Gefördert vom
bmb+f
Bundesministerium für
Bildung, Wissenschaft,
Forschung und Technologie

Verbmobil
Verbundvorhaben

Definition of the Abstract Semantic Classes for the Verbmobil Forschungsprototyp 1.0

Johan Bos
Markus Egg
Michael Schiehlen

Universität des Saarlandes, IBM Heidelberg,
Universität Stuttgart

Vm
Report 165
August, 1996

August, 1996

Johan Bos
Markus Egg
Michael Schiehlen

Computerlinguistik
Bau 17.2
Universität des Saarlandes
66041 Saarbrücken

Tel.: (0681) 302 - 4344
Fax: (0681) 302 - 4351

Gehört zum Antragsabschnitt: 8.1

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 R gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

Contents

1 Introduction

This report is an extended, updated version of Verbmobil memo 101. It describes the syntax of the abstract semantic classes used in the Verbmobil Interface Term (VIT). The VIT is a common interface format for the different modules of the VERBMOBIL translation system. It represents an underspecified semantic structure of a natural language utterance, along with it prosodic, syntactic, and discourse information.

We thank Bernd Abb, Bianka Buschbeck-Wolf, Mariana Damova, Michael Dorna, Martin Emele, Björn Gambäck, Johannes Heinecke, Michael Herweg, Uwe Küssner, Kai Lebeth, Christian Lieske, Yoshiki Mori, Manfred Pinkal, Joachim Quantz, Sabine Reinhard, Bärbel Ripplinger, CJ Rupp, Uta Schwertel, and Karsten Worm for comments on earlier versions of this document.

2 General Issues

Notational Conventions

R: Relation (Predname); I: Index; L: Label; G: Label of grouping; D: Description (designator); H: Hole (underspecified scope); T: Hole ("top"-hole); M: Label ("main"-label); arg123 means either arg1, arg2 or arg3. `add_to_group(L,G)` means that label `L` should be placed into group `G` (used by disambiguation).

Labels

A predicate's label is represented as its first argument. All predicates are uniquely labeled, except for the predicates `arg1`, `arg2`, `arg3`, `n_arg`, `s_arg`, and `o_arg`, which have the same labels as the predicate they correspond to, and the labels of `comp`, `superl`, and `nom`, which are identical to the predicate they are modifying.

Grouping

Groups of conjunctively connected relations, that enter scope relation as one single unit (as they occur e.g. as the result of intersective modification), are assigned a common group label. A grouping is represented as `sem_group(l1,[l2,l3,...,])` in the `Groupings` slot. Groupings have at least one member and can contain other groupings. Classes that never enter a grouping are: qua, ppro, dem, whq, whp, fadv, padv, comp, gr, and int (and also mood operators). Classes that are always represented in a grouping are cn, nom, pn, cop, iv, tv, dv, pcv, ipcv, dpcv, mv, prpa, part, itadv, dra, and scadv. The remaining classes are in groupings when they act as modifiers.

Free Labels

Some labels that occur as arguments of a predicate are possibly free. That is, they do not refer to a predicate. In this case there is an entry of the form `unbound(L)` for a label `L` in the `Scope` slot of the VIT.

Comparatives

Comparative constructions are represented as `[comp(L,I1,I2,I3)]`. `L` and `I1` are the label and the instance of the adjective/adverb the comparative pertains to. `I2` represents the (set of) individuals with respect to which `I1` is being compared. This (set of) individuals can be expressed by a *als*-PP, e.g., in *schöner als die meisten Linguisten*. `I3` represents the amount to which `I1` possesses a property more than (the members of) `I2` like in *drei Wochen später als die anderen*. `I2` and `I3` might be unbound.

Superlatives

Superlative constructions are represented as `[superl(L,I1,I2)]`. `L` and `I1` are the label and the instance of the adjective/adverb the superlative pertains to. `I2` represents the (set of) individuals with respect to which `I1` is being compared. This (set of) individuals can be expressed by a *von*-PP or a genitive DP, e.g., in *der schönste der Linguisten*. `I2` might be unbound.

Free Datives

Syntactically free dative expressions (e.g. "Das passt mir gut") introduce a predicate `perspective(L,I1,I2)`.

Mood

Sentence mood gets the representation `[R(L,H)/whq(L,L1,H)]`. `H` is the underspecified scope, `L` is the main label. The mood operator is introduced in the lexicon in the case of `whq`, which is part of the semantics of every *wh*-word. The introduction of other mood operators is triggered by syntactic sentence structure. `R` is `whq`, `decl`, `ynq`, `imp`, `ynq_decl` (triggered by the Japanese particle "ne") or `ynq_imp` (e.g. in German "treffen wir uns").

Nominalisations

Nominalisations of adjectives introduce the predicate `abstr_nom(L,I)`, where `L` is in the same grouping as the adjective is. Nominalisations of events introduce `nom(L,I)`, where `L` is the label of the event relation, and `I` its instance.

Relation nouns

In constructions like "die Hälfte des Jahres" the genetive DP introduces an `n_arg` relation. This used to be only allowed for a subset of common nouns, but this has been extended to all. The class relation_noun therefore lost its existence.

Gradual Prepositions

The treatment of the prepositions in phrases like "von vor drei Tagen" or "bis nach Ostern" results in a unbound instance `I1`, where the representation shows a structure like `[prep1(L,I,I1),prep2(L1,I1,I2)]`, where `L` and `L1` are members of the same groupings.

Spelling

Spellings are represented as `[spell(L,I,[D1,D2,....,Dn])]`, where `I` is the instance of the verb that is modified, and `Dx` is $\in$ {a, b, c, ..., y, z, sz, ae, oe, ue, prime, dash, hyphen}.

The `VIT`

The VIT (Verbmobil Interface Term) is a ten place term. It consists of the following parts:

```
vit(UtteranceID, Semantics,
    MainCondition, Sorts,
    Discourse, Syntax,
    TenseAndAspect, Scope,
    Prosody, Groupings).
```

The `Semantics` slot is a list of labeled predicates.

The `MainCondition` is a label that points to the label that covers the whole representation. For complete utterances it points to a predicate representing the mood of the utterance.

`Sorts` is a list of entries of the form `s_sort(I,Sort)` or `s_concept(I,Concept)`. The values of `Sort` are taken from the SemDB (Heinecke and Worm 1996).

`Discourse` is a list with entries of the form `prontype(I,D1,D2)`, `demontype(I,D)`, and `dialog_act(DialogAct)`. In the description for personal pronoun and demonstrative in the next section you will find more on this.

`Syntax` contains entries with syntactic information associated to the information in `Semantics`.

Information in `TenseAndAspect` is represented as `ta_aspect(Inst,Aspect)`, `ta_mood(Inst,Mood)` and `ta_tense(Inst,Tense)`.

`Scope` gives information on how the labeled predicates in `Semantics` fit together. There are entries of the form:

- `leq(L,H)` (a label `L` is equal to hole `H` or indirectly in the scope of `H`);
- `eq(L,H)` (a label `L` is equal to hole `H`);
- `ccom_plug(H,L)` (a label `H` is "plugged" (identified) with label `L` based on immediate syntactic subordination information);
- `eval_plug(L,H)` (a plugging performed by semantic evaluation)
- `unbound(L)` can be found in here also.

The slot for prosodic information `Prosody` contains entries of the form:

- `pros_accent(L)` (predicate `L` has (probably) an accent;
- `pros_boundary(L)` (predicate `L` is (probably) followed by a short boundary (so-called b3-boundary);
- `pros_mood(L,D)` (prosodic mood is probably `D`, where `D` is either `decl`, `prog`, or `quest`).

Finally, `Groupings` is a slot to represent groupings, which have the format:

`sem_group(L,[L1,...Ln])`

The interested reader might consult Michael Dorna's "The ADT-Package for the Verbmobil Interface Term" (Dorna 1996).

3 Abstract Semantic Classes

In this section we discuss the abstract semantic classes (or maybe better: "syn-sem" classes), their format, use, and so on. For every class we present the class name, its *suffix* (an abbreviation of the class name, used as suffix to disambiguate predicate names), *part of speech*[1], the representation in VIT, *SemDB* (the arguments that are to be obtained from SemDB), and an example.

[1]The interested reader might consult URL `http://www2.ims.uni-stuttgart.de/vmcd/STTS.htmls` for a description of the tagset

Class Name:	**common_noun**
Suffix:	`cn`
Part of Speech:	NN
Representation:	`[R(L,I)]`
SemDB:	`predname R`
Comments:	`I` is the instance variable of the noun. `L` is member of a grouping.
Example:	der *Termin* ist gut

```
vit( segment_description('cn','der termin ist gut'),
     [termin(l6,i2),                   % Semantics
      support(l5,i1,l4),
      gut(l4,i2),
      decl(l3,h1),
      def(l8,i2,l2)],
     l3,                               % Main Label
     [s_sort(i2,time_sit_poly)],       % Sorts
     [],                               % Discourse
     [gend(i2,masc),                   % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                 % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                      % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l3,decl)],             % Prosody
     [sem_group(l2,[l6]),              % Groupings
      sem_group(l1,[l5])] ).
```

CLASS NAME: **nominalisation**
SUFFIX: nom
PART OF SPEECH: NN
REPRESENTATION: [nom(L,I),R(L,I)]
SEMDB: predname R
COMMENTS: I is the instance variable of the noun. L is member of a grouping. R is also used in a verb class.
EXAMPLE: das *Treffen* ist leider am Montag

```
vit( segment_description('nom','das treffen ist leider am montag'),
     [nom(l9,i3),                          % Semantics
      treffen(l9,i3),
      support(l8,i1,l5),
      leider(l7,h2),
      decl(l6,h1),
      an(l5,i3,i2),
      dofw(l4,i2,mon),
      def(l12,i3,l3),
      def(l11,i2,l1)],
     l6,                                   % Main Label
     [s_sort(i2,time),                     % Sorts
      s_sort(i3,meeting_sit)],
     [dir(l5,no)],                         % Discourse
     [gend(i3,neut),                       % Syntax
      num(i3,sg),
      pers(i3,3),
      cas(i3,nom)],
     [ta_mood(i1,ind),                     % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h2),                          % Scope
      leq(l2,h1),
      leq(l7,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l7)],
     [pros_mood(l6,decl)],                 % Prosody
     [sem_group(l3,[l9]),                  % Groupings
      sem_group(l2,[l8]),
      sem_group(l1,[l4])] ).
```

CLASS NAME:	**proper name**
SUFFIX:	`pn`
PART OF SPEECH:	**NE**
REPRESENTATION:	`[def(L1,I,G),named(L2,I,D),sem_group(G,[L2])]`
SEMDB:	`designator D`
COMMENTS:	`L2` is member of a grouping, this grouping is in the scope of a `def` operator. `I` is the instance variable of the proper name.
EXAMPLE:	*Holger* kommt nicht

```
vit( segment_description('pn','holger kommt nicht'),
     [named(l7,i3,holger),              % Semantics
      kommen(l5,i1),
      decl(l6,h1),
      arg3(l5,i1,i3),
      neg(l4,i2,h2),
      def(l10,i3,l3)],
     l6,                                % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i3,&(human,space_time))],
     [],                                % Discourse
     [gend(i3,masc),                    % Syntax
      num(i3,sg),
      pers(i3,3),
      cas(i3,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h2),                       % Scope
      leq(l2,h1),
      leq(l1,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l1)],
     [pros_mood(l6,decl)],              % Prosody
     [sem_group(l3,[l7]),               % Groupings
      sem_group(l2,[l5]),
      sem_group(l1,[l4])] ).
```

Class Name:	**dofw**
Suffix:	`dofw`
Part of Speech:	NN
Representation:	`[dofw(L,I,D)]`
SemDB:	`designator D`
Comments:	Day of week. `I` is the instance variable of the noun. `L` can be member of a grouping. `D` $\in$ { sun, mon, tue, wed, thu, fri, sat }.
Example:	*Montag* ist okay bei mir

```
vit( segment_description('dofw','montag ist okay bei mir'),
     [support(l7,i2,l6),                % Semantics
      okay(l6,i3),
      decl(l5,h1),
      dofw(l4,i3,mon),
      bei(l3,i2,i1),
      def(l10,i3,l2),
      pron(l8,i1)],
     l5,                                % Main Label
     [s_sort(i1,human),                 % Sorts
      s_sort(i3,time)],
     [dir(l3,no),                       % Discourse
      prontype(i1,sp,std)],
     [num(i1,sg),                       % Syntax
      pers(i1,1),
      cas(i1,dat)],
     [ta_mood(i2,ind),                  % Tense and Aspect
      ta_tense(i2,pres)],
     [leq(l1,h1),                       % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l1,[l3,l7]),            % Groupings
      sem_group(l2,[l4])] ).
```

Class Name:	**mofy**
Suffix:	`mofy`
Part of Speech:	NN
Representation:	`[mofy(L,I,D)]`
SemDB:	`designator D`
Comments:	Month of year. `I` is the instance variable of the noun. `L` can be member of a grouping. `D` $\in$ { jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec }.
Example:	der *Januar* ist echt gut

```
vit( segment_description('mofy','der januar ist echt gut'),
     [support(l7,i1,l6),                % Semantics
      echt_grad(l6,l5),
      gut(l5,i2),
      decl(l4,h1),
      mofy(l3,i2,jan),
      def(l9,i2,l2)],
     l4,                                % Main Label
     [s_sort(i2,time)],                 % Sorts
     [],                                % Discourse
     [],                                % Syntax
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                       % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l4,decl)],              % Prosody
     [sem_group(l2,[l3]),               % Groupings
      sem_group(l1,[l7])] ).
```

Class Name: **yofc**
Suffix: `yofc`
Part of Speech: **syntactic**
Representation: `[yofc(L,I,D)]`
SemDB: `CARD`
Comments: Year of century. `I` is the instance variable of the noun. `L` can be in a grouping.
`D` $\in$ { 0,...,99 }.
Example: sechzehnter März *neunzehnhundert dreiundneunzig*

```
vit( segment_description('yofc','sechzehnter maerz neunzehnhundert dreiundneunzig'),
     [mofy(l8,i3,mar),                  % Semantics
      yofc(l7,i1,93),
      tloc(l6,i2,i1),
      ord(l4,i2,i3,16),
      def(l10,i2,l2),
      def(l11,i3,l3),
      def(l9,i1,l1)],
     l10,                               % Main Label
     [s_sort(i1,time),                  % Sorts
      s_sort(i2,time),
      s_sort(i3,time)],
     [dir(l6,no)],                      % Discourse
     [],                                % Syntax
     [],                                % Tense and Aspect
     [],                                % Scope
     [pros_mood(l10,decl)],             % Prosody
     [sem_group(l2,[l6,l4]),            % Groupings
      sem_group(l3,[l8]),
      sem_group(l1,[l7])] ).
```

CLASS NAME:	**clocktime**
SUFFIX:	ctime
PART OF SPEECH:	**syntactic**
REPRESENTATION:	[clocktime(L,I,D1,D2)]
SEMDB:	-
COMMENTS:	I is the instance variable of the noun. L can be in a grouping. D1 $\in$ { u,0,...,23 }, D2 $\in$ { u,0,...,59 }. The value u is used in WH-questions where is asked for a certain clocktime, or partly specified clocktime expressions (e.g. in German "um halb").
EXAMPLE:	um *fuenf vor zwoelf*

```
vit( segment_description('ctime','um fuenf vor zwoelf'),
     [um(l4,i1,i2),                         % Semantics
      clocktime(l2,i2,11,55),
      def(l5,i2,l1)],
     l6,                                    % Main Label
     [s_sort(i2,time)],                     % Sorts
     [dir(l4,no)],                          % Discourse
     [],                                    % Syntax
     [],                                    % Tense and Aspect
     [],                                    % Scope
     [pros_mood(l6,decl)],                  % Prosody
     [sem_group(l6,[l4]),                   % Groupings
      sem_group(l1,[l2])] ).
```

CLASS NAME:	**title**
SUFFIX:	`ttl`
PART OF SPEECH:	NN
REPRESENTATION:	`[title(L,I,D)]`
SEMDB:	`designator D`
COMMENTS:	`I` refers to the person addressed. `L` can be in a grouping.
EXAMPLE:	*Doktor* Brown ist ja im Urlaub

```
vit( segment_description('ttl','doktor brown ist ja im urlaub'),
     [title(l10,i3,doktor),            % Semantics
      named(l9,i3,brown),
      support(l8,i1,l4),
      ja(l7,h2),
      urlaub(l6,i2),
      decl(l5,h1),
      in(l4,i3,i2),
      def(l13,i3,l3),
      def(l12,i2,l1)],
     l5,                               % Main Label
     [s_sort(i2,time_sit_poly),        % Sorts
      s_sort(i3,human)],
     [dir(l4,no)],                     % Discourse
     [num(i3,sg),                      % Syntax
      pers(i3,3),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,dat),
      cas(i3,nom)],
     [ta_mood(i1,ind),                 % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h2),                      % Scope
      leq(l2,h1),
      leq(l7,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l7)],
     [pros_mood(l5,decl)],             % Prosody
     [sem_group(l3,[l10,l9]),          % Groupings
      sem_group(l2,[l8]),
      sem_group(l1,[l6])] ).
```

CLASS NAME:	**quant**
SUFFIX:	qua
PART OF SPEECH:	**ART,PIAT,PIDAT,PIS**
REPRESENTATION:	`[R(L,I,G,H)]`
SEMDB:	`predname R`
COMMENTS:	`I` is the variable quantified over. `L` is not in a grouping. `L` is in a `leq` relation. `G` is the restriction and covers the material in the N' constituent. If there is an empty N' constituent, or if the quantifier is syntactically a full NP (e.g., "jemand"), the predicate `abstr_nom` introduced. In every case, the main verb of the sentence is subordinated to the `H` argument of its quantified argument DPs.
EXAMPLE:	*Jedes* Problem hat eine Lösung

```
vit( segment_description('qua','jedes problem hat eine Loesung'),
     [problem(l9,i3),                      % Semantics
      haben(l6,i1),
      loesung(l8,i2),
      decl(l7,h2),
      arg2(l6,i1,i3),
      arg3(l6,i1,i2),
      jed(l5,i3,l3,h3),
      ein_card_qua(l4,i2,l13,l1,h1,1)],
     l7,                                   % Main Label
     [s_sort(i1,situation),                % Sorts
      s_sort(i2,info_content),
      s_sort(i3,abstract)],
     [],                                   % Discourse
     [gend(i3,neut),                       % Syntax
      num(i3,sg),
      pers(i3,3),
      gend(i2,fem),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i3,nom)],
     [ta_mood(i1,ind),                     % Tense and Aspect
      ta_tense(i1,pres)],
     [unbound(l13),                        % Scope
      leq(l5,h2),
      leq(l2,h3),
      leq(l2,h2),
      leq(l2,h1),
```

```
 leq(l4,h2),
 ccom_plug(h3,l4),
 ccom_plug(h2,l5),
 ccom_plug(h1,l2)],
[pros_mood(l7,decl)],                % Prosody
[sem_group(l3,[l9]),                 % Groupings
 sem_group(l2,[l6]),
 sem_group(l1,[l8])] ).
```

Class Name:	**personal_pronoun**
Suffix:	ppro
Part of Speech:	**PPER,PRF**
Representation:	`[pron(L,I)]`
SemDB:	`designator D1 designator D2`
Comments:	L is not in a grouping. Additional information in the `Discourse` slot is: `[prontype(I,D1,D2)]`. Definite article that function as NPs (in e.g. "der ist okay") also belong to this class.

D1 (reference)	explanation
`sp`	speaker
`he`	hearer
`sp_he`	speaker and hearer
`third`	third person
`top`	anything

D2 (sort)	explanation
`refl`	reflexive pronoun
`std`	standard pronoun
`refl_std`	`refl` or `std`
`recip`	reciprocal ("einander")
`imp`	impersonal pronoun ("man")
`event`	event anaphor ("das", "es")
`event_std`	`event` or `std` ("es")
`demon`	demonstrative pronoun ("der")
`demon_event`	`demon` or `event`
`zero`	topic drop (German), zero pronoun (Japanese)
`intersent`	intersentential referring pronoun (Japanese)

Example: *Ich* komme auch

```
vit( segment_description('ppro','ich komme auch'),
     [kommen(l4,i1),                    % Semantics
      decl(l5,h1),
      arg3(l4,i1,i2),
      auch(l2,l3,h2),
      pron(l7,i2)],
     l5,                                % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i2,&(human,space_time))],
     [prontype(i2,sp,std)],             % Discourse
     [num(i2,sg),                       % Syntax
      pers(i2,1),
```

```
 cas(i2,nom)],
[ta_mood(i1,ind),                  % Tense and Aspect
 ta_tense(i1,pres)],
[unbound(l3),                      % Scope
 leq(l1,h2),
 leq(l1,h1),
 leq(l2,h1),
 ccom_plug(h2,l1),
 ccom_plug(h1,l2)],
[pros_mood(l5,decl)],              % Prosody
[sem_group(l1,[l4])] ).            % Groupings
```

CLASS NAME:	**definite**
SUFFIX:	`def`
PART OF SPEECH:	**PDS**
REPRESENTATION:	`[def(L,I,G)/udef(L,I,G)]`
SEMDB:	`ART`
COMMENTS:	`L` can be in a grouping. `G` is the restriction and covers the material of the N' constituent. `I` is the variable quantified over. `def` is introduced by the definite article, `udef` is introduced by determine free NPs.
EXAMPLE:	*Der Montag* ist gut

```
vit( segment_description('def','der montag ist gut'),
     [support(l6,i1,l5),                  % Semantics
      gut(l5,i2),
      decl(l4,h1),
      dofw(l3,i2,mon),
      def(l8,i2,l2)],
     l4,                                  % Main Label
     [s_sort(i2,time)],                   % Sorts
     [],                                  % Discourse
     [],                                  % Syntax
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                         % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l4,decl)],                % Prosody
     [sem_group(l2,[l3]),                 % Groupings
      sem_group(l1,[l6])] ).
```

CLASS NAME:	**possessive**
SUFFIX:	`poss`
PART OF SPEECH:	**PPOSS, PPOSAT**
REPRESENTATION:	`[def(L1,I1,G),poss(L2,I1,I2), pron(L3,I2), sem_group(G,[L2|X])]`
SEMDB:	`prontype D`
COMMENTS:	decomposition of "der N von X". If there is an empty N' constituent (for example German "meiner"), the relation `abstr_nom` is introduced.
EXAMPLE:	Das ist *mein Glas*

```
vit( segment_description('poss','das ist mein Glas'),
     [support(l7,i1,l5),              % Semantics
      glas(l6,i3),
      equal(l5,i3,i4),
      decl(l4,h1),
      poss(l3,i3,i2),
      pron(l11,i4),
      def(l10,i3,l1),
      pron(l9,i2)],
     l4,                              % Main Label
     [s_sort(i2,human),               % Sorts
      s_sort(i3,thing)],
     [dir(l3,no),                     % Discourse
      prontype(i4,third,demon),
      prontype(i2,sp,std)],
     [gend(i4,neut),                  % Syntax
      num(i4,sg),
      pers(i4,3),
      gend(i3,neut),
      num(i3,sg),
      pers(i3,3),
      cas(i3,nom),
      cas(i4,nom)],
     [ta_mood(i1,ind),                % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h1),                     % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l4,decl)],            % Prosody
     [sem_group(l1,[l3,l6]),          % Groupings
      sem_group(l2,[l7])] ).
```

Class Name:	**demonstrative**
Suffix:	`dem`
Part of Speech:	**PDS, PDAT**
Representation:	`[demonstrative(L,I,G)]`
SemDB:	`demontype D`
Comments:	`L` is not in a grouping. `G` is the restriction and covers the material of the N' constituent. If there is an empty N' constituent, the relation `abstr_nom` is introduced. Additional information in the `Discourse` slot: `[demontype(I,D)]`, `D` is `near` (dies), `far` (jen), `ident` (derselb), `spec` (derjenig).
Example:	*Dieser Termin* ist ziemlich gut

```
vit( segment_description('dem','dieser termin ist ziemlich gut'),
     [termin(l7,i2),                    % Semantics
      support(l6,i1,l5),
      ziemlich(l5,l4),
      gut(l4,i2),
      decl(l3,h1),
      demonstrative(l9,i2,l1)],
     l3,                                % Main Label
     [s_sort(i2,time_sit_poly)],        % Sorts
     [demontype(i2,near)],              % Discourse
     [gend(i2,masc),                    % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h1),                       % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l3,decl)],              % Prosody
     [sem_group(l2,[l6]),               % Groupings
      sem_group(l1,[l7])] ).
```

Class Name: **wh_question**

Suffix: `whq`

Part of Speech: **PWS,PWAT,PWAV**

Representation: `[whq(L,G,H),....]`

SemDB: -

Comments: `L` is the main label of the VIT. `L1` is member of grouping `G`.
wann: `[...,tloc(L2,I2,I1),time(L1,I1)]`
wo: `[...,sloc(L2,I2,I1),place(L1,I1)]`
warum: `[...,cause(L2,I2,I1),reason(L1,I1)]`
weshalb: `[...,cause(L2,I2,I1),reason(L1,I1)]`
wieso: `[...,cause(L2,I2,I1),reason(L1,I1)]`
weswegen: `[...,cause(L2,I2,I1),reason(L1,I1)]`
wodurch: `[...,cause(L2,I2,I1),reason(L1,I1)]`
wie: `[...,abstr_rel(L1,I)]`
wer: `[...,person(L1,I)]`
was: `[...,thing(L1,I)]`
woher: `[...,source(L2,I2,I1),place(L1,I1)]`
wohin: `[...,goal(L2,I2,I1),place(L1,I1)]`
welch: `[...,R(L1,I)]`
wieviel: `[...,viel_sadx(L2,I1),abstr_grad(L1,L2)]`
um wieviel Uhr: `[...,um(L2,I2,I1),clocktime(L1,I1,u,u)]`
I2 corefers with the instance of the modified verb. (see also: *mood*)

Example: *Wann* machen wir dann einen Termin aus?

```
vit( segment_description('whq','wann machen wir dann einen termin aus'),
     [time(l9,i5),                     % Semantics
      whq(l8,l9,h3),
      ausmachen(l5,i2),
      termin(l7,i1),
      tloc(l6,i2,i5),
      arg1(l5,i2,i4),
      arg3(l5,i2,i1),
      ein_card_qua(l4,i1,l16,l1,h1,1),
      dann_laprep_padv(l3,i2,h2,l17,i3,h3,l2,spec),
      pron(l14,i4)],
     l8,                               % Main Label
     [s_sort(i1,&(space_time,time_sit_poly)),% Sorts
      s_sort(i2,ment_communicat_poly),
      s_sort(i4,&(human,person))],
     [dir(l6,no),                      % Discourse
      dir(l3,no),
      prontype(i4,sp_he,std)],
```

```
[num(i4,pl),                          % Syntax
 pers(i4,1),
 gend(i1,masc),
 num(i1,sg),
 pers(i1,3),
 cas(i1,acc),
 cas(i4,nom)],
[ta_mood(i2,ind),                     % Tense and Aspect
 ta_tense(i2,pres)],
[unbound(l17),                        % Scope
 unbound(l16),
 leq(l2,h3),
 leq(l2,h1),
 leq(l4,h3),
 ccom_plug(h3,l3),
 ccom_plug(h2,l4),
 ccom_plug(h1,l2)],
[pros_mood(l8,quest)],                % Prosody
[sem_group(l2,[l6,l5]),               % Groupings
 sem_group(l1,[l7])] ).
```

CLASS NAME: **wh_prep**
SUFFIX: whp
PART OF SPEECH: **PWAV**
REPRESENTATION: [whq(L,G,H),....]
SEMDB: -
COMMENTS: L is the main label of the VIT. L1 is member of grouping G.
wofuer: [...,purpose(L2,I2,I1),thing(L1,I1)]
wozu: [...,purpose(L2,I2,I1),thing(L1,I1)]
womit: [...,concomit(L2,I2,I1),thing(L1,I1)]
worueber: [...,topic(L2,I2,I1),thing(L1,I1)]
wovon: [...,topic(L2,I2,I1),thing(L1,I1)]
wogegen: [...,reject(L2,I2,I1),thing(L1,I1)]
wobei: [...,loc_derived(L2,I2,I1),thing(L1,I1)]
woran: [...,thing(L1,I1)]
worauf: [...,thing(L1,I1)]
worin: [...,thing(L1,I1)]

EXAMPLE: *Wobei* kann ich Ihnen helfen?

```
vit( segment_description('whp','wobei kann ich ihnen helfen'),
     [thing(l7,i5),                        % Semantics
      whq(l6,l7,h2),
      helfen(l3,i2),
      loc_derived(l5,i2,i5),
      koennen(l4,i1,h1),
      arg1(l3,i2,i4),
      arg3(l3,i2,i3),
      pron(l11,i4),
      pron(l10,i3)],
     l6,                                   % Main Label
     [s_sort(i1,mental_sit),               % Sorts
      s_sort(i2,action_sit),
      s_sort(i4,human)],
     [dir(l5,yes),                         % Discourse
      prontype(i4,sp,std),
      prontype(i3,he,std)],
     [num(i4,sg),                          % Syntax
      pers(i4,1),
      num(i3,pl),
      pers(i3,3),
      cas(i3,dat),
      cas(i4,nom)],
     [ta_tense(i2,infin),                  % Tense and Aspect
```

```
 ta_mood(i1,ind),
 ta_tense(i1,pres)],
[leq(l2,h2),                          % Scope
 leq(l2,h1),
 leq(l1,h2),
 ccom_plug(h2,l1),
 ccom_plug(h1,l2)],
[pros_mood(l6,quest)],                % Prosody
[sem_group(l2,[l5,l3]),               % Groupings
 sem_group(l1,[l4])] ).
```

Class Name:	**copula**
Suffix:	cop
Part of Speech:	**VA**
Representation:	[support(L,I,L1)]/[support(L,I,L1),equal(L1,I2,I3)]
SemDB:	
Comments:	L is member of a grouping. I is the eventuality instance (possibly unbound). L1 refers to equal in COP+NP case.
Example:	Das *ist* gut

```
vit( segment_description('cop','das ist gut'),
     [support(l4,i1,l3),              % Semantics
      gut(l3,i2),
      decl(l2,h1),
      pron(l6,i2)],
     l2,                              % Main Label
     [],                              % Sorts
     [prontype(i2,third,demon)],      % Discourse
     [gend(i2,neut),                  % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                     % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l2,decl)],            % Prosody
     [sem_group(l1,[l4])] ).          % Groupings
```

CLASS NAME: **intransitive_verb**
SUFFIX: `iv`
PART OF SPEECH: **VV**
REPRESENTATION: `[R(L,I),arg123(L,I,I1)]`
SEMDB: `predname R`
COMMENTS: `L` is member of a grouping. 'arg123' should be read as either 'arg1', 'arg2', or 'arg3'.
EXAMPLE: Ich *komme* nicht

```
vit( segment_description('iv','ich komme nicht'),
     [kommen(l4,i1),                    % Semantics
      decl(l5,h1),
      arg3(l4,i1,i3),
      neg(l3,i2,h2),
      pron(l8,i3)],
     l5,                                % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i3,&(human,space_time))],
     [prontype(i3,sp,std)],             % Discourse
     [num(i3,sg),                       % Syntax
      pers(i3,1),
      cas(i3,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h2),                       % Scope
      leq(l2,h1),
      leq(l1,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l1)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l2,[l4]),               % Groupings
      sem_group(l1,[l3])] ).
```

CLASS NAME: **transitive_verb**
SUFFIX: `tv`
PART OF SPEECH: **VV**
REPRESENTATION: `[R(L,I),arg123(L,I,I1),arg123(L,I,I2)]`
SEMDB: `predname R`
COMMENTS: `L` is member of a grouping. 'arg123' should be read as either 'arg1', 'arg2', or 'arg3'.
EXAMPLE: Wir *treffen* uns

```
vit( segment_description('tv','wir treffen uns'),
     [treffen(l2,i1),                   % Semantics
      decl(l3,h1),
      arg1(l2,i1,i3),
      arg3(l2,i1,i2),
      pron(l6,i3),
      pron(l5,i2)],
     l3,                                % Main Label
     [s_sort(i1,meeting_sit),           % Sorts
      s_sort(i2,&(human,person)),
      s_sort(i3,&(human,person))],
     [prontype(i3,sp_he,std),           % Discourse
      prontype(i2,sp_he,refl_std)],
     [num(i3,pl),                       % Syntax
      pers(i3,1),
      num(i2,pl),
      pers(i2,1),
      cas(i2,acc),
      cas(i3,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                       % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l3,decl)],              % Prosody
     [sem_group(l1,[l2])] ).            % Groupings
```

CLASS NAME:	**ditransitive_verb**
SUFFIX:	`dv`
PART OF SPEECH:	**VV**
REPRESENTATION:	`[R(L,I),arg1(L,I,I1),arg2(L,I,I2),arg3(L,I,I3)]`
SEMDB:	`predname R`
COMMENTS:	`L` is member of a grouping.
EXAMPLE:	Ich *schicke* dir ein Ticket

```
vit( segment_description('dv','ich schicke dir ein Ticket'),
     [schicken(l4,i1),                    % Semantics
      ticket(l6,i2),
      decl(l5,h1),
      arg1(l4,i1,i4),
      arg2(l4,i1,i3),
      arg3(l4,i1,i2),
      ein_card_qua(l3,i2,l11,l1,h2,1),
      pron(l10,i4),
      pron(l9,i3)],
     l5,                                  % Main Label
     [s_sort(i1,action_sit),              % Sorts
      s_sort(i2,thing),
      s_sort(i3,&(human,person)),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp,std),                % Discourse
      prontype(i3,he,std)],
     [num(i4,sg),                         % Syntax
      pers(i4,1),
      num(i3,sg),
      pers(i3,2),
      gend(i2,neut),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i3,dat),
      cas(i4,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [unbound(l11),                       % Scope
      leq(l2,h2),
      leq(l2,h1),
      leq(l3,h1),
      ccom_plug(h2,l2),
```

```
 ccom_plug(h1,l3)],
[pros_mood(l5,decl)],                % Prosody
[sem_group(l2,[l4]),                 % Groupings
 sem_group(l1,[l6])] ).
```

Class Name: **propositional_complement_verb**
Suffix: `pcv`
Part of Speech: **VV**
Representation: `[R(L,I1),arg12(L,I1,I2),arg3(L,I1,X)]`
SemDB: `predname R`
Comments: `L` is member of a grouping. Four cases are distinguished.

- `[arg3(L,I1,H)]` for a "dass" complement or a v2 clause. The main verb is in a `leq` relation to `H`;
- `[arg3(L,I1,L1)]` for a "ob" or whq-complement. The former is represented as `ob(L1,H)`;
- `[arg3(L,I1,I3)]` for noun phrases as argument. In this case `I3` is the instance of the noun phrase;
- `[arg3(L,I1,H),abstr_rel(L2,I4),sem_group(G,[L2]),leq(G,H)]` for adjuncts, i.e., the coercion case. Example: 'ich glaube am montag'.

Example: Ich *glaube* dass wir uns Montag treffen

```
vit( segment_description('pcv','ich glaube dass wir uns montag treffen'),
    [glauben(l7,i5),                    % Semantics
     treffen(l4,i1),
     decl(l8,h2),
     arg2(l7,i5,i6),
     arg3(l7,i5,h1),
     dofw(l6,i2,mon),
     tloc(l5,i1,i2),
     arg1(l4,i1,i4),
     arg3(l4,i1,i3),
     pron(l14,i6),
     pron(l12,i4),
     pron(l11,i3),
     def(l10,i2,l1)],
    l8,                                 % Main Label
    [s_sort(i1,meeting_sit),            % Sorts
     s_sort(i2,time),
     s_sort(i3,&(human,person)),
     s_sort(i4,&(human,person)),
     s_sort(i5,mental_sit),
     s_sort(i6,&(human,person))],
```

```
[dir(l5,no),                          % Discourse
 prontype(i6,sp,std),
 prontype(i4,sp_he,std),
 prontype(i3,sp_he,refl_std)],
[num(i6,sg),                          % Syntax
 pers(i6,1),
 num(i4,pl),
 pers(i4,1),
 num(i3,pl),
 pers(i3,1),
 cas(i3,acc),
 cas(i4,nom),
 cas(i6,nom)],
[ta_mood(i5,ind),                     % Tense and Aspect
 ta_tense(i5,pres),
 ta_mood(i1,ind),
 ta_tense(i1,pres)],
[leq(l3,h2),                          % Scope
 leq(l2,h1),
 ccom_plug(h2,l3),
 ccom_plug(h1,l2)],
[pros_mood(l8,decl)],                 % Prosody
[sem_group(l2,[l5,l4]),               % Groupings
 sem_group(l3,[l7]),
 sem_group(l1,[l6])] ).
```

CLASS NAME:	**ditransitive_prop_complement_verb**
SUFFIX:	`dpcv`
PART OF SPEECH:	**VV**
REPRESENTATION:	`[R(L,I1),arg1(L,I1,I2),arg2(L,I1,i3),arg3(L,I1,X)]`
SEMDB:	`predname R`
COMMENTS:	`L` is member of a grouping. For the different argument cases the reader is referred to the description of propositional_complement_verb.
EXAMPLE:	Ich *sage* Ihnen dass wir uns Montag treffen

```
vit( segment_description('dpcv','ich sage ihnen dass wir uns montag treffen'),
     [sagen(l7,i5),                     % Semantics
      treffen(l4,i1),
      decl(l8,h2),
      arg1(l7,i5,i6),
      arg2(l7,i5,i7),
      arg3(l7,i5,h1),
      dofw(l6,i2,mon),
      tloc(l5,i1,i2),
      arg1(l4,i1,i4),
      arg3(l4,i1,i3),
      pron(l15,i7),
      pron(l14,i6),
      pron(l12,i4),
      pron(l11,i3),
      def(l10,i2,l1)],
     l8,                                % Main Label
     [s_sort(i1,meeting_sit),           % Sorts
      s_sort(i2,time),
      s_sort(i3,&(human,person)),
      s_sort(i4,&(human,person)),
      s_sort(i5,communicat_sit),
      s_sort(i6,person),
      s_sort(i7,&(human,person))],
     [dir(l5,no),                       % Discourse
      prontype(i7,sp,std),
      prontype(i6,he,std),
      prontype(i4,sp_he,std),
      prontype(i3,sp_he,refl_std)],
     [num(i7,sg),                       % Syntax
      pers(i7,1),
      num(i6,pl),
      pers(i6,3),
```

```
 num(i4,pl),
 pers(i4,1),
 num(i3,pl),
 pers(i3,1),
 cas(i3,acc),
 cas(i4,nom),
 cas(i6,dat),
 cas(i7,nom)],
[ta_mood(i5,ind),                     % Tense and Aspect
 ta_tense(i5,pres),
 ta_mood(i1,ind),
 ta_tense(i1,pres)],
[leq(l3,h2),                          % Scope
 leq(l2,h1),
 ccom_plug(h2,l3),
 ccom_plug(h1,l2)],
[pros_mood(l8,decl)],                 % Prosody
[sem_group(l2,[l5,l4]),               % Groupings
 sem_group(l3,[l7]),
 sem_group(l1,[l6])] ).
```

CLASS NAME:	**modal_verb**
SUFFIX:	`mv`
PART OF SPEECH:	**VM**
REPRESENTATION:	`[R(L,I,H)]`
SEMDB:	`predname R`
COMMENTS:	`L` is a grouping. `I` is the eventuality of the verb, `H` is underspecified scope and covers the embedded verb phrase or `abstr_rel` in the case of single modals.
EXAMPLE:	Wir *koennen* es am Montag machen

```
vit( segment_description('mv','wir koennen es am montag machen'),
     [machen(l4,i2),                      % Semantics
      decl(l8,h2),
      koennen(l7,i1,h1),
      an(l6,i2,i3),
      dofw(l5,i3,mon),
      arg1(l4,i2,i5),
      arg3(l4,i2,i4),
      pron(l13,i5),
      pron(l12,i4),
      def(l11,i3,l2)],
     l8,                                  % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i2,situation),
      s_sort(i3,time),
      s_sort(i5,&(human,person))],
     [dir(l6,no),                         % Discourse
      prontype(i5,sp_he,std),
      prontype(i4,third,event_std)],
     [num(i5,pl),                         % Syntax
      pers(i5,1),
      gend(i4,neut),
      num(i4,sg),
      pers(i4,3),
      cas(i4,acc),
      cas(i5,nom)],
     [ta_tense(i2,infin),                 % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [leq(l3,h2),                         % Scope
      leq(l3,h1),
      leq(l1,h2),
```

```
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],              % Prosody
[sem_group(l3,[l6,l4]),            % Groupings
 sem_group(l2,[l5]),
 sem_group(l1,[l7])] ).
```

CLASS NAME: **present_participle**
SUFFIX: prpa
PART OF SPEECH: **ADJA, ADJD**
REPRESENTATION: [R(L1,I1),arg123(L1,I1,I2)]
SEMDB: predname R
COMMENTS: L1 is in a grouping. The arg123-relation in the decomposition is the one that corresponds to the role of the subject argument of the involved verb. R is also used in a verb class.
EXAMPLE: Das *beginnende* Chaos

```
vit( segment_description('prpa','das beginnende Chaos'),
     [chaos(l4,i2),                    % Semantics
      beginnen(l3,i1),
      arg1(l3,i1,i2),
      def(l6,i2,l1)],
     l6,                               % Main Label
     [s_sort(i2,;(property,situation))],% Sorts
     [],                               % Discourse
     [gend(i2,neut),                   % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom),
      cas(i2,acc)],
     [ta_tense(i1,prespart)],          % Tense and Aspect
     [],                               % Scope
     [pros_mood(l6,decl)],             % Prosody
     [sem_group(l1,[l3,l4])] ).        % Groupings
```

CLASS NAME:	**participle**
SUFFIX:	`part`
PART OF SPEECH:	**VVPP**
REPRESENTATION:	`[perf(L,I,G),R(L1,I1),arg3(L1,I1,I2)]`
SEMDB:	`predname R`
COMMENTS:	`L1` is in grouping `G`. `L` is in a grouping. `R` is also used in a verb class. This representation of the participle models its use in the perfect tense (with an auxiliary, e.g., in "Ich bin gekommen") as well as its attributive use (like in "der eingeplante Termin"). The difference between both uses of the participle, however, is the optionality of the arg3-relation. For perfect forms of verbs, this argument relation may be omitted according to general usage (if there is no overt realization of a corresponding argument position). No such omission is possible in the attributive use.
EXAMPLE:	Der *eingeplante* Termin ist schlecht

```
vit( segment_description('part','der eingeplante termin ist schlecht'),
     [termin(l9,i4),                    % Semantics
      einplanen(l4,i2),
      perf(l8,i3,l1),
      support(l7,i1,l6),
      schlecht(l6,i4),
      decl(l5,h1),
      arg3(l4,i2,i4),
      def(l13,i4,l3)],
     l5,                                % Main Label
     [s_sort(i4,time_sit_poly)],        % Sorts
     [],                                % Discourse
     [gend(i4,masc),                    % Syntax
      num(i4,sg),
      pers(i4,3),
      cas(i4,nom)],
     [ta_tense(i2,perf),                % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [leq(l2,h1),                       % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l3,[l8,l9]),            % Groupings
      sem_group(l2,[l7]),
      sem_group(l1,[l4])] ).
```

Class Name: **transitive_adverb_adjective**
Suffix: `tadx`
Part of Speech: **ADJA,ADJD**
Representation: `[R(L,I), n_arg(L,I,I1)]`
SemDB: `predname R`
Comments: `L` can be in a grouping. `I` is the instance of the modified constituent. `n_arg` is optional and appears only if the argument DP of the adjective is overtly expressed.
Example: Der Termin ist *nah* am Wochenende

```
vit( segment_description('tadxr2u1','der termin ist nah am wochenende'),
     [termin(l9,i3),                    % Semantics
      support(l8,i2,l7),
      nahe(l7,i3),
      wochenende(l6,i1),
      decl(l5,h1),
      n_arg(l7,i3,i1),
      def(l12,i3,l3),
      def(l10,i1,l1)],
     l5,                                % Main Label
     [s_sort(i1,time),                  % Sorts
      s_sort(i3,time_sit_poly)],
     [dir(l4,no)],                      % Discourse
     [gend(i3,masc),                    % Syntax
      num(i3,sg),
      pers(i3,3),
      cas(i3,nom)],
     [ta_mood(i2,ind),                  % Tense and Aspect
      ta_tense(i2,pres)],
     [leq(l2,h1),                       % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l2,[l8]),               % Groupings
      sem_group(l3,[l9]),
      sem_group(l1,[l6])] ).
```

CLASS NAME:	**ordinal**
SUFFIX:	`ord`
PART OF SPEECH:	**ORD**
REPRESENTATION:	`[ord(L,I1,I2,D)]`
SEMDB:	`designator D`
COMMENTS:	`L` can be in a grouping. `I1` is the instance of the ordinal. `I2` is the instance of the argument of the ordinal (can be unbound). `D` is a number. Two cases are covered by this representation. E.g. "der dritte Mai" is represented as `[ord(L,I1,I2,3),mofy(L1,I2,may)]`, and "die zweite Maiwoche" is represented as `[ord(L,I1,I2,2),maiwoche(L1,I1)]`.
EXAMPLE:	Der *achte Oktober* ist nicht schlecht

```
vit( segment_description('ord','der achte oktober ist nicht schlecht'),
     [support(l10,i1,l9),              % Semantics
      schlecht(l9,i4),
      decl(l8,h1),
      mofy(l7,i3,oct),
      neg(l6,i2,h2),
      ord(l5,i4,i3,8),
      def(l14,i4,l4),
      def(l13,i3,l2)],
     l8,                               % Main Label
     [s_sort(i3,time),                 % Sorts
      s_sort(i4,time)],
     [],                               % Discourse
     [],                               % Syntax
     [ta_mood(i1,ind),                 % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l3,h2),                      % Scope
      leq(l3,h1),
      leq(l1,h1),
      ccom_plug(h2,l3),
      ccom_plug(h1,l1)],
     [pros_mood(l8,decl)],             % Prosody
     [sem_group(l4,[l5]),              % Groupings
      sem_group(l3,[l10]),
      sem_group(l2,[l7]),
      sem_group(l1,[l6])] ).
```

Class Name:	**cardinal**
Suffix:	`card`
Part of Speech:	**CARD**
Representation:	`[card(L,I,D)]`
SemDB:	`designator D`
Comments:	`L` can be in a grouping. `I` is the instance of the modified constituent, `D` is a number.
Example:	*Zwei* Stunden reichen

```
vit( segment_description('card','zwei stunden reichen'),
     [card(l6,i2,2),                      % Semantics
      stunde(l5,i2),
      reichen(l3,i1),
      decl(l4,h1),
      arg3(l3,i1,i2),
      udef(l8,i2,l2)],
     l4,                                  % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i2,time)],
     [],                                  % Discourse
     [gend(i2,fem),                       % Syntax
      num(i2,pl),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                         % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l4,decl)],                % Prosody
     [sem_group(l2,[l6,l5]),              % Groupings
      sem_group(l1,[l3])] ).
```

CLASS NAME:	**scopal_adverb**
SUFFIX:	`scadv`
PART OF SPEECH:	**ADV,ADJD**
REPRESENTATION:	`[R(L,I,H)]`
SEMDB:	`predname R`
COMMENTS:	Covers negation, quantificational adverbs, modal adverbs, and scalar adverbs. `I` is an eventuality variable. `L` is in a grouping. `H` is underspecified scope.
EXAMPLE:	Das ist *vielleicht* besser

```
vit( segment_description('scadv','das ist vielleicht besser'),
     [support(l6,i1,l3),                % Semantics
      gut(l3,i5),
      decl(l5,h1),
      vielleicht(l4,i4,h2),
      comp(l3,i5,i3,i2),
      pron(l11,i5)],
     l5,                                % Main Label
     [],                                % Sorts
     [prontype(i5,third,demon)],        % Discourse
     [gend(i5,neut),                    % Syntax
      num(i5,sg),
      pers(i5,3),
      cas(i5,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h2),                       % Scope
      leq(l2,h1),
      leq(l1,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l1)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l2,[l6]),               % Groupings
      sem_group(l1,[l4])] ).
```

CLASS NAME: **focusing_adverb**
SUFFIX: `fadv`
PART OF SPEECH: **ADV,ADJD**
REPRESENTATION: `[R(L,L1,H)]`
SEMDB: `predname R`
COMMENTS: `L` is not in a grouping. `L1` is a label of the constituent in focus (semantic focus), possible unbound (in that case `unbound(L1)` in `Scope`), `H` is underspecified scope.
EXAMPLE: Das ist *auch* gut

```
vit( segment_description('fadv','das ist auch gut'),
     [support(l6,i1,l5),               % Semantics
      gut(l5,i2),
      decl(l4,h1),
      auch(l2,l3,h2),
      pron(l8,i2)],
     l4,                               % Main Label
     [],                               % Sorts
     [prontype(i2,third,demon)],       % Discourse
     [gend(i2,neut),                   % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                 % Tense and Aspect
      ta_tense(i1,pres)],
     [unbound(l3),                     % Scope
      leq(l1,h2),
      leq(l1,h1),
      leq(l2,h1),
      ccom_plug(h2,l1),
      ccom_plug(h1,l2)],
     [pros_mood(l4,decl)],             % Prosody
     [sem_group(l1,[l6])] ).           % Groupings
```

CLASS NAME:	**iterative_adverb**
SUFFIX:	`itadv`
PART OF SPEECH:	**ADV**
REPRESENTATION:	`[tloc(L,I,I1),card(L1,I1,D),mal(L2,I1)]`
SEMDB:	`designator D`
COMMENTS:	`L` is in a grouping. `L1` and `L2` are in the same grouping, this grouping is argument of a `udef` operator. `I` is an eventuality variable. `D` is a number or `x`.
EXAMPLE:	Wir fahren *dreimal* nach Mallorca

```
vit( segment_description('itadv','wir fahren dreimal nach mallorca'),
     [fahren(l6,i1),                    % Semantics
      mal(l10,i3),
      card(l9,i3,3),
      named(l8,i2,mallorca),
      decl(l7,h1),
      arg3(l6,i1,i4),
      tloc(l5,i1,i3),
      nach(l4,i1,i2),
      pron(l14,i4),
      udef(l13,i3,l2),
      def(l12,i2,l1)],
     l7,                                % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i2,geo_location),
      s_sort(i4,&(human,object))],
     [dir(l4,yes),                      % Discourse
      prontype(i4,sp_he,std)],
     [num(i4,pl),                       % Syntax
      pers(i4,1),
      gend(i2,neut),
      num(i2,sg),
      pers(i2,3),
      cas(i2,dat),
      cas(i4,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l3,h1),                       % Scope
      ccom_plug(h1,l3)],
     [pros_mood(l7,decl)],              % Prosody
     [sem_group(l3,[l4,l5,l6]),         % Groupings
      sem_group(l2,[l9,l10]),
      sem_group(l1,[l8])] ).
```

CLASS NAME:	**standard_adverb_adjective**
SUFFIX:	`sadx`
PART OF SPEECH:	**ADV,ADJD,ADJA**
REPRESENTATION:	`[R(L,I)]`
SEMDB:	`predname R`
COMMENTS:	`L` can be in a grouping. `I` is the instance of the modified constituent.
EXAMPLE:	Das passt mir *gut*

```
vit( segment_description('sadx','das passt mir gut'),
     [passen_suit_aufgeben(l3,i1),      % Semantics
      gut(l5,i1),
      decl(l4,h1),
      s_arg(l3,i1,i3),
      perspective(l2,i1,i2),
      pron(l8,i3),
      pron(l7,i2)],
     l4,                                % Main Label
     [s_sort(i1,mental_sit),            % Sorts
      s_sort(i2,human),
      s_sort(i3,person)],
     [prontype(i3,third,demon),         % Discourse
      prontype(i2,sp,std)],
     [gend(i3,neut),                    % Syntax
      num(i3,sg),
      pers(i3,3),
      num(i2,sg),
      pers(i2,1),
      cas(i2,dat),
      cas(i3,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                       % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l4,decl)],              % Prosody
     [sem_group(l1,[l2,l5,l3])] ).      % Groupings
```

Class Name:	**temporal_adverb**
Suffix:	`tadv`
Part of Speech:	**ADV,ADJD**
Representation:	`[R(L,I)]`
SemDB:	`predname R`
Comments:	`L` can be in a grouping. `I` is the instance of the modified constituent.
Example:	Wir treffen uns *morgen*

```
vit( segment_description('tadv','wir treffen uns morgen'),
     [treffen(l2,i1),                     % Semantics
      morgen_tadv(l4,i1),
      decl(l3,h1),
      arg1(l2,i1,i3),
      arg3(l2,i1,i2),
      pron(l7,i3),
      pron(l6,i2)],
     l3,                                  % Main Label
     [s_sort(i1,&(meeting_sit,time)),     % Sorts
      s_sort(i2,&(human,person)),
      s_sort(i3,&(human,person))],
     [prontype(i3,sp_he,std),             % Discourse
      prontype(i2,sp_he,refl_std)],
     [num(i3,pl),                         % Syntax
      pers(i3,1),
      num(i2,pl),
      pers(i2,1),
      cas(i2,acc),
      cas(i3,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                         % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l3,decl)],                % Prosody
     [sem_group(l1,[l4,l2])] ).           % Groupings
```

Class Name:	**quantificational_adverb**
Suffix:	`qadv`
Part of Speech:	**ADV,ADJD**
Representation:	`[R3(L1,I1,I2),R1(L2,I2,G,H)/R1(L2,I2,G),R2(L3,I2,D)/R2(L3,I2)]`
SemDB:	`predname R1 predname R2 predname R3`
Comments:	`I1` is the instance of the modified constituent. `R1` is a quantifiertype. `R2` is the relation that describes the restriction of the quantifier `L2`. `R3` is a prepositional relation. `L2` is not in a grouping.
Example:	Ich kann *irgendwann* kommen

```
vit( segment_description('qadv','ich kann irgendwann kommen'),
     [time(l9,i3),                       % Semantics
      kommen(l5,i2),
      decl(l8,h2),
      koennen(l7,i1,h1),
      tloc(l6,i2,i3),
      arg3(l5,i2,i4),
      indef(l4,i3,l2,h3),
      pron(l13,i4)],
     l8,                                 % Main Label
     [s_sort(i1,mental_sit),             % Sorts
      s_sort(i2,move_sit),
      s_sort(i4,&(human,space_time))],
     [dir(l6,no),                        % Discourse
      prontype(i4,sp,std)],
     [num(i4,sg),                        % Syntax
      pers(i4,1),
      cas(i4,nom)],
     [ta_tense(i2,infin),                % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [leq(l3,h3),                        % Scope
      leq(l3,h2),
      leq(l3,h1),
      leq(l4,h2),
      leq(l1,h2),
      ccom_plug(h3,l1),
      ccom_plug(h2,l4),
      ccom_plug(h1,l3)],
     [pros_mood(l8,decl)],               % Prosody
     [sem_group(l3,[l6,l5]),             % Groupings
      sem_group(l2,[l9]),
```

```
sem_group(l1,[l7])] ).
```

Class Name:	**pragmatic_adverb**
Suffix:	`padv`
Part of Speech:	**ADV,ADJD**
Representation:	`[R(L,H)]`
SemDB:	`predname R`
Comments:	`L` is not in a grouping. `H` is underspecified scope.
Example:	Das kann ich *doch* nicht sagen

```
vit( segment_description('padv','das kann ich doch nicht sagen'),
     [doch(l8,h4),                        % Semantics
      sagen(l4,i2),
      decl(l7,h3),
      koennen(l6,i1,h1),
      neg(l5,i3,h2),
      arg1(l4,i2,i4),
      arg3(l4,i2,i5),
      pron(l13,i5),
      pron(l12,i4)],
     l7,                                  % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i2,communicat_sit),
      s_sort(i4,&(human,person))],
     [prontype(i5,third,demon),           % Discourse
      prontype(i4,sp,std)],
     [gend(i5,neut),                      % Syntax
      num(i5,sg),
      pers(i5,3),
      num(i4,sg),
      pers(i4,1),
      cas(i4,nom),
      cas(i5,acc)],
     [ta_tense(i2,infin),                 % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [leq(l3,h4),                         % Scope
      leq(l3,h3),
      leq(l3,h2),
      leq(l3,h1),
      leq(l8,h3),
      leq(l2,h3),
      leq(l1,h3),
      ccom_plug(h4,l2),
```

```
 ccom_plug(h3,l8),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l7,decl)],                % Prosody
[sem_group(l3,[l4]),                 % Groupings
 sem_group(l2,[l5]),
 sem_group(l1,[l6])] ).
```

Class Name: **discourse_relation_adverb**
Suffix: `dra`
Part of Speech: **ADV,ADJD,KON,KOUS,KOUI**
Representation: `[R(L,I,H1,H2)/R(L,I,H2)]`
SemDB: `predname R`
Comments: The first representation is for adverbs which syntactically operate as sentence connectors, the second for adverbs which are sentence modifiers. `L` is in a grouping. `I` is an eventuality variable, `H1` is underspecified scope of the first sentential argument, and `H2` of the second.
Example: Ich komme Montag *weil* das mir besser passt

```
vit( segment_description('dra','ich komme montag weil das mir besser passt'),
     [kommen(l11,i7),                     % Semantics
      gut(l5,i1),
      passen_suit_aufgeben(l8,i1),
      decl(l12,h1),
      arg3(l11,i7,i9),
      dofw(l10,i8,mon),
      tloc(l9,i7,i8),
      s_arg(l8,i1,i5),
      perspective(l7,i1,i4),
      weil(l6,i6,h3,h2),
      comp(l5,i1,i3,i2),
      pron(l21,i9),
      def(l20,i8,l3),
      pron(l17,i5),
      pron(l16,i4)],
     l12,                                 % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i4,human),
      s_sort(i5,person),
      s_sort(i7,move_sit),
      s_sort(i8,time),
      s_sort(i9,&(human,space_time))],
     [dir(l9,no),                         % Discourse
      prontype(i9,sp,std),
      prontype(i5,third,demon),
      prontype(i4,sp,std)],
     [num(i9,sg),                         % Syntax
      pers(i9,1),
      gend(i5,neut),
      num(i5,sg),
```

```
 pers(i5,3),
 num(i4,sg),
 pers(i4,1),
 cas(i4,dat),
 cas(i5,nom),
 cas(i9,nom)],
[ta_mood(i7,ind),                  % Tense and Aspect
 ta_tense(i7,pres),
 ta_mood(i1,ind),
 ta_tense(i1,pres)],
[leq(l4,h3),                       % Scope
 leq(l2,h1),
 leq(l1,h2),
 ccom_plug(h3,l4),
 ccom_plug(h2,l1),
 ccom_plug(h1,l2)],
[pros_mood(l12,decl)],             % Prosody
[sem_group(l1,[l7,l5,l8]),         % Groupings
 sem_group(l4,[l9,l11]),
 sem_group(l3,[l10]),
 sem_group(l2,[l6])] ).
```

CLASS NAME: **coordination_particle**

SUFFIX: `coord`

PART OF SPEECH: **KON**

REPRESENTATION: `[R(L,....)]`

SEMDB: `predname R`

COMMENTS: There are five cases:

- `[R(L,H1,H2)]` for coordination of sentences without mood operators. E.g. "ich komme und ich gehe".
- `[R(L,L1,L2)]` for coordination of utterances with mood operators. E.g. "wann kommen sie und wie spaet kommen sie?"
- `[R(L,I,L1,I1,L2,I2)]` for coordination of adjectives, common nouns, NPs without holes. E.g. "jeder [montag und dienstag]".
- `[R(L,I,H,I1,I2)]` for coordination of two NPs with holes. E.g. "jeder montag und jeder dienstag".
- `[R(L,I,H,I1,L2,I2)]` for coordination of two NPs where one NP has a hole and the other hasn't. E.g. "jeder montag und der dienstag".

EXAMPLE: Ich komme Montag *und* Dienstag

```
vit( segment_description('coord','ich komme montag und dienstag'),
     [kommen(l9,i1),                    % Semantics
      decl(l10,h1),
      arg3(l9,i1,i5),
      dofw(l8,i3,mon),
      dofw(l7,i2,tue),
      tloc(l6,i1,i4),
      und(l5,i4,l3,i3,l2,i2),
      pron(l15,i5),
      def(l13,i3,l3),
      def(l14,i4,l1),
      def(l12,i2,l2)],
     l10,                               % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i2,time),
      s_sort(i3,time),
      s_sort(i5,&(human,space_time))],
     [dir(l6,no),                       % Discourse
      prontype(i5,sp,std)],
```

```
[num(i5,sg),                    % Syntax
 pers(i5,1),
 cas(i5,nom)],
[ta_mood(i1,ind),               % Tense and Aspect
 ta_tense(i1,pres)],
[leq(l4,h1),                    % Scope
 ccom_plug(h1,l4)],
[pros_mood(l10,decl)],          % Prosody
[sem_group(l4,[l6,l9]),         % Groupings
 sem_group(l3,[l8]),
 sem_group(l2,[l7]),
 sem_group(l1,[l5])] ).
```

Class Name:	**gradual**
Suffix:	`grad`
Part of Speech:	**ADV,ADJD**
Representation:	`[R(L,L1)]`
SemDB:	`predname R`
Comments:	`L` can be in a grouping. `L1` covers the adjective, adverb, preposition, or cardinal.
Example:	Das ist leider *sehr* schlecht

```
vit( segment_description('grad','das ist leider sehr schlecht'),
     [support(l6,i1,l4),              % Semantics
      leider(l5,h2),
      sehr_grad(l4,l3),
      schlecht(l3,i2),
      decl(l2,h1),
      pron(l8,i2)],
     l2,                              % Main Label
     [],                              % Sorts
     [prontype(i2,third,demon)],      % Discourse
     [gend(i2,neut),                  % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h2),                     % Scope
      leq(l1,h1),
      leq(l5,h1),
      ccom_plug(h2,l1),
      ccom_plug(h1,l5)],
     [pros_mood(l2,decl)],            % Prosody
     [sem_group(l1,[l6])] ).          % Groupings
```

CLASS NAME:	**gradual_adverb_adjective**
SUFFIX:	gadx
PART OF SPEECH:	**PIS, PIAT**
REPRESENTATION:	[R1(L,L1),R2(L1,I)]
SEMDB:	predname R1 predname R2
COMMENTS:	Decomposition into gradual and adverb/adjective. L can be in a grouping.
EXAMPLE:	Wir haben *zuviele* Termine

```
vit( segment_description('gadx','wir haben zuviele termine'),
     [haben(l3,i3),                        % Semantics
      viel(l7,i2),
      zu_grad(l6,l7),
      termin(l5,i2),
      decl(l4,h1),
      arg2(l3,i3,i4),
      arg3(l3,i3,i2),
      udef(l10,i2,l1),
      pron(l9,i4)],
     l4,                                   % Main Label
     [s_sort(i1,time_sit_poly),            % Sorts
      s_sort(i3,situation),
      s_sort(i4,human)],
     [prontype(i4,sp_he,std)],             % Discourse
     [num(i4,pl),                          % Syntax
      pers(i4,1),
      gend(i1,masc),
      num(i1,pl),
      pers(i1,3),
      cas(i1,acc),
      cas(i4,nom)],
     [ta_mood(i3,ind),                     % Tense and Aspect
      ta_tense(i3,pres)],
     [leq(l2,h1),                          % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l4,decl)],                 % Prosody
     [sem_group(l2,[l3]),                  % Groupings
      sem_group(l1,[l5,l6])] ).
```

CLASS NAME: **standard_preposition**
SUFFIX: `sprep`
PART OF SPEECH: **APPR, APPO**
REPRESENTATION: `[R(L,I1,I2)]`
SEMDB: `predname R`
COMMENTS: `L` can be in a grouping. `I1` is the instance of the modified constituent (external argument), `I2` is the instance of the argument NP (internal argument). See also the note on gradual prepositions in section two of this document. For prepositions a corresponding `dir(L,yes/no)` is compulsory.
EXAMPLE: Ich komme *an* einem Montag

```
vit( segment_description('sprep','ich komme an einem montag'),
     [kommen(l6,i1),                      % Semantics
      decl(l7,h1),
      arg3(l6,i1,i3),
      an(l5,i1,i2),
      dofw(l4,i2,mon),
      ein_card_qua(l3,i2,l11,l1,h2,1),
      pron(l10,i3)],
     l7,                                  % Main Label
     [s_sort(i1,move_sit),                % Sorts
      s_sort(i2,time),
      s_sort(i3,&(human,space_time))],
     [dir(l5,no),                         % Discourse
      prontype(i3,sp,std)],
     [num(i3,sg),                         % Syntax
      pers(i3,1),
      cas(i3,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [unbound(l11),                       % Scope
      leq(l2,h2),
      leq(l2,h1),
      leq(l3,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l3)],
     [pros_mood(l7,decl)],                % Prosody
     [sem_group(l2,[l5,l6]),              % Groupings
      sem_group(l1,[l4])] ).
```

CLASS NAME:	**definite_preposition**
SUFFIX:	`dprep`
PART OF SPEECH:	**APPRART**
REPRESENTATION:	`[R(L,I1,I2),def(L2,I2,G)]`
SEMDB:	`predname R`
COMMENTS:	decomposition: def + sprep.
EXAMPLE:	Ich komme *am* Montag

```
vit( segment_description('dprep','ich komme am montag'),
     [kommen(l5,i1),                      % Semantics
      decl(l6,h1),
      arg3(l5,i1,i3),
      an(l4,i1,i2),
      dofw(l3,i2,mon),
      pron(l9,i3),
      def(l8,i2,l1)],
     l6,                                  % Main Label
     [s_sort(i1,move_sit),                % Sorts
      s_sort(i2,time),
      s_sort(i3,&(human,space_time))],
     [dir(l4,no),                         % Discourse
      prontype(i3,sp,std)],
     [num(i3,sg),                         % Syntax
      pers(i3,1),
      cas(i3,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h1),                         % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l6,decl)],                % Prosody
     [sem_group(l2,[l4,l5]),              % Groupings
      sem_group(l1,[l3])] ).
```

Class Name:	**anaphoric_preposition**
Suffix:	aprep
Part of Speech:	**PAV**
Representation:	[R(L,I,I1),demonstrative(L1,I1,G)]
SemDB:	predname R demontype D
Comments:	decomposition: demonstrative + sprep.
Example:	*unterdessen* gehen wir

```
vit( segment_description('aprep','unterdessen gehen wir'),
     [abstr_nom(l6,i3),                   % Semantics
      gehen_move_passen(l3,i1),
      decl(l5,h1),
      waehrend_sprep(l4,i1,i3),
      s_arg(l3,i1,i2),
      demonstrative(l9,i3,l2),
      pron(l8,i2)],
     l5,                                  % Main Label
     [s_sort(i1,;(move_sit,mental_sit)),% Sorts
      s_sort(i2,human)],
     [dir(l4,no),                         % Discourse
      demontype(i3,far),
      prontype(i2,sp_he,std)],
     [num(i2,pl),                         % Syntax
      pers(i2,1),
      cas(i2,nom)],
     [ta_mood(i1,ind),                    % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                         % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l5,decl)],                % Prosody
     [sem_group(l1,[l4,l3]),              % Groupings
      sem_group(l2,[l6])] ).
```

Class Name:	**locational_anaphoric_preposition**
Suffix:	`laprep`
Part of Speech:	**PAV**
Representation:	`[R(L,I,I1),demonstrative(L1,I1,G),abstr_loc(L2,I1)]`
SemDB:	`predname R demontype D`
Comments:	
Example:	Wir kommen *hierhin*

```
vit( segment_description('laprep','wir kommen hierhin'),
     [kommen(l4,i1),                        % Semantics
      abstr_loc(l6,i2),
      decl(l5,h1),
      arg3(l4,i1,i3),
      zu(l3,i1,i2),
      pron(l9,i3),
      demonstrative(l8,i2,l1)],
     l5,                                    % Main Label
     [s_sort(i1,move_sit),                  % Sorts
      s_sort(i3,&(human,space_time))],
     [dir(l3,yes),                          % Discourse
      prontype(i3,sp_he,std),
      demontype(i2,near)],
     [num(i3,pl),                           % Syntax
      pers(i3,1),
      cas(i3,nom)],
     [ta_mood(i1,ind),                      % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h1),                           % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l5,decl)],                  % Prosody
     [sem_group(l2,[l3,l4]),                % Groupings
      sem_group(l1,[l6])] ).
```

Class Name:	**reciprocal_preposition**
Suffix:	`recprep`
Part of Speech:	**ADV**
Representation:	`[R(L1,I1,I2),pron(L3,I2)]`
SemDB:	`predname R`
Comments:	Decomposition into pronouns and preposition. In the `Discourse` slot there is `prontype(I2,top,recip)`.
Example:	*miteinander*

```
vit( segment_description('recprep',miteinander),
     [mit(l2,i1,i2),                     % Semantics
      pron(l3,i2)],
     l4,                                 % Main Label
     [],                                 % Sorts
     [prontype(i2,top,recip)],           % Discourse
     [],                                 % Syntax
     [],                                 % Tense and Aspect
     [],                                 % Scope
     [pros_mood(l4,decl)],               % Prosody
     [sem_group(l4,[l2])] ).             % Groupings
```

Class Name:	**circumposition**
Suffix:	`cir`
Part of Speech:	**APPR+APZR**
Representation:	`[R1(L1,L2), R2(L2,I1,I2)]`
SemDB:	`Predname R1 Predname R2`
Comments:	Expressions like "um ... herum", "in ... hinein", "bis ... an", that may be analyzed as circumpositions, are rendered in terms of a gradual modifying a prepositional phrase.
Example:	Ich habe *um* den siebzehnten *herum* Zeit

```
vit( segment_description(ttestr12u1,yes,'ich habe um den siebzehnten herum zeit'),
     [haben(l6,i1),                      % Semantics
      zeit(l8,i2),
      decl(l7,h1),
      arg2(l6,i1,i5),
      arg3(l6,i1,i2),
      herum(l5,l14),
      um(l14,i1,i4),
      ord(l4,i4,i3,17),
      pron(l13,i5),
      def(l12,i4,l2),
      udef(l10,i2,l1)],
     l7,                                 % Main Label
     [s_sort(i1,situation),              % Sorts
      s_sort(i2,time),
      s_sort(i4,time),
      s_sort(i5,human)],
     [dir(l5,no),                        % Discourse
      prontype(i5,sp,std)],
     [num(i5,sg),                        % Syntax
      pers(i5,1),
      gend(i2,fem),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i5,nom)],
     [ta_mood(i1,ind),                   % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l3,h1),                        % Scope
      ccom_plug(h1,l3)],
     [pros_mood(l7,decl)],               % Prosody
     [sem_group(l3,[l5,l6]),             % Groupings
```

```
sem_group(l2,[l4]),
sem_group(l1,[l8])] ).
```

CLASS NAME: **fadv_padv**
SUFFIX: `fadv_padv`
PART OF SPEECH:
REPRESENTATION: `[R_fadv_padv(L,L1,H)]`
SEMDB: `predname R`
COMMENTS: Disambiguation:

fadv: `[R_fadv(L,L1,H)]`

padv: `[R_padv(L,H)]`

EXAMPLE: wir koennen *noch* einen termin ausmachen

```
vit( segment_description('fadv_padv','wir koennen noch einen termin ausmachen'),
     [termin(l10,i3),                   % Semantics
      ausmachen(l5,i2),
      decl(l9,h3),
      koennen(l8,i1,h1),
      noch_fadv_padv(l6,l7,h4),
      arg1(l5,i2,i4),
      arg3(l5,i2,i3),
      ein_card_qua(l4,i3,l15,l2,h2,1),
      pron(l14,i4)],
     l9,                                % Main Label
     [s_sort(i1,mental_sit),            % Sorts
      s_sort(i2,ment_communicat_poly),
      s_sort(i3,&(space_time,time_sit_poly)),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp_he,std)],          % Discourse
     [num(i4,pl),                       % Syntax
      pers(i4,1),
      gend(i3,masc),
      num(i3,sg),
      pers(i3,3),
      cas(i3,acc),
      cas(i4,nom)],
     [ta_tense(i2,infin),               % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l7),                      % Scope
      unbound(l15),
      leq(l3,h4),
      leq(l3,h3),
```

```
 leq(l3,h2),
 leq(l3,h1),
 leq(l6,h3),
 leq(l4,h3),
 leq(l1,h3),
 ccom_plug(h4,l4),
 ccom_plug(h3,l6),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l9,decl)],              % Prosody
[sem_group(l3,[l5]),               % Groupings
 sem_group(l2,[l10]),
 sem_group(l1,[l8])] ).
```

CLASS NAME: **laprep_padv**

SUFFIX: `laprep_padv`

PART OF SPEECH:

REPRESENTATION: `[R_laprep_padv(L,I,H,L1,I1,T,M,D)]`

SEMDB: `demontype D, predname R`

COMMENTS: Disambiguation:

padv: `[R_padv(L,H),leq(L,T),leq(M,H)]`

laprep:
```
[R(L1,I,I1),demonstrative(L2,I1,G),
     abstr_loc(L3,I1),sem_group(G,[L3]),
     add_to_group(L1,M),demontype(I1,D),eq(L,H)]
```

EXAMPLE: wir koennen *da* einen termin ausmachen

```
vit( segment_description('laprep_padv','wir koennen da einen termin ausmachen'),
     [termin(l9,i2),                      % Semantics
      ausmachen(l6,i3),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i3,i5),
      arg3(l6,i3,i2),
      ein_card_qua(l5,i2,l16,l2,h2,1),
      da_laprep_padv(l4,i3,h4,l17,i4,h3,l3,spec),
      pron(l15,i5)],
     l8,                                  % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i2,&(space_time,time_sit_poly)),
      s_sort(i3,ment_communicat_poly),
      s_sort(i5,&(human,person))],
     [dir(l4,no),                         % Discourse
      prontype(i5,sp_he,std)],
     [num(i5,pl),                         % Syntax
      pers(i5,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i5,nom)],
     [ta_tense(i3,infin),                 % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l17),                       % Scope
```

```
 unbound(l16),
 leq(l3,h3),
 leq(l3,h2),
 leq(l3,h1),
 leq(l5,h3),
 leq(l1,h3),
 ccom_plug(h4,l5),
 ccom_plug(h3,l4),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],              % Prosody
[sem_group(l3,[l6]),               % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

Class Name: **padv_scadv**
Suffix: `padv_scadv`
Part of Speech:
Representation: `[R_padv_scadv(L,H,T,M)]`
SemDB: `predname R`
Comments: Disambiguation:

padv: `[R_padv(L,H),leq(L,T),leq(M,H)]`

scadv: `[R_scadv(L,I,H),sem_group(G,[L]),`
`leq(G,T),leq(M,H)]`

Example: wir koennen *denn* einen termin ausmachen

```
vit( segment_description('padv_scadv','wir koennen denn einen termin ausmachen'),
     [termin(l9,i3),                      % Semantics
      ausmachen(l6,i2),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i2,i4),
      arg3(l6,i2,i3),
      denn_padv_scadv(l5,h4,h3,l3),
      ein_card_qua(l4,i3,l14,l2,h2,1),
      pron(l13,i4)],
     l8,                                  % Main Label
     [s_sort(i1,mental_sit),              % Sorts
      s_sort(i2,ment_communicat_poly),
      s_sort(i3,&(space_time,time_sit_poly)),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp_he,std)],            % Discourse
     [num(i4,pl),                         % Syntax
      pers(i4,1),
      gend(i3,masc),
      num(i3,sg),
      pers(i3,3),
      cas(i3,acc),
      cas(i4,nom)],
     [ta_tense(i2,infin),                 % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l14),                       % Scope
      leq(l3,h3),
      leq(l3,h2),
```

```
 leq(l3,h1),
 leq(l4,h3),
 leq(l1,h3),
 ccom_plug(h4,l4),
 ccom_plug(h3,l5),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],               % Prosody
[sem_group(l3,[l6]),                % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

Class Name: **padv_tadv**
Suffix: `padv_tadv`
Part of Speech:
Representation: `[R_padv_tadv(L,I,H,L1,T,M)]`
SemDB: `predname R`
Comments: Disambiguation:

padv: `[R_padv(L,H),leq(L,T),leq(M,H)]`

tadv: `[R_tadv(L1,I),add_to_group(L1,M),eq(L,H)]`

Example: wir koennen *eben* einen termin ausmachen

```
vit( segment_description('padv_tadv','wir koennen eben einen termin ausmachen'),
     [termin(l9,i3),                    % Semantics
      ausmachen(l6,i2),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i2,i4),
      arg3(l6,i2,i3),
      eben_padv_tadv(l5,i2,h4,l15,h3,l3),
      ein_card_qua(l4,i3,l14,l2,h2,1),
      pron(l13,i4)],
     l8,                                % Main Label
     [s_sort(i1,mental_sit),            % Sorts
      s_sort(i2,ment_communicat_poly),
      s_sort(i3,&(space_time,time_sit_poly)),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp_he,std)],          % Discourse
     [num(i4,pl),                       % Syntax
      pers(i4,1),
      gend(i3,masc),
      num(i3,sg),
      pers(i3,3),
      cas(i3,acc),
      cas(i4,nom)],
     [ta_tense(i2,infin),               % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l15),                     % Scope
      unbound(l14),
      leq(l3,h3),
      leq(l3,h2),
```

```
 leq(l3,h1),
 leq(l4,h3),
 leq(l1,h3),
 ccom_plug(h4,l4),
 ccom_plug(h3,l5),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],                % Prosody
[sem_group(l3,[l6]),                 % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

CLASS NAME: **sadx_scadv**
SUFFIX: `sadx_scadv`
PART OF SPEECH:
REPRESENTATION: `[R_sadx_scadv(L,I,H,L1,T,M)]`
SEMDB: `predname R`
COMMENTS: Disambiguation:

sadx: `[R_sadx(L1,I),add_to_group(L1,M),eq(L,H)]`

scadv: `[R_scadv(L,I1,H),sem_group(G,[L]),`
`leq(G,T),leq(M,H)]`

EXAMPLE: wir koennen *gleich* einen termin ausmachen

```
vit( segment_description('sadx_scadv','wir koennen gleich einen termin ausmachen'),
     [termin(l9,i3),                    % Semantics
      ausmachen(l6,i2),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i2,i4),
      arg3(l6,i2,i3),
      gleich_sadx_scadv(l5,i2,h4,l15,h3,l3),
      ein_card_qua(l4,i3,l14,l2,h2,1),
      pron(l13,i4)],
     l8,                                % Main Label
     [s_sort(i1,mental_sit),            % Sorts
      s_sort(i2,ment_communicat_poly),
      s_sort(i3,&(space_time,time_sit_poly)),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp_he,std)],          % Discourse
     [num(i4,pl),                       % Syntax
      pers(i4,1),
      gend(i3,masc),
      num(i3,sg),
      pers(i3,3),
      cas(i3,acc),
      cas(i4,nom)],
     [ta_tense(i2,infin),               % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l15),                     % Scope
      unbound(l14),
      leq(l3,h3),
```

```
 leq(l3,h2),
 leq(l3,h1),
 leq(l4,h3),
 leq(l1,h3),
 ccom_plug(h4,l4),
 ccom_plug(h3,l5),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],               % Prosody
[sem_group(l3,[l6]),                % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

CLASS NAME: **fadv_sadx**
SUFFIX: `fadv_sadx`
PART OF SPEECH:
REPRESENTATION: `[R_fadv_sadx(L,I,L1,H,L2,T,M)]`
SEMDB: `predname R`
COMMENTS: Disambiguation:

fadv: `[R_fadv(L,L1,H),leq(L,T),leq(M,H)]`

sadx: `[R_sadx(L2,I),eq(L,H),add_to_group(L2,M)]`

EXAMPLE: wir koennen *direkt* einen termin ausmachen

```
vit( segment_description('fadv_sadx','wir koennen direkt einen termin ausmachen'),
     [termin(l10,i2),                    % Semantics
      ausmachen(l7,i3),
      decl(l9,h3),
      koennen(l8,i1,h1),
      arg1(l7,i3,i4),
      arg3(l7,i3,i2),
      ein_card_qua(l6,i2,l16,l2,h2,1),
      direkt_fadv_sadx(l4,i3,l5,h4,l17,h3,l3),
      pron(l15,i4)],
     l9,                                 % Main Label
     [s_sort(i1,mental_sit),             % Sorts
      s_sort(i2,&(space_time,time_sit_poly)),
      s_sort(i3,ment_communicat_poly),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp_he,std)],           % Discourse
     [num(i4,pl),                        % Syntax
      pers(i4,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i4,nom)],
     [ta_tense(i3,infin),                % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l17),                      % Scope
      unbound(l16),
      leq(l3,h3),
      leq(l3,h2),
```

```
 leq(l3,h1),
 leq(l6,h3),
 leq(l1,h3),
 ccom_plug(h4,l6),
 ccom_plug(h3,l4),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l9,decl)],                % Prosody
[sem_group(l3,[l7]),                 % Groupings
 sem_group(l2,[l10]),
 sem_group(l1,[l8])] ).
```

Class Name: **aprep_dra**
Suffix: `aprep_dra`
Part of Speech:
Representation: `[R_aprep_dra(L,I,H,L1,I1,T,M,D)]`
SemDB: `predname R, demontype D`
Comments: Disambiguation:

aprep: `[R_aprep(L1,I,I1),demonstrative(L2,I1,G),`
`add_to_group(L1,M),eq(L,H),demontype(I1,D)]`

dra: `[R_dra(L,I1,H),leq(M,H),leq(L,T)]`

Example: wir koennen *dafuer* einen termin ausmachen

```
vit( segment_description('aprep_dra','wir koennen dafuer einen termin ausmachen'),
     [termin(l9,i2),                   % Semantics
      ausmachen(l6,i4),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i4,i5),
      arg3(l6,i4,i2),
      ein_card_qua(l5,i2,l16,l2,h2,1),
      dafuer_aprep_dra(l4,i4,h4,l17,i3,h3,l3,far),
      pron(l15,i5)],
     l8,                               % Main Label
     [s_sort(i1,mental_sit),           % Sorts
      s_sort(i2,&(space_time,time_sit_poly)),
      s_sort(i4,ment_communicat_poly),
      s_sort(i5,&(human,person))],
     [dir(l4,no),                      % Discourse
      prontype(i5,sp_he,std)],
     [num(i5,pl),                      % Syntax
      pers(i5,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i5,nom)],
     [ta_tense(i4,infin),              % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l17),                    % Scope
      unbound(l16),
```

```
 leq(l3,h3),
 leq(l3,h2),
 leq(l3,h1),
 leq(l5,h3),
 leq(l1,h3),
 ccom_plug(h4,l5),
 ccom_plug(h3,l4),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],                 % Prosody
[sem_group(l3,[l6]),                  % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

Class Name: **laprep_dra**
Suffix: laprep_dra
Part of Speech:
Representation: [R_laprep_dra(L,I,H,L1,I1,T,M,D)]
SemDB: predname R, demontype D
Comments: Disambiguation:

laprep: [R_laprep(L1,I,I1),demonstrative(L2,I1,G),
abstr_loc(L3,I1),
add_to_group(L1,M),eq(L,H),demontype(I1,D)]

dra: [R_dra(L,I1,H),leq(M,H),leq(L,T)]

Example: wir koennen *daher* einen termin ausmachen

```
vit( segment_description('laprep_dra','wir koennen daher einen termin ausmachen'),
     [termin(l9,i2),                   % Semantics
      ausmachen(l6,i4),
      decl(l8,h3),
      koennen(l7,i1,h1),
      arg1(l6,i4,i5),
      arg3(l6,i4,i2),
      ein_card_qua(l5,i2,l16,l2,h2,1),
      daher_laprep_dra(l4,i4,h4,l17,i3,h3,l3,spec),
      pron(l15,i5)],
     l8,                               % Main Label
     [s_sort(i1,mental_sit),           % Sorts
      s_sort(i2,&(space_time,time_sit_poly)),
      s_sort(i4,ment_communicat_poly),
      s_sort(i5,&(human,person))],
     [dir(l4,yes),                     % Discourse
      prontype(i5,sp_he,std)],
     [num(i5,pl),                      % Syntax
      pers(i5,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i5,nom)],
     [ta_tense(i4,infin),              % Tense and Aspect
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [unbound(l17),                    % Scope
```

```
 unbound(l16),
 leq(l3,h3),
 leq(l3,h2),
 leq(l3,h1),
 leq(l5,h3),
 leq(l1,h3),
 ccom_plug(h4,l5),
 ccom_plug(h3,l4),
 ccom_plug(h2,l1),
 ccom_plug(h1,l3)],
[pros_mood(l8,decl)],              % Prosody
[sem_group(l3,[l6]),               % Groupings
 sem_group(l2,[l9]),
 sem_group(l1,[l7])] ).
```

CLASS NAME: **card_qua**
SUFFIX: card_qua
PART OF SPEECH:
REPRESENTATION: [R_card_qua(L,I,L2,G,H,D)]
SEMDB: predname R, designator D
COMMENTS: Disambiguation:

card: [udef(L2,I,G),card(L1,I,D),
add_to_group(L1,G),eq(L,H)]

qua: [R_qua(L,I,G,H)]

EXAMPLE: *ein* Termin

```
vit( segment_description('card_qua','ein termin'),
     [termin(l4,i1),                    % Semantics
      ein_card_qua(l2,i1,l7,l1,h2,1)],
     l2,                                % Main Label
     [s_sort(i1,time_sit_poly)],        % Sorts
     [],                                % Discourse
     [gend(i1,masc),                    % Syntax
      num(i1,sg),
      pers(i1,3),
      cas(i1,nom)],
     [],                                % Tense and Aspect
     [unbound(l7),                      % Scope
      unbound(l6),
      leq(l6,h2),
      ccom_plug(h2,l6)],
     [pros_mood(l2,decl)],              % Prosody
     [sem_group(l1,[l4])] ).            % Groupings
```

Class Name: **def_qua**
Suffix: `def_qua`
Part of Speech:
Representation: `[R_def_qua(L,I,G,H)]`
SemDB: `predname R`
Comments: Disambiguation:

def: `[def(L,I,G)]`

qua: `[R_qua(L,I,G,H)]`

Example: *montags*

```
vit( segment_description('def_qua',montags),
     [dofw(l5,i2,mon),                 % Semantics
      tloc(l4,i1,i2),
      jed_def(l2,i2,l1,h2)],
     l7,                               % Main Label
     [s_sort(i2,time)],                % Sorts
     [dir(l4,no)],                     % Discourse
     [],                               % Syntax
     [],                               % Tense and Aspect
     [leq(l7,h2),                      % Scope
      ccom_plug(h2,l7)],
     [pros_mood(l7,decl)],             % Prosody
     [sem_group(l7,[l4]),              % Groupings
      sem_group(l1,[l5])] ).
```

CLASS NAME: **pcv_tv**
SUFFIX: pcv_tv
PART OF SPEECH:
REPRESENTATION: [R(L,I),s_arg(L,I,I1),o_arg(L,I,I2/L1/H)]
SEMDB: predname R
COMMENTS: s_arg is corresponding to the subject argument, o_arg to the object argument.
EXAMPLE: Ich *nehme* an dass er kommt

```
vit( segment_description('pcv_tv','ich nehme an dass er kommt'),
     [annehmen_accept_assume(l4,i3),    % Semantics
      kommen(l3,i1),
      decl(l5,h2),
      o_arg(l4,i3,h1),
      s_arg(l4,i3,i4),
      arg3(l3,i1,i2),
      pron(l9,i4),
      pron(l7,i2)],
     l5,                                % Main Label
     [s_sort(i1,move_sit),              % Sorts
      s_sort(i2,space_time),
      s_sort(i3,mental_sit),
      s_sort(i4,&(human,person))],
     [prontype(i4,sp,std),              % Discourse
      prontype(i2,third,std)],
     [num(i4,sg),                       % Syntax
      pers(i4,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom),
      cas(i4,nom)],
     [ta_mood(i3,ind),                  % Tense and Aspect
      ta_tense(i3,pres),
      ta_mood(i1,ind),
      ta_tense(i1,pres)],
     [leq(l2,h2),                       % Scope
      leq(l1,h1),
      ccom_plug(h2,l2),
      ccom_plug(h1,l1)],
     [pros_mood(l5,decl)],              % Prosody
     [sem_group(l2,[l4]),               % Groupings
```

```
sem_group(l1,[l3])] ).
```

CLASS NAME:	**ambig_tv**
SUFFIX:	`atv`
PART OF SPEECH:	
REPRESENTATION:	`[R(L,I),s_arg(L,I,I1),o_arg(L,I,I2)]`
SEMDB:	`predname R`
COMMENTS:	`s_arg` is corresponding to the subject argument, `o_arg` to the object argument.
EXAMPLE:	Ich *ziehe* den Termin *vor*

```
vit( segment_description('atv','ich ziehe den termin vor'),
     [vorziehen_move_prefer(l3,i1),     % Semantics
      termin(l5,i2),
      decl(l4,h1),
      o_arg(l3,i1,i2),
      s_arg(l3,i1,i3),
      pron(l8,i3),
      def(l7,i2,l1)],
     l4,                                % Main Label
     [s_sort(i1,;(move_sit,mental_sit)),% Sorts
      s_sort(i2,time_sit_poly),
      s_sort(i3,&(human,person))],
     [prontype(i3,sp,std)],             % Discourse
     [num(i3,sg),                       % Syntax
      pers(i3,1),
      gend(i2,masc),
      num(i2,sg),
      pers(i2,3),
      cas(i2,acc),
      cas(i3,nom)],
     [ta_mood(i1,ind),                  % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l2,h1),                       % Scope
      ccom_plug(h1,l2)],
     [pros_mood(l4,decl)],              % Prosody
     [sem_group(l2,[l3]),               % Groupings
      sem_group(l1,[l5])] ).
```

Class Name:	**ambig_iv**
Suffix:	`aiv`
Part of Speech:	
Representation:	`[R(L,I),s_arg(L,I,I1)]`
SemDB:	`predname R`
Comments:	`s_arg` is corresponding to the subject argument.
Example:	Er *geht*

```
vit( segment_description('aiv','er geht'),
     [gehen_move_passen(l2,i1),          % Semantics
      decl(l3,h1),
      s_arg(l2,i1,i2),
      pron(l5,i2)],
     l3,                                 % Main Label
     [s_sort(i1,;(move_sit,mental_sit))],% Sorts
     [prontype(i2,third,std)],           % Discourse
     [gend(i2,masc),                     % Syntax
      num(i2,sg),
      pers(i2,3),
      cas(i2,nom)],
     [ta_mood(i1,ind),                   % Tense and Aspect
      ta_tense(i1,pres)],
     [leq(l1,h1),                        % Scope
      ccom_plug(h1,l1)],
     [pros_mood(l3,decl)],               % Prosody
     [sem_group(l1,[l2])] ).             % Groupings
```

Class Name:	**complementizer**
Suffix:	`comp`
Part of Speech:	**KOUS**
Representation:	`R(L,H)`
SemDB:	-
Comments:	See pcv.
Example:	Ich möchte wissen, *ob* wir uns am Montag treffen

```
vit( segment_description('comp','ich moechte wissen ob wir uns am montag treffen'),
     [moegen(l9,i5),                          % Semantics
      wissen(l8,i6),
      ob(l11,h1),
      treffen(l5,i1),
      decl(l10,h3),
      arg2(l9,i5,i7),
      arg3(l9,i5,h2),
      arg2(l8,i6,i7),
      arg3(l8,i6,l11),
      an(l7,i1,i2),
      dofw(l6,i2,mon),
      arg1(l5,i1,i4),
      arg3(l5,i1,i3),
      pron(l18,i7),
      pron(l15,i4),
      pron(l14,i3),
      def(l13,i2,l1)],
     l10,                                     % Main Label
     [s_sort(i1,meeting_sit),                 % Sorts
      s_sort(i2,time),
      s_sort(i3,&(human,person)),
      s_sort(i4,&(human,person)),
      s_sort(i5,mental_sit),
      s_sort(i6,mental_sit),
      s_sort(i7,&(human,person))],
     [dir(l7,no),                             % Discourse
      prontype(i7,sp,std),
      prontype(i4,sp_he,std),
      prontype(i3,sp_he,refl_std)],
     [num(i7,sg),                             % Syntax
      pers(i7,1),
      num(i4,pl),
      pers(i4,1),
```

```
 num(i3,pl),
 pers(i3,1),
 cas(i3,acc),
 cas(i4,nom),
 cas(i7,nom)],
[ta_tense(i6,infin),                  % Tense and Aspect
 ta_mood(i5,ind),
 ta_tense(i5,pres),
 ta_mood(i1,ind),
 ta_tense(i1,pres)],
[leq(l4,h3),                          % Scope
 leq(l4,h2),
 leq(l3,h3),
 leq(l2,h1),
 ccom_plug(h3,l3),
 ccom_plug(h2,l4),
 ccom_plug(h1,l2)],
[pros_mood(l10,decl)],                % Prosody
[sem_group(l2,[l7,l5]),               % Groupings
 sem_group(l4,[l8]),
 sem_group(l3,[l9]),
 sem_group(l1,[l6])] ).
```

CLASS NAME:	**greeting**
SUFFIX:	gr
PART OF SPEECH:	**ITJ**
REPRESENTATION:	[greeting(L,D)]
SEMDB:	designator D
COMMENTS:	L is the main label of the VIT.
EXAMPLE:	*hallo*

```
vit( segment_description('gr',hallo),
     [greeting(l1,hallo)],              % Semantics
     l1,                                % Main Label
     [],                                % Sorts
     [],                                % Discourse
     [],                                % Syntax
     [],                                % Tense and Aspect
     [],                                % Scope
     [pros_mood(l1,decl)],              % Prosody
     [] ).                              % Groupings
```

CLASS NAME:	**interjection**
SUFFIX:	`int`
PART OF SPEECH:	**ITJ,PTKANT**
REPRESENTATION:	`[excl(L,D)]`
SEMDB:	`designator D`
COMMENTS:	`L` is the main label of the VIT.
EXAMPLE:	*jawohl*

```
vit( segment_description('int',jawohl),
     [excl(l1,jawohl)],                 % Semantics
     l1,                                % Main Label
     [],                                % Sorts
     [],                                % Discourse
     [],                                % Syntax
     [],                                % Tense and Aspect
     [],                                % Scope
     [pros_mood(l1,decl)],              % Prosody
     [] ).                              % Groupings
```

References

Dorna, M.: 1996, *The ADT Package for the Verbmobil Interface Term*, Technical Report Verbmobil Report 104, Universität Stuttgart

Heinecke, J. and Worm, K.: 1996, *The Verbmobil Semantic Database*, Technical Report Verbmobil Report 107, Humboldt Universität zu Berlin and Universität des Saarlandes

Index