

TR-I-0359

英語構文生成規則解説書
-ASURA における英語構文生成知識 -
English Generation Rules for ASURA

菊井 玄一郎 関 倫彦
Gen-ichiro KIKUI Michihiko SEKI

1993.3

梗概

本報告書では ASURA の構文生成処理で用いられている生成規則について説明する。ここでは生成規則を英語の言語現象に基づいて分類し、変換から与えられる意味表現との対応を示しながら具体的に解説する。また、変換処理と生成処理との間で受け渡されるデータ構造についても説明する。

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

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

まえがき

本解説書は ASURA の英語構文生成処理で採用されている文法・辞書に関するものである。ASURA では構文生成のための言語的知識を記述する枠組として単一化に基づく木付加文法 (Unification-based Tree Adjoining Grammar:UTAG or Feature Structure-based Tree Adjoining Grammar:FTAG)[1][4] を用いている。この枠組は単一化に基づく句構造文法 (Unification-based Phrase Structure Grammar)[2] の自然な拡張となっている。

FTAG は非常に一般的な枠組なので、品詞等の素性や要素木と呼ばれるプリミティブ (粗く言えば CFG における「書き換え規則」に対応する) の定義の仕方によって、主辞駆動句構造文法 (Head-driven Phrase Structure Grammar :HPSG)[3] や狭義の TAG¹[4] など様々な言語記述の仕方が可能である。

ここで述べる生成知識はこれらの記述の仕方のいずれかに忠実に則って「規則化」したものではなく、むしろ、柔軟な記述力を利用して、文法記述者の直観に近い形で言語現象を「記述」したものである。²。従って、抽象化や記述量の低減などをあまり行っていない部分もある。特に動詞、形容詞等の語彙まわりの部分は、扱おうとする言語表現との対応の良さから、語彙化木付加文法 (lexicalized TAG)[4] に近いものとなっている。このため、HPSG などの抽象化の進んだ規則体系から見た場合、特に語彙周りの部分で規則としての洗練さに欠けている点があるように見えるかも知れない。

また、規則化に当たっての大きな制約となる、生成の入力、すなわち、英語意味構造についても注記しておく。英語意味表現は解析・変換の各処理で抽出可能な情報をもとに作成されたものであり、「英語意味表現」として必要な情報が必ずしも全て存在している訳ではない (たとえば、名詞の定・不定の情報など)。従って、生成知識の中には、不完全な情報から「無理矢理に」表層表現を決定するためのものも含まれている。

以上のような点は「11 章 付記」の項で改めて説明する。

参考文献

- [1] Vijay-Shanker, K. et al., "Feature Structure Based Tree Adjoining Grammars", in Proceedings of COLING'88, 1988
- [2] Shieber, S. M., "An Introduction to Unification-Based Approaches to Grammar", CSLI, 1986
- [3] Pollard, C. et al., "Information-based Syntax and Semantics Volume 1 Fundamentals", CSLI, 1987
- [4] Joshi, A. K., et al., "The Convergence of Mildly Context-Sensitive Grammatical Formalisms", in Sells, Peter, Shieber, Stuart, and Wasow, Tom (editors), Foundational Issues in Natural Language Processing. MIT Press, Cambridge MA., 1990

¹初期木と付加木のみによる文法

²一方ドイツ語文法は主辞駆動文法の考え方に近い記述となっている

目次

はじめに

1	使用統語カテゴリー	5
1.1	非終端カテゴリー	5
1.2	終端カテゴリー	5
2	文の基本構造	6
2.1	S-TOP (文全体)	6
2.2	S (文)	6
2.2.1	意味構造	6
2.2.2	主要統語構造	7
2.2.3	文レベルの修飾要素	8
	★意味修飾要素の扱い	8
	★統語修飾要素の扱い	9
2.2.4	文末への "too" の生成	10
2.3	平叙文	11
2.4	外置構文	12
2.5	yes/no疑問文	13
2.6	WH疑問文	13
2.7	命令文	14
2.7.1	動詞型	14
2.7.2	名詞型	15
	★NP + PLEASE	15
	★動詞の補完	16
2.8	付加疑問文	19
2.9	約束文/感情表現文/挨拶文/応答文	20
2.10	従属文 (CONJ_S)	21
2.10.1	意味構造からの接続詞の導入	21
2.10.2	接続詞の統語的決定	22
	★conj-lexの実現	22
2.10.3	従属文の主-述構造への展開	23
2.11	倒置文 (INV_S) の展開	24
2.11.1	"do" が出現する場合	24
2.11.2	"be" 動詞の倒置の場合	25
2.11.3	助動詞の倒置の場合	26
2.11.4	否定の倒置文	27
3	VP (述部) の構造	28
3.1	基本意味構造	28
3.2	基本統語構造	29
3.3	モーダル素性の扱い	30
3.4	時制素性の扱い	31
3.4.1	単節点の時制素性規則	31
3.4.2	葉を作る時制素性規則	32
3.5	否定	33
	★ "do not" による否定	33
	★ 時制素性より導かれた助動詞の否定	33
	★ モーダルの否定	34
	★ "be" 動詞の否定	35
3.6	受動態	35
3.7	任意格の扱い	37
3.7.1	基本規則形式	37
3.7.2	任意格の種類	39
3.8	VPの中核構造 (動詞+必須格)	39
3.8.1	文型別テンプレート	39
3.8.2	lex-aspeについて	41
3.8.3	:SPECIFIED 及び :NULL について	44
3.8.4	VP 中核展開レベルにおける modi 素性の使用	45
	★必須格の任意格化の防止	45
	★"be" 動詞の前に副詞が現われるのを防ぐ	46
3.8.5	コントロールについて	47

4	NP (名詞句) の構造	48
4.1	意味構造	48
4.1.1	中核意味構造	48
4.1.2	修飾意味構造	48
4.2	名詞的統語カテゴリーの種類	49
4.3	階層構造の概略	50
4.4	階層構造別の規則	50
4.4.1	NP レベル	50
	★冠詞"a", "the", 無冠詞	50
	★thatなどによる修飾	51
	★所有格NPによる修飾	52
	★PPによる後方からの修飾	52
	★REL_Sによる修飾	53
	★ADVPによる修飾	54
4.4.2	NN レベル	55
	★NUMBによる修飾	55
	★APによる修飾	56
	★INDEX素性の被覆	56
	★NNレベルで終端語が導入される場合 (複合語)	56
4.4.3	N レベル (語彙レベル)	56
4.4.4	固有名詞	58
5	AP (形容詞句) の構造	59
5.1	基本意味構造	59
5.2	基本統語構造	60
5.3	形容詞規則	60
5.3.1	叙述用法	60
5.3.2	限定用法	61
	★限定用法の時制素性の扱い	62
	★関係節との関係	63
	★任意格	63
5.3.3	叙述・限定に共通の規則	63
	★DEGRによる修飾	63
	★比較級・最上級	64
	★終端語の導入	65
6	数詞 (NUMB) の構造	67
6.1	意味構造	67
6.2	基数の生成	67
6.3	序数詞	70
6.4	NUMBがNPとして使用される場合	70
6.5	NUMBによるNNの修飾 --> NPの項参照	70
7	代名詞 (PRON) の構造	71
7.1	意味構造	71
7.2	統語構造	71
7.3	語彙項目の導入	71
7.3.1	デフォルトの代名詞	72
7.3.2	someとany	72
8	前置詞句 (PP) の構造	73
8.1	他の規則によってPのlexが決定される場合	73
8.2	意味構造にP (前置詞マーカ) が存在する場合	74
9	副詞句 (ADVP) の構造	75
10	疑問詞 (WH) の構造	76
11	付記	77
APPENDIX 1	素性一覧	
APPENDIX 2	「英語意味素性構造」一覧	
APPENDIX 3	規則適用例	

はじめに

本稿では、変換処理部で出力された英語意味構造から、実際の英語統語構造に
いかに生成されるかを規則を例示しながら、概要を述べる。

対象例文は、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、変換部を
通った文であり、これらについて生成処理を行なうものとして作成したルールを
概説した。

規則概要

(g_pd

:name ルール固有名

; 説明

:internal_structure (S NP VP) (根節点 葉節点1 葉節点2)

:annotation

```
((S  [[syn  [[cat s]....]]      根節点 syn 統語
      [sem  [[reln inform]....]]]) 根節点 sem 意味

  (NP  [[syn  [[cat np]....]]    葉節点1 syn 統語
        [sem  ?subj-sem]        葉節点1 sem 意味
        [defsem ?subj-defsem]])   葉節点1 が空()の場合の代入値 sem

  (VP  [[syn  [[cat vp]....]]    葉節点2 syn 統語
        [sem  ?obje]            葉節点2 sem 意味
        [defsem ?subj-defsem]])   葉節点2 が空()の場合の代入値 sem
```

:environment 環境制約 (入力構造のPRAG部を参照)

)

(:SET) 素性値の集合を扱う

(:NOT) 排除する素性値を指定する

:NULL 空素性[]を示す。

:SPECIFIED 空でないことを示す。

規則の説明には、生成規則 (PD) の基本的概念・操作 (被覆、単一化など) を
予備知識として想定する。

また、説明中で、同一の根節点の意味素性と葉節点の意味素性を示す際に、
手続き的に『根節点から葉節点へ伝達する』などの表現を使うことがある。

1 使用統語カテゴリー (cat 素性の値)

1.1 非終端カテゴリー

S-TOP	文全体	Ex. "I applied for the conference."
S	文核	"I applied for the conference"
CONJ_S	接続文	"if you have a question"
REL_S	関係節	topics "that will be brought up"
VP	動詞句	"applied for the conference"
TOINF	to 不定詞句	"to apply for the conference"
NP	名詞句	"the interesting topics"
NN	名詞複合	"interesting topics"
AP	形容詞句	"the same as the last year"
ADVP	副詞句	"actually"
PP	前置詞句	"in the conference hall"
IDIOM	慣用表現	"thank you"

INV_S	倒置文	"can I register as a speaker"
(独立のカテゴリーではなく、[cat s] と [inv +] で表現。)		
WH	疑問句	"what kind of procedure"
(独立のカテゴリーではなく、[cat (:SET np advp ap)] と [wh +] で表現。)		

1.2 終端カテゴリー

VERB	動詞	"apply"
BE	be 動詞	"are"
AUX	助動詞	"can"
NOUN	名詞	"conference"
PRON	代名詞	"it"
NUMB	数詞	"three"
ADJ	形容詞	"interesting"
ADV	副詞	"actually"
PREP	前置詞	"at"
ADVPRT	副詞的小辞	bring it "up"
CONJ	接続詞	"and"
DET	限定詞	"the"
TO	to	"to" apply for the conference
NOT	否定辞	"not"
GREET	挨拶	"hello"
SIGN	記号	"?"
RIPRON	疑問代名詞	"what"
RIADJ	疑問形容詞	"which" hotel
RIADV	疑問副詞	"how"

N.B. cat素性の値を示す以外は、VERB を V、NOUN を N、ADJ を A、PREP を P と省略する。

2 文の基本構造

2.1 S-TOP (文全体)

すべての文が S-TOP $\rightarrow$ S + SIGN の構造を持つ。

```
(g_pd
:name S-TOP
:文全体(S-TOP)の展開規則
:internal_structure (S-TOP S SIGN)
:annotation
  ((S-TOP    [[syn [[cat s-top][inv -]]]
              [sem ?sem]]) ; *<===== (a)
  (S        [[syn [[cat s][inv -]
                  [adv-cl -][prsp-mod +] ; *<===== (b)
                  [s-depth matrix]
                  [signlex ?signlex]]] ; *<===== (d)
              [sem ?sem]]) ; *<===== (a)
  (SIGN     [[syn [[cat sign]
                  [lex ?signlex]]]])) ; *<===== (d)
:environment
([prsp-terms [[prsp-mod (:SET dake gurai koso nado nanka nara nomi
                        null zutsu))]])) ; *<===== (c)
```

- (a) すべての文に当てはまるので、sem は無関係に葉節点（下位構造）に送られる。
 (b) 上位構造で「prsp-terms の処理が決定している」ことを下位構造に伝える。
 (c) 日本語解析による取り立て助詞『も』以外をすべて列挙する。
 環境が ([prsp-terms [[prsp-mod mo]]) の場合は S-TOP-MO を使用。
 この規則は S 節点の [prsp-mod -] 指定のみ S-TOP と異なる。
 [prsp-mod -] $\rightarrow$ S-MODI-T00 で文末に "too" を生成。
 (d) この段階では文末記号 (".", "?") は決定不可能。
 SIGN の lex を signlex として S に伝達しておき S を具体的に展開する
 ルールで値を指定する。
 (Ex. S-WH-INV_S: S[signlex "?"] $\rightarrow$ WH + INV_S)

2.2 S (文)

2.2.1 意味構造

入力意味構造は最トップレベルで以下の構成を持つ。

```
([reln (:SET inform request questionref questionif questionconf promise
          response expressive phatic) ; *<===== (a)
[agen [[label *speaker*]]] ; *<===== (b)
[recp [[label *hearer*]]] ; *<===== (c)
[obje ?obje] ; *<===== (d)
[caus ?caus] ; *<===== (e)
[conect ?conect] ; *<===== (f)
.
.
.]
```

- (a)~(d) 入力英語意味構造のトップレベルの必須要素。

(a) 発話タイプ

reln の値 (発話タイプ) によって文の種類が決定される。

inform	(デフォルトの発話タイプ)	情報提供文
request	(話者が何らかの要求を表わす発話)	命令文
questionref	(WH疑問文に対応する発話)	WH 疑問文
questionif	(YES-NO疑問文に対応する発話)	yes/no 疑問文
questionconf	(確認を表わす疑問文[～ね]に対応する発話)	確認疑問文
promise	(約束を表わす発話)	約束文
response	(質問に対する応答や相づちなどの発話)	応答文
expressive	(話者の感情表現に関するイディオム的な発話)	感情表出文
phatic	(挨拶などで用いられるイディオム的な発話)	挨拶文

- (b) 発話者（話し手）を示す。
- (c) 対話者（聞き手）を示す。
- (d) 文の中核的な命題内容を表わす。
- (e), (f)

任意の修飾要素で、存在しない場合もある。

- ・ 意味修飾要素：統語構造には現われず、文全体の意味的特性を表わすもの。
単節点の規則（葉を作らない）で処理される。

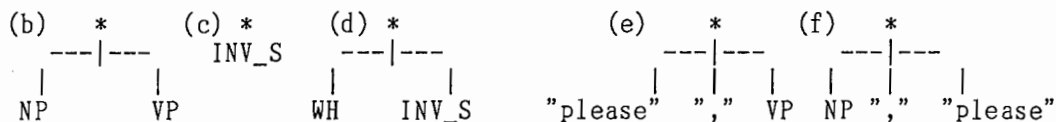
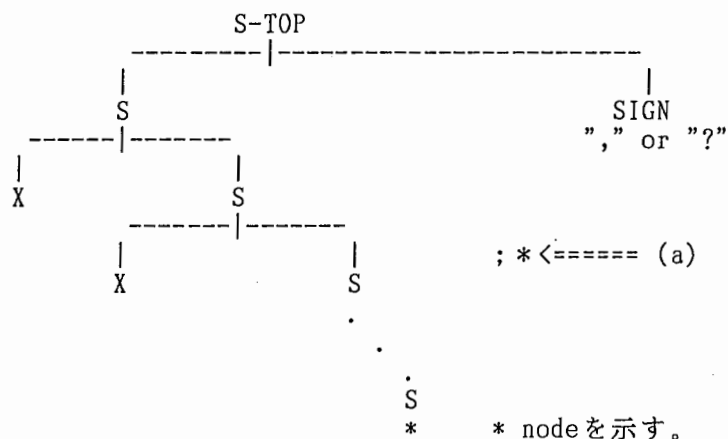
att
manner
(infmann)
(mann)

- ・ 統語修飾要素：統語構造で文レベルの修飾要素になるもの。

caus
cond
conect
conj
opps
(infattd)

まず任意の修飾要素が処理され、その後で、必須要素の reln（発話タイプ）、agen（発話者）、recp（対話者）を処理する形で、英文構造が reln の値に応じて、平叙文や疑問文などに開かれる。

2.2.2 主要統語構造



(a) 文修飾構造

S（文）に対して前方から修飾する形式（S→ X S condなど）と、
修飾要素を後方に置く形式（S→ S X causなど）がある。

N.B. 複数の文修飾要素が存在する場合がある。

N.B. 1 個の修飾要素に対し、前方修飾・後方修飾の 2 種類の規則を作成し、
その他の情報（終端語、文脈）により、選択することもできる。

- (b) 平叙文 （主部NPと述部VPに展開する）
- (c) yes/no 疑問文 （倒置文形式）
- (d) WH 疑問文 （疑問詞（句）を先頭に移動した、倒置文形式）
- (e) 命令文 （副詞 "please" を先頭に伴った命令形）
- (f) 名詞句命令文 （名詞句の後ろに副詞 "please" を伴った省略命令形）

2.2.3 文レベルの修飾要素

★意味修飾要素の扱い

```
(g_pd
:name S-COVER-MANNER
;MANNERのための被覆規則
:internal_structure (S)
:annotation
  ((S [[syn [[cat s]
    [s-modi [[s-caus -][s-cond -][s-conect -] ;*<===== (c)
    [s-conj -][s-infattd -][s-opps -]
    [too -]]]
    [negate ?negate][prsp-mod ?prsp]
    [adv-cl ?adv-cl]
    [s-depth ?s-depth][signlex ?signlex]
    [synmanner ?manner]]] ;*<===== (b)
    [sem [[manner ?manner]]]])) ;*<===== (a)
```

- (a) 任意要素 manner に対する処理を行なう。
 (b) manner が担っていた情報を syn に取り込む。
 下位構造の VP に伝達されて、V 導入規則で利用される場合がある。

Ex. VP-WANT-TOINF : VP[synmanner direct] --> want NP TOINF
 VP-WOULD-LIKE-TOINF: VP[synmanner indirect] --> would like NP TOINF

manner 以外の意味修飾要素は、それぞれ単節点の処理ルールで単に被覆される (semで参照される) だけで、syn に取り込まない。

N.B. 現在、生成において積極的役割を持たないものとして扱う。

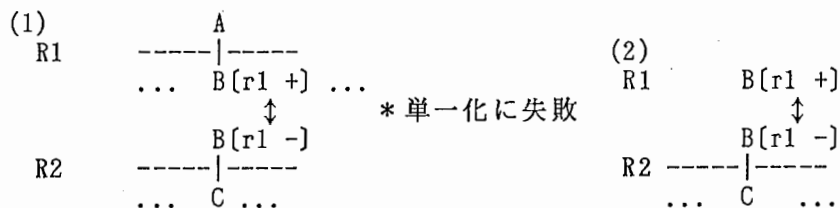
- (c) 統語修飾要素の決定状況を示す。

Ex. s-caus = 「すでに caus が処理されているかどうか
 (上位構造で、文が caus に修飾されている)」

任意要素間の被覆の順番を決定するのに使用される。

一般に、2つの規則 R1 (上位構造とする), R2 (下位構造とする) に関して、R1 と R2 を結合する適用を防ぐためには、

R1 (上位構造) の 葉節点に [r1 +]、
 R2 (下位構造) の 根節点に [r1 -] を与えておけば、単一化に失敗する。(1)
 (単節点のルールの場合も同様である(2))



N.B.

manner は、全ての統語修飾要素より、順番として先に処理される。
 また、他の意味修飾要素も同じく、全ての統語修飾要素より先に処理される。
 単節点の意味修飾要素規則間の順番はプログラムで自動的に決定される。

★統語修飾要素の扱い

```
(g_pd
:name S-MODI-CONJ
:CONJのための被覆規則
:internal_structure (S1 X COM S2)
:annotation
  ((S1 [[syn [[cat s][inv -]
            [s-modi [[s-caus -][s-cond -][s-conect ?conect] ; *<===== (a)
                    [s-conj -][s-infattd ?infattd][s-opps -]
                    [too -]]]
            [negate ?negate][prsp-mod ?prsp]
            [adv-cl ?adv-cl]
            [s-depth ?s-depth][signlex ?signlex][synmanner ?manner]]]
    [sem ?sem [[conj ?conj]]]] ; *<===== (c)
  (X [[syn [[cat (:SET advp conj_s)]]] ; *<===== (b)
    [sem ?conj]] ; *<===== (c)
  (COM [[syn [[cat sign]
            [lex ", "]]]]
  (S2 [[syn [[cat s][inv -]
            [s-modi [[s-caus -][s-cond -][s-conect ?conect]
                    [s-conj +][s-infattd ?infattd][s-opps -]
                    [too -]]]
            [negate ?negate][prsp-mod ?prsp]
            [adv-cl ?adv-cl]
            [s-depth ?s-depth][signlex ?signlex][synmanner ?manner]]]
    [sem ?sem]]))
```

- (a) caus, cond, opps の統語修飾要素の処理規則、および "too" の生成規則の (S-MODI-T00) より、順番として先に適用される。

N.B. 『順番として先に適用される』ことは、すなわち、上位構造となる。従って、上位構造での lex 指定による終端語は、下位構造の外側に位置する。

Ex. ESET P-49:

Well, although you say that way, it'll be a problem.
 <CONJ> <-----OPPS----->

- (b) 文頭に修飾要素 ?conj を付加する。

ここでは、?conj の下位構造を参照していないので、?conj のカテゴリーは、不明である。そこで、?conj のカテゴリーとして想定される ADVP と CONJ_S を集合の値(:SET)として、両方与えておく。

- (c) conj の値を変数とし、文頭に置かれた修飾要素の sem の値とする。

2.2.4 文末への "too" の生成

```
(g_pd
:name S-MODI-T00
;文末にtooを生成する規則
:internal_structure (S1 S2 COM ADV)
:annotation
  ((S1 [[syn [[cat s][inv -]
    [s-modi [[s-caus ?caus][s-cond ?cond][s-conect ?conect]
      [s-conj ?conj][s-infattd ?infattd][s-opps ?opps]
      [too -]]] ;*<===== (a)
    [negate ?negate][prsp-mod -]
    [adv-cl -]
    [s-depth matrix] ;*<===== (b)
    [signlex ?signlex][synmanner ?manner]]]
    [sem ?sem]])
  (S2 [[syn [[cat s][inv -]
    [s-modi [[s-caus ?caus][s-cond ?cond][s-conect ?conect]
      [s-conj ?conj][s-infattd ?infattd][s-opps ?opps]
      [too +]]] ;*<===== (c)
    [negate ?negate][prsp-mod +] ;*<===== (d),(e)
    [adv-cl ?adv-cl]
    [s-depth matrix][signlex ?signlex][synmanner ?manner]]]
    [sem ?sem]])
  (COM [[syn [[cat sign]
    [lex ", "]]]])
  (ADV [[syn [[cat adv]
    [lexid too-adv-1][lex "too"]]]]]))
```

(a) prsp-mod の処理が決定していない --> 2.1 S-TOP の項を参照。

```
S-TOP [prsp-mod +] (:environment M0以外) ->
S-TOP [prsp-mod -] (:environment M0) -> S-MODI-T00 [prsp-mod +]
```

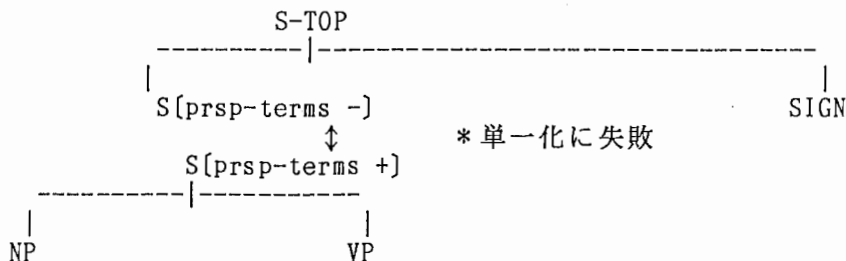
(この場合は、"too" が未出現)。

(b) 主文のみに適用されるものとする。

(c) "too" が文に付加されていることを示す。

(d) prsp-mod の処理決定を示す。
根節点の [prsp-mod -] が葉節点 S で [prsp-mod +] に変換されるので、
この規則が繰り返し適用されることはない。

(e) この規則を省略して文展開規則 (Ex. S-NP-VP-MATRIX) が適用される可能性を防ぐために、文展開規則の根節点に [prsp-terms +] を指定する。



(f) このルールは他のすべての文修飾規則の後に適用されるものとする。
従って、他のすべての文修飾規則に条件 [s-modi [[too -]]] を指定しておく。

2.3 平叙文

```
(g_pd
:name S-NP-VP-MATRIX
;主文を主部と述部に展開する規則
:internal_structure (S NP VP)
:annotation
((S [[syn [[cat s][inv -]
      [agr ?agr]
      [adv-cl -][prsp-mod +] ;*<===== (a)
      [signlex "."] ;*<===== (b)
      [synmanner ?manner]]]
  [sem [[reln inform]
        [agen [[label *speaker*]]]
        [recp [[label *hearer*]]]
        [obje ?obje [[reln (:NOT compound-sent invited-lets))]]]]]] ;*<===== (c)
(NP [[syn [[cat np]
          [case nom][agr ?agr] ;*<===== (d)
          [any-env -][arti ?arti]]]
  [sem ?subj-sem] ;*<===== (e)
  [defsem ?subj-defsem]]] ;*<===== (e)
(VP [[syn [[cat vp]
          [vform ?vform (:SET present past)][agr ?agr] ;<===== (d)
          [subj-info [[subj-sem ?subj-sem] ;*<===== (e)
                     [subj-syn [[agr ?agr][arti ?arti]]] ;*<===== (e)
                     [subj-defsem ?subj-defsem]]] ;*<===== (e)
          [any-env -][negate -]
          [adv-cl -][be-inv -][synmanner ?manner]]]
  [sem ?obje]]]]] ;*<===== (c)
```

- (a) prsp-terms の処理は決定している。
[prsp-mod mo] にもかかわらず、“too” が生成されない場合を防ぐ。
- (b) 文末記号を “.” に決定。
- (c) 中核的命題内容を VP に伝達。
- (d) NP, VP 間での agr素性の値の共有により、主語と動詞の人称・数の一致を表わす。
agr素性は person, number 素性の複合素性である。

Ex. third person, singular = [agr [[person 3][number sing]]]

NP の agr素性は N まで、VP の agr素性は V まで伝達される。

- (e) 主語は obje (述語構造) の内部構成に依存するので、この段階では決定不能。

VP に主語に関する情報を担う subj-info素性を与え、V とその必須格を直接支配する VP まで伝達する。
subj-info素性を介してこれらの値は、VP を V と必須格に展開する規則で決定される。
subj-info は以下の構成を持つ複合素性である。

[subj-info [[subj-sem ?subj-sem]	<-- 主語の意味構造
[subj-syn [[agr ?agr]	<-- 主語の統語特性 一致素性
[arti ?arti]]]	限定詞
[subj-defsem]]]	<-- 主語が 空集合[] の場合のデフォルト値

2.4 外置構文

```
(g_pd
:name S-NP-VP-EXTRAPOSI
;主文を主部と述部に展開する規則 (外置)
;signlexは"."に決定
:internal_structure (S NP VP TOINF)
:annotation
((S  [[syn ...]
      [sem ...]])
(NP  [[syn [[cat np]
             [case nom][agr ?agr]]]
      [sem [[label *it*]]]])
(VP  [[syn ...
          [subj-info [[subj-sem ?subj-sem] ;*<===== (a)
                      [subj-syn [[agr ?agr][arti ?arti]]]
                      [subj-defsem ?subj-defsem]]]
          ...]
      [sem ...]])
(TOINF [[syn [[cat toinf]
              [subj-info [[subj-sem [[label *other*] ;*<===== (b)
                                   [restr %%EMPTY%]]]]]
              [insert-v -]]]
        [sem ?subj-sem]])) ;*<===== (a)
```

N.B. ("..." で省略した箇所は S-NP-VP-MATRIX と同一)

- (a) 主語の sem を文末の TOINF に伝達する。
 (なお、that 節の外置を扱う規則は未作成。)
 文 (命題) 型の意味構造であった場合、通常の TOINF と同じように展開される。
 名詞型の意味構造であった場合、展開は失敗する。

N.B. 最終的に展開が失敗する場合でも、主語・述語の構成をなす、すべての平叙文にこの規則が適用されるので、計算量が過多となっている。

- (b) TOINF の sem で主語が *other* 及び、空集合[] でなければならない。
 [[label *other*]] だけだと、あらゆる名詞型意味構造と単一化されてしまう。

```
[[[parm ?x]
  [restr [[reln ?reln]
          [entity ?x]]]
  [label *other*]]
```

この指定により、主語が具体的に指定されている場合も展開が失敗する。
 これは、主語が「消える」のを防ぐためである。

N.B. 外置構文生成の条件が、現在は、TOINFで主語が空集合[]であるため、変換処理部での省略補完を行なう場合に、注意が必要である。
 (または、変換処理部で補完される素性を区別できるようにする)

N.B. 主語がある場合でも、外置構文の場合がありうる。
 Ex. "It is hard for a Japanese to master any European tongue."

ESETで、この外置構文が使われた例をあげる。

Ex.
 >P-274 新幹線で京都まで来るのはそれほど大変ではありません
 "To come to Kyoto by the Shinkansen isn't so hard."
 "It isn't so hard to come to Kyoto by the Shinkansen."

2.5 yes/no 疑問文

```
(g_pd
:name S-INV_S
;yes-no疑問文の展開規則
:internal_structure (S INV_S)
:annotation
((S      [[syn [[cat s][inv -] ;*<===== (a)
          [prsp-mod +] ;*<===== (b)
          [signlex "?"] ;*<===== (c)
          [synmanner ?manner]]]
      [sem [[reln questionif]
          [agen [[label *speaker*]]]
          [recp [[label *hearer*]]]
          [obje ?obje]]]] ;*<===== (d)
(INV_S [[syn [[cat s][inv +] ;*<===== (a)
          [vform (:SET present past)]
          [adv-cl -][synmanner ?manner]]]
      [sem ?obje]]]] ;*<===== (d)
```

- (a) 通常の文を倒置文に変換する。
- (b) prsp-mod の処理は決定している。
- (c) 文末記号は "?" に決定する。
- (d) 中核的命題内容は同一であることを示す。

2.6 WH 疑問文

```
(g_pd
:name S-WH-INV_S
;WH疑問文の展開規則
:internal_structure (S WH INV_S)
:annotation
((S      [[syn [[cat s][inv -] ;*<===== (a)
          [prsp-mod +] ;*<===== (b)
          [signlex "?"] ;*<===== (c)
          [synmanner ?manner]]]
      [sem [[reln questionref]
          [agen [[label *speaker*]]]
          [recp [[label *hearer*]]]
          [obje [[parm ?wh-sem] ;*<===== (d),(f)
                  [restr ?proposition]]]]]] ;*<===== (d),(g)
(WH      [[syn [[cat ?wh-cat (:SET advp np pp)][wh +] ;*<===== (e)
          [case ?case][agr ?agr]
          [prep-info [[instprep ?instprep][loctprep ?loctprep]
                      [methprep ?methprep]]]
          [insert-v ?insert-v][modi-posi ?modi-posi]
          [prep-lex ?prep-lex]]]
      [sem ?wh-sem]]] ;*<===== (f)
(INV_S [[syn [[cat s][inv +] ;*<===== (a)
          [vform (:SET present past)]
          [adv-cl -][synmanner ?manner]]]
      [sem ?proposition] ;*<===== (g)
      [slash [[syn [[cat ?wh-cat][wh +] ;*<===== (h)
          [case ?case][agr ?agr]
          [prep-info [[instprep ?instprep][loctprep ?loctprep]
                      [methprep ?methprep]]]
          [insert-v ?insert-v][modi-posi ?modi-posi]
          [prep-lex ?prep-lex]]]
          [sem ?wh-sem]]]]]]]
```

- (a) 通常文を倒置文に変換する。
- (b) prsp-mod は処理されている。
- (c) 文末記号を "?" に決定する。
- (d) WH疑問文は、意味構造で既に WH 表現に相当する部分が命題の外に置かれて、parm-restr 構造を成している。
- (e) 倒置文の文頭に疑問詞 (WH 表現) を付加する。
WH 表現は [wh +] で示す。
- (f) parm の値が WH の意味になる。
- (g) restr の値が INV_S の値になる。
- (h) restr の値は下位構造に parm の値と等しい部分を持つ。(タグで示される)
これは統語構造では 文頭の WH 表現に相当する欠損要素である。

slash 素性は、ここでは ?wh-semで明示されるように、表層に表われている WH表現の統語や意味情報をすべて持っており、下位構造内にあるそれと単一化可能な節点が、空範疇となることを表わす。

N.B. slash 素性は、次の宣言で指定される葉節点のカテゴリーには、根節点と同一の slash 素性が自動的に与えられる。
(def_binding_nodes (S VP PP AP NP ADVP REL_S TOINF ADJ))
但し、テンプレートを使用する場合は、根節点・葉節点における同一の slash 素性を明示しなければならない。

2.7 命令文

2.7.1 動詞型

中核的命題が動詞型の意味構造の場合、S --> please + VP という展開を行なう。

```
(g_pd
:name S-PLEASE-VP
;動詞句の前方にpleaseを付加して命令文を作る
;signlexは"."に決定
:internal_structure (S ADV VP)
:annotation
((S      [[syn  [[cat s][inv -]
                  [synmanner ?manner]
                  [prsp-mod +]      ;*<===== (a)
                  [signlex ".")] ] ;*<===== (b)
          [sem  [[reln request]
                  [agen  [[label *speaker*]]]
                  [recp  [[label *hearer*]]]
                  [obje  ?obje]]]] ] ;*<===== (c)
(ADV  [[syn  [[cat adv]
              [lexid please-interj-1][lex "please"]]]]]
(VP   [[syn  [[cat vp]
              [agr  [[person 2][number sing]]]
              [vform bse]] ] ;*<===== (d)
          [sem  ?obje]]]] ] ;*<===== (c)
```

- (a) prsp-mod の処理は決定している。
- (b) 文末記号は "." に決定。
- (c) 中核的命題内容を VP の意味とする。
- (d) 動詞は原形指定。

2.7.2 名詞型

★ NP + PLEASE

中核的命題が名詞型意味構造の場合
S --> NP , please という展開を行なう。

Ex. MSET da-11: ご住所とお名前をお願いします。

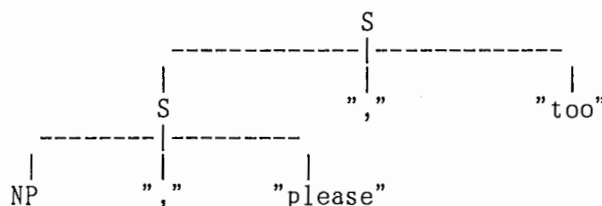
```
(g_pd
:name S-NP-PLEASE
;名詞句の後ろにpleaseを付加して命令文を作る
:internal_structure (S NP COM ADV)
:annotation
((S      [[syn  [[cat s][inv -]
                [s-modi [[s-caus ?caus][s-cond ?cond][s-conect ?connect]
                        [s-conj ?conj][s-infattd ?infattd][s-opps ?opps]
                        [too -]]] ; * <===== (b)
                [synmanner ?manner]
                [prsp-mod +][signlex "."]]]
      [sem  [[reln request]
                [agen [[label *speaker*]]]
                [recp [[label *hearer*]]]
                [obje ?obje]]]]) ; * <===== (a)
(NP      [[syn  [[cat np]
                [agr [[person 3]]]
                [case acc]]]
      [sem ?obje]]) ; * <===== (a)
(COM     [[syn  [[cat sign][lex ","]]]])
(ADV     [[syn  [[cat adv]
                [lexid please-interj-1][lex "please"]]]]]))
```

(a) 中核的命題は NP の意味になる。

(b) 上位構造で "too" が生成されていない。

NP + please 型の命令文に "too" が付加される場合、
次のような構造を生成しないよう S-MODI-T00 の使用を禁止する。

Ex. MSET d10-33: The phone number, too, please.
*The phone number, please, too.



NP と "please" の間に "too" を挿入する特殊規則を使用する。

```
(g_pd
:name S-NP-T00-PLEASE
;名詞句の後ろにtoo, pleaseを付加して命令文を作る
;tooはpleaseの前に現われるのでS-MODI-T00は使用できない
:internal_structure (S NP COM1 too COM2 ADV)
:annotation
((S      [[syn  [[cat s][inv -]
                [synmanner ?manner]
                [prsp-mod -] ; * <===== (a)
                [signlex "."]]]
      [sem  [[reln request]
                [agen [[label *speaker*]]]
                [recp [[label *hearer*]]]
                [obje ?obje]]]])
```



```

(NP  [[syn  [[cat np][case acc]
             [agr [[person 3]]]]]
      [sem ?obje]])
(COM1 [[syn  [[cat sign][lex ","]]]])
(too  [[syn  [[cat adv] ; *<===== (b)
             [lexid too-adv-1][lex "too"]]]])
(COM2 [[syn  [[cat sign]
             [lex ","]]]])
(ADV  [[syn  [[cat adv]
             [lexid please-interj-1][lex "please"]]]])

```

- (a) prsp-mod の処理はまだ終了していない（現われるべき "too" が未出現）。
 S-MODI-TOO が適用されると [prsp-mod +] に変わるので、S-MODI-TOO,
 S-TOO-NP-PLEASE の両方が適用されることはない。

★動詞の補完

生成英文の多様化のため、命令文意味構造が名詞型の時、
 動詞を補う操作も行う。

『NPをお願いします』 -> "NP,please." 以外のパターンを生成する。

Ex. MSET da-11: The address and the name, please. -->
 May I have the address and the name?
 Please give me the address and the name.

MSET d10-18: Then, the single room of seven thousand yen of
 Kyoto Prince Hotel, please. -->
 Then, please reserve the single room of seven thousand yen of
 Kyoto Prince Hotel.

```

(g_pd
:name S-MAY-I-HAVE-NP
;May I have +名詞句+?という構造の命令文
;signlexは"? "に決定
;internal_structure (S (INV_S AUX NP1 (VP V NP2)))
;annotation
((S      [[syn  [[cat s][inv -]
                [synmanner ?manner]
                [prsp-mod +][signlex "?"]]]]
          [sem  [[reln request]
                [agen [[label *speaker*]]]
                [recp [[label *hearer*]]]
                [obje ?obje]]]] ; *<===== (b)
(INV_S [[syn  [[cat s][inv +]]]] ;
(AUX  [[syn  [[cat aux]
              [vform present][agr ?agr [[person 1]
                                         [number sing]]]
              [lex "may"]]]]] ;
(NP1  [[syn  [[cat np]
              [case nom]]] ;
          [sem  [[label *speaker*]]]] ; *<===== (a)
(VP   [[syn  [[cat vp]
              [vform bse]]]] ;
(V    [[syn  [[cat verb]
              [vform bse][agr ?agr]
              [lex "have"]]]]] ;
(NP2  [[syn  [[cat np]
              [case acc][insert-v have]]] ; *<===== (c)
          [sem  ?obje]]]] ; *<===== (b)

```

- (a) "May I have" を補う。
- (b) 中核的命題が "have" の目的語になる。
- (c) この補完は個々の名詞に依存する。
insert-v 素性で補完される動詞の種類を指定して N レベルまで伝達し、
個々の N が持つ insert-v の値と一致させる。
- すなわち、この規則は名詞のinsert-v素性がhaveの時にしか適用できない。

次の2つの規則は S-PLEASE-VP と共に使用され、名詞型の中核的命題内容が VP に一致して、"give", "reserve for me" を補完して VP を生成する。

```
(g_pd
:name VP-GIVE-ME-NP
;VPが名詞型の意味構造の場合にgive meを補う (命令文)
:internal_structure (VP V NP1 NP2)
:annotation
((VP [[syn [[cat vp]
              [vform bse]
              [aspt-cover -][tense-sem-aspe-cover -]
              [insert-v give]]]
      [sem ?vp-sem]])
 (V [[syn [[cat verb][vform bse][lex "give"]]]])
 (NP1 [[syn [[cat np][case acc][agr ?agr1]]]
       [sem [[label *speaker*]]]])
 (NP2 [[syn [[cat np]
              [case acc][agr ?agr2]
              [insert-v give]]] ; *<===== (d)
       [sem ?vp-sem]]]))

(g_pd
:name VP-RESERVE-NP-FOR-ME
;VPが名詞型の意味構造の場合にreserve ... for meを補う (命令文)
:internal_structure (VP1 (VP2 V NP) PP)
:annotation
((VP1 [[syn [[cat vp]
              [vform bse]
              [aspt-cover -][tense-sem-aspe-cover -]
              [insert-v reserve]]]
      [sem ?vp-sem]])
 (VP2 [[syn [[cat vp]]])
 (V [[syn [[cat verb]
              [vform bse]
              [lex "reserve"]]]])
 (NP [[syn [[cat np]
              [case acc][agr ?agr2]
              [insert-v reserve]]] ; *<===== (d)
      [sem ?vp-sem]])
 (PP [[syn [[cat pp]
              [prep-lex "for"]]]]
      [sem [[label *speaker*]]]]))

(d) 適切な insert-v の指定。
```

insert-v の値を持つ終端語レベルの規則を次に示す。

```
(g_pd
:name N-ADDRESS
;名詞address
:type :instance
:using N-COMMON-N-TEMPLATE
:internal_structure (N)
:annotation
((N [[syn [[agr [[number (:SET sing pl)]]]
      [arti (:SET the)][insert-v (:SET give have)] ;*<===== (e)
      [lexid address-n-1][lex "address"]]]]
  [sem [[restr [[reln (:SET address-n-1 address-n-2)]]]]]]))
```

```
(g_pd
:name NN-SINGLE_ROOM
;名詞複合single room
:internal_structure (NN A N)
:annotation
((NN [[syn [[cat nn]
             [case ?case][agr ?agr [[person 3][number (:SET sing pl)]]]
             [prep-info [[modiprep "of"]]]
             [arti (:SET a the)][insert-v reserve]]] ;*<===== (e)
      [sem [[parm ?x]
             [restr [[reln single_room-n-1]
                     [entity ?x]]]]]]))
(A [[syn [[cat adj]
          [lexid single-adj-1][lex "single"]]]])
(N [[syn [[cat noun]
          [case ?case][agr ?agr]
          [lexid room-n-1][lex "room"]]]]]))
```

- (e) これらの名詞が、名詞型の命令文意味構造で用いられた場合、用意される insert-v で与えられた動詞を補完することができる。

2.8 付加疑問文

トップレベルのquestionconfにより、次の2つの構造が作られる。

- (1) S-S-RIGHT によって文末に "right?" を付加する。
- (2) S-S-TAG によって文末に "isn't it?"などを生成する。

```
(1)
(g_pd
:name S-S-RIGHT
;文の後ろにright?を付加して確認疑問文を作る
:internal_structure (S1 S2 COM ADJ)
:annotation
((S1      [[syn  [[cat s][inv -]
                  [prsp-mod +]
                  [signlex "?"]
                  [synmanner ?manner]]]
           [sem  [[reln questionconf]
                  [agen ?agen]
                  [recp ?recp]
                  [obje ?obje]]]])
 (S2 [[syn  [[cat s][inv -]
             [s-depth subord] [synmanner ?manner]
             [adv-cl -]]]
      [sem ?obje]])
 (COM [[syn  [[cat sign]
             [lex ", "]]]])
 (ADJ  [[syn  [[cat adj]
             [lex "right"]]]]]))

(2)
(g_pd
:name S-S-TAG
;付加疑問文
:internal_structure (S1 S2 COM AUX NOT PRON)
:annotation
((S1      [[syn  [[cat s][inv -]
                  [synmanner ?manner]
                  [prsp-mod +]
                  [signlex "?"]]]]
           [sem  [[reln questionconf]
                  [agen ?agen]
                  [recp ?recp]
                  [obje ?obje]]]])
 (S2 [[syn  [[cat s][inv -]
             [agr ?agr]           ; *<===== (b)
             [subj-sem ?subj-sem]
             [tag-aux ?tag-aux]  ; *<===== (a)
             [s-depth subord]
             [synmanner ?manner]
             [adv-cl -]]]
      [sem ?obje]])
 (COM [[syn  [[cat sign]
             [lex ", "]]]])
 (AUX  [[syn ?tag-aux]]           ; *<===== (a)
 (NOT  [[syn [[cat not]
             [lex "not"]]]]])
 (PRON [[syn [[cat pron]
             [case nom][agr ?agr]]] ; *<===== (b)
        [sem [[label *tag-np*]]]]))
```

- (a) 付加疑問に現われる助動詞的要素 (have, will, be, etc.) に関する情報を下位へ伝達する。

助動詞的要素を導入する VP 展開ルール (Ex. VP-PERF) によって値が与えられる --> 本文と付加疑問中の助動詞的要素の一致が起こる。

- (b) 代名詞の決定

PRON の sem に [[label *tag-pron*]] を与える。

[[label *tag-pron*]] と agr の組合せから適切な代名詞を導く。

Ex. [agr [[person 3][number sing]]] + [label *tag-pron*] --> "it"
--> PRON の項を参照。

N.B. 以下の付加疑問文については、現在、規則を作成していない。

前ページ(1)の "~,right?" のパターンのみが生成される。

- ・ 主節が一般動詞の場合の "do" を用いた付加疑問文
"He likes opera, doesn't he?"
- ・ 否定文に付く肯定の付加疑問文 (be動詞、助動詞、一般動詞)
"You haven't met my wife, have you?"
"Your wife doesn't smoke, does she?"
- ・ 否定語を含む文に対する肯定の付加疑問文
"Nobody phoned, did they?"
- ・ I am の後の付加疑問文
"I'm late, aren't I?"

N.B. 付加疑問文においては、縮約形を使うことが制約としてあるが、通常の縮約形と同様、MGネット (形態素処理ルーチン) で取り扱っている。

(MGネットの通常の縮約形は、縮約したほうがよい程度の制約)

2.9 約束文、感情表現文、挨拶文、応答文

- ・ 約束文

発話タイプ promise からは、意志未来の "will" を導く。

[reln promise] S-PROMISE: S --> "I" + "will" + VP

- ・ 感情表現文、挨拶文、応答文

発話タイプ expressive phatic response は、イディオム処理とする。

[reln (:SET expressive phatic response)] S-IDIOM: S --> IDIOM

N.B. 発話タイプ expressive で、感謝を表わす THANK_YOU-IDIOM などの場合、

:environment で、politeness (丁寧度) の数値によって、

Thanks./Thank you./Thank you very much./I'll really appreciate it.

などのvariationを生成することもできる。(現在は、"Thank you."のみ)

(g_pd

:name IDIOM-THANK_YOU

;慣用表現 (Thank you.)

:internal_structure (IDIOM V PRON)

:annotation

```
((IDIOM      [[syn [[cat idiom]]]
               [sem [[reln thank_you-idiom]
                     [agen [[label *speaker*]]]
                     [recp [[label *hearer*]]]
                     [aspt ?aspt]
                     [sem-aspe ?sem-aspe]
                     [tense ?tense]]]])
```

```
(V      [[syn [[cat verb]
               [vform present][agr [[person 1][number sing]]]
               [lexid thank-v-1][lex "thank"]]]])
```

```
(PRON [[syn [[cat pron]
              [case acc][agr [[person 2][number sing]]]
              [lex "you"]]]])
```

2.10 従属文 (CONJ_S)

従属文は文修飾規則、VP 修飾規則、動詞語彙項目導入規則によって導入される。

Ex. S-MODI-OPPS: S --> CONJ_S + "," + S
VP-CAUS: VP --> VP + CONJ_S or PP
VP-THINK-S: VP --> "think" + S or CONJ_S)

2.10.1 意味構造からの接続詞の導入

意味構造に接続詞が存在する場合は次のような規則で CONJ + S に展開する。

```
(g_pd
:name CONJ_S-IF
;ifによって導かれる副詞節
:internal_structure (CONJ_S CONJ S)
:annotation
((CONJ_S [[syn [[cat conj_s]
                [adv-cl +] ; *<===== (d)
                [conj-lex -]]] ; *<===== (f)
          [sem [[reln if-conj-1] ; *<===== (a)
                [obje ?obje]]]])
(CONJ [[syn [[cat conj]
              [lexid if-conj-1][lex "if"]]]]]
(S [[syn [[cat s][inv -]
           [s-depth subord] ; *<===== (c)
           [adv-cl +] ; *<===== (d)
           [any-env +]]] ; *<===== (e)
    [sem ?obje]]]))
```

(a) 意味構造に CONJ (接続詞マーカー) が存在する。

(b) CONJ を統語構造に具体化する。

(c) 「埋め込み文である」という [s-depth subord] を与える。

(d) adv-cl 素性は
「未来形の代用として現在形を使用する時間、または条件の副詞節かどうか」を意味する。
"if" 文はこれに相当する副詞節である。
この素性は VP に伝達されて V の活用形態の決定の際に参照される。

(e) 「some --> any という変化を引き起こす環境かどうか」を示す。
"if" 節の中では "some" の代わりに "any" が使用される。
この素性は主語や目的語にまで伝えられ、PRON 導入規則に利用される。

(f) この CONJ_S を導入した規則によって与えられた CONJ の語彙項目。
意味構造に CONJ がある場合は、直前の規則によっては指定されない。
この場合、"- " で表わす。

2.10.2 接続詞の統語的決定

意味構造に接続詞マーカーが存在しない場合は、
(意味役割が tdes-1 の時、"think" などの補文の時)
その CONJ_S を導入する規則で conj-lex として決定されている。

N.B. 意味構造中には、通常、"that" などの補文標識は表現されないが、
同格の場合など、一部、変換出力で示される場合がある。(接続詞マーカー)

```
(g_pd
:name VP-Tf
;動詞+ (that) 節を支配する動詞句
:type :template
:using nil
:internal_structure (VP V S)
:annotation
((VP [[syn [[cat vp]
            [agr ?agr][vform ?vform]
            [subj-info [[subj-sem ?agen]
                        [subj-syn ?subj-syn]
                        [subj-defsem [[label *def-speaker*]]]]]]
      [sem [[agen ?agen]
            [cont ?cont]]]])
 (V [[syn [[cat verb]
            [agr ?agr][vform ?vform]]]])
 (S [[syn [[cat (:SET s conj_s)][inv -]
            [adv-cl -] ; *<===== (a)
            [conj-lex "that"]] ; *<===== (b)
      [sem ?cont]]]))
```

(a) that 節中では「未来」は通常どおり未来形で表わされる。

(b) 接続詞を "that" と指定する。

★conj-lex の実現

```
(g_pd
:name CONJ_S
;副詞節
:internal_structure (CONJ_S CONJ S)
:annotation
((CONJ_S [[syn [[cat conj_s]
                [adv-cl ?adv-cl] ; *<===== (a)
                [conj-lex ?lex (:NOT -)]] ; *<===== (b)
            [sem ?sem]]])
 (CONJ [[syn [[cat conj]
              [lex ?lex]]]] ; *<===== (b)
 (S [[syn [[cat s][inv -]
           [s-depth subord][adv-cl ?adv-cl]] ; *<===== (a)
      [sem ?sem]]]))
```

(a) adv-cl の伝達。

(b) conj-lex を CONJ の lex とし、語彙項目を決定する。
意味構造から導かれる場合は適用されないように "-" でないことを条件とする。

2.10.3 従属文の主-述構造への展開

```
(g_pd
:name S-NP-VP-SUBORD
;従属文を主部と述部に展開する規則
:internal_structure (S NP VP)
:annotation
((S  [[syn  [[cat s][inv -]
             [agr ?agr]
             [any-env ?any-env][adv-cl ?adv-cl]
             [rel-cl ?rel-cl][s-depth subord] ; * <===== (a)
             [synmanner ?manner]
             [tag-aux ?tag-aux]]]
      [sem ?sem]]) ; * <===== (b)
(NP  [[syn  [[cat np]
             [case nom][agr ?agr]
             [any-env -][arti ?arti]]]
      [sem ?subj-sem]
      [defsem ?subj-defsem]])
(VP  [[syn  [[cat vp]
             [vform ?vform (:SET present past)][agr ?agr]
             [subj-info [[subj-sem ?subj-sem]
                         [subj-syn [[agr ?agr][arti ?arti]]]
                         [subj-defsem ?subj-defsem]]]
             [any-env ?any-env][negate -]
             [adv-cl ?adv-cl][be-inv -][rel-cl ?rel-cl]
             [s-depth subord] ; * <===== (a)
             [synmanner ?manner][tag-aux ?tag-aux]]]
      [sem ?sem]])) ; * <===== (b)
```

- (a) 埋め込み文だという情報。主文には適用されない。
埋め込み文には下に示す最トップレベルの意味構造が存在しないので
S-NP-VP-MATRIX は使用できない。

```
[[reln inform]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

従属文以外でも S-S-T00 (S --> S + ", " + "too")によって導入される
S などでもすでに上位の構造は被覆されているので、
S-NP-VP-SUBORD を使用する。
従って、[s-depth subord] と指定しておく。

- (b) 意味構造の内部は見ない。
主語の決定、数・人称の一致の扱いは S-NP-VP-MATRIX と同じである。

2.1.1 倒置文(INV_S)の展開

2.1.1.1 "do" が出現する場合

```
(g_pd
:name INV_S-DO
;do+主語+動詞句からなる倒置文
:internal_structure (INV_S AUX NP VP)
:annotation
((INV_S [[syn [[cat s][inv +]]] ;*<===== (a)
          [sem ?sem [[sem-aspe (:SET stative simple)] ;*<===== (b),(h)
                    [tense ?vform (:SET present past)]]]]))
(do [[syn [[cat aux] ;*<===== (c)
          [agr ?agr][vform ?vform]
          [lexid do-v-1][lex "do"]]]])
(NP [[syn [[cat np]
          [case nom][agr ?agr] ;*<===== (e)
          [arti ?arti]
          [any-env +]]] ;*<===== (f)
     [sem ?subj-sem] ;*<===== (e)
     [defsem ?subj-defsem]])
(VP [[syn [[cat vp][vform bse] ;*<===== (d)
          [subj-info [[subj-sem ?subj-sem] ;*<===== (e)
                    [subj-syn [[agr ?agr][arti ?arti]]
                    [subj-defsem ?subj-defsem]]]
          [any-env +] ;*<===== (f)
          [aux-do +]]] ;*<===== (g)
     [sem ?sem]]]) ;*<===== (h)
```

- (a) 倒置文である。
- (b) 中核的命題内容の内部を参照して、単純な（進行形、完了形ではない）現在、過去形であることを確認する。
[sem-aspe perf] であれば "have" を、[tense future] であれば "will" を生成する必要があるためである。
- (c) 助動詞 "do" を出す。動詞の形態素性 "vform" の値として根節点 sem の tense の値を代入する。
agr を主語と共有することで人称・数の一致を表わす。
- (d) 本動詞は原形。
- (e) 主語の決定は平叙文と同じく subj-info による。
- (f) 疑問文中では "some" の代わりに "any" が使用される。
- (g) "do" が現われたことを示す。
この規則で導入される VP が VP-BE-NP (VP --> BE + NP) などの "be" 動詞導入規則で展開されないように VP-BE-NP などには根節点に [aux-do -] と指定する。
- (h) 中核的命題内容を本動詞句の sem とする。
sem-aspe, tense はこのルールで扱い、:SETで指定した値の場合のみ、このルールが適用される。

2.1.1.2 "be" 動詞の倒置の場合

```
(g_pd
:name INV_S-BE-NP
;主語+be+補語名詞句のbeが前置した倒置文 (using ADJUNCTION)
:internal_structure (INV_S V NP1 ((VP1 VP2) NP2))
:annotation
((INV_S [[syn [[cat s][inv +]
               [vform ?tense]]]
         [sem ?sem ; * <===== (a)
           [[reln be-v-copula]
            [tense ?tense (:SET present past)]
            [obje ?obje]
            [iden ?iden]
            [aspt ?aspt]
            [sem-aspe (:SET stative simple)]]]]]]
 (V [[syn [[cat be]
           [vform ?tense][agr ?agr]
           [lexid be-v-copula][lex "be"]]]]]
 (NP1 [[syn [[cat np]
            [case nom][agr ?agr]
            [arti the]]]
       [sem ?obje]
       [defsem [[label *it*]]]])
 (VP1 [[syn [[cat vp]
            [agr ?agr]]]
       [sem ?sem]] ; * <===== (c)
 (VP2 [[syn [[cat vp]
            [agr ?agr]]]
       [sem ?sem]]
 (NP2 [[syn [[cat np]
            [case nom]]]
       [sem ?iden]]))
```

- (a) "「be」 + 補語」の意味構造である。
N.B. コピュラとしての"be"動詞の場合である。
- (b) "be" 動詞の「倒置」。
- (c) 任意格 (VP 修飾要素) は VP --> VP + X , VP --> X + VP という形の規則で扱われる。
従って、VP 節点が必要。

任意格の処理のために adjunction によって VP 節点を用意する。
ここでは (a) で処理されている
素性以外のもの (任意格) だけが扱われる。

N.B. 否定疑問文の場合は、adjunction の処理ができない。
-> 問題点参照

2.11.3 助動詞の倒置の場合

```
(g_pd
:name INV_S-CAN
;can+主語+動詞句からなる倒置文
:internal_structure (INV_S AUX NP VP)
:annotation
((INV_S [[syn [[cat s][inv +]]]
          [sem [[reln possibility]
                [aspt ?aspt]
                [tense ?tense(:SET present)]
                [sem-aspe ?sem-aspe]
                [expr ?expr]
                [obje ?obje]]]]) ; *<===== (a)
(AUX [[syn [[cat aux]
            [vform present][agr ?agr]
            [lex "can"]]]]])
(NP [[syn [[cat np]
           [case nom][agr ?agr]
           [arti ?arti]
           [any-env +]]]
     [sem ?subj-sem]
     [defsem ?subj-defsem]])
(VP [[syn [[cat vp]
           [vform bse][agr ?agr]
           [subj-info [[subj-sem ?subj-sem]
                        [subj-syn [[agr ?agr][arti ?arti]]]
                        [subj-defsem ?subj-defsem]]]]]
     [sem ?obje]]]]) ; *<===== (a)
```

(a) 助動詞構文に相当する意味構造は一般的に次のように構成されている。

```
[[reln ?reln]
 [xxxxx ?xxxxx]
 [obje ?obje]
 [aspt ?aspt]
 [sem-aspe ?sem-aspe]
 [tense ?tense]]
```

(?reln は possibility, obligation など。
xxxxx は agen または expr。)

文全体からモーダルを除いた意味内容を表わす obje を本動詞句の semとする。

N.B. 変換処理で、任意格要素などは、モーダルより一段深い動詞句レベルに一律に「引き下げられ」ている。

N.B. 助動詞のテンスについては、上の規則では、lex指示で生成しているが、過去形の場合もこの規則に追加して、MGネットで処理しなければならない。

2.11.4 否定の倒置文

```
(g_pd
:name INV_S-NEG-DO
;don't + 主語 + 動詞句からなる倒置文
:internal_structure (INV_S don't NP VP)
:annotation
((INV_S [[syn [[cat s][inv +]]]
          [sem [[reln negate]
                [obje ?obje]
                [aspt ?aspt]
                [sem-aspe (:SET stative unreal)]
                [tense present]]]])
 (don't [[syn [[cat aux]
               [vform present][agr ?agr]
               [lex "don't"]]]])
 (NP [[syn [[cat np][case nom]
            [agr ?agr][arti ?arti]]]
      [sem ?agen]
      [defsem [[label *def-hearer*]]])
 (VP [[syn [[cat vp]
            [vform bse][agr ?agr]
            [subj-info [[subj-sem ?agen]
                        [subj-syn [[agr ?agr][arti ?arti]]]
                        [subj-defsem ?subj-defsem]]]
            [any-env +][aux-do +]]]
      [sem ?obje]]]))
```

N.B. 現在形の場合の例である。
過去形の場合もこの規則に追加して、MGネットで処理する。

N.B. "don't" は、本来、MGネットで生成されるべき。

3 VP (述部) の構造

3.1 基本意味構造

Ex.

[[reln ?reln]	; *	<=====	(a)
[agen ?agen]	;		
[obje ?obje]	;		
.	;		
.	;		
.	;		

[loct ?loct]	;		
[tloc ?tloc]	;		
.	;		
.	;		
.	;		

[aspt ?aspt]	;		
[sem-aspe ?sem-aspe]	;		
[tense ?tense]	;		

- (a) 動詞の語義を示す。
語彙項目 (単語) が導き出される。(Ex. attend-v-1 -> "attend")
- (b) 必須格。 agen,obje,expr,recp などは、必須格であるが、
動詞語義によっては、loct や cond が必須格となる場合がある。
- (c) 任意格。 必須格と同様、任意格であるかどうかは、動詞語義規則の指定に依る。
agen,obje,expr,recp などは、少なくとも任意格とはならない。
- (d) 時制関係素性。
aspt は、生成ではアクティブには扱わない。
単節点の規則で、常に処理される。

-> 3.4.1 単節点の時制素性規則

sem-aspe,tense は、それぞれ、アスペクトと仮のテンスを意味し、
常に、組み合わせたセットで処理している。

(1) 否定文、(2) モーダル文、(3) 受動文ではこのような構造の上位に
否定、法、受動の意味構造がかぶさっており、
そのレベルに時制関係素性が「引き上げられている」。

- (1)

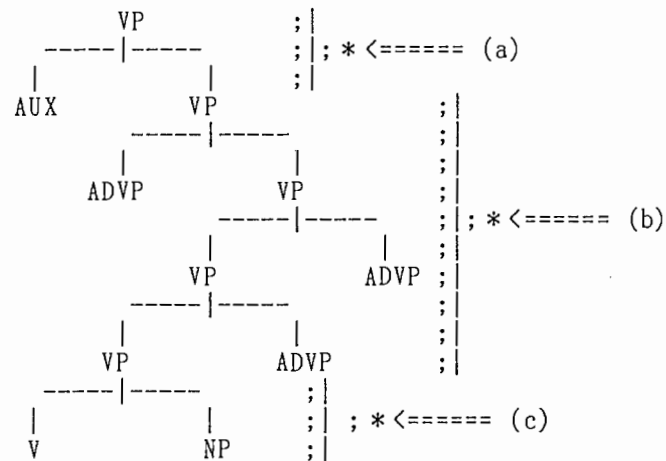
[[reln negate]	;	*	<=====	(a)
[obje ?obje]	;			
[aspt ?aspt]	;			
[sem-aspe ?sem-aspe]	;			
[tense ?tense]	;			
- (2)

[[reln obligation-must]	;	*	<=====	(a)
[obje ?obje]	;			
[aspt ?aspt]	;			
[sem-aspe ?sem-aspe]	;			
[tense ?tense]	;			
- (3)

[[reln passive]	;	*	<=====	(a)
[obje ?obje]	;			
[aspt ?aspt]	;			
[sem-aspe ?sem-aspe]	;			
[tense ?tense]	;			

- (a) ここに動詞語義の関係名、必須格、任意格の複合が入る。

3.2 基本統語構造



- (a) モーダルや時制素性から導かれた助動詞。

- (b) VP の任意格による修飾。

(a), (b) のレベルでは VP が展開されて新たな VP が導入される。従って、同一規則群の適用に際して、任意格の組み合わせの数だけ、複数の適用順序で、生成文ができてしまう。これを防ぐため、modi 素性を使用する。

--> 3.7 任意格について を参照。

N. B. すべての任意格要素の順番を、任意格の種類だけで決定しているが、
終端語や任意格の文字列の長さなどを考慮しなければならない。

生成出力後、構造調整などで、任意格の順序の変更が必要である。

- (c) VP を V とその必須格に展開する。

VP を導入しないので、必然的に時制、モーダル、否定、任意格などより後に、処理されることになる。

3.3 モーダル素性の扱い

```
(g_pd
:name VP-MUST
;義務を表わすmustが前方に付加された動詞句
:internal_structure (VP1 AUX VP2)
:annotation
((VP1 [[syn [[cat vp][agr ?agr]
              [vform (:NOT bse)] ;;;willなどが既に付加されたVPには不適用
              [subj-info ?subj-info [[subj-sem ?agen] ;*<===== (a)
                                      [subj-syn ?subj-syn]
                                      [subj-defsem ?subj-defsem]]]
              [tag-aux ?tag-aux]]] ;*<===== (b)
      [sem [[reln obligation-must] ;|
            [agen ?agen] ;|
            [obje ?obje] ;*(f) ;| ;*<===== (c)
            [tense (:SET present future)] ;|
            [sem-aspe ?sem-aspe]]]])
(AUX [[syn ?tag-aux [[cat aux] ;*<===== (b),(d)
                    [agr ?agr][vform present]
                    [lex "must"]]]])
(VP2 [[syn [[cat vp]
            [vform bse] ;*<===== (e)
            [subj-info ?subj-info]]]
      [sem ?obje]]))
```

- (a) 主語が決定される。
- (b) 文末の付加疑問に現われるのは "must"。
- (c) モーダルと時制関係素性は常にセットで扱われる。
- (d) モーダルが VP の前方に付加される。
- (e) 本動詞は、助動詞 "must"があるので、原形である。
vform は中心語の V まで同一である。
- (f) 任意格は ?obje の下位にあるので、そこで処理される。

3.4 時制素性の扱い

時制素性には、tense, sem-aspe と aspt の3種類がある。

3.4.1 単節点の時制素性規則

```
(g_pd
:name VP-SIMPLE-PRESENT-PAST
;動詞の活用のみで表わされる単純な現在、過去時制
:internal_structure (VP)
:annotation
((VP [[syn [[cat vp]
          [vform ?tense] ;*<===== (a)
          [agr ?agr]
          [synmanner ?manner][subj-info ?subj-info]
          [s-depth ?subord][tag-aux ?tag-aux][any-env ?any-env]
          [rel-cl ?rel-cl][negate ?negate][be-inv ?be-inv]
          [modi [[accmmodi -][conjmodi -][compmodi -] ;*<===== (b)
                [duramodi -][instmodi -][loctmodi -]
                [mannmodi -][mannmodi-front -][degrmodi -]
                [degrmodi-front -]
                [deptmodi -][rolemodi -][tdepmodi -]
                [taspmodi -][taspmodi-front -][tdeslmodi -][purpmodi -]
                [tdes2modi -][tlocmodi -][tlocmodi-front -]
                [tdep-tdeslmodi -][rangmodi -]]]]]]
      [sem [[tense ?tense (:SET present past)] ;*<===== (a)
            [sem-aspe (:SET simple stative nill)]]]]))
```

- (a) tense の値を vform の値とする。
aspt は 生成に関与しないと見なし、VP-ASPT-COVER で単純に処理する。
->次のルール
- (b) modi 素性は VP の修飾状況を示す複合素性。
例えば、[accmmodi -] は上位 VP で accom に修飾されていないことを示す。
VP 修飾規則で決定される。
(Ex. VP[modi [[accmmodi -]]] --> VP[modi [[accmmodi +]]] + PP(with ...))
ここでの指定は「この段階で VP はいかなる任意格にも修飾されていない」、
つまり「時制素性の処理はすべての任意格の処理に先行する」ことを示す。

N. B.

時制素性のうち、aspt素性については、生成処理では積極的に用いない。
これは、解析処理からの素性であり、比較・デバッグ作業のために、
変換処理後も、素性構造内に残してあるためである。

aspt素性は、次の規則で扱われる。

```
(g_pd
:name VP-ASPT-COVER
;VPのasptを無条件に被覆する（時制、アスペクトはtense, sem-aspeによって決定）
:internal_structure (VP)
:annotation
((VP [[syn [[cat vp]
          [vform ?vform][agr ?agr]
          [subj-info ?subj-info]
          [modi [[accmmodi -][causmodi -][compmodi -]
                .....
                [tloc-ordermodi -]]]
          [tense-sem-aspe-cover ?tense-sem-aspe-cover]
          [any-env ?any-env][negate ?negate][adv-cl ?adv-cl]
          [synmanner ?manner][tag-aux ?tag-aux]
          [perf -][prog -][will -]
          [aux-do ?aux-do][s-depth ?subord]
          [aspt-cover +]
          [lex-aspe ?lex-aspe]
          [rel-cl ?rel-cl]
          [be-inv -][have-inf ?have-inf]]]
      [sem [[aspt ?aspt]]]]]]);*<===== asptを無条件に参照するのみ。
```


3.4.2 葉を作る時制素性規則

```
(g_pd
:name VP-FUTURE
:動詞句の前方に未来を表わす助動詞willが付加された構造
:internal_structure (VP1 will VP2)
:annotation
((VP1 [[syn [[cat vp]
            {vform (:NOT bse prp)}[agr ?agr]
            {adv-cl -} ; * <===== (a)
            {negate ?negate}
            {any-env ?any-env}
            {subj-info ?subj-info}[tag-aux ?will] ; * <===== (b)
            {modi [{accmmodi -}[causmodi -][compmodi -] ; * <===== (c)
                    {condmodi -}[conjmodi -][degrmodi -]
                    {degrmodi-front -}[deptmodi -][destmodi -]
                    {duramodi -}[freqmodi -][goalmodi -]
                    {instmodi -}[loctmodi -][mannmodi -]
                    {mannmodi-front -}[methmodi -][purpmodi -][rangmodi -]
                    {recpmodi -}[rolemodi -][slocmodi -]
                    {sourmodi -}[taspmodi -][tdepmodi -]
                    {tdeslmodi -}[tdes2modi -][tdep-tdeslmodi -]
                    {tlocmodi -}[tlocmodi-front -][tloc-ordermodi -]]]]]
            {sem ?vp-sem [{tense future} ; * <===== (d)
                           {sem-aspe unreal}]]])
(will [[syn ?will [[cat aux] ; * <===== (b),(e)
                    {vform present}[agr ?agr]
                    {lex "will"}]]])
(VP2 [[syn [[cat vp]
            {vform bse}[agr ?agr] ; * <===== (f)
            {negate ?negate}
            {any-env ?any-env}[be-inv -]
            {subj-info ?subj-info}
            {modi [{accmmodi -}[causmodi -][compmodi -] ; * <===== (c)
                    {condmodi -}[conjmodi -][degrmodi -]
                    {degrmodi-front -}[deptmodi -][destmodi -]
                    {duramodi -}[freqmodi -][goalmodi -]
                    {instmodi -}[loctmodi -][mannmodi -]
                    {mannmodi-front -}[methmodi -][purpmodi -][rangmodi -]
                    {recpmodi -}[rolemodi -][slocmodi -]
                    {sourmodi -}[taspmodi -][tdepmodi -]
                    {tdeslmodi -}[tdes2modi -][tdep-tdeslmodi -]
                    {tlocmodi -}[tlocmodi-front -][tloc-ordermodi -]]]
            {aux +}]] ; * <===== (g)
            {sem ?vp-sem}]]])
```

- (a) 「未来」を現在形で表わす副詞節でないことが条件。
[adv-cl +] の場合は VP-FUTURE-ADV-CL で現在形になる
(葉を作らず vform を "+" に指定する操作のみを行う)。
- (b) 文末の付加疑問には "will" が現われる。
- (c) この規則はあらゆる任意格規則に先行する。つまり、"will" は全ての任意格より上位で VP に付加されている。
- (d) tense, sem-aspe の指定。
[tense future]と[sem-aspe unreal]の組み合わせで、未来形を生成する。
- (e) "will" の VP への付加。
- (f) 本動詞は、"will"の後ろに来るので、原形である。
- (g) 助動詞が、処理されていることを示す。

3.5 否定

★ "do not" による否定

```
(g_pd
:name VP-NEG-DO-NOT
;do notによって否定される動詞句 (現在形/過去形)
:internal_structure (VP1 AUX not VP2)
:annotation
((VP1 [[syn [[cat vp]
               [agr ?agr][vform ?vform]
               [subj-info ?subj-info]]]
      [sem [[reln negate]
             [sem-aspe (:SET stative simple)]
             [tense ?vform (:SET present past)]
             [obje ?obje [[reln (:NOT possibility be-v-exist be-v-copula)]]]]]]))
(AUX [[syn [[cat aux]
            [vform ?vform][agr ?agr]
            [lex "do"]]]]]
(not [[syn [[cat not][lex "not"]]]])
(VP2 [[syn [[cat vp]
            [vform bse][agr ?agr]
            [subj-info ?subj-info]
            [aux-do +]]]
      [sem ?obje]]]))
```

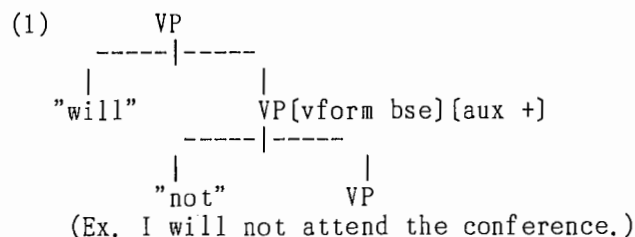
N.B. "never"による否定は、[reln negate-never]で同様に処理する。
他の副詞や代名詞、限定詞による否定は、現在、扱っていない。

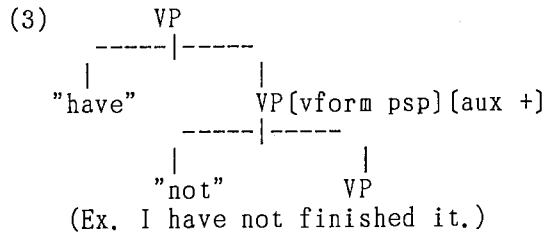
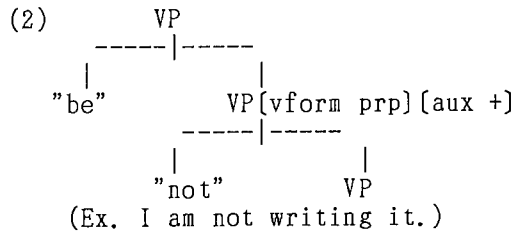
N.B. 時制素性 (tense, sem-aspe) は、独立して扱われるべきだが、
時間的な事情で、現在は否定規則とともに扱っている。

★ 時制素性より導かれた助動詞の否定

```
(g_pd
:name VP-NEG-AFTER-AUX
;助動詞文の否定
:internal_structure (VP1 not VP2)
:annotation
((VP1 [[syn [[cat vp]
               [vform ?vform (:SET bse prp psp)] ; *<===== (a)
               [agr ?agr]
               [subj-info ?subj-info]
               [be-inv ?be-inv]
               [aux +]]] ; *<===== (b)
      [sem [[reln negate]
             [obje ?obje]]]]))
(not [[syn [[cat not] ; *<===== (c)
            [lex "not"]]]])
(VP2 [[syn [[cat vp]
            [vform ?vform][agr ?agr]
            [negate +][any-env +]
            [subj-info ?subj-info]
            [be-inv ?be-inv]
            [aux +]]]
      [sem ?obje]]]))
```

(a) 動詞は (1)原形、(2)現在分詞、(3)過去分詞。





(b) 助動詞が上位に現われている。命令文に適用されるのを防ぐ。
命令文は VP-NEG-REQUEST で扱う。

(c) VP に "not" を付加する。

★ モーダルの否定

モーダルの否定では、否定、時制、モーダルに関する素性をまとめて処理する。

```

(g_pd
:name VP-NEG-POSSIBILITY-PRESENT
;canが付加された動詞句がnotによって否定される構造
:internal_structure (VP1 can not VP2)
:annotation
((VP1 [[syn [[cat vp]
              [vform (:NOT bse)][agr ?agr]
              [subj-info [[subj-sem ?expr]
                          [subj-syn ?subj-syn]
                          [subj-defsem ?subj-defsem]]]
              [modi [[tlocmodi-front -]]]]] ;;canの前にstillなどは来ない
      [sem [[reln negate]
            [obje [[reln possibility]
                  [expr ?expr :SPECIFIED]
                  [obje ?obje :SPECIFIED]]]
            [sem-aspe stative]
            [tense present]]]])
(can [[syn [[cat aux]
            [vform present][agr ?agr]
            [lex "can"]]]]])
(not [[syn [[cat not]
            [lex "not"]]]]])
(VP2 [[syn [[cat vp]
            [vform bse][agr ?agr]
            [subj-info [[subj-sem ?expr]
                        [subj-syn ?subj-syn]
                        [subj-defsem ?subj-defsem]]]]]
      [sem ?obje]]]])
  
```

★ "be" 動詞の否定

"be" 動詞の否定の場合、"not" が "be" の直後に現われるので、
VP に "not" を付加する規則では扱えず、特殊規則が必要。

```
(g_pd
:name VP-NEG-BE-NP
;be動詞+補語名詞句がnotによって否定される動詞句（現在形）
:internal_structure (VP be not NP)
:annotation
((VP [[syn [[cat vp]
          [vform ?vform][agr ?agr [[person ?person1][number ?number]]]
          [subj-info [[subj-sem ?obje]
                      [subj-syn [[agr ?agr][arti ?arti1]]]
                      [subj-defsem [[label *it*]]]]]]
      [sem [[reln negate]
            [sem-aspe (:SET simple stative)]
            [tense ?tense (:SET present past)]
            [obje [[reln be-v-copula]
                  [obje ?obje]
                  [iden ?iden]]]]]]))
(be [[syn [[cat be]
          [vform ?tense][agr ?agr]
          [lexid be-v-copula][lex "be"]]]]]))
(not [[syn [[cat not]
           [lex "not"]]]]]))
(NP [[syn [[cat np]
          [case nom][agr [[person ?person2][number ?number]]]
          [any-env +][arti ?arti2]]]
     [sem ?iden]]]))
```

3.6 受動態

```
(g_pd
:name VP-PASSIVE
;受動態の動詞句
:internal_structure (VP1 be VP2)
:annotation
((VP1 [[syn [[cat vp]
          [agr ?agr][vform ?vform]
          [subj-info ?pass-subj-info] ; *<===== (a)
          [aux-do -] ; *<===== (b)
          [lex-aspe ?lex-aspe]]]
      [sem [[reln passive] ; *<===== (c)
            [obje ?proposition]]]]))
(be [[syn [[cat be] ; *<===== (d)
          [agr ?agr][vform ?vform]
          [lexid be-v-1][lex "be"]]]]]))
(VP2 [[syn [[cat vp]
          [vform psp] ; *<===== (e)
          [subj-info [[subj-sem [[label *other*] ; *<===== (f)
                                [restr %%EMPTY%%]]]
                    [subj-defsem ?subj-defsem]]]
                    ;;[label *other*]および[]とは単一化できる
          [pass-subj-info ?pass-subj-info] ; *<===== (a)
          [pass-pp ?prep]
          [lex-aspe ?lex-aspe]]]
     [sem ?proposition]
     [slash [[syn [[cat np][case acc]]] ; *<===== (g)
             [sem ?pass-subj-sem]]]]]))
```

```

(g_pd
:name VP-PASSIVE-PP
;受動態の動詞句
:internal_structure (VP1 be VP2 PP)
:annotation
((VP1 [[syn [[cat vp]
               [agr ?agr][vform ?vform]
               [subj-info ?pass-subj-info] ;*<===== (a)
               [aux-do -] ;*<===== (b)
               [lex-aspe ?lex-aspe]]]
      [sem [[reln passive] ;*<===== (c)
             [obje ?proposition]]]])
(be [[syn [[cat be] ;*<===== (d)
           [agr ?agr][vform ?vform]
           [lexid be-v-1][lex "be"]]]])
(VP2 [[syn [[cat vp]
            [vform psp] ;*<===== (e)
            [subj-info [[subj-sem ?act] ;*<===== (f')
                        [subj-defsem ?subj-defsem]]]
            [pass-subj-info ?pass-subj-info] ;*<===== (a)
            [pass-pp ?prep] ;*<===== (h)
            [lex-aspe ?lex-aspe]]]
     [sem ?proposition]
     [slash [[syn [[cat np][case acc]]] ;*<===== (g)
              [sem ?pass-subj-sem]]]])
(PP [[syn [[cat pp][prep-lex ?prep]]] ;*<===== (h)
     [sem ?act]]]) ;*<===== (f')

```

- (a) subj-info を pass-subj-info に変換する。
VP を 個々の V とその必須要素に展開する規則には主語になる必須格の情報が能動文 (subj-info)、受動文 (pass-subj-info) 別に記載されている。能動文の場合は上位からの subj-info がそのまま VP から V を導入する PD の subj-info とマッチする。
この操作で能動文・受動文間の主語の変換を行う。
 - (b) "do" 疑問文には使えない。
 - (c) aspt, sem-aspe, tense は他の VP の場合と全く同様に処理される。
残った reln と obje をこの規則で処理する。
 - (d) "be" 動詞の付加。
 - (e) 動詞を過去分詞にする。
 - (f) 対応する能動文で主語になる意味格は 空集合[] か、
受動態の場合に省略補完された [[label *other*]] である。
具体的な意味を持つ主格が統語構造に現われず「消えてしまう」のを防ぐ。
 - (f') 対応する能動文で主語になる意味格を文末の PP として生成する。
空集合[], [[label *other*]] の場合は、統語要素に変換する規則がないので、PP の展開が失敗する。
 - (g) 受動文の VP 内部には主語に相当する「ギャップ」が存在する。
これをトレースとするために、subj-sem の値を slash の意味素性として導入。
 - (h) PP を VP の後方に生成。
lex は個々の動詞が pass-pp の値として持っているものに決定される。
- N.B. 受動態の主語は、通常、前置詞 "by" で導かれるが、動詞によっては、
"by" 以外の前置詞を用いる場合があるので、ここでは特定しない。

3.7 任意格の扱い

3.7.1 基本規則形式

任意格は、後方からと前方からのそれぞれの修飾パターンのテンプレートを2つ用意する。

```
(g_pd
:name VP-MODI-TEMPLATE
;動詞句の後方からの修飾構造
:type :template
:using nil
:internal_structure (VP1 VP2 MODI)
:annotation
((VP1 [[syn [[cat vp]
              [vform ?vform][agr ?agr]
              [subj-info ?subj-info]
              [pass-subj-info ?pass-subj-info]
              [negate ?negate]
              [tag-aux ?tag-aux]
              [pass-pp ?pass-pp]
              [adv-cl ?adv-cl]
              [be-inv ?be-inv]
              [negate ?negate]
              [synmanner ?manner]
              [aux-do ?aux-do]
              [lex-aspe ?lex-aspe]
              [s-depth ?subord]
              [any-env ?any-env]]]]
      [slash ?slash]])
 (VP2 [[syn [[cat vp][negate ?negate]
              [agr ?agr][vform ?vform]
              [tag-aux ?tag-aux]
              [subj-info ?subj-info]
              [pass-subj-info ?pass-subj-info]
              [pass-pp ?pass-pp]
              [adv-cl ?adv-cl]
              [be-inv ?be-inv]
              [synmanner ?manner]
              [aux-do ?aux-do]
              [lex-aspe ?lex-aspe]
              [any-env ?any-env]]]]
      [slash ?slash]])
 (MODI [[syn [[cat (:SET pp toinf advp conj_s)]
              [any-env ?any-env][modi-posi back]]] *<===== (g)
      [slash ?slash]]))
```

N.B. slash 素性は、テンプレートで使用する場合は、根節点と葉節点それぞれに明示しなければならない。

```

(g_pd
:name VP-TLOC-BACK
;時間を表わす副詞的語句に後方から修飾される動詞句
:type :instance
:using VP-MODI-TEMPLATE
:internal_structure (VP1 VP2 MODI)
:annotation
((VP1 [[syn [[modi [[accmmodi ?accm][causmodi ?caus] ;|
                        [destmodi -][instmodi ?inst][freqmodi -] ;|
                        [loctmodi -][mannmodi ?mann][quantmodi -] ;| *<===== (a)
                        [recpmodi -][tdepmodi ?tdep][tdesmodi -] ;|
                        [tlocmodi -]]]]]] ;|
      [sem ?proposition [[tloc ?tloc]]]]) ; *<===== (b),(f)
(VP2 [[syn [[modi [[accmmodi ?accm][causmodi ?caus]
                        [destmodi -][instmodi ?inst][freqmodi -]
                        [loctmodi -][mannmodi ?mann][quantmodi -]
                        [recpmodi -][tdepmodi ?tdep][tdesmodi -]
                        [tlocmodi +]]]]]] ; *<===== (c)
      [sem ?proposition]])
(MODI [[syn [[cat ?tloc-cat (:SET advp pp conj_s)] ; *<===== (d)
              [prep-info [[tlocprep ?prep-lex]]] ; *<===== (f)
              [prep-lex ?prep-lex (:SET "on" "in" "at"))]] ; *<===== (e)
      [sem ?tloc]]]]) ; *<===== (f)

```

- (a) modi で複数の任意格規則間の処理の順番を決定する。
tloc は dest, freq, loct などより順番として、先に適用される。
- (b) tloc の処理。
- (c) tloc が処理された（上位ですでに tloc が VP を修飾している）。
- (d) tloc は ADVP, PP, CONJ_S のいずれかのカテゴリーになる。
- (e) prep-lex は PP レベルでそれが直接支配する P の lex を指定するのに使用される。
tloc が PP に具現化される場合 P の lex は "on", "in", "at" のどれかになる。
- (f) prep-info は個々の N が任意格別に要求する P の lex の情報
(tlocprep は tloc を担った時に、現われる P) を表わし、N まで伝達される。
個々の N を生成する規則にはそれぞれの値が指定されている。
- (g) VP-MODI-TEMPLATE では修飾要素の syn に [modi-posi back] が与えられている。
これは修飾位置が VP の後方であることを示す。

VP --> X + VP という形の規則では、修飾要素 X には
[modi-posi front] が与えられている。

修飾要素が ADVP の場合 (PP, CONJ_S は必ず [modi-posi back])、
個々の ADVP の持つ、modi-posi と一致させる。

3.7.2 任意格の種類

	修飾位置	modi
accm	back	accmmodi
caus	back	causmodi
comp	back	compmodi
conect	back	conectmodi
conj	back	conjmodi
degr	back, front	degrmodi, degrmodi-front
dept	back	deptmodi
dest	back	destmodi
duration	back	duramodi
freq	back	freqmodi
goal	back	goalmodi
infmann	back	infmannmodi
inst	back	instmodi
loct	back	loctmodi
mann	back, front	mannmodi, mannmodi-front
meth	back	methmodi
purp	back	purpmodi
purpose	back	purposemodi
quant	back	quantmodi
rang	back	rangmodi
recp	back	recpmodi
role	back	rolemodi
tasp	back, front	taspmodi, taspmodi-front
tdep	back	tdepmodi
tdes-1	back	tdes1modi
tdes-2	back	tdes2modi
tloc	back	tlocmodi
tloc-order	back	tloc-ordermodi

N.B. 任意格要素の順番は、任意格の種類だけでなく、他の情報（終端語や任意格の文字列の長さなど）によって、変更が必要な場合がある。特に、mann格の要素には、いろいろな語句が入るので、一概に、他の任意格との順序を決定するのは難しい。生成出力後、構造調整などによる調整が必要である。

3.8 VP の中核構造（動詞＋必須格）

3.8.1 文型別テンプレート

文型別テンプレート（『Oxford Advanced Learner's Dictionary』に基づく）を使用する。VP を V とその必須格に展開するとともに、上位構造で導入されるが値は個々の述語に依存する素性の値が決定される。任意格規則と違い、葉節点には VP は現われないので必然的にそれより後に適用される。

```
(g_pd
:name VP-Tn.p
;動詞＋副詞的小辞＋目的語名詞句を支配する動詞句テンプレート
:type :template
:using nil
:internal_structure (VP V ADVPRT NP)
:annotation
((VP [[syn [[cat vp]
               [agr ?agr][vform ?vform]]]])
 (V [[syn [[cat verb]
            [agr ?agr][vform ?vform]]]])
 (ADVPRT [[syn [[cat advprt]]]])
 (NP [[syn [[cat np]
            [case acc]]]]))
```



```

(g_pd
:name VP-BRING_UP
;動詞bring+名詞句+upを支配する動詞句の語義規則
:type :instance
:using VP-Tn.p
:internal_structure (VP V ADVPART NP)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen] ;*<===== (a)
                        [subj-defsem [[label *unknown*]]]]] ;*<===== (b)
                        [pass-subj-info [[pass-subj-sem ?obje] ;*<===== (c)
                        [pass-subj-defsem ?obje-defsem]]] ;*<===== (d)
                        [pass-pp "by"] ;*<===== (e)
                        [lex-aspe act]]] ;*<===== (f)
[sem [[reln bring_up-v-1]
      [agen ?agen]
      [obje ?obje]]]])
(V [[syn [[lexid bring-v-1][lex "bring"]]])
(ADVPART [[syn [[lex "up"]]])
(NP [[sem ?obje] ;*<===== (g)
     [defsem ?obje-defsem [[label *it*]]]])] ;*<===== (h),(d)

```

- (a) 能動文の主語は agen。
- (b) 能動文の主語が 空集合[] の場合は、[[label *unknown*]] --> "I"。
- (c) 受動文では obje が主語になる。
- (d) 受動文のデフォルト主語は能動文の目的語のデフォルトと同じ。
- (e) 受動文では対応する能動文の主語は "by" を伴って現われる。
- (f) 個々の V が持つ語彙的アスペクトの指定。--> 下を参照。
- (g) 能動文では obje が目的語となる。
- (h) 目的語のデフォルトは "it"。

```

(g_pd
:name VP-I
;動詞 1 語を支配する動詞句テンプレート
:type :template
:using nil
:internal_structure (VP V)
:annotation
((VP [[syn [[cat vp]
            [agr ?agr][vform ?vform]]]])
(V [[syn [[cat verb]
          [agr ?agr][vform ?vform]]]]))

```

```

(g_pd
:name VP-MEET-AGEN
;動詞meet 1 語を支配する動詞句の語義規則
:type :instance
:using VP-I
:internal_structure (VP V)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen] ;*<===== (a)
                        [subj-syn [[agr [[number pl]]]]] ;*<===== (b)
                        [subj-defsem [[label *we*]]]]] ;*<===== (c)
                        [lex-aspe achiev]]]
[sem [[reln meet-v-1]
      [agen ?agen]]]])
(V [[syn [[lexid meet-1][lex "meet"]]]])

```

- (a) agen が主語になる。
- (b) 主語は必ず複数。
- (c) 主語のデフォルトは "we" 。

★文型別テンプレート一覧

『Oxford Advanced Learner's Dictionary(OALD) Fourth Edition』の
VERB PATTERN SCHEME に従う。

VP-La:	VP --> V + ADJ
VP-Ln:	VP --> V + NP
VP-I:	VP --> V
VP-Ip:	VP --> V + ADVPRT
VP-In/pr:	VP --> V + NP/PP
VP-Ipr:	VP --> V + P + NP
VP-It:	VP --> V + TOINF
VP-Tn:	VP --> V + NP
VP-Tn.pr:	VP --> V + NP + P + NP
VP-Tn.p:	VP --> V + NP + ADVPRT
VP-Tf:	VP --> V + S/CONJ_S[conj-lex "that"]
VP-Tw:	VP --> V + CONJ_S
VP-Tt:	VP --> V + TOINF
VP-Tnt:	VP --> V + NP + TOINF
VP-Tg:	VP --> V + VP[vform prp]
VP-Tsg:	VP --> V + NP[case poss] + VP[vform prp]
VP-Tng:	VP --> V + NP[case acc] + VP[vform prp]
VP-Tni:	VP --> V + NP + VP[vform bse]
VP-Cn.a:	VP --> V + NP + AP
VP-Cn.n/a:	VP --> V + NP + PP[prep-lex "as"]
VP-Cn.t:	VP --> V + NP + TOINF
VP-Cn.g:	VP --> V + NP + VP[vform prp]
VP-Cn.i:	VP --> V + NP + VP[vform bse]
VP-Dn.n:	VP --> V + NP + NP
VP-Dn.pr-TO:	VP --> V + NP + "to" + NP
VP-Dn.pr-FOR:	VP --> V + NP + "for" + NP
VP-Dn.f:	VP --> V + NP + CONJ_S[conj-lex "that"]
VP-Dpr.f:	VP --> V + "to" + NP + CONJ_S[conj-lex "that"]
VP-Dn.w:	VP --> V + NP + CONJ_S
VP-Dpr.w:	VP --> V + "to" + NP + CONJ_S
VP-Dn.t:	VP --> V + NP + TOINF
VP-Dpr.t:	VP --> V + "to" + NP + TOINF

3.8.2 lex-aspe について

個々の V について指定される情報の 1 つに lex-aspe がある。
この素性は時制関係素性の被覆のさいに参照される。

stative	...	状態
achiev	...	状態の変化を目標とし、時間の幅を持たない瞬間的行為
accomp	...	状態の変化を目標とし、時間の幅を持つ行為
act	...	状態の変化を目標とはしない行為

記載例

```
(g_pd
:name VP-HAVE
;動詞have+目的語名詞句を支配する動詞句
:type :instance
:using VP-Tn
:internal_structure (VP V NP)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen]
                        [subj-defsem [[label *unknown*]]]]]
      [pass-subj-info [[pass-subj-sem ?obje]
                        [pass-subj-defsem ?obje-defsem]]]
      [pass-pp "by"]
      [lex-aspe stative] ; *<===== (a)
      [synmanner ?synmanner]
      [any-env ?any-env]]]
  [sem [[reln have-v-1]
        [agen ?agen]
        [obje ?obje]]]])
(V [[syn [[lexid have-1] [lex "have"]]])
(NP [[syn [[any-env ?any-env] [synmanner ?synmanner]]]
     [sem ?obje]
     [defsem ?obje-defsem [[LABEL *it*]]]]))
```

(a) "have" は状態動詞。

進行形は [sem-aspe continuous] から導くが、状態動詞 ([lex-aspe stat]) が進行形になるのを防ぐために、進行形の規則に [lex-aspe stative] には使用できないという条件を入れる。

(Ex.

ESET P-504: I have the problem in making the reservation of the plane.

"have" の意味構造 (略式)。

```
[[reln have-v-1]
 [aspt prog]
 [tense present]
 [agen ?agen]
 [obje ?obje]
 [caus ?caus]
 [sem-aspe continuous]] )
```

```

(g_pd
:name VP-PROG
;進行形の動詞句
:internal_structure (VP1 be VP2)
:annotation
((VP1 [[syn [[cat vp]
[vform ?tense] ; * <===== (a)
[agr ?agr]
[subj-info ?subj-info][tag-aux ?be]
[adv-cl ?adv-cl]
[modi [[mannmodi -][mannmodi-front -][loctmodi -]
[tlocmodi -]
[tlocmodi-front -]]]
[prog -]]]
[sem ?vp-sem [[tense ?tense (:SET present past)] ; * <===== (a)
[sem-aspe continuous]]]]]) ; * <===== (b)
(be [[syn ?be [[cat be]
[vform ?tense] ; * <===== (a),(e)
[agr ?agr]
[lexid be-v-1][lex "be"]]]])
(VP2 [[syn [[cat vp]
[vform prp] ; * <===== (c)
[subj-info ?subj-info]
[lex-aspe (:NOT stative)] ; * <===== (d)
[modi [[mannmodi -][mannmodi-front -][loctmodi -]
[tlocmodi -]
[tlocmodi-front -]]]
[prog +]]]
[sem ?vp-sem]]])

```

- (a) 時制の決定。
- (b) 継続相。
- (c) 現在分詞。
- (d) 状態動詞ではない。
- (e) "be" 動詞の VP への付加。

状態動詞の場合は単純な現在形・過去形にする。

```

(g_pd
:name VP-CONTINUOUS-STATIVE
:internal_structure (VP)
:annotation
((VP [[syn [[cat vp]
[vform ?tense]
[agr ?agr]
[subj-info ?subj-info]
[adv-cl ?adv-cl]
[tag-aux ?tag-aux]
[lex-aspe stative] ; * <===== (a)
[modi [[causmodi -][degrmodi -][mannmodi-front -][taspmodi -]]]]]
[sem ?vp-sem [[tense ?tense (:SET present past)]
[sem-aspe continuous]]]]])

```

3.8.3 :SPECIFIED 及び :NULL について

(特に、VP のみに関する事柄ではないが、ここで扱うことにする。)
意味素性の被覆の際に、その素性の値が「埋まっているべき」か「空であるべき」かをそれぞれ :SPECIFIED, :NULL で指定できる。
この2つの記号の使用によって素性の値がある場合とない場合で異なる構造を生成することができる。
なお、この表記は根節点の sem にのみ使用可能。

```
(g_pd
:name VP-ASK-3-NP
;動詞ask+質問の相手を表わす名詞句を支配する動詞句
:type :instance
:using VP-Tn
:internal_structure (VP V NP)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen]
                        [subj-defsem [[label *unknown*]]]]]
      [pass-subj-info [[pass-subj-sem ?recp]
                        [pass-subj-defsem ?recp-defsem]]]
      [modi [[rangmodi -]]]
      [pass-pp "by"]
      [lex-aspe accomp]]]
  [sem [[reln ask-v-3]
        [agen ?agen]
        [recp ?recp :SPECIFIED]]]) *<===== (a)
(V [[syn [[lexid ask-v-1][lex "ask"]]]])
(NP [[sem ?recp]
     [defsem ?recp-defsem [[label *def-speaker*]]]]) *<===== (a)
```

```
(g_pd
:name VP-ASK-3
;動詞askを支配する動詞句
:type :instance
:using VP-I
:internal_structure (VP V)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen]
                        [subj-defsem [[label *unknown*]]]]]
      [pass-subj-info [[pass-subj-sem [[label *it*]]]]]
      [pass-pp "by"]
      [lex-aspe act]]]
  [sem [[reln ask-v-3]
        [agen ?agen]
        [recp :NULL]]]) *<===== (b)
(V [[syn [[lexid ask-v-3][lex "ask"]]]])
```

(a) recp (ここでは、「質問の受け手」) に値がある時は、それを目的語として生成する(他動詞)。

(b) recp に値がない時は、目的語は生成されない(自動詞)。

3.8.4 VP 中核展開レベルにおける modi素性の使用

modi 素性は VP 修飾要素同士の構造上の上位／下位関係の順番の決定に使用されることを既に述べた。

VP 中核の展開においても重要な役割を果たす場合がある。

★必須格の任意格化の防止

```
(g_pd
:name VP-RECEIVE-NP
;動詞receive+目的語名詞句を支配する動詞句
:type :instance
:using VP-Tn
:internal_structure (VP V NP)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen]
                        [subj-defsem [[label *unknown*]]]]
                        [pass-subj-info [[pass-subj-sem ?obje]
                        [pass-subj-defsem ?obje-defsem]]]
                        [pass-pp "by"]
                        [lex-aspe act]
                        [modi [[deptmodi -]]]] * <===== (b)
[sem [[reln receive-v-1] ;|
      [agen ?agen]       ;| * <===== (a)
      [obje ?obje]]]]] ;|
(V [[syn [[lexid receive-v-1][lex "receive"]]]])
(NP [[sem ?obje]
    [defsem ?obje-defsem [[label *it*]]]]))

(g_pd
:name VP-RECEIVE-NP-FROM
;動詞receive+目的語名詞句+from ...を支配する動詞句
:type :instance
:using VP-Tn.pr
:internal_structure (VP V NP1 P NP2)
:annotation
((VP [[syn [[subj-info [[subj-sem ?agen]
                        [subj-defsem [[label *unknown*]]]]
                        [pass-subj-info [[pass-subj-sem ?obje]
                        [pass-subj-defsem ?obje-defsem]]]
                        [pass-pp "by"]
                        [lex-aspe act]]]
[sem [[reln receive-v-1] ;|
      [agen ?agen]       ;|
      [obje ?obje]       ;| * <===== (a)
      [dept ?dept]]]]] ;|
(V [[syn [[lexid receive-v-1][lex "receive"]]]])
(NP1 [[sem ?obje]
      [defsem ?obje-defsem [[label *it*]]]])
(P [[syn [[lex "from"]]]])
(NP2 [[sem ?dept]]))
```

(a) receive-v-1 の必須格構成は (1) agen-obje、または (2) agen-obje-dept。

(b) [modi [[deptmodi -]]] と指定すれば、既に任意格 dept が処理された構造には適用できない。
 入力意味構造が (2) の形の時、dept が任意格として処理された後、VP-RECEIVE-NP で agen, obje のみが必須格として処理されるのを防ぐ。

N. B. 変換以降の必須格と任意格の処理については、問題となる。

この例は、receive-v-1 を語義とする場合である。

(必須格パターンについては、receive-v-1 は、決定しない場合)

★"be" 動詞の前に副詞が現われるのを防ぐ

副詞は "be" 動詞の前ではなく後ろに現われる。
(Ex. *This just is an application form.
This is just an application form.)

[modi-posi front] と指定された ADVP が "be" 動詞の前に現われるのを防ぐために、
VP を展開し、"be" を導入する規則の根節点に条件として
「そのような ADVP に上位で修飾されていない」と modi 素性を使って指定する。

```
(g_pd
:name VP-BE-NP
;be動詞+補語名詞句を支配する動詞句 (using ADJUNCTION)
:internal_structure (VP1 be ((VP2 VP3) NP))
:annotation
((VP1 [[syn [[cat vp]
[agr ?agr [[person ?person1][number ?number]]]
[vform ?vform]
[subj-info [[subj-sem ?obje]
[subj-syn [[agr ?agr]]]
[subj-defsem [[label *it*]]]]]
[lex-aspe stative]
[modi [[degrmodi-front -][mannmodi-front][taspmodi-front] *(a)
[accmmodi -][causmodi -][compmodi -] ;|
[connectmodi -][conjmodi -][degrmodi -] ;|
[deptmodi -][destmodi -][duramodi -] ;|
[freqmodi -][goalmodi -][infmannmodi -] ;| * (b)
[instmodi -][loctmodi -][mannmodi -] ;|
[methmodi -][purpmodi -][purposemodi -] ;|
[rangmodi -][recpmodi -][rolemodi -] ;|
[taspmodi -][tdepmodi -][tdes1modi -] ;|
[tdes2modi -][tlocmodi -][tloc-ordermodi -]]] ;|
[aux-do -][be-inv -]
[tag-aux ?be]]]
[sem ?sem [[reln be-v-copula] ;|
[obje ?obje] ;| *<===== (c)
[iden ?iden]]]]] ;|
(be [[syn ?be [[cat be]
[agr ?agr][vform ?vform]
[lexid be-v-copula][lex "be"]]]]]
(VP2 [[syn [[cat] ;|
[agr ?agr]]] ;|
[sem ?sem]]] ;| *<===== (d)
(VP3 [[syn [[cat] ;|
[agr ?agr]]] ;|
[sem ?sem]]] ;|
(NP [[syn [[cat np]
[case nom][agr [[person ?person2][number ?number]]]]]
[sem ?iden]]]))
```

- (a) 既に、前方から TASP や TLOC に修飾されていてはならない。
前方修飾の ADVP は adjunction 節点 VP2/VP3 の前方に付加されることになる。
従って、"be" と補語の間に現われる。
- (b) adjunction 節点を設けたために、後方修飾の ADVP や PP に関して
修飾レベルの複数の可能性 (VP1 と VP2/VP3) が生ずる。
任意格は一括して VP2/VP3 に付加されるように、根節点は後方からも
修飾されていないと指定しておく。
- (c) 任意格もこのトップレベルに現われる。
- (d) adjunction 節点。

3.8.5 コントロールについて

NP が同じ文中の他の NP と同じ物を指す時、表に現われない場合がある。

(Ex. I want you to send me an application form. ... "send" の主語は "you".
I want to send you an application form. ... "send" の意味的主語は
文全体の主語と同じ "I".)

このような場合、音形を持たない要素 <PRO> が現われていると考える。
<PRO> になる条件は control 素性で指定する。指定された条件の場合、
プログラムで自動的に<PRO> が生成される。
<PRO> は形態素処理を通過すると消える。

```
(g_pd
:name VP-WANT-TOINF
;動詞want+名詞句+to不定詞句を支配する動詞句
:type :instance
:using VP-Tnt
:internal_structure (VP V NP TOINF)
:annotation
((VP [[syn [[subj-info [[subj-sem ?expr]
                        [subj-defsem [[label *def-speaker*]]]]]]
      [synmanner direct]
      [lex-aspe stative]])
  [sem [[reln want-desire]
        [expr ?expr ]
        [obje ?obje]]]])
(V [[syn [[lexid want-3][lex "want"]]]])
(NP [[control [[sem ?expr]]] *<===== (a)
      [sem ?subj-sem] *<===== (b)
      [defsem ?expr]]]
  (TOINF [[syn [[subj-info [[subj-sem ?subj-sem] *<===== (b)
                        [subj-defsem ?subj-defsem]]]]]
          [sem ?obje]]]))
```

- (a) この NP は TOINF の意味上の主語である。(b) を見よ。
これが 主文の主語 expr と同じ (単一化できる) 時、<PRO> になる。

4 NP (名詞句) の構造

4.1 意味構造

4.1.1 中核意味構造

```
[[parm ?x]
 [restr [[reln ?reln]
         [entity ?x]]]
 [index ?index]]
```

4.1.2 修飾意味構造

(1) 限定詞

(冠詞以外の限定詞は、限定詞マーカー[構造]が付与される)

```
[[parm ?x] ; * <===== (a)
 [restr [[reln (:SET that-det-1 this-det-1 those-det-1
                  each-det-1 every-det-1 such-det-1
                  some-det-1 some-det-1 any-det-1
                  another-det-1 a_few-det-1))
         [obje ?x]]]]]
```

(a) ?x は基本的意味構造。被修飾要素 (統語構造の中心語に相当)。

N.B. parm の ?x には、:SPECIFIEDの指定が、(2)の場合のように必要。
入力構造に対して、非構造を許さないための方法である。
以下、(3)(4)(5)(6) も同様である。

(2) 所有格

```
[[parm ?x :SPECIFIED] ; * <===== (a)
 [restr [[reln possessive]
         [agen ?agen] ; * <===== (b)
         [obje ?x]]]]]
```

(a) ?x は基本的意味構造。被修飾要素。

(b) 修飾要素。所有格名詞になる。

(3) PP や ADVP などによる修飾

```
[[parm ?x] ; * <===== (a)
 [restr [[reln modify]
         [arg1 ?y] ; * <===== (b)
         [arg2 ?x]]]]]
```

(a) ?x は基本的意味構造。被修飾要素。

(b) 修飾要素の意味構造。

(4) PP による修飾

```
[[parm ?x] ; *<===== (a)
 [restr [[reln (:SET modify-unit modify-loct modify-loct-near
                  modify-goal-for modify-pod modify-tloc
                  modify-tloc-after modify-dest modify-dest-for
                  modify-dept modify-rang modify-before
                  modify-agen-by modify-purp-for modify-in_charge_of
                  modify-prior-to modify-conj-and))]
          [arg1 ?y] ; *<===== (b)
          [arg2 ?x]]]]
```

(a) ?x は基本的意味構造。被修飾要素。

(b) 修飾要素の意味構造。

(5) REL_S, AP による修飾

```
[[parm ?x] ; *<===== (a)
 [restr ?restr] ; *<===== (b)
```

(a) ?x は基本的意味構造。被修飾要素。

(b) 修飾要素の意味構造。命題型の構造（関係名と必須、任意格）を持つ。

(6) NUMB による修飾

```
[[parm ?x] ; *<===== (a)
 [restr [[reln modify]
          [arg1 [[parm ?y]
                  [restr [[reln number] ;|
                          [iden ?iden] ; *<===== (b) ;| ; *<===== (c)
                          [entity ?y]]]]] ;|
          [arg2 ?x]]]]]
```

(a) ?x は基本的意味構造。被修飾要素。

(b) 修飾要素の意味構造。

(c) このレベルに [unit ?unit] が存在する場合もある。

4.2 名詞的統語カテゴリーの種類

NP..... 名詞句

名詞を中心とする完結した句で、それ自体で主語や目的語として使用可能
the intelligent boy, the report which you sent me, Mr. Ken Brown,

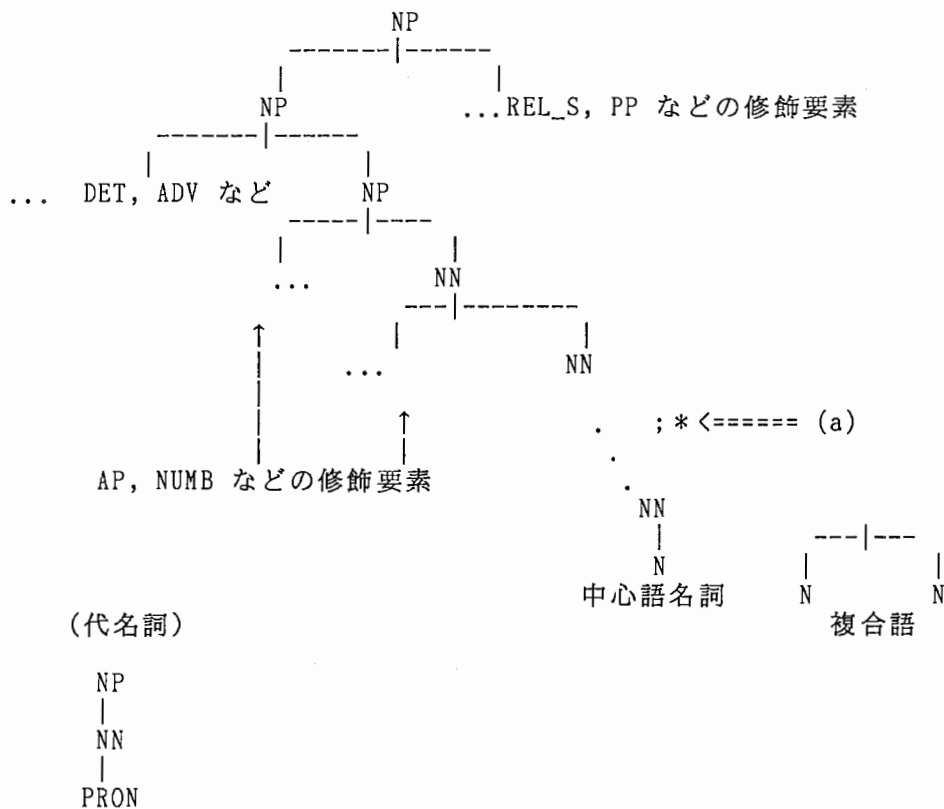
NN..... NPとNの中間レベル, NPから限定詞（冠詞や指示詞）を除いたもの
intelligent boy, conference office

N 名詞

単語レベル

boy, conference, desk, Ken, bus

4.3 階層構造の概略



(a) NN レベルは繰り返しが必要。
(Ex. tall handsom intelligent boy)

4.4 階層構造別の規則

4.4.1 NPレベル

NP は、完結した名詞句で、主語や目的語になることができる。

★ 冠詞 "a", "the"、無冠詞

"a", "the" は意味構造に存在しない。

NP-NN (NP --> NN), NP-A-NN (NP --> "a" + NN) などの規則は

sem に無関係に適用される。

適用の際に必要な素性を導入し、それを N レベルまで伝達し、
N が固有に持つ素性値と照合することによって、不適切な構造を排除する。

N.B. 不定冠詞の "a" から "an" への形態変形は、形態素処理において、
プログラムで行なわれる。

N.B. NP -> the + NN
NP -> a + NN
NP -> NN

の3つのルールを統合して、NP -> DETER + NN とし、

DETER に、必要な素性を集めておいて、
MGネットなどで、"" , "the", "a" を出力する方法も考えられる。

```

(g_pd
:name NP-A-NN
;冠詞a+NNを支配する名詞句
:internal_structure (NP DET NN)
:annotation
((NP [[syn [[cat np][wh -]
      [case ?case] ;*<===== (a)
      [agr ?agr [[person 3][number sing]]] ;*<===== (b)
      [prep-info ?prep-info] ;*<===== (a)
      [arti a] ;*<===== (c)
      [insert-v ?insert-v]]] ;*<===== (a)
      [sem ?sem]])
 (DET [[syn [[cat det] ;*<===== (d)
              [lex "a"]]]])
 (NN [[syn [[cat nn]
            [case ?case] ;*<===== (a)
            [agr ?agr] ;*<===== (b)
            [prep-info ?prep-info] ;*<===== (a)
            [arti a] ;*<===== (c)
            [insert-v ?insert-v]]] ;*<===== (a)
            [sem ?sem]]]))))

```

- (a) 格、前置詞情報、挿入動詞情報を下位に伝える。
 (b) "a" が付加されれば、3人称、単数。--> 下位に伝達。
 (a), (b) が N が固有に持つ素性と矛盾すれば、その派生は失敗する。
 (c) 「冠詞 "a" が付加された」という情報を伝達する。
 上位で決定されている可能性もあり。
 (d) "a" の生成。

★that などによる修飾

"that" "each" などの名詞限定要素は意味構造中にマーカーとして、存在する。

```

(g_pd
:name NP-THAT-NN
;that+NNを支配する名詞句
:internal_structure (NP DET NN)
:annotation
((NP [[syn [[cat np][wh -]
      [case ?case][agr ?agr [[person 3][number sing]]] ;*<===== (a)
      [prep-info ?prep-info]]]
      [sem [[parm ?x] ;|
            [restr [[reln that-det-1] ;| ;*<===== (a)
                    [obje ?x]]]]]] ;|
 (DET [[syn [[cat det]
              [lexid that-det-1][lex "that"]]]])
 (NN [[syn [[cat nn]
            [case ?case][agr ?agr]
            [prep-info ?prep-info]]]
      [sem ?x]]]))))

```

- (a) "that" の意味構造の被覆。
 (b) 3人称単数。

N.B. 日本語文中に、『その本』というように、explicitに現われる場合である。

★所有格 NP による修飾

```
(g_pd
:name NP-POSS-NN-1
;名詞句が所有格の代名詞に修飾されている構造
:internal_structure (NP POS NN)
:annotation
((NP [[syn [[cat np][wh -]
           [case ?case][agr ?agr [[person ?pers][number sing]]]
           [prep-info ?prep-info]
           [insert-v ?insert]]]
      [sem [[parm ?x :SPECIFIED]
            [restr [[reln possessive]
                    [agen ?agen] ;*<=====代名詞の場合
                    [obje ?x]]]]]])
(POS [[syn [[cat (:SET np)]
            [case gen]]]
      [sem ?agen]])
(NN [[syn [[cat nn]
           [case ?case][agr ?agr]
           [prep-info ?prep-info]
           [insert-v ?insert]]]
     [sem ?x]]]))
```

N. B. 一般名詞の場合のアポスロフィ処理は、MGネットで行なわれる。

★PP による後方からの修飾

```
(g_pd
:name NP-NP-PP-MODIFY
;名詞句が前置詞句に修飾されている構造
:internal_structure (NP1 NP2 PP)
:annotation
((NP1 [[syn [[cat np][wh -]
           [case ?case][agr ?agr]
           [prep-info ?prep-info]
           [any-env ?any-env][arti ?arti][insert-v ?insert-v]]]
      [sem [[parm ?x2] ;*<===== (a)
            [restr [[reln modify]
                    [arg1 ?x1 [[restr [[reln (:NOT tomorrow-adv-1
                                         number hand_writing-n-1
                                         about-adv-1
                                         ordinary-number
                                         as_many_as-adj-1
                                         only-adv-1]]]]]]]] ;* <===== (b)
            [arg2 ?x2]]]]]])
(NP2 [[syn [[cat np][wh -]
           [case ?case][agr ?agr]
           [prep-info ?prep-info]
           [any-env ?any-env][arti ?arti][insert-v ?insert-v]]]
      [sem ?x2]]] ;* <===== (a)
(PP [[syn [[cat pp]
          [prep-info [[modiprep ?modiprep (:SET "of" "on" "for" "by")]]] ;* <===== (d)
          [prep-lex ?modiprep]]]
     [sem ?x1]]])) ;* <===== (b)
```

- (a) 被修飾要素（中心語）の意味の伝達。
- (b) 修飾要素（PP）の意味の伝達。
- (c) modify 意味構造は PP 以外の要素に具現化される場合がある。
予め、除外できるものを指定しておく。
- (d) modify 構造が PP として現われる場合の P の lex を modiprep で現わす。
個々の N が持つ modiprep の値と照合することによって、1つに決定される。

```

(g_pd
:name NP-NP-PP-LOCT
;名詞句が場所を表わす前置詞句に修飾されている構造
:internal_structure (NP1 NP2 PP)
:annotation
((NP1 [[syn [[cat np][wh -]
               [case ?case][agr ?agr]
               [prep-info ?prep-info]
               [arti ?arti]]]
      [sem [[parm ?x]
             [restr [[reln modify-loct]
                     [arg1 ?loct]
                     [arg2 ?x]]]]]])
(NP2 [[syn [[cat np][wh -]
               [case ?case][agr ?agr]
               [prep-info ?prep-info]
               [arti ?arti]]]
      [sem ?x]])
(PP [[syn [[cat pp]
            [prep-info [[loctprep ?loctprep (:SET "at" "in")]]] ;*<===== (a)
            [prep-lex ?loctprep]]]
     [sem ?loct]]]))

```

(a) loct (場所格) の時、現われる P を loctprep で指定する。N レベルで決定。

N.B. 前置詞マーカーがないので、意味格名（この場合は、LOCT格）から、前置詞 P を生成しなければならない。すなわち、この方法は、前置詞とその前置詞がとる目的語との共起関係を見ている。

★REL_S による修飾

```

(g_pd
:name NP-NP-REL
;NPが関係節によって修飾されている構造
:internal_structure (NP1 NP2 REL_S)
:annotation
((NP1 [[syn [[cat np]
               [case ?case][agr ?agr]
               [prep-info ?prep-info]
               [any-env ?any-env][arti ?arti][insert-v ?insert-v]
               [synmanner ?synmanner]]] ;*<===== (a)
      [sem [[parm ?y :SPECIFIED]
             [restr ?restr [[reln (:NOT modify modify-conj-and)]]]]]] ;*<===== (b)
(NP2 [[syn [[cat np][wh ?wh] ;*<===== (c)
               [case ?case][agr ?agr]
               [prep-info ?prep-info]
               [any-env ?any-env][arti ?arti][insert-v ?insert-v]]]
      [sem ?y]])
(REL_S [[syn [[cat rel_s]
               [vform ?vform]
               [synmanner ?synmanner] ;*<===== (a)
               [wh-ante ?wh]]] ;*<===== (c)
        [sem ?restr]
        [slash [[syn [[cat np]
                       [agr ?agr]
                       [prep-info ?prep-info]]]
                 [sem ?y]]]]])) ;*<===== (d)

```

(a) 上位から S --> VP --> 目的語 NP と伝達されてきた synmanner を REL_S に伝える。synmanner は文全体が「直接的な発話か、間接的な発話か」を示すが、この情報が REL_S の内部で統語的に実現される場合がある。
(Ex. ESET p-10: Well, I have something that I just <would like to> ask you.)

(b) 最終的には失敗する派生を予め防いでおく。

(c) wh-ante は「先行詞が WH かどうか」を表わす。
先行詞が WH ("what") の場合、先行詞と関係代名詞が「融合」しているので
"that" を生成しない。(*what that I said ...)
(Ex. ESET p-588: It seems that what I said wasn't comprehended.)
このために、REL_S 展開には次の 2 つの規則を用意する。

```
REL_S-1: REL_S[wh-ante -] --> "that" + S
REL_S-2: REL_S[wh-ante +] --> S
```

(d) REL_S 内部には先行詞に対応する「ギャップ」が存在する。

★ ADVP による修飾

```
(g_pd
:name NP-NP-ADVP
;NPが副詞句に後方から修飾される構造
:internal_structure (NP1 NP2 ADVP)
:annotation
((NP1 [[syn [[cat np][wh -]
              [case ?case][agr ?agr]
              [prep-info ?prep-info]
              [arti the]]]
      [sem [[parm ?x]
            [restr [[reln modify]
                    [arg1 ?arg1 [[parm ?y]
                                  [restr [[reln (:SET last_year-adv-1
                                                         next_month-adv-1
                                                         next_time-adv-1
                                                         next_year-adv-1
                                                         this_year-adv-1
                                                         this_time-adv-1
                                                         today-adv-1
                                                         tomorrow-adv-1))
                                  (entity ?y))]]]]]]]]
      {arg2 ?x}]]]]))
(NP2 [[syn [[cat np][wh -]
              [case ?case][agr ?agr]
              [prep-info ?prep-info]
              [arti the]]]
      [sem ?x]])
(ADVP [[syn [[cat advp]]]
       [sem ?arg1]]))
```

(a) 後方から NP を修飾する ADVP を指定する。
(Ex. the conference this time)

N.B. (a)に終端レベルの語義名を指定しなければならない。
副詞と名詞の構造が区別がつかないことによる。
tomorrow-ADV-1 と book-N-1 とが、全く同じ構造を持つからである。

4.4.2 NNレベル ★NUMB による修飾

```
(g_pd
:name NN-NUMB-NN
;NNが数詞に修飾されている構造
:internal_structure (NN1 NUMB NN2)
:annotation
((NN1 [[syn [[cat nn]
[case ?case][agr ?agr]
[prep-info [[loctprep ?loctprep][tloctprep ?tloctprep]
[modiprep "of"]]]]
[arti ?arti (:SET - the)][insert-v ?insert-v]]]
[sem [[parm ?x [[restr [[reln (:NOT yen-n-1 dollar-n-1)]]]]]] ;*<===== (a)
[restr [[reln modify]
[arg1 [[parm ?y]
[restr [[reln number] ;|
[iden ?iden] ;| ;*<===== (b)
[entity ?y]]]]] ;|
[arg2 ?x]]]]]]))
(NUMB [[syn [[cat numb]
[case ?case]
[arti -][agr ?agr]]] ;*<===== (c)
[sem ?iden]])
(NN2 [[syn [[cat nn]
[case ?case][agr ?agr] ;*<===== (c)
[prep-info [[loctprep ?loctprep][tloctprep ?tloctprep]]]
[arti ?arti][insert-v ?insert-v]]]
[sem ?x]]]))
```

- (a) 被修飾語が「金銭の単位」の場合は適用しない。--> 下を参照。
(b) unit 素性が存在しても、トップレベルではないので被覆の必要はない。
(c) NUMB と NN 間の数の一致をとらえる。

```
(g_pd
:name NN-NUMB-NN-MONEY
;NNが数詞に修飾されている構造
:internal_structure (NN1 NUMB NN2)
:annotation
((NN1 [[syn [[cat nn]
[case ?case][agr [(person 3)(number sing)]]] ;*<===== (a)
[prep-info [[loctprep ?loctprep][tloctprep ?tloctprep]
[modiprep "of"]]]]
[arti ?arti (:SET -)][insert-v ?insert-v]]]
[sem [[parm ?x [[restr [[reln (:SET yen-n-1 dollar-n-1)]]]]]] ;*<===== (b)
[restr [[reln modify]
[arg1 [[parm ?y]
[restr [[reln number]
[iden ?iden]
[entity ?y]]]]]
[arg2 ?x]]]]]]))
(NUMB [[syn [[cat numb]
[case ?case]
[arti -][agr ?agr]]] ;*<===== (c)
[sem ?iden]])
(NN2 [[syn [[cat nn]
[case ?case][agr ?agr] ;*<===== (c)
[prep-info [[loctprep ?loctprep][tloctprep ?tloctprep]]]
[arti ?arti][insert-v ?insert-v]]]
[sem ?x]]]))
```

- (a) 金銭の単位を表わす名詞は必ず単数扱いにする。
(Ex. p-266: <Does> fifty thousand yen meet the accomodation fee.)
(b) 被修飾語の指定。
(c) 形態的には複数。

★ AP による修飾

--> AP の項を見よ。

★ INDEX 素性の被覆

INDEX 素性は NN レベルで以下の規則で被覆する。

現在、生成では INDEX 素性は被覆するだけで、積極的に活用してはいない。

```
(g_pd
:name NN-INDEX-COVER
;NNレベルで名詞のINDEXを被覆する
:internal_structure (NN)
:annotation
((NN [[syn [[cat nn]
             [case ?case][agr ?agr]
             [prep-info ?prep-info]
             [arti ?arti][insert-v ?insert-v]]]
      [sem [[index ?index]]]]))
```

★ NN レベルで終端語が導入される場合 (複合語)

```
(g_pd
:name NN-ACCOMODATION_FEE
;名詞複合accomodation_fee
:internal_structure (NN N1 N2)
:annotation
((NN [[syn [[cat nn]
             [case ?case][agr ?agr [[person 3][number sing]]]
             [prep-info [[modiprep "of"]]]
             [arti the]]]
      [sem [[parm ?x]
            [restr [[reln accomodation_fee-n-1]
                    [entity ?x]]]]]]))
(N1 [[syn [[cat noun]
           [case nom][agr [[person 3][number sing]]]
           [lexid accomodation-n-1][lex "accomodation"]]]])
(N2 [[syn [[cat noun]
           [case ?case][agr ?agr]
           [lexid fee-n-1][lex "fee"]]]]]))
```

4.3.3 Nレベル (語彙のレベル)

名詞型の中心意味構造を被覆して終端語を生成する。

以下の素性が NP, NN レベルから伝達されて来て、(指定があれば) 決定されたり、照合されたりする。

- (a) [case ?case]
- (b) [agr ?agr]
- (c) [arti ?arti]
- (d) [prep-info [[instprep ?instprep][methprep ?methprep][modiprep ?modiprep]
 [loctprep ?loctprep][tlocprep ?tlocprep]]]
- (e) [insert-v ?insert-v]

大抵の場合、N 導入規則は N-COMMON-N-TEMPLATE との組み合わせで使用。

N-COMMON-N-TEMPLATE では modiprep のデフォルトとして、"of" を指定する。

```
(g_pd
:name N-COMMON-N-TEMPLATE
;普通名詞
:type :template
:using nil
:internal_structure (N)
:annotation
((N [[syn [[cat noun]
          [case ?case][agr [[person 3]]]
          [prep-info [[modiprep "of"]]]]]]
  [sem [[parm ?x]
        [restr [[entity ?x]]]]]]))
```

```
(g_pd
:name N-ADDRESS
;名詞address
:type :instance
:using N-COMMON-N-TEMPLATE
:internal_structure (N)
:annotation
((N [[syn [[agr [[number (:SET sing pl)]]] ;*<===== (a)
          [arti (:SET the)] ;*<===== (b)
          [insert-v (:SET give have)] ;*<===== (c)
          [lexid address-n-1][lex "address"]]]]
  [sem [[restr [[reln (:SET address-n-1 address-n-2)]]]]]]))
```

- (a) "address" は単数の場合と複数の場合、両方が可能。
 (b) 冠詞は "the" と指定する (暫定的)。
 (c) "address" の意味のみが VP をなす場合、補われる V は "give" または "have"。
 (Ex. MSET d8-13:
 Then, I'll send you an application form, so please give me the address.
 Then, I'll send you an application form, so may I have the address.)

```
(g_pd
:name N-BEGINNING
;名詞beginning
:type :instance
:using N-COMMON-N-TEMPLATE
:internal_structure (N)
:annotation
((N [[syn [[agr [[number sing]]] ;*<===== (a)
          [prep-info [[loctprep "at"]][tlocprep "at"]]] ;*<===== (b)
          [arti the]
          [lexid beginning-n-1][lex "beginning"]]]]
  [sem [[restr [[reln beginning-n-1]]]]]]))
```

- (a) 単数。
 (b) loctprep, tlocprep とともに "at" を使用する。

4.4.4 固有名詞

★基本意味構造

```
[[parm ?x]
 [restr [[reln named]
         [iden ?iden]
         [entity ?x]]]]]
```

★規則

```
(g_pd
:name N-PROPN-TEMPLATE
;固有名詞
:type :template
:using nil
:internal_structure (N)
:annotation
((N [[syn [[cat noun]
            [case ?case][agr [[person 3][number sing]]]
            [prep-info [[modiprep "of"]]]]]]
 [sem [[parm ?x]
       [restr [[reln named]
               [entity ?x]]]]]])))]
```

```
(g_pd
:name N-AMERICA
;名詞America
:type :instance
:using N-PROPN-TEMPLATE
:internal_structure (N)
:annotation
((N [[syn [[arti -]
            [lexid America-propn][lex "America"]]]]
 [sem [[restr [[iden America-propn]]]]]])))]
```

5 AP (形容詞句) の構造

5.1 基本意味構造

VP と同じ形の意味構造で、reln を見なければ形式からは区別できない。

```
Ex.  [[reln ?reln] ; * <===== (a)
      [obje ?obje] ; |
      .           ; | ; * <===== (b)
      .           ; |
      [loct ?loct] ; |
      [tloc ?tloc] ; |
      .           ; | ; * <===== (c)
      [aspt ?aspt ] ; |
      [sem-aspe ?sem-aspe] ; | ; * <===== (d)
      [tense ?tense]] ; |
```

- (a) Ex. same-adj-1
- (b) 必須格。
- (c) 任意格。
- (d) 時制関係素性。

次に示すのは AP が NP を修飾する場合の意味構造。

```
[[parm ?x] ; * <===== (a)
 [restr ?restr]] ; * <===== (b)
```

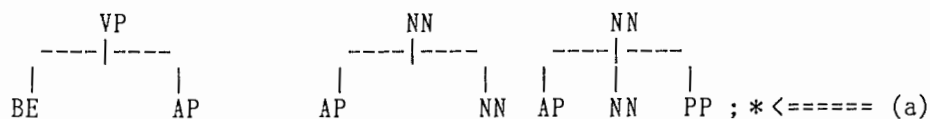
- (a) ?x は被修飾要素の意味構造。
?x に対しては、:SPECIFIED が必要と思われる。
- (b) AP の意味構造。
関係節となる可能性もある。
(Ex. an interesting topic
an topic that is interesting)

5.2 基本統語構造

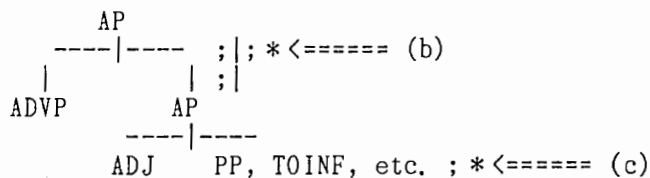
(1) 叙述用法

(2) 限定用法

me) 1. (Ex. ESET p-422: the same way as the last ti



(3) 内部構成



- (a) PP は形容詞の必須格。
- (b) 強意語 (Ex. "very") による修飾。
- (c) 必須格構造。限定用法の場合は (2-1) の構造になる。

5.3 形容詞規則

5.3.1 叙述用法

時制、DEGR 以外の任意格は VP と全く同様の仕組みで扱われる。
その後、次のような規則で "be" 動詞と AP に展開される。

```
(g_pd
:name VP-BE-AP
;be動詞+補語形容詞句を支配する動詞句
:internal_structure (VP be AP)
:annotation
((VP [[syn [[cat vp]
[agr ?agr][vform ?vform]
[subj-info ?subj-info [[subj-sem ?subj-sem]
[subj-syn [[agr ?agr][arti ?arti]]]
[subj-defsem ?subj-defsem]]]
[modi ?modi [[degrmodi -][degrmodi-front -][goalmodi ?goalmodi]
[tlocmodi-front -]]]
[any-env ?any-env][aux-do -][lex-aspe stative]
[be-inv -][pre-poss-cover -]
[rel-cl ?rel-cl]]] ; *<===== (a)
[sem ?sem]])
(be [[syn [[cat be]
[vform ?vform][agr ?agr]
[lexid be-v-1][lex "be"]]]])
(AP [[syn [[cat ap]
[subj-info ?subj-info]
[modi ?modi]
[any-env ?any-env]
[adj-posi pred] ; *<===== (b)
[rel-cl ?rel-cl]]] ; *<===== (a)
[sem ?sem]]]))
```

- (a) 「関係節の内部かどうか」を表わす。中心語の A まで伝達し、A と照合する。
例えば、"many" に [rel-cl -] を指定しておけば、
"participants that are many" などを防げる。
- (b) 「叙述用法である」という情報。

5.3.2 限定用法

```
(g_pd
:name NN-AP-NN-1
;形容詞句+NNを支配するNN
:internal_structure (NN1 AP NN2)
:annotation
((NN1 [[syn [[cat nn]
               [case ?case][agr ?agr [(person 3)[number ?adj-num]]] ;*<===== (b)
               [prep-info ?prep-info]
               [arti ?arti][insert-v ?insert-v]]]
      [sem [[parm ?x [[parm ?y]
                       [restr [[reln ?reln]]]]]
            [restr ?restr [[reln (:NOT modify modify-conj-and about-adv-1))] ]]]])
(AP [[syn [[cat ap]
            [adj-arti ?arti] ;*<===== (a)
            [adj-num ?adj-num] ;*<===== (b)
            [adj-posi pre-nn]]] ;*<===== (c)
     [sem ?restr]])
(NN2 [[syn [[cat nn]
            [case ?case][agr ?agr]
            [prep-info ?prep-info]
            [arti ?arti] ;*<===== (a)
            [insert-v ?insert-v]]]
     [sem ?x]]]))
```

- (a) この AP の中心語の A が被修飾 NN に要求する冠詞。
(Ex. "entire" ... [adj-arti the])
- (b) この AP の中心語の A が被修飾 NN に要求する数。
(Ex. "various" ... [adj-num pl])
- (a), (b) でADJ, N 間の一致を扱う。
- (c) NN の前の位置 (限定用法)。

```
(g_pd
:name NN-AP-NN-PP
;形容詞句+NN+前置詞句を支配するNN (the same ... as ..., etc.)
:internal_structure (NN1 AP NN2 PP)
:annotation
((NN1 [[syn [[cat nn]
               [case ?case][agr ?agr [(person 3)[number ?adj-num]]]
               [prep-info ?prep-info]
               [arti ?arti][insert-v ?insert-v]]]
      [sem [[parm ?x [[parm ?y]
                       [restr [[reln ?reln]]]]]
            [restr ?restr [[reln (:NOT modify modify-conj-and))] ]]]])
(AP [[syn [[cat ap]
            [adj-arti ?arti]
            [adj-num ?adj-num]
            [adj-posi split] ;*<===== (a)
            [adj-pp-sem ?adj-pp-sem] ;*<===== (b)
            [adj-prep-lex ?adj-prep-lex]]] ;*<===== (c)
     [sem ?restr]])
(NN2 [[syn [[cat nn]
            [case ?case][agr ?agr]
            [prep-info ?prep-info]
            [arti ?arti][insert-v ?insert-v]]]
     [sem ?x]])
(PP [[syn [[cat pp]
            [prep-lex ?adj-prep-lex]]] ;*<===== (c)
     [sem ?adj-pp-sem]])) ;*<===== (b)
```

- (a) 中心語 (ADJ) と必須格 (PP) が NN を挟んで分離している。
 (b) PP の意味。個々の A を導入する規則で決定される。--> 下の (b) を見よ。
 (c) P の lex。個々の A を導入する規則で決定される。--> 下の (a) を見よ。
 (b), (c) によって、叙述用法では AP 内部に現われる 必須格の情報が
 限定用法では NN の後ろに転送する。

```
(g_pd
:name ADJ-SAME-ATTRI
:sameを支配する形容詞
:type :instance
:using ADJ-TEMPLATE
:internal_structure (ADJ)
:annotation
((ADJ [[syn [[adj-arti the] ;*<===== (a)
            [adj-posi split] ;*<===== (b)
            [adj-prep-lex "as"] ;*<===== (c)
            [adj-pp-sem ?comp] ;*<===== (d)
            [lexid same-adj-1][lex "same"]]]
      [sem [[reln same-adj-1]
            [comp ?comp]]]])) ;*<===== (d)
```

(ADJ-TEMPLATE とともに使用。必須格の obje はテンプレートで被覆。)

- (a) "same" は "the" を伴う。
 (b) 限定用法の場合のみ使用される。
 叙述用法の場合は AP-SAME-PRED (AP --> "the" + "same" + "as ...") を使用。
 (c) NN の後ろに現われる補語必須格の P は "as"。
 (d) NN の後ろに現われる補語必須格は comp。

★限定用法の時制素性の扱い

NN-AP-NN-1, NN-AP-NN-PP では、動詞型の意味構造がそのまま
 AP に伝達されるので、
 時制関係素性はまだ未処理である。
 このため、次の単節点の規則が必要となる。

```
(g_pd
:name AP-TENSE-COVER
:internal_structure (AP)
:annotation
((AP [[syn [[cat ap]
            [subj-info ?subj-info]
            [ap-modi -] ;*<===== (a)
            [adj-num ?adj-num]
            [adj-posi (:NOT pred)] ;*<===== (b)
            [degree ?degree]
            [adj-prep-lex ?adj-prep-lex]
            [adj-pp-sem ?adj-pp-sem]]]
      [sem [[aspt ?aspt] ;|
            [tense ?tense] ;| ;*<===== (c)
            [sem-aspe ?sem-aspe]]]])) ;|
```

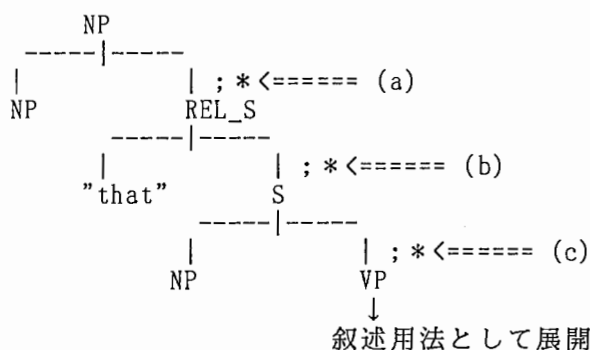
- (a) AP は既に修飾されてはならない。
 AP が DEGR によって修飾されている場合、DEGR より先に被覆される。
 適用順序の多義性をなくす。
 (b) 叙述用法の場合は適用されない。
 (c) 限定用法の場合、時制関係素性は単に被覆されるだけ。

★関係節との関係

AP が名詞を修飾する意味構造は次のように関係節としても具現化されうる。

(Ex. ESET p-31: The fastest means of transportation is the Shinkansen.

The means of transportation that is the fastest is the Shinkansen.)



(a) NP-NP-REL による。

(b) REL_S による。

(c) S-SUBORD による。

--> NP の項を参照。

★任意格

DEGR, INFMANN 以外の任意格を AP レベルで被覆する規則は存在しない。

従って、それらが現われた場合、関係節構造の VP レベルで被覆されなければならない、AP が直接 NN を修飾する構造 (限定用法) は許されない。

5.3.3 叙述・限定両用法に共通の規則

★ DEGR による修飾

(AP の修飾規則には他に INFMANN による修飾を扱う AP-INFMANN-AP もある。)

```

(g_pd
:name AP-DEGR-AP
;程度副詞句に修飾される形容詞句
:internal_structure (AP1 ADVP AP2)
:annotation
((AP1 [[syn [[cat ap]
              [subj-info ?subj-info]
              [adj-num ?adj-num]
              [adj-posi ?adj-posi (:SET pred pre-nn)]
              [rel-cl ?rel-cl]]]
       [sem ?sem [[degr ?degr [[restr [[reln (:SET quite-adv-1
                                                rather-adv-1
                                                very-adv-1)]]]]]]]]))

(ADVP [[syn [[cat advp]
              [modi-posi adj-front]]] ; * <===== (b)
       [sem ?degr]])

(AP2 [[syn [[cat ap]
              [subj-info ?subj-info]
              [ap-modi +] ; * <===== (a)
              [adj-num ?adj-num]
              [adj-posi ?adj-posi]
              [rel-cl ?rel-cl]]]
       [sem ?sem]]))
  
```



```

(g_pd
:name AP-AP-DEGR
;程度副詞句に後方から修飾される形容詞句
:internal_structure (AP1 AP2 ADVP)
:annotation
((AP1 [[syn [[cat ap]
               [subj-info ?subj-info]
               [adj-num ?adj-num]
               [adj-posi ?adj-posi (:SET pred pre-nn)]
               [rel-cl ?rel-cl]]]
      [sem ?sem [[degr ?degr [[restr [[reln (:SET at_all-adv-1
                                                enough-adv-1))]]]]]]))
 (AP2 [[syn [[cat ap]
               [subj-info ?subj-info]
               [ap-modi +] ; *<===== (a)
               [adj-num ?adj-num]
               [adj-posi ?adj-posi]
               [rel-cl ?rel-cl]]]
      [sem ?sem]])
 (ADVP [[syn [[cat advp]
               [modi-posi back]]] ; *<===== (b)
      [sem ?degr]]))

```

- (a) 既に上位で修飾されている。
(b) ADVP の位置の指定。

★比較級・最上級

```

(g_pd
:name AP-MORE-COVER
;比較級の形容詞句
:internal_structure (AP)
:annotation
((AP [[syn [[cat ap]
               [subj-info ?subj-info]
               [adj-num ?adj-num]
               [adj-posi (:SET pred pre-nn)]
               [degree comparative] ; *<===== (a)
               [adj-arti ?adj-arti]]]
      [sem [[degr [[parm ?x] ;|
                    [restr [[reln more-adv-1] ;| ; *<===== (b)
                    [entity ?x]]]]]]]])) ;|

```

- (a) MG ネット で形態素処理は行なわれる。
(b) "more" の副詞構造 を指定して処理する。

```

(g_pd
:name AP-MOST-COVER
;最上級の形容詞句
:internal_structure (AP)
:annotation
((AP [[syn [[cat ap]
               [subj-info ?subj-info]
               [adj-num ?adj-num]
               [adj-posi pre-nn] ; *<===== (a)
               [degree superlative] ; *<===== (b)
               [adj-arti the]]] ; *<===== (c)
      [sem [[degr [[parm ?x] ;|
                    [restr [[reln most-adv-1] ;| ; *<===== (d)
                    [entity ?x]]]]]]]])) ;|

```

- (a) 限定用法の場合のみ使用。
(b) MG ネット で形態素処理は行なわれる。
(c) 最上級は修飾する NN に "the" を要求する。
(d) "most" の副詞構造 を指定して処理する。

```

(g_pd
:name AP-MOST-COVER-PRED
;最上級の形容詞句
:internal_structure (AP1 DET AP2)
:annotation
((AP1 [[syn [[cat ap]
               [subj-info ?subj-info]
               [adj-num ?adj-num]
               [adj-posi pred] ; * <===== (a)
               [rel-cl ?rel-cl]]]
      [sem ?sem [[degr [[parm ?x]
                        [restr [[reln most-adv-1]
                                [entity ?x]]]]]]]]))
(DET [[syn [[cat det][lex "the"]]]]) ; * <===== (b)
(AP2 [[syn [[cat ap][subj-info ?subj-info]
            [adj-num ?adj-num]
            [adj-posi pred]
            [rel-cl ?rel-cl]
            [degree superlative]]]
     [sem ?sem]]))

```

(a) 叙述用法の場合のみに使用。

(b) 限定用法の場合のように "the" を付加する NN が存在しないので、AP に付加する。

★ 終端語の導入

AP が A のみを支配する時は、AP-ADJ でまず A レベルに落とす。

```

(g_pd
:name AP-ADJ
;形容詞一語を支配する形容詞句
:internal_structure (AP ADJ)
:annotation
((AP [[syn [[cat ap]
             [subj-info ?subj-info]
             [modi ?modi]
             [adj-num ?adj-num]
             [adj-posi ?adj-posi (:SET pred pre-nn split)]
             [adj-arti ?adj-arti]
             [degree ?degree]
             [rel-cl ?rel-cl]]]
     [sem ?sem]])
 (ADJ [[syn [[cat adj]
             [subj-info ?subj-info]
             [modi ?modi]
             [adj-num ?adj-num]
             [adj-posi ?adj-posi]
             [adj-arti ?adj-arti]
             [degree ?degree]
             [rel-cl ?rel-cl]]]
      [sem ?sem]]))

```

次に、終端語導入規則で reln と 必須格を被覆する。

ここで、V や N の場合と同じように、上位から伝達されてきた素性の値が個々の単語の特殊性に基づいて、ここで決定される（場合がある）。多くの場合、テンプレートが使用される。

```
(g_pd
:name ADJ-TEMPLATE
;形容詞
:type :template
:internal_structure (ADJ)
:annotation
((ADJ [[syn [[cat adj]
               [subj-info [[subj-sem ?obje] ;*<===== (a)
                           [subj-defsem [[label *it*]]]]]] ;*<===== (b)
               [sem [[obje ?obje]]]])) ;*<===== (a)
```

- (a) 大抵の場合、obje が主語になる。
 (b) 主語のデフォルトは "it".
 (a), (b) とは異なる性質を持つ A はテンプレートは使用できない。

```
(g_pd
:name ADJ-ENTIRE
;形容詞entire
:type :instance
:using ADJ-TEMPLATE
:internal_structure (ADJ)
:annotation
((ADJ [[syn [[lexid entire-adj-1][lex "entire"]
               [adj-posi pre-nn] ;*<===== (a)
               [adj-arti the]]] ;*<===== (b)
               [rel-cl -] ;*<===== (c)
               [sem [[reln entire-adj-1]]]]))
```

- (a) "entire" は限定用法のみ。
 (b) "entire" は必ず "the" を伴う。
 (c) 関係節を構成して名詞を修飾することはない。
 (Ex. the entire contents vs. *the contents that are entire)

補語必須格を持つ場合、AP レベルから「個々の終端語＋必須格」に展開される。

```
(g_pd
:name AP-SORRY-THAT
:sorry that ... を支配する形容詞句
:internal_structure (AP ADJ CONJ_S)
:annotation
((AP [[syn [[cat ap]
               [adj-posi pred] ;*<===== (a)
               [subj-info [[subj-sem ?agen] ;*<===== (b)
                           [subj-defsem [[label *def-speaker*]]]]] ;*<===== (c)
               [lex-aspe stative]]]
       [sem [[reln sorry-adj-2]
               [agen ?agen]
               [recp [[label *hearer*]]]
               [cont ?cont]]]]] ;*<===== (d)
       (ADJ [[syn [[cat adj]
                     [lexid sorry-adj-2][lex "sorry"]]]]]
       (CONJ_S [[syn [[cat conj_s] ;|
                     [conj-lex that]]] ;| ;*<===== (d)
                 [sem ?cont]]]] ;|
```

- (a) "sorry" は叙述用法のみ。
 (b) agen が主語になる。
 (c) デフォルトの主語の指定。
 (d) 必須格 cont は CONJ_S として実現する。

6 数詞 (NUMB) の構造

6.1 意味構造

(1) 基数

```
[[column-1 ?number-1]
 [column-10 ?number-10]
 [column-100 ?number-100]
 .
 .
 .
```

(2) 序数

```
[[parm ?x]
 [restr [[reln named]]]
 [iden ?iden] ; * <===== (a)
 [entity ?x]]
```

]])

(a) 1st-number, 2nd-number, 3rd-number, ...

6.2 基数の生成

まず、次のような桁を分ける規則を適用する。

```
(g_pd
:name NUMB-1-10
;数詞の桁を分ける
:internal_structure (NUMB NUMB1 NUMB2)
:annotation
((NUMB [[syn [[cat numb]
               [agr [[person 3][number pl]]]]]] ; * <===== (a)
       [sem [[column-1 ?number-1]
               [column-10 ?number-10]]]]))
(NUMB1 [[syn [[cat numb]]]
        [sem [[column-10 ?number-10]]]]))
(NUMB2 [[syn [[cat numb]]]
        [sem [[column-1 ?number-1]]]]))
```

```
(g_pd
:name NUMB-1-10-100
;数詞の桁を分ける
:internal_structure (NUMB NUMB1 NUMB2)
:annotation
((NUMB [[syn [[cat numb]
               [agr [[person 3][number pl]]]]]] ; * <===== (a)
       [sem [[column-1 ?number-1]
               [column-10 ?number-10]
               [column-100 ?number-100]]]]))
(NUMB1 [[syn [[cat numb]]]
        [sem [[column-100 ?number-100]]]]))
(NUMB2 [[syn [[cat numb]]]
        [sem [[column-1 ?number-1]
               [column-10 ?number-10]]]]))
```

(a) 桁が2つ以上あるときは必ず複数。

次に、それぞれの分節から具体的な数字を導く。

```
(g_pd
:name NUMB-1
;数詞1
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]
               [agr [[person 3][number sing]]]
               [lex "one"]]
       [sem [[column-1 1-number]]]]))
```

```
(g_pd
:name NUMB-11
;数詞11
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]
               [agr [[person 3][number pl]]]
               [lex "eleven"]]]
  [sem [[column-1 1-number]
        [column-10 1-number]]]]))
```

```
(g_pd
:name NUMB-30
;数詞30
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]
               [agr [[person 3][number pl]]]
               [lex "thirty"]]]
  [sem [[column-10 3-number]]]]))
```

「ゼロ」は次のような規則によって [lex ""] で表示する。

```
(g_pd
:name NUMB-0
;数詞0
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]
               [lex ""]]
  [sem [[column-1 0-number]]]]))
```

```
(g_pd
:name NUMB-00
;数詞00
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]
               [lex ""]]
  [sem [[column-1 0-number]
        [column-10 0-number]]]]))
```

N.B. lex ""については、no dict data として、現在は
WARNING が出力される。

"77530" = "seventy seven thousand five hundred thirty"

```
[[column-1 0-number]
 [column-10 3-number]
 [column-100 5-number]
 [column-1000 7-number]
 [column-10000 7-number]]
```

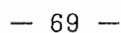
↓ ; * <===== NUMB-10000-1000

↓ ; * <===== NUMB-1-10

; * NUMB-500

; * NUMB-30

; * NUMB-0



6.3 序数詞

```
(g_pd
:name NUMB-1ST
;数詞1st
:internal_structure NUMB
:annotation
((NUMB [[syn [[cat numb]{lexid 1st-number}
               [lex "1st"]]]
      [sem [[parm ?x]
            [restr [[reln named]
                    [iden 1st-number]
                    [entity ?x]]]]]]))
```

6.4 NUMB が NP として使用される場合

統語構造

```
      NP
      |
      NN
      | ; * <===== NN-NUMB
      NUMB
```

使用規則

```
(g_pd
:name NN-NUMB
;数詞を支配する NN
:internal_structure (NN NUMB)
:annotation
((NN [[syn [[cat nn]
            [case ?case]{agr ?agr}
            [prep-info ?prep-info]
            [arti -]{insert-v ?insert-v}]]
      [sem [[parm ?x]
            [restr [[reln number]
                    [iden ?iden]
                    [entity ?x]]]]]]))
(NUMB [[syn [[cat numb]
            [case ?case]{agr ?agr}
            [prep-info ?prep-info]
            [insert-v ?insert-v]]
      [sem ?iden]]))
```

6.5 NUMB による NN の修飾 --> NP の項 参照

7 代名詞(PRON)の構造

7.1 意味構造

1. 通常の名詞型

(Ex.)
[[parm ?x]
 [restr [[reln this-pron-1]
 [entity ?x]]]]

2. label 素性型

(Ex.)
[[label *speaker*]]

7.2 統語構造

NP
| ; *<===== (a)
NN
| ; *<===== (b)
PRON

(a), (b) 単にカテゴリーを変換するだけの規則 NP-NN, NN-PRON による展開。

7.3 語彙項目の導入

```
(g_pd
:name PRON-THIS
;代名詞 this
:internal_structure PRON
:annotation
((PRON [[syn [[cat pron]
               [case ?case][agr [[person 3][number sing]]]
               [prep-info [[instprep "by"][modiprep "of"]]]
               [lexid this-pron-1][lex "this"]]]]
      [sem [[parm ?x]
             [restr [[reln this-pron-1]
                     [entity ?x]]]]]]]))

(g_pd
:name PRON-SPEAKER-I
;代名詞 I
:internal_structure PRON
:annotation
((PRON [[syn [[cat pron]
               [case ?case][agr [[person 1][number sing]]]
               [prep-info [[loctprep -][modiprep "of"]]]
               [lex "I"]]]]
      [sem [[label (:SET *speaker* *tag-np*)]]]]])) ;*<===== (a)
```

(a) *tag-np* は S-S-TAG で導入され、agr の値に応じて異なる代名詞に具体化される。
ここでは、[agr [[person 1][number sing]]] なので、“I” として具現化されている。

7.3.1 デフォルトの代名詞

defsem で "I", "you" を指定する場合、デフォルトであることを明らかにするため、"

N.B. defsem は、生成で空[]の格を埋める動作を行なう。

```
(g_pd
:name PRON-UNKNOWN-DEFAULT-SPEAKER
;代名詞 (デフォルト, unknown)
:internal_structure PRON
:annotation
((PRON [[syn [[cat pron]
               [case ?case][agr [[person 1][number sing][human +]]]
               [lex "<I>"]]]]
 [sem [[label *def-speaker*]]]]))
```

7.3.2 some と any

否定文、疑問文、時・条件の副詞節には [any-env +] が、それ以外の環境には [any-env -] が付与されている。PRON の lex 導入には、この情報を使用する。

```
(g_pd
:name PRON-ANYONE
;代名詞 anyone
:internal_structure PRON
:annotation
((PRON [[syn [[cat pron]
               [case ?case][agr [[person 3][number sing]]]
               [any-env +] ; *<===== (a)
               [lexid anyone-pron-1][lex "anyone"]]]]
 [sem [[parm ?x] ;|
       [restr [[reln someone-pron-1] ;| ; *<===== (b)
               [entity ?x]]]]]]]])) ;|
```

```
(g_pd
:name PRON-SOMEONE
;代名詞 someone
:internal_structure PRON
:annotation
((PRON [[syn [[cat pron][case ?case]
               [agr [[person 3][number sing]]]
               [any-env -] ; *<===== (a)
               [lexid someone-pron-1][lex "someone"]]]]
 [sem [[parm ?x] ;|
       [restr [[reln someone-pron-1] ;| ; *<===== (b)
               [entity ?x]]]]]]]])) ;|
```

(a) 環境が異なる。

(b) sem は同じ。

8 前置詞句 (PP) の構造

8.1 他の規則によって P の lex が決定される場合

```
(g_pd
:name PP
:前置詞句の展開
:internal_structure (PP P NP)
:annotation
((PP [[syn [[cat pp]
            [prep-info ?prep-info]
            [prep-lex ?prep-lex]]] ; *<===== (a)
      [sem ?sem :SPECIFIED]])
(P [[syn [[cat prep]
          [lex ?prep-lex]]]] ; *<===== (a)
(NP [[syn [[cat (:SET np advp)] ; *<===== (b)
         [case acc][agr ?agr]
         [prep-info ?prep-info]
         [gov-p ?prep-lex]]] ; *<===== (c)
     [sem ?sem]]]))
```

- (a) この PP を導入した規則によって与えられた P の lex。

(Ex. VP-ROLE: VP --> VP + PP[prep-lex "as"])

PP レベルでは prep-lex で現わす。

この素性の値を P の lex と同一とすることで、P を決定する。

PP が prep-info の中に指定してある意味役割を担っている場合、prep-lex と該当する素性は値を共有する。

(Ex. pp = locat

[prep-lex ?prep-lex]

[prep-info [[locatprep ?prep-lex (:SET "at" "in" "on"))]])

- (b) P の目的語は ADVP の場合もある。

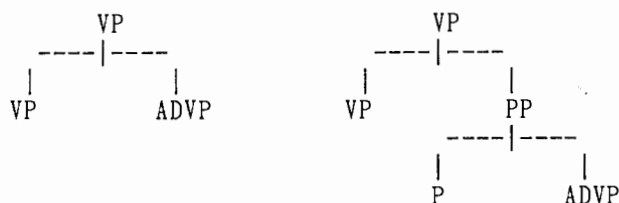
(Ex. MSET d9-8: You can take a bus that will go to the International Conference Center from there.)

- (c) ADVP を支配することができる P を ADVP ごとに指定しておく。

当然、PP の prep-lex、P の lex と値を共有する。

ADVP が VP を修飾する場合、(1) 直接、(2) PP を介しての2つの場合ができてしまう時がある。

(1) (2)



(Ex. VP-LOCT: VP --> VP + ADVP/PP[prep-lex "at" "in" "on"]

1. send the application form today
2. *send the application form at today
3. *send the application form in today
4. *send the application form on today)

この時、ADVP "today" に [gov-p -] (P の目的語にはなれない) と指定しておけば、(2) の構造 (例文 (2,3,4) に対応する) は失敗する。

"from there" 生成のためには、"there" に [gov-p "from"] と指定すれば良い。

8.2 意味構造に P (前置詞マーカー) が存在する場合

```
(g_pd
:name PP-AFTER
;afterを中心語とする前置詞句
:internal_structure (PP P NP)
:annotation
((PP [[syn [[cat pp]
             [modi-posi back]]]
      [sem [[reln after-prep-1] ; *<===== (a)
            [obje ?obje]]]])
 (P  [[syn [[cat prep]
             [lex "after"]]]])
 (NP [[syn [[cat np]
            [case acc][agr ?agr]]]
      [sem ?obje]]]))
```

(a) "after" の意味。

9 副詞句 (ADVP) の構造

★意味構造

```
[[parm ?x]
 [restr [[reln after_all]
         [entity ?x]]]]
```

★規則

```
(g_pd
:name ADVP-TEMPLATE
;副詞句
:type :template
:using nil
:internal_structure (ADVP ADV)
:annotation
((ADVP [[syn [[cat adv]]]
        [sem [[parm ?x]
              [restr [[entity ?x]]]]]])
 (ADV [[syn [[cat adv]]]]))
```

```
(g_pd
:name ADVP-AFTER_ALL
;副詞句after_all
:type :instance
:using ADVP-TEMPLATE
:internal_structure (ADVP ADV)
:annotation
((ADVP [[syn [[modi-posi back] ;*<===== (a)
              [gov-p -]]] ;*<===== (b)
        [sem [[restr [[reln after_all-adv-1]]]]]])
 (ADV [[syn [[lexid after_all-adv-1][lex "after all"]]]]))
```

- (a) 修飾位置を表わす。"after all" は VP の後方が修飾位置。
(b) いかなる P にも支配されない。

10 疑問詞 (WH) の構造

句レベルでは [cat np], [cat advp], [cat ap] と [wh +] の組み合わせで表現する。

[wh +] のカテゴリーは疑問詞の終端カテゴリー [cat ripron], [cat riadj], [cat riadv] を支配する。

```
(g_pd
:name WH-WHAT
;WH名詞句 (what)
:internal_structure (NP RIPRON)
:annotation
((NP      [[syn [[cat np][wh +]
                 {agr [[person 3][number sing]]}
                 {insert-v do}]]]
           [sem [[parm ?x]
                 {restr [[reln what-pron-1]
                         {entity ?x}}]]]])
(RIPRON [[syn [[cat ripron]
               {lexid what-ripron-1}{lex "what"}]]])
```

```
(g_pd
:name WH-WHAT-N
;WH名詞句 (what + N)
:internal_structure (NP ADJ NN)
:annotation
((NP      [[syn [[cat np][wh +]
                 {case ?case}{agr ?agr [[person 3][number (:SET sing pl)]]}]]]
           [sem [[parm ?parm]
                 {restr [[reln what-riadj-1]
                         {obje ?parm}}]]]])
(ADJ      [[syn [[cat riadj]
                 {lexid what-riadj-1}{lex "what"}]]])
(NN       [[syn [[cat nn]
                 {case ?case}{agr ?agr}]]]
           [sem ?parm]])
```

11 付記

本解説書で述べた英語生成規則はある程度の厳密さとロバスト性を備えている。しかし、開発途中でのインタフェースの変更や試行錯誤的な修正などのために、「まえがき」でも述べた通り、必ずしも簡潔で見通しの良い構成にはなっていない。

本章では前段とのインタフェースである意味素性構造および生成規則に関する補足と改良すべき点について述べる。

11.1 文の基本構造

11.1.1 IFT

現在のインタフェースは一文に一つの IFT が与えられているが、複文などを考えると一文に一つの IFT では適切に表現できない場合がある。

11.1.2 文に対する付加的修飾要素

1. 意味表現について

動詞句に対する修飾と文に対する修飾を区別するため、付加要素が IFT のレベルの素性になっている場合がある。このため、IFT のレベルで付加語を扱わなければならない。意味表現の改良が必要かも知れない(文(粗く言って“event”)のレベルと動詞句(同じく“action”)の部分とを階層的にするなど)。

2. 順序について

本来、「選好」によって制御すべき付加的修飾要素の順序を素性値による「制約」によって制御しているのは若干問題である(付加語の項を参照されたい)。

11.1.3 WH 疑問文

現在の意味構造において、WH のスコープをはっきりさせる関係上 WH 句を前に移動させている。しかし、実際は IFT などでもスコープは明らかなので WH を前置する必要はないように思われる。

11.1.4 名詞型命令文

“NP, please.” のような文を「名詞型命令文」とよぶ。この文に関しては意味表現があきらかに変なので改善する必要があると思われる。

この意味構造に対応して行なわれる「動詞の補完」の規則などは生成における「過激な処理」の例である。

11.1.5 外置構文

ここでは外置構文を単一化文法の枠組で扱ったが、このような構造は、外置抜き的基本的な構文生成が終った後、変形処理によって生成した方がよいと思われる¹。

11.2 述部 (VP) の構造

11.2.1 時制、相

時制や相に関しては、まず、意味表現の問題がある。現在の英語意味表現と日本語の表層表現との間にはギャップがあるため、適切な素性値が与えられていない場合がしばしばある。

日英翻訳における(もちろんその逆も)時制と相の決定の問題は本質的に難しいものの一つであろう。

¹当初の生成処理系の設計には、このための「構造調整処理」が含まれていた

11.2.2 否定

否定に関しても意味構造の問題がある。現在、否定を関係名とする構造に文や句の構造が埋め込まれている。通常、時制や相の情報は動詞(文)のレベルに記述されるが、否定の中に動詞(文)が埋め込まれている場合、時制や相は否定のレベルに記述される場合がある。これは、日本語の表層構造が簡抜けになって英語の意味構造になっているためと思われる(e.g. 「((彼はそこに行か)なかった)」)。

11.2.3 付加語の構造

一つの付加語を表出する規則自体は特に問題ないが、複数の付加語が存在した場合にこれらの間の順序の決定が問題である。現在は XXXXmodi 素性による「制約」によって規則の適用位置(順序)を制御しているが、これはコーディングが間接的であるし、何よりも、選好関係を適切に処理できないのが問題である。補語の順序は LP 原則などの何らかの外付けルーチンで処理するのが現実的であろう。

11.2.4 受動態

態は意味表現における関係名として指定されている。態を「情報の新旧」「情報の縄張り」などの素性に分解して表現することも考えられるが、これらの素性が日本語から決定できるかどうかは疑問である。トランスファー式の翻訳のインターフェースとしてはこのような意味表現で良いだろう。

標準的な句構造文法と異なり、この生成文法では受動態と能動態を同一の語彙エントリーに記述しているが、実効的には複数の語彙エントリーが存在するのと同じである。

11.2.5 VP の中核構造

動詞とこの動詞に対する補語の作る構造は動詞ごとに木を定義することによって記述している。この方法は実際に出現する文構造の記述に近いという利点はあるが、記述が冗長になるという問題がある。記述を簡潔にする

1. GPSG 風の記述にする

現在の「文型別テンプレート」をインスタンスレベルの規則とし、動詞語彙の部分に subcat 素性を設けその値を文型の名前(たとえば TN)とする。

```
-----
:name VP-TN                               ;;; 他動詞用 VP 規則
:internal_structure (VP V NP)
:annotation
((VP [[syn [[cat VP][agr ?agr][vform ?vform]
           [lex-aspe ?lex-aspe]]]
     [subj-info ?subj]
     [sem ?sem]])
 (V [[syn [[cat V][subcat TN]
          [lex-aspe ?lex-aspe]]]
    [subj-info ?subj]
    [obje-info ?obje]
    [sem ?sem]])
 (NP [[syn [[cat NP][case ACC]]]
      [sem ?obje]]))
-----
```

一方、動詞の語彙規則は次のようにする。

```
-----
:name V-HAVE
:internal_structure V
:annotation
((V [[syn [[cat V][subcat TN]
          [lex-aspe stative]]]
    [subj-info ?agen]
    [obje-info ?obje]
    [sem [[reln have-v-1]
          [agen ?agen]
          [obje ?obje]]]])
-----
```

以上によって動詞の語彙 (V-HAVE) が他動詞用の VP 規則 (VP-TN) の V の部分と単一化して適切な構造が生成される。

2. HPSG 風の記述にする

ドイツ語生成規則の記述形式である。

基本的には

VP → VP XP

という規則を再帰的に用いて (例えば目的格と場所格を取る動詞には2回適用して) 様々な格を表現する。XP の品詞や意味素性は動詞の語彙エントリに記述された subcat 素性によって与えられるする。

```
-----
:name VP-VP-XP
:internal_structure (VP1 VP2 XP)
:annotation
((VP1 [[syn [[cat VP][agr ?agr][vform ?vform]
              [lex-aspe ?lex-aspe]]]
      [subcat ?rest]
      [sem ?sem]])
 (VP2 [[syn [[cat VP][agr ?agr][vform ?vform]
              [lex-aspe ?lex-aspe]]]
      [subcat [[first [[syn ?comp-syn]
                       [sem ?comp-sem]]]
               [rest ?rest]]
      [sem ?sem]])
 (XP [[syn ?comp-syn]
      [sem ?comp-sem]]))
-----
```

VP と V(語彙) を結ぶために次のような規則を用意する。

```
-----
:name VP-V
:internal_structure (VP V)
:annotation
((VP1 [[syn [[cat VP][agr ?agr][vform ?vform]
              [lex-aspe ?lex-aspe]]]
      [subcat ?subcat]
      [sem ?sem]])
 (V [[syn [[cat VP][agr ?agr][vform ?vform]
           [lex-aspe ?lex-aspe]]]
    [subcat ?subcat]
    [sem ?sem]])
-----
```

一方、語彙規則は次のようになる。

```
-----
:name V-HAVE
:internal_structure V
:annotation
((V [[syn [[cat V][subcat TN]
           [lex-aspe ?lex-aspe]]]
    [subcat [[first [[syn [[cat NP][case ACC]]]
                     [sem ?obje]]]
             [rest [[first [[syn [[cat NP][case NOM]]]
                             [sem ?agen]]]
                     [rest :END]]]]]
    [sem [[reln have-v-1]
          [agen ?agen]
          [obje ?obje]]]])
-----
```

また、主語を生成するために次のような規則を用いる。


```

-----
:name VP-XP-VP
:internal_structure (VP1 XP VP2)
:annotation
((VP1 [[syn [[cat VP][agr ?agr][vform ?vform]
             [lex-aspe ?lex-aspe]]]
      [subcat :END]
      [sem ?sem]])
 (XP [[syn ?somp-syn]
      [sem ?comp-sem]])
 (VP2 [[syn [[cat VP][agr ?agr][vform ?vform]
             [lex-aspe ?lex-aspe]]]
      [subcat [[first [[syn ?comp-syn]
                       [sem ?comp-sem]]]
               [rest :END]]
      [sem ?sem]]))
-----

```

ここに示したような HPSG 風の記述を用いると、語彙規則以外の規則は全て他の様々な動詞と組み合わせ利用でき、記述量の減少に効果がある。但し、このような構造は階層が深くなりがちであるので、付加語の扱いが煩雑になる可能性がある。

11.3 名詞句の構造

現在の名詞句の規則で特に問題となっているのは、冠詞や数の決定についてである。ここで述べた規則では arti 素性によって冠詞 (定/不定) を制約している。arti 素性は名詞の語彙、連体修飾に関する幾つかの規則、名詞句を支配する構造によって制約されている。これは、定/不定、単数/複数を区別する情報が意味表現に欠如しているために行なっている暫定的な処置である。理屈の上で望ましい方法は、英語意味構造で

1. 指示物が単一物か複数の集合か
2. この指示物が聞き手 (話し手) の心の中でどのように同定されうるか

といったことを表現し、生成規則では、この意味情報と名詞の持つ性質 (固有名詞/普通名詞、加算/不加算) などから表層形を決定することであろう。

もちろん、このような意味表現を日本語入力から作成することが可能かどうかは別問題である²。

11.4 数詞の構造

ここでは、現在の意味構造に対して英単語による数表現を出力するところまで単一化による規則記述で記述したが、そこまでは不要かも知れない (アラビア数字で十分)。

²現在のレベルではコーパスから得られた統計的なデータで一応のデフォルトを決めるのがせいぜい可能な処理であろう

APPENDIX 1

統語素性一覧

★adj-arti (形容詞がそれが修飾する名詞に要求する冠詞)

(値)	
the	定冠詞
a	不定冠詞
-	無冠詞

★adj-num (形容詞がそれが修飾する名詞に要求する数)

(値)	
sing	単数
pl	複数

★adj-posi (AP の位置)

(値)	
pred	述部 (叙述用法)
pre-nn	NN の前 (限定用法)
pre-np	NP の前 (限定用法)
split	ADJ と その補部が被修飾 NN を挟んで分離している (限定用法)

★adj-pp-sem (NN 後方に現われる形容詞補語の PP の意味)

(値)	
	意味構造

★adj-prep-lex (NN 後方に現われる形容詞補語の PP の中心語 P の語彙項目)

(値)	
"as"	
:	

★adv-cl (未来時制を現在形で表わす副詞節)

(値)	
+	
-	

★agr (主語 - 動詞などの一致を司る)

★person (人称)

(値)	
1	
2	
3	

★number (数)

(値)	
sing	単数
pl	複数

★any-env (some --> any の変化を引き起こす環境)

(値)	
+	
-	

★ap-modi (既に上位で修飾されている AP)

(値)	
+	
-	

★arti (冠詞情報)

(値)	
a	不定冠詞
the	定冠詞
-	無冠詞

★aux (VP に助動詞が付加されている)

(値)

+

-

★aux-do (疑問文で "do" が現われている)

(値)

+

-

★be-inv ("be-v-copula" は既に上位で具現されており、単に被覆するだけ)

(値)

+

-

★case (名詞の格)

(値)

nom

主格

acc

目的格

gen

所有格

★cat (品詞)

(値)

advp

副詞句

ap

形容詞句

conj_s

接続文

idiom

慣用表現

nn

名詞複合

np

名詞句

pp

前置詞句

rel_s

関係節

s

文 (文全体から文末記号を除いたもの)

s-top

文全体

toinf

to 不定詞句

vp

動詞句

adj

形容詞

adv

副詞

advprt

副詞的小辞

aux

助動詞

be

be 動詞

conj

接続詞

det

決定辞

greet

挨拶

numb

数詞

not

否定辞

noun

名詞

prep

前置詞

pron

代名詞

riadj

疑問形容詞

riadv

疑問副詞

ripron

疑問代名詞

sign

記号

to

to

verb

動詞

★conj-lex (接続詞の語彙項目)

(値)

"after"

"that"

-

★degree (ADJ の語形変化)

(値)

comparative 比較級
superlative 最上級

★gov-p (ADVP を支配することができる P)

(値)

"from"

"in"

:

-

★have-inf (to 不定詞の完了形)

(値)

+

-

★insert-v (NP に対して補われる V)

(値)

give

have

make

reserve

★inv (倒置文)

(値)

+

-

★lex (語彙項目)

(値)

単語

"apply"

"conference"

:

★lex-aspe (動詞のアスペクト分類)

(値)

stative

状態動詞

act

動作動詞で下のいずれでもないもの

accomp

持続的達成動詞

achiev

瞬間的達成動詞

★lexid (語義番号)

(値)

語義番号

apply-v-1

conference-n-1

:

★modi (VP の任意格による修飾状況)

★degrmodi-front (VP は既に上位で DEGR に前方から修飾されている)

(値)

+

-

★mannmodi-front (VP は既に上位で MANN に前方から修飾されている)

(値)

+

-

★taspmodi-front (VP は既に上位で TASP に前方から修飾されている)

(値)

+

-

★accmodi (VP は既に上位で ACCM に後方から修飾されている)
 (値)
 +
 -
 ★causmodi (VP は既に上位で CAUS に後方から修飾されている)
 (値)
 +
 -
 ★condmodi (VP は既に上位で COND に後方から修飾されている)
 (値)
 +
 -
 ★conectmodi (VP は既に上位で CONECT に後方から修飾されている)
 (値)
 +
 -
 ★conjmodi (VP は既に上位で CONJ に後方から修飾されている)
 (値)
 +
 -
 ★degrmodi (VP は既に上位で DEGR に後方から修飾されている)
 (値)
 +
 -
 ★deptmodi (VP は既に上位で DEPT に後方から修飾されている)
 (値)
 +
 -
 ★destmodi (VP は既に上位で DEST に後方から修飾されている)
 (値)
 +
 -
 ★duramodi (VP は既に上位で DURATION に後方から修飾されている)
 (値)
 +
 -
 ★freqmodi (VP は既に上位で FREQ に後方から修飾されている)
 (値)
 +
 -
 ★goalmodi (VP は既に上位で GOAL に後方から修飾されている)
 (値)
 +
 -
 ★infmannmodi (VP は既に上位で INFMANN に後方から修飾されている)
 (値)
 +
 -
 ★instmodi (VP は既に上位で INST に後方から修飾されている)
 (値)
 +
 -
 ★loctmodi (VP は既に上位で LOCT に後方から修飾されている)
 (値)
 +
 -
 ★mannmodi (VP は既に上位で MANN に後方から修飾されている)
 (値)
 +
 -
 ★methmodi (VP は既に上位で METH に後方から修飾されている)
 (値)
 +
 -

- ★purpmodi (VP は既に上位で PURP に後方から修飾されている)
(値)
+
-
- ★purposemodi (VP は既に上位で PURPOSE に後方から修飾されている)
(値)
+
-
- ★quantmodi (VP は既に上位で QUANT に後方から修飾されている)
(値)
+
-
- ★rangmodi (VP は既に上位で RANG に後方から修飾されている)
(値)
+
-
- ★recpmodi (VP は既に上位で RECP に後方から修飾されている)
(値)
+
-
- ★rolemodi (VP は既に上位で ROLE に後方から修飾されている)
(値)
+
-
- ★taspmodi (VP は既に上位で TASP に後方から修飾されている)
(値)
+
-
- ★tdepmodi (VP は既に上位で TDEP に後方から修飾されている)
(値)
+
-
- ★tdes1modi (VP は既に上位で TDES-1 に後方から修飾されている)
(値)
+
-
- ★tdes2modi (VP は既に上位で TDES-2 に後方から修飾されている)
(値)
+
-
- ★tlocmodi (VP は既に上位で TLOC に後方から修飾されている)
(値)
+
-
- ★tloc-ordermodi (VP は既に上位で TLOC-ORDER に後方から修飾されている)
(値)
+
-
- ★modi-posi (ADVP の現われる位置)
(値)
front VP の前
back VP の後ろ
adj-front 形容詞の前
adj-back 形容詞の後ろ
- ★negate (否定の動詞句)
(値)
+
-

★pass-pp (受動文で対応する能動文の主語が P と共に現われる場合の P)
(値)
"by"
"in"
:

★pass-subj-info (受動文の主語に関する情報)
★pass-subj-sem (受動文の主語の意味情報)
(値)
意味構造
★pass-subj-syn (受動文の主語の統語情報)
★agr (一致)
★arti (冠詞)
★pass-subj-defsem (受動文の主語のデフォルト意味情報)
(値)
意味構造

★prsp-mod (PRAG の prsp-mod の処理が終了した)
(値)
+
-

★prep-info (P に関する情報)
★instprep (INST を具現化する P の語彙項目)
(値)
"by"
"in"
"on"
"using"
★loctprep (LOCT を具現化する P の語彙項目)
(値)
"at"
"in"
★methprep (METH を具現化する P の語彙項目)
(値)
"in"
"through"
★modiprep (NP を修飾する PP の中心語の P の語彙項目)
(値)
"by"
"for"
"of"
"on"
★tloctprep (TL0C を具現化する P の語彙項目)
(値)
"at"
"in"
"on"

★prep-lex (PP の中心語 P の語彙項目)
(値)
"at"
"by"
"in"
:

★pre-poss-cover ("possibility" は既に上位で具現されており、単に被覆するだけ)
(値)
+
-

★rel-cl (関係節)

(値)

+

-

★s-depth (文の深さによる分類)

(値)

matrix

主文

subord

従属文

★signlex (文末記号の種類)

(値)

"."

"?"

★s-modi (S の修飾状況)

★s-caus (S は既に上位で CAUS に修飾されている)

(値)

+

-

★s-cond (S は既に上位で COND に修飾されている)

(値)

+

-

★s-connect (S は既に上位で CONECT に修飾されている)

(値)

+

-

★s-conj (S は既に上位で CONJ に修飾されている)

(値)

+

-

★s-infattd (S は既に上位で INFATTD に修飾されている)

(値)

+

-

★s-opps (S は既に上位で OPPS に修飾されている)

(値)

+

-

★too (S は既に上位で "too" に修飾されている)

(値)

+

-

★subj-info (主語に関する情報)

★subj-sem (主語の意味情報)

(値)

意味構造

★subj-syn (主語の統語情報)

★agr (一致)

★arti (冠詞)

★subj-defsem (主語のデフォルト意味情報)

(値)

意味構造

★synmanner (S を修飾する MANNER)

(値)

direct

indirect

★tag-aux (付加疑問文中に現われる助動詞に関する情報)

(値)

助動詞の syn 素性の集合

★vform (動詞の語形)

(値)

present	現在形
past	過去形
bse	原形
prp	現在分詞
psp	過去分詞

★wh (句レベルでの疑問詞と通常カテゴリーの識別)

(値)

+
-

★wh-ante (関係節の先行詞が "what")

(値)

+
-

APPENDIX 2

「英語意味素性構造」一覧 日英変換と英語生成との間のインタフェースについて

1 まえがき

変換方式の考え方から行けば、英語意味素性構造は生成すべき英語文の「意味」を表現するのに用いられるものであり、理想的には英語側の論理で作られるべきものである。ところが、実際に日本語音声入力から結合したシステムを考えると、生成すべき英語の意味を表すのに必要な情報が全て自動的に得られることは期待できない。従って、ここで説明する英語意味素性構造もある程度前段の処理能力に依存したものになっていることをあらかじめ断っておく。

2 トップレベル

関係名の値として、IFT（発話タイプ）が示される。引数として、agen, recp, objeをとる。
agenは発話の行為者（値は必ず[[label *speaker*]]）、recpは発話の受け手（値は必ず[[label *hearer*]]）を表し、objeの下には原則として文の情報（ごくまれに名詞句の情報）が記述される。

2.1 発話タイプ

a. 情報伝達文

```
[[reln inform]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

b. 要求文

```
[[reln request]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

c. WH疑問文

```
[[reln questionref]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

疑問詞の意味表現は下のように命題の外に置かれる。

```
[[reln questionref]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje [[parm ?parm]
        [restr ?restr]]]]
```

d. yes/no疑問文

```
[[reln questionif]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

e. 確認疑問文

```
[[reln questionconf]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

f. 約束文

```
[[reln promise]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

g. 応答文

```
[[reln response]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

h. 感情表現文

```
[[reln expressive]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

i. 挨拶文

```
[[reln phatic]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]]
```

2.2 トップレベルを修飾する自由格（任意格）

a. 種類

attd
caus
cond
conect
conj
infattd
inmann
mann
manner
opps

b. 関係名と同レベルに記述される

例：

```
[[reln inform]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje]
 [conect ?conect]
 [caus ?caus]]
```

3 重文、複文

3.1 重文

a. sent-1が第一文、sent-2が第二文の意味表現。

```
[[reln inform]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje [[reln compound-sent]
        [sent1 ?sent1]
        [sent2 ?sent2]]]]]
```

b. obje1が第一文、obje2が第二文の意味表現。

```
[[reln ?reln]
 [agen [[label *speaker*]]]
 [recp [[label *hearer*]]]
 [obje ?obje1]
 [conj [[reln first-sent]
        [obje ?obje2]]]]]
```

3.2 複文

"think" など補文をとる動詞の必須格、COND, PURPOSE などの任意格として現れる文型意味表現が複文になる。

例：

```
[cont [[reln subject-volition]
       [aspt unr1]
       [tense future]
       [expr ?x1]
       [obje [[reln attend-v-1]
               [agen ?x1]
               [loct []]]]
       [sem-aspe unreal]]]]]

[cond [[reln if-conj-1]
       [obje [[reln have-v-1]
               [aspt unr1]
               [tense future]
               [agen ?x2]
               [obje [[parm ?x1[]]
                       [restr [[reln question-n-1]
                               [entity ?x1]]]
                       [index [[defint infef]
                               [gender none]
                               [number sing-pl]
                               [person third]]]]]
               [sem-aspe unreal]]]]]]]
```

任意格は殆どの場合、関係名として接続詞が指定されている。引数objeの値が複文の内容。

例：

```
[[reln that-conj-1]
 [obje ?obje]]

[[reln if-conj-1]
 [obje ?obje]]
```

```
[[reln since-conj-3]
 [obje ?obje]]
```

```
[[reln although-conj-1]
 [obje ?obje]]
```

```
[[reln while-conj-1]
 [obje ?obje]]
```

4 文レベル

4.1 基本命題

関係名とその引数を並べて記述する。

a. 動作

例：

```
[[reln apply-v-1]
 [agen ?agen]
 [obje ?obje]]])
```

b. 存在 (there構文)

i. 場所格がある場合

```
[[reln be-v-exist]
 [obje ?obje]
 [loct ?tloc]]
```

ii. 場所格がない場合

```
[[reln be-v-exist]
 [obje ?obje]]
```

c. 状態

例：

```
[[reln want-desire]
 [expr ?expr]
 [obje ?obje]]
```

```
[[reln guess-seem]
 [obje ?obje]]
```

d. コピュラ

```
[[reln be-v-copula]
 [obje ?obje]
 [iden ?iden]]
```

"be" + ADJ 構文

```
[[reln fine-adj-1]
 [obje ?obje]]
```

埋まっていない要素は空集合[]で表されている。

例：

```
[[reln apply-v-1]
 [agen ?agen]
 [obje []]]])
```

4.2 任意格

a. 任意格は関係名と同一のレベルに記述する。

例：

```
[[reln send-v-1]
 [agen ?agen]
 [recp ?recp]
 [tloc ?tloc]] 任意格
```

b. 任意格の種類

格名	意味	対応する英語
accm	随伴	with
caus	原因	for, because...
comp	比較	than, as
connect	結合	in short, but, then, in that case, ...
conj	接続	well, yes, no, all right, ...
degr	程度	more, most, ...
dept	起点	from
dest	終点	to
duration	継続	during
freq	頻度	again, often, ...
goal	目標	to
inmann	陳述副詞	maybe, probably, please
inst	手段	by, on, in, using
loct	場所	at, in
mann	様態	in advance, that way, properly
meth	方法	in, through
purp	目的	for
purpose	目的 (命題)	to..., in order to...
quant	量	once, twice, many times, ...
rang	～について	about
role	役割	as
tasp	様相副詞	already, still, right away, ...
tdep	時間的起点	from
tdep-1	時間的終点 (まで)	untill
tdep-2	時間的終点 (までに)	by
tloc	時刻	at, in, on
tloc-order	順番	first of all, ...

c. 意味表現に前置詞・接続詞が現れる場合もある。

例：

```
[loct [[reln on-prep-1]
      [obje [[parm ?x2[]]
             [restr [[reln desk-n-1]
                     [entity ?x2[]]]]
            [index [[defint indef]
                    [gender none]
                    [number sing-pl]
                    [person third]]]]]]])
```

```

[tloc [(reln after-prep-1)
      [obje [[parm ?x4[]]
            [restr [(reln conference-n-1)
                    [entity ?x4]]]
            [index [(defint indef)
                    [gender none]
                    [number sing-pl]
                    [person third]]]]]]]]

```

4.3 様相、法

様相、法の情報は命題を引数とする関係名として記述する。
命題の他に、agenまたはexprを引数とする場合が多い。

例

```

[(reln obligation)
 [obje [(reln send-1)
        [agen ?agen]
        [recp ?recp]
        [tloc ?tloc]]]

```

a. 義務

```

[(reln obligation)
 [agen ?agen]
 [obje ?obje]]

[(reln duty-should)
 [agen ?agen]
 [obje ?obje]]

[(reln obligation-should)
 [agen ?agen]
 [obje ?obje]]

[(reln obligation-must)
 [agen ?agen]
 [obje ?obje]]

[(reln had_better-obligation)
 [agen ?agen]
 [obje ?obje]]

```

b. 可能性

```

[(reln possibility)
 [agen ?agen]
 [obje ?obje]]

```

c. 必要性

```

[(reln necessity)
 [expr ?expr]
 [obje ?obje]]

```

d. 許可

```

[(reln permission)
 [agen ?agen]
 [obje ?obje]]

```

e. その他

```
[[reln suggestion-may]
 [obje ?obje]]

[[reln presumption-could]
 [obje ?obje]]
```

4.4 否定・肯定

否定は命題を引数とする関係名として記述する。

```
[[reln negate]
 [obje ?obje]]
```

4.5 時制・相

基本命題の関係名、様相・法、または否定と同レベルに記述される。

```
[[reln ?reln]
 [...]
 [...]
 [aspt ?aspt]
 [sem-aspe ?sem-aspe]
 [tense ?tense]]
```

例：

```
[[reln possibility]
 [aspt stat]
 [tense present]
 [expr ?x3()]
 [obje [[reln go-v-2]
        [agen ?x3]
        [tloc [[parm ?x2()]
                [restr [[reln tomorrow-adv-1]
                        [entity ?x2]]]]]]
        [dest [[parm ?x1()]
                [restr [[reln there-adv-1]
                        [entity ?x1]]]]]]]]
 [sem-aspe stative]]
```

5 名詞句レベル

5.1 普通名

```
[[parm ?x]
 [restr [[reln ?reln]
        [entity ?x]]]
 [index [defint ?defint]
        [gender ?gender]
        [number ?number]
        [person ?person]]]
```

INDEXには定・不定、性、数、人称の情報が記述される。

例：

```
[[parm ?x]
 [restr [[reln application-n-1]
        [entity ?x]]]
 [index [defint indef]
        [gender none]
        [number sing-pl]
        [person third]]]
```


5.2 組織名、地名、建造物名などの固有名

```
[[parm ?x]
 [restr [{reln named}
         {iden ?iden}
         {entity ?x}]]
 [index ?index]]
```

5.3 日付

a.

```
[[parm ?x]
 [restr [{reln date-month-day}
         {day ?day}
         {month ?month}
         {entity ?x}]]]]
```

b.

```
[[parm ?x]
 [restr [{reln date-month-day}
         {month ?month}
         {entity ?x}]]]]
```

5.4 住所

a.

```
[[parm ?x]
 [restr [{reln named}
         {iden {reln address}
               {city ?city}
               {town ?town}
               {ward ?ward}
               {chome ?chome}
               {else ?else}}
         {entity ?x}]]]]
```

b.

```
[[parm ?x]
 [restr [{reln named}
         {iden {reln address}
               {city ?city}
               {town ?town}
               {ward ?ward}
               {else ?else}}
         {entity ?x}]]]]
```

c.

```
[[parm ?x]
 [restr [{reln named}
         {iden {reln address}
               {pref ?pref}
               {town ?town}
               {ward ?ward}
               {chome ?chome}
               {banchi ?banchi}
               {gou ?gou}}
         {entity ?x}]]]]
```

d.

```
[[parm ?x]
 [restr [[reln named]
          [iden [reln address]
                 [pref ?pref]
                 [town ?town]
                 [azal ?azal]
                 [else ?else]]
          [entity ?x]]]]]
```

e.

```
[[parm ?x]
 [restr [[reln named]
          [iden [reln address]
                 [pref ?pref]
                 [entity ?x]]]]]
```

5.5 時間

```
[[parm ?x]
 [restr [[reln modify-time]
          [arg1 ?arg1]
          [arg2 ?arg2]]]]]
```

5.6 人名

a.

```
[[parm ?x]
 [restr [[reln name]
          [family-name ?family-name]
          [entity ?x]]]]]
```

b.

```
[[parm ?x]
 [restr [[reln name]
          [first-name ?first-name]
          [entity ?x]]]]]
```

c.

```
[[parm ?x]
 [restr [[reln name]
          [family-name ?family-name]
          [first-name ?first-name]
          [entity ?x]]]]]
```

5.7 名詞修飾構造

a. 限定詞

```
[[parm ?x]
 [restr [[reln (:SET that-det-1 this-det-1 those-det-1
                    each-det-1 every-det-1 such-det-1
                    some-det-1 some-det-1 any-det-1
                    another-det-1 a_few-det-1))
          [obje ?x]]]]]
```

b. 所有格

```
[[parm ?x]
 [restr [[reln possessive]
          [agen ?agen]
          [obje ?x]]]]]
```

c. PP や ADVP などによる修飾

```
[[parm ?x]
 [restr [[reln modify]
         [arg1 ?y]
         [arg2 ?x]]]]]
```

d. PP による修飾

```
[[parm ?x]
 [restr [[reln (:SET modify-unit modify-loct modify-loct-near
                    modify-goal-for modify-pod modify-tloc
                    modify-tloc-after modify-dest modify-dest-for
                    modify-dept modify-rang modify-before
                    modify-agen-by modify-purp-for modify-in_charge_of
                    modify-prior-to modify-conj-and))
         [arg1 ?y]
         [arg2 ?x]]]]]
```

e. REL_S, AP による修飾

```
[[parm ?x]
 [restr ?restr]]  restrは命題型意味表現
```

5.8 限量

a. NUMB による限量

```
[[parm ?x]
 [restr [[reln modify]
         [arg1 [[parm ?y]
                [restr [[reln number]
                        [iden ?iden]
                        [entity ?y]]]]]]
         [arg2 ?x]]]]]
```

b. 限定詞による限量

例：

```
[[parm ?x]
 [restr [[reln all-det-1]
         [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln all_of-det-1]
         [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln all_over-det-1]
         [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln most_of-det-1]
         [obje ?x]]]]]
```

6 副詞

6.1 動詞修飾の副詞

例：

```
[sem [[parm ?x]
      [restr [[reln again-adv-1]
               [entity ?x]]]]]
```

6.2 名詞修飾の副詞

例：

```
[sem [[parm ?x]
      [restr [[reln about-adv-1]
               [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln just-adv-1]
         [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln more_than-adv-1]
         [obje ?x]]]]]
```

```
[[parm ?x]
 [restr [[reln only-adv-1]
         [obje ?x]]]]]
```

7 形容詞

関係名と引数を並べて記述する。
形式上は全く動詞と区別がない。

例：

```
[[reln unexpected-adj-1]
 [obje ?obje]]
```

```
[[reln full-adj-1]
 [obje ?obje]
 [inst ?inst]]
```

```
[[reln full-adj-1]
 [obje ?obje]
 [inst ?inst]]
```

動詞と同じように、時制や任意格素性が関係名レベルに存在してもよい。

例：

```
[[reln unexpected-adj-1]
 [aspt stat]
 [tense present]
 [obje ?x4]
 [degr [[parm ?x5()]
        [restr [[reln so-adv-1]
                 [entity ?x5]]]]]]
 [sem-aspe stative]]
```

8 語用論的情報

tooの扱い

例：

```
{sem ?sem}
{prag [{benefit none}
      {hearer ?x1}
      {intention inform}
      {politeness [{degree 2}]}
      {prsp-terms [{prsp-mod mo}]}
      {speaker ?x3}
      {topic [{topic-mod null}]}}}]]
```

9 特殊構文

a. 比較級

形容詞の関係名と同レベルにあるDEGR "more"によって表現される。

例：

```
{[reln expensive-adj-1]
 {obje ?obje}
 {degr [{parm ?x}
       {restr [{reln more-adv-1}
                {entity ?x}]}}}]]
```

a. 最上級

形容詞の関係名と同レベルにあるDEGR "most"によって表現される。

例：

```
{[reln fast-adj-1]
 {obje ?obje}
 {degr [{parm ?x}
       {restr [{reln most-adv-1}
                {entity ?x}]}}}]]
```

10 非分析的表現

挨拶、応答など (IFTはexpressive, phatic, response)

例：

```
{[reln expressive]
 {agen ?x1[{label *speaker*}]}
 {recp ?x2[{label *hearer*}]}
 {obje [{reln thank_you-idiom}
       {aspt -}
       {tense present}
       {sem-aspe null}
       {agen ?x1}
       {recp ?x2}]]]}

{[reln phatic]
 {agen ?x1[{label *speaker*}]}
 {recp ?x2[{label *hearer*}]}
 {obje [{reln good_bye-interj-1}
       {agen ?x1}
       {recp ?x2}]]]}
```

```
[[reln response]
 [agen ?x1[[label *speaker*]]]
 [recp ?x2[[label *hearer*]]]
 [obje [[reln certainly-idiom-1]
        [agen ?x1]
        [recp ?x2]]]]]
```



```

    (SUBJ-DEFSEM []))
  (ANY-ENV -)
  (DECLARATIVE [])
  (NEGATE -)
  (ADV-CL -)
  (BE-INV -)
  (SYNMANNER INDIRECT)
  (TAG-AUX []))
(SEM [(RELN HAVE-V-1)
  (AGEN !X4[(LABEL *SPEAKER*)])
  (OBJE [(PARM !X3[(PARM !X2())
    (RESTR [(RELN SOMETHING-PRON-1)
      (ENTITY !X2)])
    (INDEX [(DEFINT INDEF)
      (GENDER NONE)
      (NUMBER SING)
      (PERSON THIRD)))]))
  (RESTR [(RELN WANT-DESIRE)
    (EXPR !X4)
    (OBJE [(RELN ASK-V-1)
      (AGEN !X4)
      (RECP [])
      (OBJE !X3)])
    (ASPT STAT)
    (TENSE PRESENT)
    (DEGR [(PARM !X1())
      (RESTR [(RELN JUST-ADV-1)
        (ENTITY !X1)])])
    (SEM-ASPE STATIVE)))]))
  (ASPT STAT)
  (TENSE PRESENT)
  (SEM-ASPE STATIVE))]
(SLASH []))
**** re-activated (INV_S-BE-AP%VP1 PP VP-ASPT-COVER <VP-HAVE!VP-TN> AP-TENSE-COVER NP-TOINF NP/PP-TENSE-ASPECT-COVER INV_S-DO
VP-TENSE-ASPECT-COVER-BSE-ING VP-SIMPLE-PRESENT-PAST INV_S-BE-AP S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN) ****

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
|--S [#2544 EXPANDED NIL]
|  |--(ADVP CONJ_S) [#2545 ACTIVE NIL]
|  |--SIGN [#2546 LEXIFIED "," NIL]
|  |--S [#2547 EXPANDED NIL]
|    |--NP [#2548 ACTIVE NIL]
|    |--VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|      |--VERB [#2548 LEXIFIED "have" NIL]
|      |--NP [#2547 ACTIVE NIL]
|--SIGN [#2550 LEXIFIED "." NIL]
**** expanding NP [#2687 ACTIVE NIL] ****
[[SYN [(ANY-ENV -)
  (SYNMANNER INDIRECT)
  (CAT NP)
  (CASE ACC)]]]
(SEM [(PARM !X4[(PARM !X3())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X3)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD)))]))
  (RESTR [(RELN WANT-DESIRE)
    (EXPR !X2[(LABEL *SPEAKER*)])
    (OBJE [(RELN ASK-V-1)
      (AGEN !X2)
      (RECP [])
      (OBJE !X4)])
    (ASPT STAT)
    (TENSE PRESENT)
    (DEGR [(PARM !X1())
      (RESTR [(RELN JUST-ADV-1)
        (ENTITY !X1)])])
    (SEM-ASPE STATIVE)))]))
  (DEFSEM [(LABEL *IT*)])
  (SLASH []))
**** re-activated (PP <NP-SOMETHING-AP> <CONJ_S-WH-1> <NP-AP-NP> <NP-NP-REL> <NN-AP-NN-1> <NN-AP-NN-PP> S-MODI-TOO
S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV
TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (NP-NP-REL) ****

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
|--S [#2544 EXPANDED NIL]
|  |--(ADVP CONJ_S) [#2545 ACTIVE NIL]
|  |--SIGN [#2546 LEXIFIED "," NIL]
|  |--S [#2547 EXPANDED NIL]
|    |--NP [#2548 ACTIVE NIL]
|    |--VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|      |--VERB [#2548 LEXIFIED "have" NIL]
|      |--NP [#2547 EXPANDED (NP-NP-REL)]
|        |--NP [#2548 ACTIVE NIL]
|        |--REL_S [#2549 ACTIVE NIL]
|--SIGN [#2550 LEXIFIED "." NIL]
**** expanding REL_S [#2549 ACTIVE NIL] ****
[[SYN [(CAT REL_S)
  (VFORM )]]]

```



```

(SYNMANNER INDIRECT)
(WH-ANTE ())
(SEM [(RELN WANT-DESIRE)
  (EXPR !X4[(LABEL *SPEAKER*)])
  (OBJE [(RELN ASK-V-1)
    (AGEN !X4)
    (RECP ())
    (OBJE !X3[(PARM !X2())
      (RESTR [(RELN SOMETHING-PRON-1)
        (ENTITY !X2)])
      (INDEX [(DEFINT INDEF)
        (GENDER NONE)
        (NUMBER SING)
        (PERSON THIRD)])))]))])

(ASPT STAT)
(TENSE PRESENT)
(DEGR [(PARM !X1())
  (RESTR [(RELN JUST-ADV-1)
    (ENTITY !X1)])])
(SEM-ASPE STATIVE)])
(SLASH [(SYN [(CAT NP)
  (AGR ())
  (PREP-INFO ())])
  (SEM !X3)])
**** re-activated (INV_S-BE-AP%VP1 PP VP-ASPT-COVER <VP-WOULD-LIKE-TOINF> <VP-WANT-TOINF!VP-TNT> VP-DEGR-FRONT VP-DEGR!VP-MODI-TEMPLATE
AP-TENSE-COVER NP-TOINF NP/PP-TENSE-ASPECT-COVER INV_S-DO VP-TENSE-ASPECT-COVER-BSE-ING VP-SIMPLE-PRESENT-PAST INV_S-BE-AP
S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 REL_S REL_S-2 VP-MAKE-NP VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV
TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF) ****
**** Applying (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF) ****

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
--S (#2544 EXPANDED NIL)
|--(ADVP CONJ_S) (#2545 ACTIVE NIL)
|--SIGN (#2546 LEXIFIED ", " NIL)
|--S (#2547 EXPANDED NIL)
|  |--NP (#2548 ACTIVE NIL)
|  |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|    |--VERB (#2686 LEXIFIED "have" NIL)
|    |--NP (#2687 EXPANDED (NP-NP-REL))
|      |--NP (#2688 ACTIVE NIL)
|      |--REL_S (#2689 ACTIVE NIL)
|        +--REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|          + |--CONJ (#2691 LEXIFIED "that" NIL)
|          + |--S (#2692 EXPANDED NIL)
|            + |--NP (#2693 ACTIVE NIL)
|            + |--VP (#2694 EXPANDED NIL)
|              + |--ADVP (#2695 ACTIVE NIL)
|              + |--VP (#2696 EXPANDED NIL)
|                + |--AUX (#2697 LEXIFIED "would" NIL)
|                + |--VERB (#2698 LEXIFIED "like" NIL)
|                + |--NP (#2699 ACTIVE NIL)
|                + |--TOINF (#2700 ACTIVE NIL)
|          +--REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|            |--S (#2702 EXPANDED NIL)
|              |--NP (#2703 ACTIVE NIL)
|              |--VP (#2704 EXPANDED NIL)
|                |--ADVP (#2705 ACTIVE NIL)
|                |--VP (#2706 EXPANDED NIL)
|                  + |--AUX (#2707 LEXIFIED "would" NIL)
|                  + |--VERB (#2708 LEXIFIED "like" NIL)
|                  + |--NP (#2709 ACTIVE NIL)
|                  + |--TOINF (#2710 ACTIVE NIL)
|            |--SIGN (#2550 LEXIFIED ", " NIL)
|            **** expanding TOINF (#2710 ACTIVE NIL) ****
|            [(SYN [(CAT TOINF)
|              (VFORM BSE)
|              (AGR ())
|              (SUBJ-INFO [(SUBJ-SEM ())
|                (SUBJ-DEFSEM ()))]))])
|            (SEM [(RELN ASK-V-1)
|              (AGEN [(LABEL *SPEAKER*)])
|              (RECP ())
|              (OBJE !X2[(PARM !X1())
|                (RESTR [(RELN SOMETHING-PRON-1)
|                  (ENTITY !X1)])
|                (INDEX [(DEFINT INDEF)
|                  (GENDER NONE)
|                  (NUMBER SING)
|                  (PERSON THIRD)])))]))])
|            (SLASH [(SYN [(CAT NP)
|              (AGR ())
|              (PREP-INFO ())])
|              (SEM !X2)])
|            **** re-activated (PP VP-RECP!VP-MODI-TEMPLATE <VP-ASK-1-NP-ABOUT!VP-TN.PR> <VP-ASK-1-NP-NP!VP-DN.N> S-MODI-TOO S-NP-VP-SUBORD
CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN
NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR) ****
**** condition (Slash No.1) is satisfied on [#2717] ****
**** fs before application ****
|            [(SEM !X2[(PARM !X1())
|              (RESTR [(RELN SOMETHING-PRON-1)
|                (ENTITY !X1)])
|              (INDEX [(DEFINT INDEF)
|                (GENDER NONE)
|                (NUMBER SING)
|                (PERSON THIRD)])))]))

```

```

                (GENDER NONE)
                (NUMBER SING)
                (PERSON THIRD))))))
(DEFSEM ((LABEL *IT*)))
(SYN [(CAT NP)
      (CASE ACC)])
(SLASH [(SYN [(CAT NP)
              (AGR [])
              (PREP-INFO ())])
        (SEM !X2))])
**** Slash No.1 is applied ****
**** fs on the slash node after application
[(SEM !X3[(PARM !X2())
          (RESTR [(RELN SOMETHING-PRON-1)
                  (ENTITY !X2)])
          (INDEX [(DEFINT INDEF)
                  (GENDER NONE)
                  (NUMBER SING)
                  (PERSON THIRD))])])
        (DEFSEM ((LABEL *IT*)))
        (SYN !X1[(CAT NP)
                  (AGR [])
                  (PREP-INFO [])
                  (CASE ACC)
                  (LEX ""')])
        (SLASH [(SYN !X1)
                  (SEM !X3)])])
**** Applying (TOINF VP-ASK-1-NP-NP!VP-DN.N) ****
**** condition (Slash No.1) is satisfied on [#2723] ****
**** fs before application ****
[(SEM !X2[(PARM !X1())
          (RESTR [(RELN SOMETHING-PRON-1)
                  (ENTITY !X1)])
          (INDEX [(DEFINT INDEF)
                  (GENDER NONE)
                  (NUMBER SING)
                  (PERSON THIRD))])])
        (SYN [(CAT NP)
              (CASE ACC)])
        (SLASH [(SYN [(CAT NP)
                      (AGR [])
                      (PREP-INFO ())])
                  (SEM !X2)])])
**** Slash No.1 is applied ****
**** fs on the slash node after application
[(SEM !X3[(PARM !X2())
          (RESTR [(RELN SOMETHING-PRON-1)
                  (ENTITY !X2)])
          (INDEX [(DEFINT INDEF)
                  (GENDER NONE)
                  (NUMBER SING)
                  (PERSON THIRD))])])
        (SYN !X1[(CAT NP)
                  (AGR [])
                  (PREP-INFO [])
                  (CASE ACC)
                  (LEX ""')])
        (SLASH [(SYN !X1)
                  (SEM !X3)])])

```

```

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
--S [#2544 EXPANDED NIL]
|  |--(ADVP CONJ.S) [#2545 ACTIVE NIL]
|  |--SIGN [#2546 LEXIFIED "," NIL]
|  |--S [#2547 EXPANDED NIL]
|  |  |--NP [#2548 ACTIVE NIL]
|  |  |--VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|  |  |  |--VERB [#2586 LEXIFIED "have" NIL]
|  |  |  |--NP [#2687 EXPANDED (NP-NP-REL)]
|  |  |  |  |--NP [#2688 ACTIVE NIL]
|  |  |  |  |--REL_S [#2689 ACTIVE NIL]
|  |  |  |  |  +--REL_S [#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|  |  |  |  |  +  |--CONJ [#2691 LEXIFIED "that" NIL]
|  |  |  |  |  +  |--S [#2692 EXPANDED NIL]
|  |  |  |  |  +  |  |--NP [#2693 ACTIVE NIL]
|  |  |  |  |  +  |  |--VP [#2694 EXPANDED NIL]
|  |  |  |  |  +  |  |  |--ADVP [#2695 ACTIVE NIL]
|  |  |  |  |  +  |  |  |--VP [#2696 EXPANDED NIL]
|  |  |  |  |  +  |  |  |  |--AUX [#2697 LEXIFIED "would" NIL]
|  |  |  |  |  +  |  |  |  |--VERB [#2698 LEXIFIED "like" NIL]
|  |  |  |  |  +  |  |  |  |--NP [#2699 ACTIVE NIL]
|  |  |  |  |  +  |  |  |  |--TOINF [#2700 ACTIVE NIL]
|  |  |  |  +--REL_S [#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|  |  |  |  |  |--S [#2702 EXPANDED NIL]
|  |  |  |  |  |  |--NP [#2703 ACTIVE NIL]
|  |  |  |  |  |  |--VP [#2704 EXPANDED NIL]
|  |  |  |  |  |  |  |--ADVP [#2705 ACTIVE NIL]
|  |  |  |  |  |  |  |--VP [#2706 EXPANDED NIL]
|  |  |  |  |  |  |  |  |--AUX [#2707 LEXIFIED "would" NIL]
|  |  |  |  |  |  |  |  |--VERB [#2708 LEXIFIED "like" NIL]
|  |  |  |  |  |  |  |  |--NP [#2709 ACTIVE NIL]
|  |  |  |  |  |  |  |  |--TOINF [#2710 ACTIVE NIL]
|  |  |  |  |  |  |  |  +--TOINF [#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
|  |  |  |  |  |  |  |  +  |--TO [#2712 LEXIFIED "to" NIL]

```

```

+ |--VP (#2713 EXPANDED NIL)
+ |--VERB (#2714 LEXIFIED "ask" NIL)
+ |--NP (#2715 ACTIVE NIL)
+ |--PREP (#2716 LEXIFIED "about" NIL)
+ |--NP (#2717 TRACE "" NIL)
+---TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
+ |--TO (#2719 LEXIFIED "to" NIL)
+ |--VP (#2720 EXPANDED NIL)
+ |--VERB (#2721 LEXIFIED "ask" NIL)
+ |--NP (#2722 ACTIVE NIL)
+ |--NP (#2723 TRACE "" NIL)

|--SIGN (#2550 LEXIFIED "." NIL)
**** expanding ADVP (#2705 ACTIVE NIL) ****
[[SYN [(CAT ADVP)
  (MODI-POST FRONT)]]]
[SEM [(PARM !X2())
  (RESTR [(RELN JUST-ADV-1)
    (ENTITY !X2)])]]]
[SLASH [(SYN [(CAT NP)
  (AGR [])
  (PREP-INFO [])]]]
[SEM [(PARM !X1())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X1)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD)])]]]]]
**** re-activated (<ADVP-JUST!ADVP-TEMPLATE>) ****
**** Applying (ADVP-JUST!ADVP-TEMPLATE) ****

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
|--S (#2544 EXPANDED NIL)
+ |--(ADVP CONJ_S) (#2545 ACTIVE NIL)
+ |--SIGN (#2546 LEXIFIED "," NIL)
+ |--S (#2547 EXPANDED NIL)
+ |--NP (#2548 ACTIVE NIL)
+ |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
+ |--VERB (#2686 LEXIFIED "have" NIL)
+ |--NP (#2687 EXPANDED (NP-NP-REL))
+ |--NP (#2688 ACTIVE NIL)
+ |--REL_S (#2689 ACTIVE NIL)
+---REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
+ + |--CONJ (#2691 LEXIFIED "that" NIL)
+ + |--S (#2692 EXPANDED NIL)
+ + |--NP (#2693 ACTIVE NIL)
+ + |--VP (#2694 EXPANDED NIL)
+ + |--ADVP (#2695 ACTIVE NIL)
+ + |--VP (#2696 EXPANDED NIL)
+ + |--AUX (#2697 LEXIFIED "would" NIL)
+ + |--VERB (#2698 LEXIFIED "like" NIL)
+ + |--NP (#2699 ACTIVE NIL)
+ + |--TOINF (#2700 ACTIVE NIL)
+---REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
+ |--S (#2702 EXPANDED NIL)
+ |--NP (#2703 ACTIVE NIL)
+ |--VP (#2704 EXPANDED NIL)
+ |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
+ |--ADV (#2724 LEXIFIED "just" NIL)
+ |--VP (#2706 EXPANDED NIL)
+ |--AUX (#2707 LEXIFIED "would" NIL)
+ |--VERB (#2708 LEXIFIED "like" NIL)
+ |--NP (#2709 ACTIVE NIL)
+ |--TOINF (#2710 ACTIVE NIL)
+---TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
+ + |--TO (#2712 LEXIFIED "to" NIL)
+ + |--VP (#2713 EXPANDED NIL)
+ + |--VERB (#2714 LEXIFIED "ask" NIL)
+ + |--NP (#2715 ACTIVE NIL)
+ + |--PREP (#2716 LEXIFIED "about" NIL)
+ + |--NP (#2717 TRACE "" NIL)
+---TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
+ |--TO (#2719 LEXIFIED "to" NIL)
+ |--VP (#2720 EXPANDED NIL)
+ |--VERB (#2721 LEXIFIED "ask" NIL)
+ |--NP (#2722 ACTIVE NIL)
+ |--NP (#2723 TRACE "" NIL)

|--SIGN (#2550 LEXIFIED "." NIL)
**** expanding NP (#2703 ACTIVE NIL) ****
[[SYN [(CAT NP)
  (CASE NOM)
  (AGR [])
  (ANY-ENV -)
  (ARTI [])]]]
[SEM [(LABEL *SPEAKER*)]]
[DEFSEM [(LABEL *DEF-SPEAKER*)]]
[SLASH [(SYN [(CAT NP)
  (AGR [])
  (PREP-INFO [])]]]
[SEM [(PARM !X1())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X1)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)

```

```

      (NUMBER SING)
      (PERSON THIRD))))))
**** re-activated (PP <PRON-SPEAKER-I> <ADVP-SPEAKER-HERE> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (NP-NN NN-PRON PRON-SPEAKER-I) ****

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
--S (#2544 EXPANDED NIL)
|--(ADVP CONJ_S) (#2545 ACTIVE NIL)
|--SIGN (#2546 LEXIFIED ", " NIL)
|--S (#2547 EXPANDED NIL)
|   |--NP (#2548 ACTIVE NIL)
|   |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|       |--VERB (#2686 LEXIFIED "have" NIL)
|       |--NP (#2687 EXPANDED (NP-NP-REL))
|           |--NP (#2688 ACTIVE NIL)
|           |--REL_S (#2689 ACTIVE NIL)
|               +--REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|                   + |--CONJ (#2691 LEXIFIED "that" NIL)
|                   + |--S (#2692 EXPANDED NIL)
|                   + |   |--NP (#2693 ACTIVE NIL)
|                   + |   |--VP (#2694 EXPANDED NIL)
|                   + |       |--ADVP (#2695 ACTIVE NIL)
|                   + |       |--VP (#2696 EXPANDED NIL)
|                   + |           |--AUX (#2697 LEXIFIED "would" NIL)
|                   + |           |--VERB (#2698 LEXIFIED "like" NIL)
|                   + |           |--NP (#2699 ACTIVE NIL)
|                   + |           |--TOINF (#2700 ACTIVE NIL)
|                   +--REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|                       |--S (#2702 EXPANDED NIL)
|                           |--NP (#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|                               |--NN (#2725 EXPANDED NIL)
|                               |--PRON (#2726 LEXIFIED "I" NIL)
|                               |--VP (#2704 EXPANDED NIL)
|                                   |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
|                                   |--ADV (#2724 LEXIFIED "just" NIL)
|                                   |--VP (#2706 EXPANDED NIL)
|                                       |--AUX (#2707 LEXIFIED "would" NIL)
|                                       |--VERB (#2708 LEXIFIED "like" NIL)
|                                       |--NP (#2709 ACTIVE NIL)
|                                       |--TOINF (#2710 ACTIVE NIL)
|                                           +--TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|                                           + |--TO (#2712 LEXIFIED "to" NIL)
|                                           + |--VP (#2713 EXPANDED NIL)
|                                           + |   |--VERB (#2714 LEXIFIED "ask" NIL)
|                                           + |   |--NP (#2715 ACTIVE NIL)
|                                           + |   |--PREP (#2716 LEXIFIED "about" NIL)
|                                           + |   |--NP (#2717 TRACE "" NIL)
|                                           +--TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|                                               |--TO (#2719 LEXIFIED "to" NIL)
|                                               |--VP (#2720 EXPANDED NIL)
|                                                   |--VERB (#2721 LEXIFIED "ask" NIL)
|                                                   |--NP (#2722 ACTIVE NIL)
|                                                   |--NP (#2723 TRACE "" NIL)
|--SIGN (#2550 LEXIFIED ", " NIL)
**** expanding TOINF (#2700 ACTIVE NIL) ****
[[SYN [(CAT TOINF)
  (VFORM BSE)
  (AGR [])
  (SUBJ-INFO [(SUBJ-SEM [])
    (SUBJ-DEFSEM [])])])
  (SEM [(RELN ASK-V-1)
    (AGEN [(LABEL *SPEAKER*)])
    (RECP [])
    (OBJE !X2[(PARM !X1())
      (RESTR [(RELN SOMETHING-PRON-1)
        (ENTITY !X1)])
      (INDEX [(DEFINT INDEF)
        (GENDER NONE)
        (NUMBER SING)
        (PERSON THIRD)]))])
    (SLASH [(SYN [(CAT NP)
      (AGR [])
      (PREP-INFO [])]
      (SEM !X2))])
    (**** re-activated (PP VP-RECP!VP-MODI-TEMPLATE <VP-ASK-1-NP-ABOUT!VP-TN.PR> <VP-ASK-1-NP-NP!VP-DN.N> S-MODI-TOO S-NP-VP-SUBORD
CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN
NN-N NN-PRON AP-ADJ) ****
**** Applying (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR) ****
**** condition (Slash No.1) is satisfied on (#2733) ****
**** fs before application ****
[(SEM !X2[(PARM !X1())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X1)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD)]))])
  (DEFSEM [(LABEL *IT*)])
  (SYN [(CAT NP)
    (CASE ACC)])
  (SLASH [(SYN [(CAT NP)
    (AGR [])
    (PREP-INFO [])])
    (SEM !X2))])

```

```

(SEM !X2))))
**** Slash No.1 is applied ****
**** fs on the slash node after application
[(SEM !X3[(PARM !X2())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X2)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD))])]
  (DEFSEM [(LABEL *IT*)])
  (SYN !X1[(CAT NP)
    (AGR [])
    (PREP-INFO [])
    (CASE ACC)
    (LEX """)])]
  (SLASH [(SYN !X1)
    (SEM !X3)]))
**** Applying (TOINF VP-ASK-1-NP-NP!VP-DN.N) ****
**** condition (Slash No.1) is satisfied on [#2739] ****
**** fs before application ****
[(SEM !X2[(PARM !X1())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X1)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD))])]
  (SYN [(CAT NP)
    (CASE ACC)])
  (SLASH [(SYN [(CAT NP)
    (AGR [])
    (PREP-INFO [])])
    (SEM !X2)]))
**** Slash No.1 is applied ****
**** fs on the slash node after application
[(SEM !X3[(PARM !X2())
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X2)])
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD))])]
  (SYN !X1[(CAT NP)
    (AGR [])
    (PREP-INFO [])
    (CASE ACC)
    (LEX """)])]
  (SLASH [(SYN !X1)
    (SEM !X3)]))

```

```

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
--S [#2544 EXPANDED NIL]
|--(ADVP CONJ_S) [#2545 ACTIVE NIL]
|--SIGN [#2546 LEXIFIED "," NIL]
|--S [#2547 EXPANDED NIL]
|  |--NP [#2548 ACTIVE NIL]
|  |--VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|    |--VERB [#2686 LEXIFIED "have" NIL]
|    |--NP [#2687 EXPANDED (NP-NP-REL)]
|      |--NP [#2688 ACTIVE NIL]
|      |--REL_S [#2689 ACTIVE NIL]
|        +--REL_S [#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|          + |--CONJ [#2691 LEXIFIED "that" NIL]
|          + |--S [#2692 EXPANDED NIL]
|            + |--NP [#2693 ACTIVE NIL]
|              + |--VP [#2694 EXPANDED NIL]
|                + |--ADVP [#2695 ACTIVE NIL]
|                  + |--VP [#2696 EXPANDED NIL]
|                    + |--AUX [#2697 LEXIFIED "would" NIL]
|                      + |--VERB [#2698 LEXIFIED "like" NIL]
|                        + |--NP [#2699 ACTIVE NIL]
|                          + |--TOINF [#2700 ACTIVE NIL]
|                            +--TOINF [#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
|                              + |--TO [#2728 LEXIFIED "to" NIL]
|                                + |--VP [#2729 EXPANDED NIL]
|                                  + |--VERB [#2730 LEXIFIED "ask" NIL]
|                                    + |--NP [#2731 ACTIVE NIL]
|                                      + |--PREP [#2732 LEXIFIED "about" NIL]
|                                        + |--NP [#2733 TRACE "" NIL]
|                                          +--TOINF [#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
|                                            + |--TO [#2735 LEXIFIED "to" NIL]
|                                              + |--VP [#2736 EXPANDED NIL]
|                                                + |--VERB [#2737 LEXIFIED "ask" NIL]
|                                                  + |--NP [#2738 ACTIVE NIL]
|                                                    + |--NP [#2739 TRACE "" NIL]
|              +--REL_S [#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|                |--S [#2702 EXPANDED NIL]
|                  |--NP [#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|                    |--NN [#2725 EXPANDED NIL]
|                      |--PRON [#2726 LEXIFIED "I" NIL]
|                        |--VP [#2704 EXPANDED NIL]
|                          |--ADVP [#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|                            |--ADV [#2724 LEXIFIED "just" NIL]

```

```

|--VP [#2706 EXPANDED NIL]
|--AUX [#2707 LEXIFIED "would" NIL]
|--VERB [#2708 LEXIFIED "like" NIL]
|--NP [#2709 ACTIVE NIL]
|--TOINF [#2710 ACTIVE NIL]
+--TOINF [#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
+ |--TO [#2712 LEXIFIED "to" NIL]
+ |--VP [#2713 EXPANDED NIL]
+ +--VERB [#2714 LEXIFIED "ask" NIL]
+ +--NP [#2715 ACTIVE NIL]
+ +--PREP [#2716 LEXIFIED "about" NIL]
+ +--NP [#2717 TRACE "" NIL]
+--TOINF [#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
+ |--TO [#2719 LEXIFIED "to" NIL]
+ |--VP [#2720 EXPANDED NIL]
+ +--VERB [#2721 LEXIFIED "ask" NIL]
+ +--NP [#2722 ACTIVE NIL]
+ +--NP [#2723 TRACE "" NIL]
--SIGN [#2550 LEXIFIED "." NIL]
**** expanding ADVP [#2695 ACTIVE NIL] ****
([SYN [(CAT ADVP)
      (MODI-POSI FRONT)]])
(SEM [(PARM !X2())
      (RESTR [(RELN JUST-ADV-1)
              (ENTITY !X2)])])
(SLASH [(SYN [(CAT NP)
              (AGR [])
              (PREP-INFO [])])
        (SEM [(PARM !X1())
              (RESTR [(RELN SOMETHING-PRON-1)
                      (ENTITY !X1)])
              (INDEX [(DEFINT INDEF)
                      (GENDER NONE)
                      (NUMBER SING)
                      (PERSON THIRD)])])])])
**** re-activated (<ADVP-JUST!ADVP-TEMPLATE>) ****
**** Applying (ADVP-JUST!ADVP-TEMPLATE) ****

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
--S [#2544 EXPANDED NIL]
|--(ADVP CONJ_S) [#2545 ACTIVE NIL]
|--SIGN [#2546 LEXIFIED "," NIL]
--S [#2547 EXPANDED NIL]
|--NP [#2548 ACTIVE NIL]
|--VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
+--VERB [#2686 LEXIFIED "have" NIL]
+--NP [#2687 EXPANDED (NP-NP-REL)]
+--NP [#2688 ACTIVE NIL]
+--REL_S [#2689 ACTIVE NIL]
+--REL_S [#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
+ |--CONJ [#2691 LEXIFIED "that" NIL]
+ |--S [#2692 EXPANDED NIL]
+ +--NP [#2693 ACTIVE NIL]
+ +--VP [#2694 EXPANDED NIL]
+ +--ADVP [#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
+ +--ADV [#2740 LEXIFIED "just" NIL]
+ +--VP [#2696 EXPANDED NIL]
+ +--AUX [#2697 LEXIFIED "would" NIL]
+ +--VERB [#2698 LEXIFIED "like" NIL]
+ +--NP [#2699 ACTIVE NIL]
+ +--TOINF [#2700 ACTIVE NIL]
+ +--TOINF [#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
+ + |--TO [#2728 LEXIFIED "to" NIL]
+ + |--VP [#2729 EXPANDED NIL]
+ + +--VERB [#2730 LEXIFIED "ask" NIL]
+ + +--NP [#2731 ACTIVE NIL]
+ + +--PREP [#2732 LEXIFIED "about" NIL]
+ + +--NP [#2733 TRACE "" NIL]
+ +--TOINF [#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
+ + |--TO [#2735 LEXIFIED "to" NIL]
+ + |--VP [#2736 EXPANDED NIL]
+ + +--VERB [#2737 LEXIFIED "ask" NIL]
+ + +--NP [#2738 ACTIVE NIL]
+ + +--NP [#2739 TRACE "" NIL]
+--REL_S [#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|--S [#2702 EXPANDED NIL]
+--NP [#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
+--NN [#2725 EXPANDED NIL]
+--PRON [#2726 LEXIFIED "I" NIL]
+--VP [#2704 EXPANDED NIL]
+--ADVP [#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
+--ADV [#2724 LEXIFIED "just" NIL]
+--VP [#2706 EXPANDED NIL]
+--AUX [#2707 LEXIFIED "would" NIL]
+--VERB [#2708 LEXIFIED "like" NIL]
+--NP [#2709 ACTIVE NIL]
+--TOINF [#2710 ACTIVE NIL]
+--TOINF [#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
+ |--TO [#2712 LEXIFIED "to" NIL]
+ |--VP [#2713 EXPANDED NIL]
+ +--VERB [#2714 LEXIFIED "ask" NIL]
+ +--NP [#2715 ACTIVE NIL]
+ +--PREP [#2716 LEXIFIED "about" NIL]
+ +--NP [#2717 TRACE "" NIL]

```

```

      +---TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
      |---TO (#2719 LEXIFIED "to" NIL)
      |---VP (#2720 EXPANDED NIL)
      |   |--VERB (#2721 LEXIFIED "ask" NIL)
      |   |--NP (#2722 ACTIVE NIL)
      |   |--NP (#2723 TRACE "" NIL)
--SIGN (#2550 LEXIFIED "." NIL)
**** expanding NP (#2693 ACTIVE NIL) ****
([SYN [(CAT NP)
      (CASE NOM)
      (AGR [])
      (ANY-ENV -)
      (ARTI [])])
 (SEM [(LABEL *SPEAKER*)])
 (DEFSEM [(LABEL *DEF-SPEAKER*)])
 (SLASH [(SYN [(CAT NP)
               (AGR [])
               (PREP-INFO [])])
          (SEM [(PARM !X1())
                (RESTR [(RELN SOMETHING-PRON-1)
                        (ENTITY !X1)])
                (INDEX [(DEFINT INDEF)
                        (GENDER NONE)
                        (NUMBER SING)
                        (PERSON THIRD)]))])])])
**** re-activated (PP <PRON-SPEAKER-I> <ADVP-SPEAKER-HERE> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (NP-NN NN-PRON PRON-SPEAKER-I) ****
S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
--S (#2544 EXPANDED NIL)
|---(ADVP CONJ_S) (#2545 ACTIVE NIL)
|---SIGN (#2546 LEXIFIED "," NIL)
|---S (#2547 EXPANDED NIL)
|   |--NP (#2548 ACTIVE NIL)
|   |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|       |--VERB (#2686 LEXIFIED "have" NIL)
|       |--NP (#2687 EXPANDED (NP-NP-REL))
|           |--NP (#2688 ACTIVE NIL)
|           |--REL_S (#2689 ACTIVE NIL)
|               +---REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|               + |---CONJ (#2691 LEXIFIED "that" NIL)
|               + |---S (#2692 EXPANDED NIL)
|               + |   |--NP (#2693 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|               + |   |--NN (#2741 EXPANDED NIL)
|               + |   |--PRON (#2742 LEXIFIED "I" NIL)
|               + |   |--VP (#2694 EXPANDED NIL)
|               + |       |--ADVP (#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
|               + |       |--ADV (#2740 LEXIFIED "just" NIL)
|               + |       |--VP (#2696 EXPANDED NIL)
|               + |           |--AUX (#2697 LEXIFIED "would" NIL)
|               + |           |--VERB (#2698 LEXIFIED "like" NIL)
|               + |           |--NP (#2699 ACTIVE NIL)
|               + |           |--TOINF (#2700 ACTIVE NIL)
|               + |               +---TOINF (#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|               + |               + |---TO (#2728 LEXIFIED "to" NIL)
|               + |               + |---VP (#2729 EXPANDED NIL)
|               + |               + |   |--VERB (#2730 LEXIFIED "ask" NIL)
|               + |               + |   |--NP (#2731 ACTIVE NIL)
|               + |               + |   |--PREP (#2732 LEXIFIED "about" NIL)
|               + |               + |   |--NP (#2733 TRACE "" NIL)
|               + |               +---TOINF (#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|               + |               + |---TO (#2735 LEXIFIED "to" NIL)
|               + |               + |---VP (#2736 EXPANDED NIL)
|               + |               + |   |--VERB (#2737 LEXIFIED "ask" NIL)
|               + |               + |   |--NP (#2738 ACTIVE NIL)
|               + |               + |   |--NP (#2739 TRACE "" NIL)
|               +---REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|               |---S (#2702 EXPANDED NIL)
|                   |--NP (#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|                   |--NN (#2725 EXPANDED NIL)
|                   |--PRON (#2726 LEXIFIED "I" NIL)
|                   |--VP (#2704 EXPANDED NIL)
|                       |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
|                       |--ADV (#2724 LEXIFIED "just" NIL)
|                       |--VP (#2706 EXPANDED NIL)
|                           |--AUX (#2707 LEXIFIED "would" NIL)
|                           |--VERB (#2708 LEXIFIED "like" NIL)
|                           |--NP (#2709 ACTIVE NIL)
|                           |--TOINF (#2710 ACTIVE NIL)
|                           +---TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|                           + |---TO (#2712 LEXIFIED "to" NIL)
|                           + |---VP (#2713 EXPANDED NIL)
|                           + |   |--VERB (#2714 LEXIFIED "ask" NIL)
|                           + |   |--NP (#2715 ACTIVE NIL)
|                           + |   |--PREP (#2716 LEXIFIED "about" NIL)
|                           + |   |--NP (#2717 TRACE "" NIL)
|                           +---TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|                           + |---TO (#2719 LEXIFIED "to" NIL)
|                           + |---VP (#2720 EXPANDED NIL)
|                           + |   |--VERB (#2721 LEXIFIED "ask" NIL)
|                           + |   |--NP (#2722 ACTIVE NIL)
|                           + |   |--NP (#2723 TRACE "" NIL)
|                           +---SIGN (#2550 LEXIFIED "." NIL)

```

**** expanding NP [#2688 ACTIVE NIL] ****

```
[[SYN [(CAT NP)
  [WH []]
  [CASE ACC]
  [AGR []]
  [PREP-INFO []]
  [ANY-ENV -]
  [ARTI []]
  [INSERT-V []]]]
[SEM [(PARM !X1[])]
  (RESTR [(RELN SOMETHING-PRON-1)
    (ENTITY !X1[])]
  (INDEX [(DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD)]))]]
[SLASH []]]
```

**** re-activated (PP <CONJ_S-WH-1> <PRON-SOMETHING> <PRON-ANYTHING> NN-INDEX-COVER S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING) ****

S-TOP [#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]

```
--S [#2544 EXPANDED NIL]
| --(ADVP CONJ_S) [#2545 ACTIVE NIL]
| --SIGN [#2546 LEXIFIED "," NIL]
| --S [#2547 EXPANDED NIL]
|   --NP [#2548 ACTIVE NIL]
|   --VP [#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|     --VERB [#2686 LEXIFIED "have" NIL]
|     --NP [#2687 EXPANDED (NP-NP-REL)]
|       --NP [#2688 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING)]
|         --NN [#2743 EXPANDED NIL]
|         --PRON [#2744 LEXIFIED "something" NIL]
|       --REL_S [#2689 ACTIVE NIL]
|         +-REL_S [#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|           + --CONJ [#2691 LEXIFIED "that" NIL]
|           + --S [#2692 EXPANDED NIL]
|             --NP [#2693 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|               --NN [#2741 EXPANDED NIL]
|               --PRON [#2742 LEXIFIED "I" NIL]
|             --VP [#2694 EXPANDED NIL]
|               --ADVP [#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|                 --ADV [#2740 LEXIFIED "just" NIL]
|               --VP [#2696 EXPANDED NIL]
|                 --AUX [#2697 LEXIFIED "would" NIL]
|                 --VERB [#2698 LEXIFIED "like" NIL]
|                 --NP [#2699 ACTIVE NIL]
|                 --TOINF [#2700 ACTIVE NIL]
|               +-TOINF [#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
|                 + --TO [#2728 LEXIFIED "to" NIL]
|                 + --VP [#2729 EXPANDED NIL]
|                   --VERB [#2730 LEXIFIED "ask" NIL]
|                   --NP [#2731 ACTIVE NIL]
|                   --PREP [#2732 LEXIFIED "about" NIL]
|                   --NP [#2733 TRACE "" NIL]
|                 +-TOINF [#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
|                   --TO [#2735 LEXIFIED "to" NIL]
|                   --VP [#2736 EXPANDED NIL]
|                     --VERB [#2737 LEXIFIED "ask" NIL]
|                     --NP [#2738 ACTIVE NIL]
|                     --NP [#2739 TRACE "" NIL]
|               +-REL_S [#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|                 --S [#2702 EXPANDED NIL]
|                   --NP [#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|                     --NN [#2725 EXPANDED NIL]
|                     --PRON [#2726 LEXIFIED "I" NIL]
|                   --VP [#2704 EXPANDED NIL]
|                     --ADVP [#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|                       --ADV [#2724 LEXIFIED "just" NIL]
|                     --VP [#2706 EXPANDED NIL]
|                       --AUX [#2707 LEXIFIED "would" NIL]
|                       --VERB [#2708 LEXIFIED "like" NIL]
|                       --NP [#2709 ACTIVE NIL]
|                       --TOINF [#2710 ACTIVE NIL]
|                     +-TOINF [#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
|                       + --TO [#2712 LEXIFIED "to" NIL]
|                       + --VP [#2713 EXPANDED NIL]
|                         --VERB [#2714 LEXIFIED "ask" NIL]
|                         --NP [#2715 ACTIVE NIL]
|                         --PREP [#2716 LEXIFIED "about" NIL]
|                         --NP [#2717 TRACE "" NIL]
|                     +-TOINF [#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
|                       --TO [#2719 LEXIFIED "to" NIL]
|                       --VP [#2720 EXPANDED NIL]
|                         --VERB [#2721 LEXIFIED "ask" NIL]
|                         --NP [#2722 ACTIVE NIL]
|                         --NP [#2723 TRACE "" NIL]
|                 --SIGN [#2550 LEXIFIED "," NIL]
```

**** expanding NP [#2548 ACTIVE NIL] ****

```
[[SYN [(CAT NP)
  [CASE NOM]
  [AGR []]
  [ANY-ENV -]
  [ARTI []]]]
```



```

(SEM [(LABEL *SPEAKER*)])
(DEFSEM [(LABEL *UNKNOWN*)])
(SLASH [])
**** re-activated (PP <PRON-SPEAKER-I> <ADVP-SPEAKER-HERE> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
**** Applying (NP-NN NN-PRON PRON-SPEAKER-I) ****

```

```

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
|--S (#2544 EXPANDED NIL)
|  |--(ADVP CONJ_S) (#2545 ACTIVE NIL)
|  |--SIGN (#2546 LEXIFIED " " NIL)
|  |--S (#2547 EXPANDED NIL)
|     |--NP (#2548 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|     |  |--NN (#2745 EXPANDED NIL)
|     |  |  |--PRON (#2746 LEXIFIED "I" NIL)
|     |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|     |  |--VERB (#2686 LEXIFIED "have" NIL)
|     |  |--NP (#2687 EXPANDED (NP-NP-REL))
|     |     |--NP (#2688 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING))
|     |     |  |--NN (#2743 EXPANDED NIL)
|     |     |  |  |--PRON (#2744 LEXIFIED "something" NIL)
|     |  |--REL_S (#2689 ACTIVE NIL)
|     |     +--REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|     |     +  |--CONJ (#2691 LEXIFIED "that" NIL)
|     |     +  |--S (#2692 EXPANDED NIL)
|     |        +  |--NP (#2693 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|     |        +  |  |--NN (#2741 EXPANDED NIL)
|     |        +  |  |  |--PRON (#2742 LEXIFIED "I" NIL)
|     |        +  |--VP (#2694 EXPANDED NIL)
|     |        |  |--ADVP (#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
|     |        |  |  |--ADV (#2740 LEXIFIED "just" NIL)
|     |        |--VP (#2696 EXPANDED NIL)
|     |        |  |--AUX (#2697 LEXIFIED "would" NIL)
|     |        |  |--VERB (#2698 LEXIFIED "like" NIL)
|     |        |  |--NP (#2699 ACTIVE NIL)
|     |        |  |--TOINF (#2700 ACTIVE NIL)
|     |        |  +--TOINF (#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|     |        |  +  |--TO (#2728 LEXIFIED "to" NIL)
|     |        |  +  |--VP (#2729 EXPANDED NIL)
|     |        |  |  |--VERB (#2730 LEXIFIED "ask" NIL)
|     |        |  |  |--NP (#2731 ACTIVE NIL)
|     |        |  |  |--PREP (#2732 LEXIFIED "about" NIL)
|     |        |  |  |--NP (#2733 TRACE "" NIL)
|     |        |  +--TOINF (#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|     |        |  |  |--TO (#2735 LEXIFIED "to" NIL)
|     |        |  |  |--VP (#2736 EXPANDED NIL)
|     |        |  |  |--VERB (#2737 LEXIFIED "ask" NIL)
|     |        |  |  |--NP (#2738 ACTIVE NIL)
|     |        |  |  |--NP (#2739 TRACE "" NIL)
|     |  +--REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|     |  |--S (#2702 EXPANDED NIL)
|     |     |--NP (#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|     |     |  |--NN (#2725 EXPANDED NIL)
|     |     |  |  |--PRON (#2726 LEXIFIED "I" NIL)
|     |     |--VP (#2704 EXPANDED NIL)
|     |     |  |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
|     |     |  |  |--ADV (#2724 LEXIFIED "just" NIL)
|     |     |--VP (#2706 EXPANDED NIL)
|     |     |  |--AUX (#2707 LEXIFIED "would" NIL)
|     |     |  |--VERB (#2708 LEXIFIED "like" NIL)
|     |     |  |--NP (#2709 ACTIVE NIL)
|     |     |  |--TOINF (#2710 ACTIVE NIL)
|     |     |  +--TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|     |     |  +  |--TO (#2712 LEXIFIED "to" NIL)
|     |     |  +  |--VP (#2713 EXPANDED NIL)
|     |     |  |  |--VERB (#2714 LEXIFIED "ask" NIL)
|     |     |  |  |--NP (#2715 ACTIVE NIL)
|     |     |  |  |--PREP (#2716 LEXIFIED "about" NIL)
|     |     |  |  |--NP (#2717 TRACE "" NIL)
|     |     |  +--TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|     |     |  |  |--TO (#2719 LEXIFIED "to" NIL)
|     |     |  |  |--VP (#2720 EXPANDED NIL)
|     |     |  |  |--VERB (#2721 LEXIFIED "ask" NIL)
|     |     |  |  |--NP (#2722 ACTIVE NIL)
|     |     |  |  |--NP (#2723 TRACE "" NIL)
|  |--SIGN (#2550 LEXIFIED " " NIL)

```

```

**** expanding (ADVP CONJ_S) (#2545 ACTIVE NIL) ****

```

```

[[SYN [(CAT (:SET ADVP CONJ_S))]]
(SEM [(PARM !X1())
      (RESTR [(RELN WELL-INTERJ-1)
               (ENTITY !X1())])])

```

```

**** activated (PP <CONJ_S-WH-1> <ADVP-WELL-INTERJ!ADVP-TEMPLATE> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP
VP-GIVE-ME-NP VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****

```

```

**** Applying (CONJ_S-WH-1) ****

```

```

**** Applying (ADVP-WELL-INTERJ!ADVP-TEMPLATE) ****

```

```

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
|--S (#2544 EXPANDED NIL)
|  |--(ADVP CONJ_S) (#2545 ACTIVE NIL)
|  +--CONJ_S (#2747 ASSUMED (CONJ_S-WH-1))
|  +  |--BOTTOM (#2748 ACTIVE NIL)
|  +  |--S (#2749 ACTIVE NIL)
|  +--ADVP (#2750 ASSUMED (ADVP-WELL-INTERJ!ADVP-TEMPLATE))
|     |--ADV (#2751 LEXIFIED "well" NIL)

```

```

|--SIGN (#2546 LEXIFIED "," NIL)
|--S (#2547 EXPANDED NIL)
  |--NP (#2548 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
  | |--NN (#2745 EXPANDED NIL)
  | | |--PRON (#2746 LEXIFIED "I" NIL)
  |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
  | |--VERB (#2686 LEXIFIED "have" NIL)
  | |--NP (#2687 EXPANDED (NP-NP-REL))
  | | |--NP (#2688 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING))
  | | | |--NN (#2743 EXPANDED NIL)
  | | | | |--PRON (#2744 LEXIFIED "something" NIL)
  | |--REL_S (#2689 ACTIVE NIL)
  | +--REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
  | + | |--CONJ (#2691 LEXIFIED "that" NIL)
  | + | +--S (#2692 EXPANDED NIL)
  | + | | |--NP (#2693 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
  | + | | | |--NN (#2741 EXPANDED NIL)
  | + | | | | |--PRON (#2742 LEXIFIED "I" NIL)
  | + | | |--VP (#2694 EXPANDED NIL)
  | + | | | |--ADVP (#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
  | + | | | | |--ADV (#2740 LEXIFIED "just" NIL)
  | + | | | |--VP (#2696 EXPANDED NIL)
  | + | | | | |--AUX (#2697 LEXIFIED "would" NIL)
  | + | | | | |--VERB (#2698 LEXIFIED "like" NIL)
  | + | | | | |--NP (#2699 ACTIVE NIL)
  | + | | | | |--TOINF (#2700 ACTIVE NIL)
  | + | | +--TOINF (#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
  | + | | + | |--TO (#2728 LEXIFIED "to" NIL)
  | + | | + | | |--VP (#2729 EXPANDED NIL)
  | + | | + | | | |--VERB (#2730 LEXIFIED "ask" NIL)
  | + | | + | | | |--NP (#2731 ACTIVE NIL)
  | + | | + | | | |--PREP (#2732 LEXIFIED "about" NIL)
  | + | | + | | | |--NP (#2733 TRACE "" NIL)
  | + | | +--TOINF (#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
  | + | | | |--TO (#2735 LEXIFIED "to" NIL)
  | + | | | |--VP (#2736 EXPANDED NIL)
  | + | | | | |--VERB (#2737 LEXIFIED "ask" NIL)
  | + | | | | |--NP (#2738 ACTIVE NIL)
  | + | | | | |--NP (#2739 TRACE "" NIL)
  +--REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
  | |--S (#2702 EXPANDED NIL)
  | | |--NP (#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
  | | | |--NN (#2725 EXPANDED NIL)
  | | | | |--PRON (#2726 LEXIFIED "I" NIL)
  | |--VP (#2704 EXPANDED NIL)
  | | |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
  | | | |--ADV (#2724 LEXIFIED "just" NIL)
  | | |--VP (#2706 EXPANDED NIL)
  | | | |--AUX (#2707 LEXIFIED "would" NIL)
  | | | |--VERB (#2708 LEXIFIED "like" NIL)
  | | | |--NP (#2709 ACTIVE NIL)
  | | | |--TOINF (#2710 ACTIVE NIL)
  | | +--TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
  | | + | |--TO (#2712 LEXIFIED "to" NIL)
  | | + | |--VP (#2713 EXPANDED NIL)
  | | + | | |--VERB (#2714 LEXIFIED "ask" NIL)
  | | + | | |--NP (#2715 ACTIVE NIL)
  | | + | | |--PREP (#2716 LEXIFIED "about" NIL)
  | | + | | |--NP (#2717 TRACE "" NIL)
  | | +--TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
  | | | |--TO (#2719 LEXIFIED "to" NIL)
  | | | |--VP (#2720 EXPANDED NIL)
  | | | | |--VERB (#2721 LEXIFIED "ask" NIL)
  | | | | |--NP (#2722 ACTIVE NIL)
  | | | | |--NP (#2723 TRACE "" NIL)
|--SIGN (#2550 LEXIFIED "," NIL)
**** expanding S (#2749 ACTIVE NIL) ****
((SYN ((CAT S)
  (INV -)
  (ADV-CL -)
  (S-DEPTH SUBORD)))
  (SEM ((RELN WELL-INTERJ-1)
    (ENTITY !X1()))
  (SLASH ((SYN ((CAT ))
    (WH +)
    (AGR []))
    (SEM !X1))))
#### nearly activated (PP S-TOP S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 REL_S REL_S-2 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ####
#### uncovered top arcs (ENTITY) ####
++++ activated NIL ++++
#### Sub tree (#2747 ASSUMED (CONJ_S-WH-1)) was removed ####

S-TOP (#2543 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
|--S (#2544 EXPANDED NIL)
  |--ADVP (#2750 EXPANDED (ADVP-WELL-INTERJ!ADVP-TEMPLATE))
  | |--ADV (#2751 LEXIFIED "well" NIL)
  |--SIGN (#2546 LEXIFIED "," NIL)
  |--S (#2547 EXPANDED NIL)
  | |--NP (#2548 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
  | | |--NN (#2745 EXPANDED NIL)
  | | | |--PRON (#2746 LEXIFIED "I" NIL)
  |--VP (#2549 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
  | |--VERB (#2686 LEXIFIED "have" NIL)

```

```

|--NP (#2687 EXPANDED (NP-NP-REL))
|--NP (#2688 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING))
| |--NN (#2743 EXPANDED NIL)
| |--PRON (#2744 LEXIFIED "something" NIL)
|--REL_S (#2689 ACTIVE NIL)
+--REL_S (#2690 ASSUMED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
+ |--CONJ (#2691 LEXIFIED "that" NIL)
+ |--S (#2692 EXPANDED NIL)
+ |--NP (#2693 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
+ |--NN (#2741 EXPANDED NIL)
+ |--PRON (#2742 LEXIFIED "I" NIL)
+ |--VP (#2694 EXPANDED NIL)
+ |--ADVP (#2695 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
+ |--ADV (#2740 LEXIFIED "just" NIL)
+ |--VP (#2696 EXPANDED NIL)
+ |--AUX (#2697 LEXIFIED "would" NIL)
+ |--VERB (#2698 LEXIFIED "like" NIL)
+ |--NP (#2699 ACTIVE NIL)
+ |--TOINF (#2700 ACTIVE NIL)
+ +--TOINF (#2727 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
+ + |--TO (#2728 LEXIFIED "to" NIL)
+ + |--VP (#2729 EXPANDED NIL)
+ + |--VERB (#2730 LEXIFIED "ask" NIL)
+ + |--NP (#2731 ACTIVE NIL)
+ + |--PREP (#2732 LEXIFIED "about" NIL)
+ + |--NP (#2733 TRACE "" NIL)
+ +--TOINF (#2734 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
+ + |--TO (#2735 LEXIFIED "to" NIL)
+ + |--VP (#2736 EXPANDED NIL)
+ + |--VERB (#2737 LEXIFIED "ask" NIL)
+ + |--NP (#2738 ACTIVE NIL)
+ + |--NP (#2739 TRACE "" NIL)
+--REL_S (#2701 ASSUMED (REL_S-2 S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
| |--S (#2702 EXPANDED NIL)
| |--NP (#2703 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
| |--NN (#2725 EXPANDED NIL)
| |--PRON (#2726 LEXIFIED "I" NIL)
| |--VP (#2704 EXPANDED NIL)
| |--ADVP (#2705 EXPANDED (ADVP-JUST!ADVP-TEMPLATE))
| |--ADV (#2724 LEXIFIED "just" NIL)
| |--VP (#2706 EXPANDED NIL)
| |--AUX (#2707 LEXIFIED "would" NIL)
| |--VERB (#2708 LEXIFIED "like" NIL)
| |--NP (#2709 ACTIVE NIL)
| |--TOINF (#2710 ACTIVE NIL)
| +--TOINF (#2711 ASSUMED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
| + |--TO (#2712 LEXIFIED "to" NIL)
| + |--VP (#2713 EXPANDED NIL)
| + |--VERB (#2714 LEXIFIED "ask" NIL)
| + |--NP (#2715 ACTIVE NIL)
| + |--PREP (#2716 LEXIFIED "about" NIL)
| + |--NP (#2717 TRACE "" NIL)
| +--TOINF (#2718 ASSUMED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
| + |--TO (#2719 LEXIFIED "to" NIL)
| + |--VP (#2720 EXPANDED NIL)
| + |--VERB (#2721 LEXIFIED "ask" NIL)
| + |--NP (#2722 ACTIVE NIL)
| + |--NP (#2723 TRACE "" NIL)
--SIGN (#2550 LEXIFIED "." NIL)

```

----- Expansion was Terminated -----

----- New Expansion Started -----
 **** Control schema No.1 has been applied ****
 **** fs on controlled node

```

((SYN ((CAT NP)
  (CASE ACC)
  (AGR [])
  (LEX ""))))
(CONTROL ((SEM !X2((LABEL *SPEAKER*))))))
(SEM !X2)
(DEFSEM !X2)
(SLASH ((SYN ((CAT NP)
  (AGR ((PERSON 3)
    (NUMBER SING))))
  (PREP-INFO [])
  (CASE ACC)
  (LEX ""))))
(SEM ((PARM !X1())
  (RESTR ((RELN SOMETHING-PRON-1)
    (ENTITY !X1)))
  (INDEX ((DEFINT INDEF)
    (GENDER NONE)
    (NUMBER SING)
    (PERSON THIRD))))))
S-TOP (#2788 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
|--S (#2789 EXPANDED NIL)
| |--ADVP (#2790 EXPANDED (ADVP-WELL-INTERJ!ADVP-TEMPLATE))
| |--ADV (#2791 LEXIFIED "well" NIL)
| |--SIGN (#2792 LEXIFIED "," NIL)
| |--S (#2793 EXPANDED NIL)
| |--NP (#2794 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
| |--NN (#2795 EXPANDED NIL)
| |--PRON (#2796 LEXIFIED "I" NIL)
| |--VP (#2797 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))

```

```

|--VERB [#2798 LEXIFIED "have" NIL]
|--NP [#2799 EXPANDED (NP-NP-REL)]
|--NP [#2800 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING)]
|--NN [#2801 EXPANDED NIL]
|--PRON [#2802 LEXIFIED "something" NIL]
--REL_S [#2803 EXPANDED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|--CONJ [#2804 LEXIFIED "that" NIL]
--S [#2805 EXPANDED NIL]
|--NP [#2806 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|--NN [#2807 EXPANDED NIL]
|--PRON [#2808 LEXIFIED "I" NIL]
--VP [#2809 EXPANDED NIL]
|--ADVP [#2810 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|--ADV [#2811 LEXIFIED "just" NIL]
--VP [#2812 EXPANDED NIL]
|--AUX [#2813 LEXIFIED "would" NIL]
--VERB [#2814 LEXIFIED "like" NIL]
--NP [#2815 PRO "" NIL]
--TOINF [#2816 EXPANDED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
--TO [#2817 LEXIFIED "to" NIL]
--VP [#2818 EXPANDED NIL]
--VERB [#2819 LEXIFIED "ask" NIL]
--NP [#2820 ACTIVE NIL]
--NP [#2821 TRACE "" NIL]
--SIGN [#2822 LEXIFIED "." NIL]
----- Expansion was Terminated -----

```

**** actualize defsem [#2820 ACTIVE NIL] ****

----- New Expansion Started -----

S-TOP [#2788 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]

```

--S [#2789 EXPANDED NIL]
|--ADVP [#2790 EXPANDED (ADVP-WELL-INTERJ!ADVP-TEMPLATE)]
|--ADV [#2791 LEXIFIED "well" NIL]
--SIGN [#2792 LEXIFIED "," NIL]
--S [#2793 EXPANDED NIL]
|--NP [#2794 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|--NN [#2795 EXPANDED NIL]
|--PRON [#2796 LEXIFIED "I" NIL]
--VP [#2797 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
--VERB [#2798 LEXIFIED "have" NIL]
--NP [#2799 EXPANDED (NP-NP-REL)]
--NP [#2800 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING)]
--NN [#2801 EXPANDED NIL]
--PRON [#2802 LEXIFIED "something" NIL]
--REL_S [#2803 EXPANDED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
--CONJ [#2804 LEXIFIED "that" NIL]
--S [#2805 EXPANDED NIL]
|--NP [#2806 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|--NN [#2807 EXPANDED NIL]
|--PRON [#2808 LEXIFIED "I" NIL]
--VP [#2809 EXPANDED NIL]
|--ADVP [#2810 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|--ADV [#2811 LEXIFIED "just" NIL]
--VP [#2812 EXPANDED NIL]
|--AUX [#2813 LEXIFIED "would" NIL]
--VERB [#2814 LEXIFIED "like" NIL]
--NP [#2815 PRO "" NIL]
--TOINF [#2816 EXPANDED (TOINF VP-ASK-1-NP-NP!VP-DN.N)]
--TO [#2817 LEXIFIED "to" NIL]
--VP [#2818 EXPANDED NIL]
--VERB [#2819 LEXIFIED "ask" NIL]
--NP [#2820 ACTIVE NIL]
--NP [#2821 TRACE "" NIL]
--SIGN [#2822 LEXIFIED "." NIL]

```

**** expanding NP [#2820 ACTIVE NIL] ****

[(SEM !X2[(LABEL *DEF-HEARER*)])]

(DEFSEM !X2)

(SYN [(CAT NP)

(CASE ACC)])

(SLASH [(SYN [(CAT NP)

(AGR [(PERSON 3)

(NUMBER SING)])

(PREP-INFO [])

(CASE ACC)

(LEX ""))]

(SEM [(PARM !X1())

(RESTR [(RELN SOMETHING-PRON-1)

(ENTITY !X1)])

(INDEX [(DEFINT INDEF)

(GENDER NONE)

(NUMBER SING)

(PERSON THIRD)])))]

**** activated (PP <PRON-UNKNOWN-DEFAULT-HEARER> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP

VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****

**** Applying (NP-NN NN-PRON PRON-UNKNOWN-DEFAULT-HEARER) ****

S-TOP [#2788 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]

```

--S [#2789 EXPANDED NIL]
|--ADVP [#2790 EXPANDED (ADVP-WELL-INTERJ!ADVP-TEMPLATE)]
|--ADV [#2791 LEXIFIED "well" NIL]
--SIGN [#2792 LEXIFIED "," NIL]
--S [#2793 EXPANDED NIL]
|--NP [#2794 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|--NN [#2795 EXPANDED NIL]

```

```

|--PRON (#2796 LEXIFIED "I" NIL)
|--VP (#2797 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|--VERB (#2798 LEXIFIED "have" NIL)
|--NP (#2799 EXPANDED (NP-NP-REL))
|--NP (#2800 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING))
|--NN (#2801 EXPANDED NIL)
|--PRON (#2802 LEXIFIED "something" NIL)
|--REL_S (#2803 EXPANDED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
|--CONJ (#2804 LEXIFIED "that" NIL)
|--S (#2805 EXPANDED NIL)
|--NP (#2806 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|--NN (#2807 EXPANDED NIL)
|--PRON (#2808 LEXIFIED "I" NIL)
|--VP (#2809 EXPANDED NIL)
|--ADV (#2810 EXPANDED (ADV-JUST!ADV-TEMPLATE))
|--ADV (#2811 LEXIFIED "just" NIL)
|--VP (#2812 EXPANDED NIL)
|--AUX (#2813 LEXIFIED "would" NIL)
|--VERB (#2814 LEXIFIED "like" NIL)
|--NP (#2815 PRO "" NIL)
|--TOINF (#2816 EXPANDED (TOINF VP-ASK-1-NP-NP!VP-DN.N))
|--TO (#2817 LEXIFIED "to" NIL)
|--VP (#2818 EXPANDED NIL)
|--VERB (#2819 LEXIFIED "ask" NIL)
|--NP (#2820 EXPANDED (NP-NN NN-PRON PRON-UNKNOWN-DEFAULT-HEARER))
|--NN (#2823 EXPANDED NIL)
|--PRON (#2824 LEXIFIED "<you>" NIL)
|--NP (#2821 TRACE "" NIL)
--SIGN (#2822 LEXIFIED "." NIL)
----- Expansion was Terminated -----

```

----- New Expansion Started -----

**** Control schema No.1 has been applied ****

**** fs on controlled node

```

[[SYN [(CAT NP)
  (CASE ACC)
  (AGR [])
  (LEX "" )]]]
(CONTROL [(SEM !X2[(LABEL *SPEAKER*)]]])
(SEM !X2)
(DEFSEM !X2)
(SLASH [(SYN [(CAT NP)
  (AGR [(PERSON 3)
    (NUMBER SING)])
  (PREP-INFO [])
  (CASE ACC)
  (LEX "" )]]]
  (SEM [(PARM !X1[])
    (RESTR [(RELN SOMETHING-PRON-1)
      (ENTITY !X1)])
    (INDEX [(DEFINT INDEF)
      (GENDER NONE)
      (NUMBER SING)
      (PERSON THIRD)]))])])
S-TOP (#2752 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX))
--S (#2753 EXPANDED NIL)
|--ADV (#2754 EXPANDED (ADV-WELL-INTERJ!ADV-TEMPLATE))
|--ADV (#2755 LEXIFIED "well" NIL)
|--SIGN (#2756 LEXIFIED "," NIL)
--S (#2757 EXPANDED NIL)
|--NP (#2758 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|--NN (#2759 EXPANDED NIL)
|--PRON (#2760 LEXIFIED "I" NIL)
|--VP (#2761 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN))
|--VERB (#2762 LEXIFIED "have" NIL)
|--NP (#2763 EXPANDED (NP-NP-REL))
|--NP (#2764 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING))
|--NN (#2765 EXPANDED NIL)
|--PRON (#2766 LEXIFIED "something" NIL)
--REL_S (#2767 EXPANDED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF))
--CONJ (#2768 LEXIFIED "that" NIL)
--S (#2769 EXPANDED NIL)
|--NP (#2770 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I))
|--NN (#2771 EXPANDED NIL)
|--PRON (#2772 LEXIFIED "I" NIL)
|--VP (#2773 EXPANDED NIL)
|--ADV (#2774 EXPANDED (ADV-JUST!ADV-TEMPLATE))
|--ADV (#2775 LEXIFIED "just" NIL)
|--VP (#2776 EXPANDED NIL)
|--AUX (#2777 LEXIFIED "would" NIL)
|--VERB (#2778 LEXIFIED "like" NIL)
|--NP (#2779 PRO "" NIL)
|--TOINF (#2780 EXPANDED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR))
|--TO (#2781 LEXIFIED "to" NIL)
|--VP (#2782 EXPANDED NIL)
|--VERB (#2783 LEXIFIED "ask" NIL)
|--NP (#2784 ACTIVE NIL)
--PREP (#2785 LEXIFIED "about" NIL)
--NP (#2786 TRACE "" NIL)
--SIGN (#2787 LEXIFIED "." NIL)
----- Expansion was Terminated -----

```

**** actualize defsem (#2784 ACTIVE NIL) ****

----- New Expansion Started -----

```

S-TOP [#2752 EXPANDED (S-TOP S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX)]
--S [#2753 EXPANDED NIL]
|---ADVP [#2754 EXPANDED (ADVP-WELL-INTERJ!ADVP-TEMPLATE)]
|   |---ADV [#2755 LEXIFIED "well" NIL]
|   |---SIGN [#2756 LEXIFIED "," NIL]
|   --S [#2757 EXPANDED NIL]
|       |---NP [#2758 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|       |   |---NN [#2759 EXPANDED NIL]
|       |   |---PRON [#2760 LEXIFIED "I" NIL]
|       |---VP [#2761 EXPANDED (VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-HAVE!VP-TN)]
|       |   |---VERB [#2762 LEXIFIED "have" NIL]
|       |   |---NP [#2763 EXPANDED (NP-NP-REL)]
|       |       |---NP [#2764 EXPANDED (NP-NN NN-INDEX-COVER NN-PRON PRON-SOMETHING)]
|       |       |   |---NN [#2765 EXPANDED NIL]
|       |       |   |---PRON [#2766 LEXIFIED "something" NIL]
|       |       |---REL_S [#2767 EXPANDED (REL_S S-NP-VP-SUBORD VP-SIMPLE-PRESENT-PAST VP-ASPT-COVER VP-DEGR-FRONT VP-WOULD-LIKE-TOINF)]
|       |       |   |---CONJ [#2768 LEXIFIED "that" NIL]
|       |       |   --S [#2769 EXPANDED NIL]
|       |           |---NP [#2770 EXPANDED (NP-NN NN-PRON PRON-SPEAKER-I)]
|       |           |   |---NN [#2771 EXPANDED NIL]
|       |           |   |---PRON [#2772 LEXIFIED "I" NIL]
|       |           |---VP [#2773 EXPANDED NIL]
|       |               |---ADVP [#2774 EXPANDED (ADVP-JUST!ADVP-TEMPLATE)]
|       |               |   |---ADV [#2775 LEXIFIED "just" NIL]
|       |               |   |---VP [#2776 EXPANDED NIL]
|       |               |       |---AUX [#2777 LEXIFIED "would" NIL]
|       |               |       |---VERB [#2778 LEXIFIED "like" NIL]
|       |               |       |---NP [#2779 PRO "" NIL]
|       |               |---TOINF [#2780 EXPANDED (TOINF VP-ASK-1-NP-ABOUT!VP-TN.PR)]
|       |               |   |---TO [#2781 LEXIFIED "to" NIL]
|       |               |   |---VP [#2782 EXPANDED NIL]
|       |               |       |---VERB [#2783 LEXIFIED "ask" NIL]
|       |               |       |---NP [#2784 ACTIVE NIL]
|       |               |       |---PREP [#2785 LEXIFIED "about" NIL]
|       |               |       |---NP [#2786 TRACE "" NIL]
|       |---SIGN [#2787 LEXIFIED "." NIL]
**** expanding NP [#2784 ACTIVE NIL] ****
((SYN [(ARTI (:NOT -))
      (CAT NP)
      (CASE ACC)])
 (SEM !X2[(LABEL *DEF-HEARER*)])
 (DEFSEM !X2)
 (SLASH [(SYN [(CAT NP)
               (AGR [(PERSON 3)
                     (NUMBER SING)])
               (PREP-INFO [])
               (CASE ACC)
               (LEX ""))]
      (SEM [(PARM !X1())
            (RESTR [(RELN SOMETHING-PRON-1)
                    (ENTITY !X1)])
            (INDEX [(DEFINT INDEF)
                    (GENDER NONE)
                    (NUMBER SING)
                    (PERSON THIRD)]))]))))
**** activated (PP <PRON-UNKNOWN-DEFAULT-HEARER> S-MODI-TOO S-NP-VP-SUBORD CONJ_S-TILL CONJ_S-THAT-1 VP-MAKE-NP VP-GIVE-ME-NP
VP-RESERVE-NP-FOR-ME VP-BE-AP VP-BE-AP-INV TOINF NP-NN NP-A-NN NP-THE-NN NN-N NN-PRON AP-ADJ) ****
#### Whole tree failed in expansion ####

```

----- Expansion was Terminated -----

-(31)-----
well , I have something that I just would like <PRO> to ask <you> <trace> .

**** (ADVP-JUST!ADVP-TEMPLATE ADVP-WELL-INTERJ!ADVP-TEMPLATE NN-INDEX-COVER NN-PRON NN-PRON NN-PRON NN-PRON NP-NN NP-NN NP-NN NP-NN
NP-NP-REL PRON-SOMETHING PRON-SPEAKER-I PRON-SPEAKER-I PRON-UNKNOWN-DEFAULT-HEARER REL_S S-COVER-MANNER S-MODI-CONJ S-NP-VP-MATRIX
S-NP-VP-SUBORD S-TOP TOINF VP-ASK-1-NP-NP!VP-DN.N VP-ASPT-COVER VP-ASPT-COVER VP-DEGR-FRONT VP-HAVE!VP-TN VP-SIMPLE-PRESENT-PAST
VP-SIMPLE-PRESENT-PAST VP-WOULD-LIKE-TOINF) were applied ****

WARNING: yyy for node (" NP) in MG_NET-MAIN
WARNING: yyy for node ("<you>" PRON) in MG_NET-PRON
WARNING: yyy for node (" NP) in MG_NET-MAIN

=====
"well, I have something that I just would like to ask <you>."
=====