; |  | 2(1):> PRAG012
;;; |  | 2(1):< PRAG012 Success
;;; |  | 2(1):> PRAG013

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((FIRST ((FOCUS ?FOCUS)
          (PRSP-MOD ?MOD)
          (SCOPE ?SCOPE)))
  (REST ())) and
 ()
;;; |  | 2(1):< PRAG013 Fail
;;; |  | 2(1):> PRAG015
```

```
;;; |  | 2(1):< PRAG015 Success
;;; |  | 2(1):> PRAG020

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((FIRST ((FOCUS ?FOCUS)
          (SCOPE ?SCOPE)
          (TOPIC-MOD ?MOD)))
  (REST ())) and
 ()
;;; |  | 2(1):< PRAG020 Fail
;;; |  | 2(1):> PRAG016
;;; |  | 2(1):< PRAG016 Success
;;; |  | 2(1):> PRAG017
;;; --- PATTERN MATCHING FAIL. ---
;;; Not token identical
 ()
and
 ().
;;; |  | 2(1):< PRAG017 Fail
;;; |  | 2(1):> PRAG019
;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (POLITENESS SPEAKER HEARER) and (RESTR);;; |  | 2(1):< PRAG019 Fail
Parameter                    Value
---------                    -----
:PHASE                       :PRAG-DEFAULT-1
:IFT                         :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
  (PRAG ((HEARER !X2)
         (POLITENESS ((DEGREE 5)))
         (SPEAKER !X1))))
;;; |  | 2(1):> PRAG021

;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (POLITENESS SPEAKER HEARER) and (TOPIC HEARER SPEAKER)
;;; |  | 2(1):< PRAG021 Success
Parameter                    Value
---------                    -----
:PHASE                       :PRAG-DEFAULT-2
:IFT                         :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
  (PRAG ((HEARER !X2)
         (POLITENESS ((DEGREE 5)))
         (SPEAKER !X1)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> PRAG022

;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (HEARER SPEAKER POLITENESS TOPIC) and (INTENTION HEARER SPEAKER)
;;; |  | 2(1):< PRAG022 Success
Parameter                    Value
---------                    -----
:PHASE                       :PRAG-DEFAULT-3
:IFT                         :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
```

```
                (ASPT UNRL)
                (AGEN !X1)
                (RECP !X2)
                (OBJE ((RELN まとめる-1)
                       (AGEN !X2)
                       (OBJE ((PARM !X3())
                              (RESTR ((RELN 要点-1)
                                      (ENTITY !X3)))))
                       (RANG ((PARM !X5((PARM !X4())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X4)))))
                              (RESTR ((RELN 発表する-1)
                                      (ASPT UNRL)
                                      (AGEN !X2)
                                      (OBJE !X5)))))))))))
   (PRAG ((BENEFIT NONE)
          (HEARER !X2)
          (INTENTION INFORM)
          (POLITENESS ((DEGREE 5)))
          (SPEAKER !X1)
          (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> PRAG024
;;; --- PATTERN MATCHING FAIL. ---
;;; Matching FAIL. Defferent feature set (TOPIC POLITENESS SPEAKER HEARER BENEFIT INTENTION) and (PRSP-TERMS HEARER SPEAKER)
;;; |  | 2(1):< PRAG024 Success
Parameter                       Value
---------                       -----
:PHASE                          :PRAG-DEFAULT-4
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
   (PRAG ((BENEFIT NONE)
          (HEARER !X2)
          (INTENTION INFORM)
          (POLITENESS ((DEGREE 5)))
          (PRSP-TERMS ((PRSP-MOD NULL)))
          (SPEAKER !X1)
          (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> PRAG023
;;; |  | 2(1):< PRAG023 Fail
Parameter                       Value
---------                       -----
:PHASE                          :JAPANESE
:PRSP                           :INIT
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                      (RESTR ((RELN テーマ-1)
                                              (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
   (PRAG ((BENEFIT NONE)
          (HEARER !X2)
          (INTENTION INFORM)
          (POLITENESS ((DEGREE 5)))
          (PRSP-TERMS ((PRSP-MOD NULL)))
          (SPEAKER !X1)
          (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :JAPANESE
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
```

```
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X2)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X1)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :IDIOM
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X1((LABEL *SPEAKER*)))
        (RECP !X2((LABEL *HEARER*)))
        (OBJE ((RELN てもらう-RECEIVE_FAVOR)
               (ASPT UNRL)
               (AGEN !X1)
               (RECP !X2)
               (OBJE ((RELN まとめる-1)
                      (AGEN !X2)
                      (OBJE ((PARM !X3())
                             (RESTR ((RELN 要点-1)
                                     (ENTITY !X3)))))
                      (RANG ((PARM !X5((PARM !X4())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X4)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X2)
                                     (OBJE !X5)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X2)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X1)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> IDIOM004
;;; |  | 2(1):< IDIOM004 Success
;;; |  | 2(1):> IDIOM112
;;; |  | 2(1):< IDIOM112 Success
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :PRE-GENERAL
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X2)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :GENERAL
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
```

```
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X2)))))
                             (RESTR ((RELN 発表する-1)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> GEN197

;;; --- PATTERN MATCHING FAIL. ---
;;; FSs are different type.
 ((PARM !X1())
  (RESTR ((RELN テーマ-1)
          (ENTITY !X1)))) and
 ()
;;; |  | 2(1):< GEN197 Success
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :DEFAULT-1
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X2)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :DEFAULT-2
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2())
                                       (RESTR ((RELN テーマ-1)
                                               (ENTITY !X2)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> DEFN264
;;; |  | 2(1):< DEFN264 Success
Parameter                       Value
---------                       -----
:PHASE                          :J-E
:TYPE                           :DEFAULT-3
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
```

```
                (EXPR !X4)
                (OBJE ((RELN SUMMARIZE-V-1)
                       (AGEN !X1)
                       (OBJE ((PARM !X3((PARM !X2()))
                                       (RESTR ((RELN SUBJECT-N-1)
                                               (ENTITY !X2)))
                                       (INDEX ((DEFINT INDEF)
                                               (GENDER NONE)
                                               (NUMBER SING-PL)
                                               (PERSON THIRD)))))
                              (RESTR ((RELN PRESENT-V-2)
                                      (ASPT UNRL)
                                      (AGEN !X1)
                                      (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :ENGLISH
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2()))
                                      (RESTR ((RELN SUBJECT-N-1)
                                              (ENTITY !X2)))
                                      (INDEX ((DEFINT INDEF)
                                              (GENDER NONE)
                                              (NUMBER SING-PL)
                                              (PERSON THIRD)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :ASPECT-INIT
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X4((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT UNRL)
               (EXPR !X4)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X3((PARM !X2()))
                                      (RESTR ((RELN SUBJECT-N-1)
                                              (ENTITY !X2)))
                                      (INDEX ((DEFINT INDEF)
                                              (GENDER NONE)
                                              (NUMBER SING-PL)
                                              (PERSON THIRD)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X3)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X4)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> ASPECT001
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 WANT-DESIRE and  THINK-V-1

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 WANT-DESIRE and  COST-V-1

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 WANT-DESIRE and  RELATE-V-1
```

```
;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 WANT-DESIRE and  POSSIBILITY
;;; |  | 2(1):< ASPECT001 Success
;;; |  | 2(1):> ASPECT001

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  THINK-V-1

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  COST-V-1

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  RELATE-V-1

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  POSSIBILITY

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  WANT-DESIRE

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 PRESENT-V-2 and  NEGATE
;;; |  | 2(1):< ASPECT001 Success
Parameter                       Value
---------                       -----
:PHASE                          :ASPECT-CHANGE
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT STAT)
               (EXPR !X2)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4((PARM !X3())
                                       (RESTR ((RELN SUBJECT-N-1)
                                               (ENTITY !X3)))
                                       (INDEX ((DEFINT INDEF)
                                               (GENDER NONE)
                                               (NUMBER SING-PL)
                                               (PERSON THIRD)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X4)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X2)
         (TOPIC ((TOPIC-MOD NULL))))))
Parameter                       Value
---------                       -----
:PHASE                          :ASPECT
:IFT                            :INFORM
 ((SEM ((RELN INFORM)
        (AGEN !X2((LABEL *SPEAKER*)))
        (RECP !X1((LABEL *HEARER*)))
        (OBJE ((RELN WANT-DESIRE)
               (ASPT STAT)
               (EXPR !X2)
               (OBJE ((RELN SUMMARIZE-V-1)
                      (AGEN !X1)
                      (OBJE ((PARM !X4((PARM !X3())
                                       (RESTR ((RELN SUBJECT-N-1)
                                               (ENTITY !X3)))
                                       (INDEX ((DEFINT INDEF)
                                               (GENDER NONE)
                                               (NUMBER SING-PL)
                                               (PERSON THIRD)))))
                             (RESTR ((RELN PRESENT-V-2)
                                     (ASPT UNRL)
                                     (AGEN !X1)
                                     (OBJE !X4)))))))))))
  (PRAG ((BENEFIT NONE)
         (HEARER !X1)
         (INTENTION INFORM)
         (POLITENESS ((DEGREE 5)))
         (PRSP-TERMS ((PRSP-MOD NULL)))
         (SPEAKER !X2)
         (TOPIC ((TOPIC-MOD NULL))))))
;;; |  | 2(1):> ASPECT017

;;; --- PATTERN MATCHING FAIL. ---
;;; Defferent feature set (EXPR OBJE ASPT RELN) and (OBJE ASPT RELN)
```

```
;;; |  | 2(1):< ASPECT017 Success
;;; |  | 2(1):> ASPECT027

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 - and  STAT
;;; |  | 2(1):< ASPECT027 Fail
;;; |  | 2(1):> ASPECT021
;;; |  | 2(1):< ASPECT021 Success
;;; |  | 2(1):> ASPECT027

;;; --- PATTERN MATCHING FAIL. ---
;;; Atomic type FSs can't match.
 - and  UNRL
;;; |  | 2(1):< ASPECT027 Fail
Parameter                    Value
---------                    -----
:PHASE                       :INDEX-REMOVED
:IFT                         :INFORM
 [[SEM [[RELN INFORM]
        [AGEN !X2[[LABEL *SPEAKER*]]]
        [RECP !X1[[LABEL *HEARER*]]]
        [OBJE [[RELN WANT-DESIRE]
               [ASPT STAT]
               [TENSE PRESENT]
               [EXPR !X2]
               [OBJE [[RELN SUMMARIZE-V-1]
                      [AGEN !X1]
                      [OBJE [[PARM !X4[[PARM !X3[]]
                                       [RESTR [[RELN SUBJECT-N-1]
                                               [ENTITY !X3]]]
                                       [INDEX [[DEFINT INDEF]
                                               [GENDER NONE]
                                               [NUMBER SING-PL]
                                               [PERSON THIRD]]]]]
                             [RESTR [[RELN PRESENT-V-2]
                                     [ASPT UNRL]
                                     [TENSE FUTURE]
                                     [AGEN !X1]
                                     [OBJE !X4]
                                     [SEM-ASPE UNREAL]]]]]]]
               [SEM-ASPE STATIVE]]]]]
  [PRAG [[BENEFIT NONE]
         [HEARER !X1]
         [INTENTION INFORM]
         [POLITENESS [[DEGREE 5]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X2]
         [TOPIC [[TOPIC-MOD NULL]]]]]]
;;; | 1(1):< MAINRULE Success
;;;=========== Transfer Result ============
 [[SEM [[RELN INFORM]
        [AGEN !X4[[LABEL *SPEAKER*]]]
        [RECP !X2[[LABEL *HEARER*]]]
        [OBJE [[RELN WANT-DESIRE]
               [ASPT STAT]
               [TENSE PRESENT]
               [EXPR !X4]
               [OBJE [[RELN SUMMARIZE-V-1]
                      [AGEN !X2]
                      [OBJE [[PARM !X3[[PARM !X1[]]
                                       [RESTR [[RELN SUBJECT-N-1]
                                               [ENTITY !X1]]]
                                       [INDEX [[DEFINT INDEF]
                                               [GENDER NONE]
                                               [NUMBER SING-PL]
                                               [PERSON THIRD]]]]]
                             [RESTR [[RELN PRESENT-V-2]
                                     [ASPT UNRL]
                                     [TENSE FUTURE]
                                     [AGEN !X2]
                                     [OBJE !X3]
                                     [SEM-ASPE UNREAL]]]]]]]
               [SEM-ASPE STATIVE]]]]]
  [PRAG [[BENEFIT NONE]
         [HEARER !X2]
         [INTENTION INFORM]
         [POLITENESS [[DEGREE 5]]]
         [PRSP-TERMS [[PRSP-MOD NULL]]]
         [SPEAKER !X4]
         [TOPIC [[TOPIC-MOD NULL]]]]]]
;;; ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```