The Semantic Representation of Locatives in Machine Translation

Thesis presented for the degree of Cand. Philol.

Department of Linguistics

University of Oslo

submitted by:

Fredrik Jørgensen

fredrik.jorgensen@sensewave.com

November 10, 2004

Contents

Acknowledgments

First and foremost, I wish to thank my supervisor, Jan Tore Lønning. The time and effort he has put down to guide me through the writing of this thesis, is far beyond what one could expect or hope for. In the many long and enlightening sessions in his office, I've enjoyed both his immense academic capacity, and his equally well developed sense of humor.

I would also like to thank Stephan Oepen. Acting as a co-supervisor in the last stage of the writing process, I have benefited greatly from his extensive knowledge of HPSG, the LKB system, MRS and about everything else that has to do with computational linguistics.

I would like to thank everyone at LOGON, for letting me be a part of the project and attend the conferences in Oslo and Os. I would in particular like to thank Dan Flickinger for finding time to discuss prepositional phrases with me, and Lars Hellan and Dorothee Beermann for inviting me to the workshop on directionals in Trondheim, which I found very profiting.

I would also like to thank Lilja, Iselin, Bjørnar and all other inhabitants of 'hundremeterskogen', for making my years on SLI a memorable time of my life.

I wish to thank my brother, Christian, for proof-reading and useful comments, and my parents and mother-in-law for helping out with baby-sitting and other practical things during the most stressful periods.

And finally, I would like to thank my beautiful little family, Karen, Oskar and Vilja, for no special reason.

Chapter 1

Introduction

The study of how space and spatial relations are expressed in natural language is a fascinating topic. Space is a very concrete subject, and obviously central to both the experiences and the language of human beings. But despite this, it is rather difficult to model the relationship between space and language. Natural language expressions of spatial relations are realized differently across the human languages, although the spatial entities we describe are the same.

The study of the meaning of natural language expressions, known as semantics, is of value in trying to formalize what we mean to express when we use language. But it is also of value in contrastive analyses, when trying define how a concept is described across different languages.

In natural language, we find ways of describing both regions of space and motion in space. We can locate objects, and we can refer to locations. I will in this thesis describe a novel approach to understanding how space is expressed in language, as presented in the article 'On the Semantics of Locatives', by Marcus Kracht (2002). The novelty of this article lies in its attempt to account for the different spatial expressions in natural language in a uniform manner, by claiming that spatial expressions (locatives) consist of two separate layers; one layer referring to a region in space, and the other layer expressing motion or the location of an object with respect to this region.

The central topic of the thesis will be: How can the central features of the theory on locatives in Kracht (2002) be implemented in a specific computational framework, namely the 'Linguistic Knowledge Building' (LKB) system?

The language for semantic representation in the implementation I will present is an underspecified meta-level language describing semantic structures, known as Minimal Recursion Semantics (MRS). The semantic framework found in Kracht (2002), however, is lambda calculus, as in the Montogovian tradition. We then face the following challenge: How can the lambda calculus expressions in Kracht (2002) be given a representation in the meta-level language MRS? And what will the relationship between the meta-level (MRS) and object-language (lambda calculus) representations be?

Furthermore, I wish to explore the consequences of applying this approach to locatives on Norwegian spatial prepositions: Does the theory in Kracht (2002) adequately describe the semantic properties of Norwegian locatives, and can the concepts of the theory be applied in generalizing over the class of Norwegian locatives?

Apart from the monolingual perspective, I also wish to explore the contrasts between Norwegian and English locatives, and the translation of locatives from Norwegian to English. To what degree can the model of Norwegian locatives developed in this thesis contribute to making accurate predictions regarding the translation of locative expressions?

Overview

Chapter 2 contains a presentation of prepositions as a part-of-speech category, and locatives as a subclass of this category.

In Chapter 3, I will present an approach to the truth-conditional semantics of locatives, as given in Kracht (2002). This chapter describes the mathematical entities we can use to model the meaning of locatives, and how these mathematical entities interact with other mathematical entities. I will also present an alternative theory regarding how three-dimensional space can be modeled, taking vectors to be the basic mathematical entity rather than points in space. Chapter 3 is meant to describe the deeper semantic structures of locatives, whereas the semantic predicates we use in the remainder of the thesis can be viewed as abstractions over these structures. For readers not interested in the formal semantics of locatives, understanding the details of this chapter is not essential for the reading of the rest of the thesis.

In Chapter 4, I describe machine translation system architectures in general, and one particular instance of such a system, the LOGON project. I will describe the central properties of the theories employed in the LOGON project, on which our implementation also will be based. I will describe Head-driven Phrase Structure Grammar (HPSG), as the syntactical framework for the grammar fragment I develop in the thesis, and Minimal Recursion Semantics (MRS) as the framework for the semantic representations.

Chapter 5 is the junction of the preceding chapters. In this chapter, I describe the implementation of the semantic theory from Chapter 3 in the frameworks described in Chapter 4. We face the challenges that arise, both in re-formulating the lambda calculus expressions from Chapter 3 in the meta-level language MRS, and assigning syntactic structure to the contexts Norwegian locative expressions appear in.

In Chapter 6, I implement an important principle, the Emptiness Principle from Kracht (2002), which enables locative prepositional phrases to be selected as different semantic entities. I argue that the possibilities that arise from this interpretation of locatives gives us more accurate semantic descriptions of the relationship between certain types of verbs and their locative complements.

In Chapter 7, I apply the model we have constructed for locatives, to classify Norwegian locatives. We also see the impact this model has on the translation of locatives from Norwegian to English.

Appendix A contains the resources for the grammar fragment I have developed. I encourage the reader familiar with LKB and the Matrix to examine the implementation, as not all aspects of the implementation are discussed in the present thesis.

The accompanying CD-ROM contains the grammar fragment developed in the thesis, as well as a Windows-version of the LKB system. I also encourage the reader to try the grammar fragment in the LKB. This will only be enclosed the copies of the thesis submitted as a partial fulfillment of the Cand. Philol. degree at the University of Oslo. For information on how to acquire the LKB system, visit `http://www.delph-in.net/lkb/`. To receive a copy of my grammar fragment, please contact me.

Chapter 2

Background

In this chapter, I will present prepositions in general, and locatives as a special type of prepositions. Furthermore, I will describe the difference between static and directional locatives, and look at two theories which try to capture the relation between static and directional locatives, namely the theories of Jackendoff (1990) and Kracht (2002).

2.1 Prepositions

The class of prepositions is considered a part of speech category with the following properties. Morphologically, prepositions take no inflection. The class of prepositions is a closed class, along with auxiliaries, conjunctions, determiners and pronouns. These classes consist of finite sets of words which can be exhaustively listed, and they do not admit new members. It is difficult to distinguish prepositions by any formal features, but they typically syntactically precede a nominal phrase (NP), hence the name *pre*position. Most prepositions consist of one word, called *simple* prepositions. But we also find *complex prepositions*, consisting of two or more words, e.g. *in front of* and *to the right of*.

It is normal to divide prepositions into two classes, based on their semantic type. *Semantically full* prepositions, also called lexical prepositions, refer to something external to language, e.g. a spatial relationship between two objects, as seen in (1). Semantically full prepositions are heads of prepositional phrases (PPs), and usually followed by a prepositional complement. They may also be preceded within the PP by a an adjugated phrase (typically an adjective phrase or a nominal phrase with a measure noun). *Semantically empty* prepositions, sometimes referred to as *selected prepositions*, are prepositions which are strongly bound by a predicate, such that they cannot be substituted by semantically close prepositions. These prepositions are by some approaches considered to be case markers, lexically predicted by the phrasal head, and with no semantic contribution to the sentence they appear in, as *på* in (2).

(1) *Semantically full preposition:*

Musa løp under bordet
Mouse.DEF ran under table.DEF

'The mouse ran under the table'

(2) *Semantically empty preposition:*

Petter stoler på bruktbilselgeren
Peter trusts on car-seller.DEF

'Peter trusts the car salesman'

PPs occur in several positions. According to Faarlund et al. (1997, p. 411), prepositions may be complements of verbs (3a) or prepositions (3b), or adjuncts to either verb phrases (3c), noun phrases (3d) or adjective phrases (3e)[1] .

(3) a. Andreas legger boken på bordet
Andrew lays book.DEF on table.DEF
'Andrew lays the book on the table.'

b. Han hørte alt fra under bordet
He heard everything from under table.DEF
'He heard everything from under the table.'

c. Musa danser på bordet
Mouse.DEF dances on table.DEF
'The mouse dances on the table.'

d. Boken på bordet er tung
Book.DEF on table.DEF is heavy
'The book on the table is heavy.'

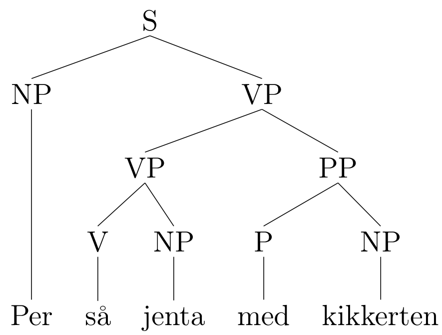
e. Elevene er svake i geografi
Students.DEF are weak in geography
'The students are weak in geography'

Furthermore, we find structural ambiguities, the well known problem of PP attachment. These ambiguities typically arise when deciding whether the PP is in adverbial or adnominal position, as shown in (4).

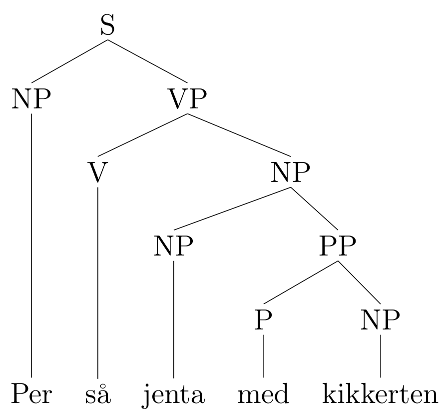
[1]It may be argued that the PP in this sentence is a complement of the adjective, and not an adjunct.

(4) a. Per så jenta med kikkerten
Peter saw girl.DEF with binoculars.DEF

b. *PP attached to the VP:*



c. *PP attached to the NP:*



In addition to the structural ambiguities that arise, prepositions are notoriously polysemous, and the possible interpretations of a single sentence may be quite a few. Just consider the preposition *med* in (5)

(5) a. *NP adjunct, expressing an object quality:*

Johannes spiser pizza med pepperoni.
John eats pizza with pepperoni.

'John eats pizza with pepperoni on it.'

b. *VP adjunct, expressing the instrument of the event:*

Johannes spiser pizza med kniv og gaffel.
John eats pizza with knife and fork.

'John eats pizza using knife and fork.'

c. *VP adjunct, expressing accompaniment:*
Johannes spiser pizza med sjefen sin.
John eats pizza with boss his.
'John eats pizza (together) with his boss.'

Prepositions have a wide range of semantics, illustrated by the following classification of prepositions proposed by the EC-sponsored EAGLES Lexicon Interest Group[2]:

1. Modifiers of predicative heads, i.e. verbs and predicative nouns, behave like adverbs and are divided into *Place-Position, Place-Goal, Place-Origin, Place-Path, Cause, Aim, Concern, Accompaniment, Instrument, Benefactive, Substitute, Manner, Function, Measure, Comparison* and *Time*.

2. Modifiers of non-predicative heads, i.e. non-predicative nouns, behave like adjectives and are *Place (further subdivided in Position, Goal, Origin and Path), Whole, Stuff, Quality (further subdivided in Inherent and State), Concern, Aim, Specification, Function* and *Measure*.

2.2 Locatives

Prepositional phrases expressing location in space, called *place* in the above categorization, are often referred to as *locative*[3]prepositions. A *locative* can be said to locate an object, often called *figure*, in a relation to one or several objects, the *ground*. The natural meaning of a locative is a relation in space.

As stated earlier, prepositions typically come before an NP, i.e. they take nominal complements. Prepositions also take PP complements, as we shall see later. Both of these are locative prepositions. But are words which take no complements, but have a locative meaning, prepositions? Or put differently; is the set of locative words a subset of prepositions?

According to *Norsk Referansegrammatikk* (Faarlund et al., 1997), the 'Norwegian Reference Grammar', they are. Faarlund et al. consider words like *inn* ('in') and *ut* ('out'), formerly considered to be adverbs, to be intransitive prepositions. I will in the thesis adopt this categorization, as we will treat locative words as a subclass of prepositions, with variation with respect

[2]http://www.ilc.cnr.it/EAGLES96/rep2/node13.html

[3]The term *locative preposition* are sometimes used in the meaning *static* preposition. We use the term here to include all spatial prepositions, i.e. directional *and* static prepositions.

to transitivity. We then find several similarities between verbs and prepositions. We have intransitive and transitive prepositions, the latter taking either nominal or prepositional complements, as we have for verbs. Prepositions also have, as verbs, case marking properties, as the complement of both the verb and preposition is marked for the accusative case in the sentences in (6).

(6) a. Jon slo ham
 'John hit him'
 b. Jon satt bak ham
 'John sat behind him'

Locative prepositions are divided into static and directional prepositions, with different semantics and different syntactic distribution. This is also reflected in the EAGLES classification above. For instance, directional locatives do not combine well with NPs (7a), unless the NP denotes certain types of events (7b) or path-like objects (7c). Directional locatives also combine well with motion verbs (7d), but not so well with static verbs (7e)[4].

(7) a. ?Mannen opp til fjellene.
 ?Man.DEF up to mountains.DEF.
 ?'The man up to the mountains.'
 b. Veien opp til fjellene.
 Road.DEF up to mountains.DEF.
 'The road up to the mountains.'
 c. Reisen opp til fjellene.
 ?Man.DEF up to mountains.DEF.
 ?'The man up to the mountains.'
 d. Mannen løp opp til fjellene.
 Man.DEF run.PST up to mountains.DEF.
 'The man ran up to the mountains.'
 e. ?Mannen satt opp til fjellene.
 ?Man.DEF sit.PST up to mountains.DEF.
 ?'The man sat up to the mountains.'

Directional prepositions can be divided further, based on their semantic properties. Directionals can express goal of motion, origin or source of motion,

[4]I find it difficult to tell whether the problematic sentences below really are ungrammatical, uninterpretable, or just rarely encountered. It might be possible to contextualize sufficiently to interpret some of them. For instance, sentence (7e) may not be so hard to contextualize, whereas (7a) seems much harder to imagine being uttered or successfully interpreted.

and parts of the path on which motion is taking place. There seems to be no general consensus in the literature on how to classify path-denoting prepositions. I have simply presented them as a single group in this hierarchy.

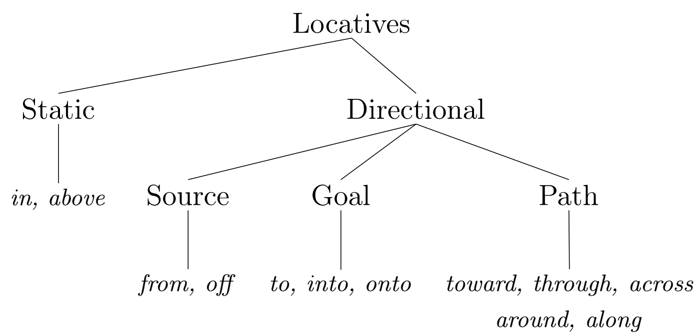


Figure 2.1: Hierarchy of locatives

2.3 Static and Directional Locatives

Apart from dividing locatives into two separate categories, there also seems to be a systematic relationship between the static and directional locatives. In this section, we take a look at two proposals for expressing this relationship, and see how they differ in some central respects.

2.3.1 Jackendoff (1987, 1990) and Conceptual Semantics

Ray Jackendoff's "Conceptual Semantics" is a decompositional theory of meaning, inspired by and borrowing notions from Generative Syntax. The motivation for the theory is thus similar to the argument of creativity we find in Generative Syntax: Given the indefinitely large variety of objects to be represented by lexical concepts, how can a finite brain encode all these objects? Jackendoff argues that these objects are not encoded as a list of those objects previously encountered, but are encoded as some sort of finite schema that can be compared to mental representations of arbitrary new objects to produce a judgment of conformance or non-conformance. This view is formulated in the *mentalist postulate*:

> "Meaning in natural language is an information structure that is mentally encoded by human beings."
> (Jackendoff, 1987, p. 122)

Thus, the goal of Conceptual Semantics is "the characterization of the mental resources that make possible human knowledge and experience of the world" (Jackendoff, 1990, p. 8). This characterization or grammar of concepts consists of innate formation rules, based on a repertoire of major conceptual categories; the "semantic parts of speech". These can be elaborated into a function-argument organization, some of the most important ones listed for the spatial domain listed in (8).

(8) a. [PLACE] $\rightarrow$ $[_{Place}$ PLACE-FUNCTION([THING])]

b. [PATH] $\rightarrow$ $\left[_{Path} \left\{ \begin{array}{l} \text{TO} \\ \text{FROM} \\ \text{TOWARD} \\ \text{AWAY-FROM} \\ \text{VIA} \end{array} \right\} \left(\left[\left\{ \begin{array}{l} \text{THING} \\ \text{PLACE} \end{array} \right\} \right] \right) \right]$

c. [EVENT] $\rightarrow$ $\left\{ \begin{array}{l} [_{Event} \text{ GO([THING],[PATH])]} \\ [_{Event} \text{ STAY([THING],[PLACE])]} \end{array} \right\}$

d. [STATE] $\rightarrow$ $\left\{ \begin{array}{l} [_{State} \text{ BE([THING],[PLACE])]} \\ [_{State} \text{ ORIENT([THING],[PATH])]} \\ [_{State} \text{ EXT([THING],[PATH])]} \end{array} \right\}$

e. [EVENT] $\rightarrow$ $\left[_{Event}\text{CAUSE} \left(\left[\left\{ \begin{array}{l} \text{THING} \\ \text{EVENT} \end{array} \right\} \right], \text{[EVENT]} \right) \right]$

Conceptual Semantics also borrow ideas from X-bar syntax, especially the idea of cross-categorical generalizations, here across the major ontological categories (or "the semantic parts of speech"). Jackendoff lists six points of cross-categorical similarities (e.g. the correspondence between the "syntactic and semantic parts of speech", deictic use etc.). He shows how concepts of spatial location and motion can be generalized to other semantic fields, e.g. possession and properties. Jackendoff also shows how aggregation and boundedness can be generalized to apply for events and objects in a similar X-bar fashion.

Locatives in Conceptual Semantics

As we saw in (8), function-argument pairs play a crucial role in Jackendoff's theory. All major syntactic constituents are mapped into function-argument structures, as in (9).

(9) a. *Syntactic structure*

$[_{S}[_{NP}$ John $][_{VP}$ ran $[_{PP}$ into $[_{NP}$ the room $]]]]$

b. *Conceptual structure*

$[_{Event}$ GO $([_{Thing}$ JOHN $],$
$[_{Path}$ TO $([_{Place}$ IN $([_{Thing}$ ROOM $])])])]$

These mappings are achieved through lexical entries containing Lexical Conceptual Structures, LCSs. The correspondence between the syntactic and conceptual constituents is expressed through co-indexation. Particularly interesting in our case is the treatment of static and directional locatives, and of motion verbs. Static locatives have lexical entries with a Place function in the LCS, as in (10).

$$(10)\quad \begin{bmatrix} \text{in} \\ \text{P} \\ \underline{\quad\;}\ \text{NP}_j \\ [_{Place}\ \text{IN}\ (\ [_{Thing}\qquad]_j\ )\ ] \end{bmatrix}$$

Thus, static locatives can be said to have one layer, opposed to directional locatives, which have two layers; a Path function as the outer function, and a Place function as the inner function, as in (11).

$$(11)\quad \begin{bmatrix} \text{into} \\ \text{P} \\ \underline{\quad\;}\ \text{NP}_j \\ [_{Path}\ (\ [_{Place}\ \text{IN}\ (\ [_{Thing}\qquad]_j\ )\ ]\ )\ ] \end{bmatrix}$$

Furthermore, Jackendoff (1990) treats directional locatives as arguments to motion verbs, such that *run*, in (12), subcategorizes for an optional directional locative argument, represented in angle brackets in the LCS. Through a mechanism Jackendoff calls Argument Fusion, the LCS of the prepositional complement is substituted for the part of the verb's LCS it is coindexed with. In this case, the PP's LCS will substitute $[_{Path}\quad]_j$ in (12).

$$(12)\quad \begin{bmatrix} \text{run} \\ \text{V} \\ \underline{\qquad}\langle \text{PP}_j \rangle \\ [_{Event}\ \text{GO}\ (\ [_{Thing}\quad]_i,\ \ [_{Path}\quad]_j\ )\ ] \end{bmatrix}$$

One advantage with this approach to locatives is that the correct argument is predicated to be traversing the trajectory denoted by the Path function, as in (12), where the moving entity is coindexed i with the subject.

Jackendoff claims that Place functions denote regions:

> "...a conceptual constituent belonging to the category Place can be elaborated as a Place-function plus an argument that belongs to the category Thing. The argument serves as a spatial reference point, in terms of which the Place-function defines region."
> (Jackendoff, 1990, p. 44)

The Path function similarly elaborates a trajectory through the five Path functions, which map a reference Thing or Place into a related trajectory. This interpretation of locatives, as I understand it, forces Jackendoff to treat locatives as arguments of verbs. This works perfectly well for locatives when they in fact are arguments of verbs, but when they act as predicative or modifying, we need a different interpretation of locatives.

Discussion of locatives in Conceptual Semantics

Bierwisch (1988) notes two different positions with respect to the denotations of local PPs. First, we have *the referential interpretation*, advocated by Jackendoff, where local PPs denote regions, just like NPs denote things. And then we have *the modificational interpretation*, where the local PPs denote properties of being located at a certain place. Bierwisch (1988) notes further that

> "The referential interpretation seems to be appropriate for PPs in argument position, as e.g. *Hans liegt im Bett* can plausibly be said to as express a relation between *Hans* and a place denoted by *im Bett*. It is difficult to see, however, how on this account PPs can serve as modifiers or predicatives - unless a place is construed as a property, but that would violate the gist of the referential interpretation. The modificational interpretation, on the other hand, concerns itself with PPs as adjuncts, but seems to be in trouble with PPs in argument position. From this, one might be tempted to draw the conclusion that both interpretations are partially right and that they both are needed."
> (Bierwisch, 1988, p. 8)

Wunderlich (1991) assumes the opposite position of Jackendoff, and argues that locatives are one-place predicates.

> "The predicative role of a PP has to be considered the fundamental one. All other uses of PP (as a modifier or as an argument)

> must respect its predicative nature. [...] The PP does not refer to this region, as has been assumed by Jackendoff and his followers. Such a conception would immediately lead to a non-compositional semantics."
> (Wunderlich, 1991, p. 600)

Jackendoff uses locatives in subject position to argue for the referential position, as seen in (13a). But Wunderlich regards a sentence like (13a) only to be an abbreviation for (13b), and that the PP actually is an NP.

(13) a. Under the tree is a good place to sleep.
     b. (The place) under the tree is a good place to sleep. (Wunderlich, 1991, p. 620)

As we see, there seem to be problems with the denotation of locatives. Should the denotation be regions of space, or rather properties of individuals (and events)? While Bierwisch (1988) only note these problems, Wunderlich (1991) argues for the modificational position. Let us now turn to a theory which tries to meet both these challenges semantically, but also claim to be cross-linguistically valid from a syntactic point of view.

2.3.2 Kracht (2002) and The Semantics of Locatives

Marcus Kracht (2002) follows Jackendoff to some extent in the division of spatial functions into Path and Place functions, but the two positions differ on some crucial points. Kracht's main proposal is that locatives consist of two layers. The first layer, *configuration*, defines a location or region, similar to Jackendoff's Place function. The second layer, *mode* defines a movement (i.e. a set of motion events) with respect to this configuration, and not a trajectory, as Jackendoff's Path function. The elements defining these layers are called *localisers* (L) and *modalisers* (M).

> "From a syntactical and semantical point of view a locative expression is structured as follows
>
> $[M[L\ DP^{5}]]$
>
> where M is a modaliser (specifying the mode), L a localiser(specifying the configuration) and DP a determiner phrase."
> Kracht (2002, p. 159)

[5] I will consequently refer to this type of phrases as NPs.

The layers are manifest as either adpositions or case, and the typical case is that $M{+}L$ forms a unit.

Kracht differs from Jackendoff in that, basically, all locatives are adjuncts. One consequence of treating all locative PPs as modificational adjuncts, is that the entity moving (e.g. the subject for the verb *run*) is not directly accessible to the locative PP in the compositional process. This is not a problem for Jackendoff, where the verb expresses a relation between a moving entity and a trajectory, but Kracht needs some additional machinery for predicting which entity is located by the locative (cf. Section 3.3).

When locatives are treated as adjuncts, it is the mode that expresses the relation between the region on the one hand and either the event or the moving entity in the event (adverbial adjuncts) or the entity (adnominal adjuncts) on the other. In analyzing modalisers, Kracht identifies five modes for adverbial adjuncts[6] and one for adnominal adjuncts (a-loc). These are listed in Table 2.1 (with examples and the corresponding Jackendoff Path functions).

Kracht	*Jackendoff*	*Examples*
static	$[_{Place}\quad]$	i ('in'), under ('under'), bak ('behind')
coinitial	$[_{Path}$ FROM ($[_{Place}\quad]$)]	fra ('from'), ut av ('out of')
cofinal	$[_{Path}$ TO ($[_{Place}\quad]$)]	til ('to'), inn i ('into')
transitory	$[_{Path}$ VIA ($[_{Place}\quad]$)]	gjennom ('through'), forbi ('past')
approximative	$[_{Path}$ TOWARD ($[_{Place}\quad]$)]	mot ('toward')
(recessive)	$[_{Path}$ AWAY-FROM ($[_{Place}\quad]$)]	vekk fra ('away from')
a-loc	$[_{Place}\quad]$	i ('in'), under ('under'), bak ('behind')

Table 2.1: Kracht's modes

All the modes, except the static, are defined through identifying the mover of the event. Verbs denote event types, and language has a small set of semantic roles, among them the role of a *mover*. Kracht assumes a function to pick out the mover of the event types of verbs of motion, and to predicate the motion of this entity. This will be explained in Chapter 3.

Locative PPs normally figure as MPs, and are intersective modifiers and adjuncts to the VP, according to Kracht. But Kracht observes that verbs can subcategorize for LPs, and that LPs also occur in a few other contexts. Examples of these are given in (14).

[6] It is actually *minimally* five modes. Kracht suggests there may be evidence for two additional modes, *recessive* mode corresponding to Jackendoff's AWAY-FROM, and *terminative* mode expressing motion that terminates right next to the region defined by the LP.

(14) a. The mouse came [$_{MP}$ from [$_{LP}$ under the bed]]
b. Hans is lying [$_{LP}$ under the bed]
c. [$_{LP}$ Under the bed] is a good place to hide

Under this assumption, Kracht can provide an explanation for the cases the modificational interpretation had problems with, namely argument positions (noted by Bierwisch (1988)) and sentence initial use (noted by Wunderlich (1991)). These observations seem to explain the referential properties of some locatives, and thus, the disagreement described by Bierwisch (1988) may in fact be the result of mistakingly treating MPs and LPs as belonging to the same category, according to Kracht's analysis of locatives.

2.3.3 Discussion

As we have seen, the division of locatives into two layers may explain both the predicative and referential properties of locative PPs. Treating locatives as adjuncts does not necessarily follow from this division, however. One could still choose to treat directional locatives as arguments to verbs of motion. Choosing the argument approach, one would have to (i) refine the classification of verbs into motion and static verbs, and (ii) enrich the argument structure of motion verbs to accommodate (optional) directional PP complements. The adjunct approach, on the other hand, would require the same refinement of classification of verbs (i.e. identifying the verbs which have an argument with the semantic role of *mover*). But this approach would not call for an augmentation of the argument structure.

As described above, Kracht identifies different modes for static locatives in adverbial and adnominal position. This is of both semantic and syntactic significance, Kracht claims, as locatives cannot be used adnominally in Finnish and Hungarian. In these languages, one would use a combination with 'to be' instead. But directional locatives also modify some NPs, e.g. path-denoting, event denoting and public transport denoting NPs.

(15) a. Motorveien til/fra/gjennom Lillehammer.
'The highway to/from/through Lillehammer.'
b. Turen til/fra/gjennom Lillehammer.
'The journey to/from/through Lillehammer.'
c. Ekspressbussen til/fra/gjennom Lillehammer.
'The express bus to/from/through Lillehammer.'

What should the denotation of these locatives be? They seem to define a path-like region which then in turn defines the trajectory or extension of the main NP. This could mean that directional locatives must be given a second interpretation, as NP adjuncts. We may then end up with two interpretations of all locatives, depending on their site of adjugation. This may in turn prove to be more unprincipled than the region and trajectory interpretations of Jackendoff. On the other hand, Jackendoff would require these locatives to be complements of the NPs, or define a separate rule for this type of adjugation.

Still, the two-layering of locatives proposed by Kracht seem to explain some of the 'slippery' nature of locatives, and I will try to show that it explains some of the differences between Norwegian and English in expressing spatial relations.

2.4 Summary

In this chapter, I have described the class of prepositions, some of its functions in language, and proceeded to describe one of its subclasses, locatives. We saw how Jackendoff (1987, 1990) described the relationship between static and directional locatives. I have also discussed two different semantic properties of locatives, i.e. their referential and modificational properties. Finally, we looked briefly at how Kracht (2002) overcame the differences between these two positions, by giving locatives both referential properties (for LPs) and modificational properties (for MPs), depending on the context.

In the next chapter, we will look more closely at the semantic theory described in Kracht (2002). We will start out by describing a formal framework for semantic representation: truth-conditional semantics. From this, we will proceed to investigate (i) two theories trying to formalize the spatial relations expressed by locatives, and (ii) how locatives describe motion. Finally, we will see in detail how the semantic representation for a sentence containing a locative PP is composed.

Chapter 3

The Semantics of Locatives

3.1 Truth-Conditional Semantics

In truth-conditional semantics, the meaning of natural language expressions is determined by their influence on the truth conditions of the sentences they are part of. These truth conditions are defined formally in terms of models. Consider a sentence like (16).

(16) John walks

We first define a model, as in (17). The interpretation of a sentence, shown in (18), is consists of relating the sentence to this model. The interpreted expression is conventionally placed inside double square brackets, and the interpretation is expressed in terms of set theory. The sentence in (16) is true if the object that 'John' refers to is a member of the set of objects that walk. Each expression is interpreted in relation to a model M, hence the superscript on the square brackets.

(17) Model M is defined as follows:
$[\![\text{John}]\!]^M = \text{j}$
$[\![\text{walks}]\!]^M = \text{the set of entities walking} = \text{e.g. } \{\text{j}\}$
$[\![[_S\text{NP VP}]]\!]^M = 1$ (true) *iff* $[\![\text{NP}]\!]^M \in [\![\text{VP}]\!]^M$, otherwise 0 (untrue)

(18) $[\![[_{NP}\text{John}][_{VP}\text{walks}]]\!]^M = 1$
$\Leftrightarrow \quad [\![\text{John}]\!]^M \in [\![\text{walks}]\!]^M$
$\Leftrightarrow \quad \text{j} \in \{\text{j}\}$
This, we see, is the case in model M.

Instead of interpreting into set theoretic expressions, we can translate the expressions into predicate logic expressions by using the lambda operator, which can be seen as a set abstractor. And rather than speaking of sets of entities, we speak of their characteristic functions, such that the characteristic function of a set yields 1 for every entity that is a member of the set, and 0 otherwise. The only rule (at least to start with) is that of functional application, where we in our previous variant needed one rule for each pair of syntactic categories. We then receive the following translation of (16) into a lambda calculus.

(19) $[\text{John}]' = \text{j}$
$[\text{walks}]' = \lambda x[\text{walk}'(x)]$
$[_s\text{NP VP}]' = \text{VP}'(\text{NP}')$

(20) *Translation:*
$[[_{NP}\text{John}][_{VP}\text{walks}]]'$
$= \quad \lambda x[\text{walk}'(x)](\text{j})$
$= \quad \text{walk}'(\text{j})'$

In predicate logic we also have the usual connectives and quantifiers, and add to functional application the rule of intersective modification, expressed through conjunction. Hence, we can translate sentences as the ones in (21).

(21) $[\text{Every man walks}]' \Leftrightarrow \forall x(\text{man}'(x) \rightarrow \text{walks}'(x))$
$[\text{A white horse runs}]' \Leftrightarrow \exists x(\text{horse}'(x) \wedge \text{white}'(x) \wedge \text{runs}'(x))$

We may furthermore extend our calculus with the theory of types, such that each expression is assigned a (basic or complex) type. The theory of types was primarily introduced to get around the paradox of self-membership, i.e. Russell's paradox, which it does by locating entities at distinguished levels, which in turn restricts the compositional process. This entitles us to formulate higher order properties, e.g. *red* (a first order property) being a *color* (a second order property), and predicate modifiers, e.g. *quickly* (predicate modifier) as a certain manner of *walking* (first order predicate).

Types are defined from a set of basic types, and type constructors. For instance, if we have that e and t are basic types, and that $\rightarrow$ is a binary type constructor, then types may be defined as follows:

> **T**, the set of types, are defined as the smallest set such that:
> (i) $e, t \in \mathbf{T}$
> (ii) if $a, b \in \mathbf{T}$, then $(\text{a} \rightarrow \text{b}) \in \mathbf{T}$.
> (Gamut, 1991, p. 79)

For example, if e is the type of entities, individual constants and variables familiar from standard predicate-logic, and t the type of truth-values, we have that expressions may be typed as in (22).

(22)

Kind of expression	*Type*	*Examples*
Individual expression	e	John
One-place first-order predicate	$e \rightarrow t$	man, walks, red
Sentence	t	A man walks
Predicate modifier	$(e \rightarrow t) \rightarrow (e \rightarrow t)$	quickly
One-place second-order predicate	$(e \rightarrow t) \rightarrow t$	color
⋮ (examples from Gamut (1991, p. 81))		

The inventory of our semantics may be extended and refined in numerous ways, by generalizing quantifiers, type lifting proper names, adding modal and intensional operators, adding types for events, time and space points, and so on.

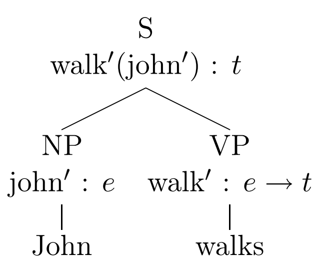
Implicitly assumed above, the input to semantics is syntax, such that the syntactic structures generated from parsing determines the way rules are applied to the constituents. Each part (or subtree) of the syntactic structure (or tree) should be assigned a type and a denotation.

Consider now (19) again. Assigning types to the syntactic constituents, and using a context-free grammar for determining the syntactic structure, we have the following compositional process.

(23) a. *Syntax*
NP → John
VP → walks
S → NP VP

b. *Semantics*
john′ : e
walk′ : $e \rightarrow t$

c.



Giving a locative an interpretation into truth-conditional semantics then means giving each of the syntactic constituents a denotation in the model. NPs can denote entities of type e, as we saw above. But what about LPs and MPs? In the previous chapter we saw arguments for giving LPs a referential interpretation, referring to a region. This means LPs denote regions of some sort. As for MPs, Kracht advocated the view that these should be treated as modificational adjuncts, with intersective semantics. MPs must then have the same type of denotation as the phrase they modify.

3.2 The Semantics of Localiser Phrases

According to Kracht (2002), localiser phrases consist of a localiser and an NP complement, on the form [L NP]. If we assume that NPs denote entities of type e, then the localiser must denote functions from type e to something that is a region-like entity, i.e. L: $e \rightarrow$ REGION. But how is this region defined? As it is a region of space, it must refer to points of space in some way. It is normal to base the interpretation of locatives on a loc' function, that returns for a NP, the region of space it occupies, or it's "eigenplace" (Wunderlich, 1991, p. 597). But other factors apart from the eigenplace of the prepositional complements must be taken into account when determining the denotation of LPs. We will look at two proposals, *parameterized neighborhoods* (Kracht, 2002) and *vector space semantics* (Zwarts and Winter, 2000).

3.2.1 Parameterized Neighborhoods

In this section I will go through Kracht's proposal for the denotation of LPs, along with the arguments for extending the LP denotation from a region to a parameterized neighborhood.

First of all, Kracht assumes an ontology where we find the following basic types: e (objects or entities), i (time points), p (spatial points), v (events) and t (truth-values). Kracht also defines a notation for functions from a type to truth-values.

> "DEFINITION 2. If α is a type, $\alpha^\bullet \stackrel{def}{=} \alpha \rightarrow t$ is a type, the type of **groups** over α."
> (Kracht, 2002, p. 177)

Note that groups are represented in the semantics as sets, so that a set with members of type α will have the type $\alpha^\bullet$. By not distinguishing between a set and its characteristic function the notation is made simpler.

In addition to the basic types above, we have r, regions or path connected subsets of the three-dimensional space, a subtype of $p^\bullet$, and j, intervals, a subtype of $i^\bullet$.

Regions seem like a good place to start when defining the denotation of LPs. Regions are sets of spatial points, of type r. It would then be true of an object x that it is located in a region L if it is contained in this region, i.e.

the set of spatial points defined by $\mathsf{loc}'(x)$ is a subset of the spatial points defined by L.

The notion of a region and containment in this is not sufficient to express contact between two objects, however, as it is normally expressed by the prepositions *på* (Norwegian), *auf* (German) and *on* (English). Consider a bird in a cage on a table. The cage is *on* the table. If it was sufficient for a locative to denote regions, the bird and the cage would be contained in the same region, and hence both would be on the table, no matter where in the cage the birds is. But if we say that LPs denote sets of regions, we can say that the location of the bird, which is a region, must be member of the set of regions in contact with the table, for the sentence *The bird is on the table* to be true when the bird is in a cage on the table.

Thus, we need to speak of sets of regions, or *neighborhoods*, when trying to define the denotation of LPs. Neighborhoods are sets of regions, hence typed $r^\bullet$.

But Kracht gives evidence that these need to be time-dependent, as they change if the landmark is in motion. In German, *vor* governs dative if used statically, and accusative if it is used with locative goal phrases. Consider now the contrast between the dative *dem Auto* in (24a) and the accusative *das Auto* in (24b).

(24) a. Während des ganzen Rennens fuhr Häkkinen vor dem Auto von Schumacher.

Throughout the entire race, Häkkinen was driving in front of Schumacher's car.

b. Am Anfang des Rennens fuhr Häkkinen vor das Auto von Schumacher.

At the beginning of the race, Häkkinen was driving to in front of Schumacher's car.

In (24a), the relationship between the two cars is considered to be static, even though the location of the cars is constantly changing. This means that we cannot speak of location simpliciter, but must compute the local relationship at each moment. Therefore, we arrive at a *parameterized neighborhood*, the denotation of an LP, which is a function from time points to neighborhoods, of type $i \to r^\bullet$.

In addition, we have the loc' function, which is a function from entities to a subset of the four-dimensional time/space, of type $e \to (i \to r)$.

Localisers are in Kracht (2002) based on local relations, which are binary relations on the set of regions. To achieve this compositionally, we need a denotation for the LP constituent of the sentence, and must replace the binary relation by a different construct, based on a restricted form of flexible type assignment. Kracht defines this construct below.

> "There is a bijection from $\wp(N)$ to $N \to \{0,1\}$ defined by $S \mapsto \chi_S$, where $\chi_S(x) = 1$ iff $x \in S$. (χ_s is called the characteristic function of S.) For sets M, N and P, there is a natural bijection β from $M \times N \to P$ to $M \to (N \to P)$ given by $\beta(f) \overset{def}{=} \lambda x \lambda y[f(\langle x, y\rangle)]$. A special instance is $P = \{0,1\}$. ($\beta(f)$ is also called the 'Carrying' [sic.[1]] of the function of f). Let $R \subseteq M \times N$ be a relation from M to N. Then $\chi_R : (M \times N) \to \{0,1\}$, and hence $\beta(\chi_R) : M \to (N \to \{0,1\})$. Now, exchanging $\wp(N)$ for $\{0,1\}$ we obtain, finally a correspondence between relations from M to N and functions from M to $\wp(N)$.
>
> PROPOSITION 4. Let M and N be sets. There is a bijective correspondence between subsets of $M \times N$ and functions from M to $\wp(N)$ given by
>
> $$R \mapsto R^{\spadesuit} \overset{def}{=} \lambda x\{y : \langle x, y\rangle \in R\}."$$
>
> (Kracht, 2002, p. 177-178)

A local relation is then a function from regions to sets of regions. This means, in turn, that we need to map the complement of the localiser, of type e, to its *eigenplace*, through the loc' function. Now given a function N from regions to neighborhoods, this is done as in (25).

(25) $N^{\heartsuit} \overset{def}{=} \lambda x \lambda t N(\mathsf{loc}'(x)(t))$

The following example shows how one can proceed to define these sets of regions.

> "Let r be a region and $\iota(r)$ denote the convex hull of r minus the region r itself. Let $i(r, s)$ iff $s \subseteq \iota(r)$. Using this, the semantics of *in* becomes

[1] 'Currying', after the logician Haskell Curry. The technique of transforming a function taking multiple arguments into a function that takes a single argument (the first of the arguments to the original function) and returns a new function which takes the remainder of the arguments and returns the result

$$\begin{aligned}\mathsf{in}' &= i^{\spadesuit\heartsuit}\\ &= \lambda x\lambda t[i^{\spadesuit}(\mathsf{loc}'(x)(t))]\\ &= \lambda x\lambda t\{r\ :\ i(\mathsf{loc}'(x)(t), r), r \text{ a region}\}\\ &= \lambda x\lambda t\{r\ :\ r \subseteq \iota(\mathsf{loc}'(x)(t)), r \text{ a region}\}\text{''}\end{aligned}$$

Kracht (2002, p. 187-188)

Other localisers are analyzed in a similar way, sometimes depending on a metric and orientation in addition to the set theoretic notions. The general semantics for an arbitrary localiser *local* is defined as in (26).

(26)
$$\begin{aligned}\mathsf{local}' &= l^{\spadesuit\heartsuit}\\ &= \lambda x\lambda t[l^{\spadesuit}(\mathsf{loc}'(x)(t))]\\ &= \lambda x\lambda t\{r\ :\ l(\mathsf{loc}'(x)(t), r), r \text{ a region}\}\end{aligned}$$

Anticipating events just a bit, locatives as adnominal adjuncts are interpreted by use of the function $\mathsf{a} - \mathsf{loc}'$ in (27), which turns a parameterized neighborhood into a parameterized property of individuals (or a set of individuals, if you wish). In e.g. *the mouse under the table*, the sets of individuals denoted by *the mouse* and *under the table* are intersected. The set of individuals denoted by *under the table* in adnominal position is derived in (28).

(27) $\mathsf{a} - \mathsf{loc}' = \lambda L\lambda t\{x\ :\ \mathsf{loc}'(x)(t) \in L(t)\}$

(28) $[\text{under the table}]'$
$$\begin{aligned}&= \mathsf{a} - \mathsf{loc}'(\mathsf{under}'(\mathsf{thetable}'))\\ &= \mathsf{a} - \mathsf{loc}'(\lambda t\{r :\ \mathsf{under}'(r,\ \mathsf{loc}'(\mathsf{thetable}')(t)), r \text{ a region}\})\\ &= \lambda t\{x :\ \mathsf{loc}'(x)(t) \in \{r :\ \mathsf{under}'(r,\ \mathsf{loc}'(\mathsf{thetable}')(t)), r \text{ a region}\}\}\end{aligned}$$

3.2.2 Vector Spaces

In this section, I will describe an alternative to parameterized neighborhoods, namely *vector spaces*. This theory provides an alternative to Kracht's proposal, and has an elegant treatment of PP modification, treating both locative PPs and PP modifiers as denoting the same type of mathematical properties, *vectors*.

Zwarts and Winter (2000) note that locatives may be modified by phrases expressing distance or direction, e.g. a measure phrase (*10 meters*), an adverb (*diagonally*) or a dimensional adjective (*far*), which I group under the term PP modifiers here. These are normally taken to be adjugated to the PP, hence modificational adjuncts. But if the denotation of locatives is taken to be relations between sets of points, these PP modifiers cannot be given a compositional analysis.

Zwarts and Winter (2000) provide the following example. Let us assume that the preposition *outside* denotes the the set of regions disjoint to the ground, as in (29).

(29) $\text{outside}' = \lambda x\{r : r \cap \mathsf{loc}'(x) = \emptyset\}$

If a PP with the locative *outside* is modified by a PP modifier like e.g. *10 meters*, we would run into problems giving this a compositional treatment. This because we would need a function to measure the distance between the figure and the ground. But the ground isn't contained in the set of regions defined in (29), which is the denotation to the phrase the PP modifier is adjugated to. We could solve this problem by reproducing x, the ground, by measuring the distance between the figure and the complement of $\cup\text{outside}'(x)$. But this ad hoc strategy would not work for other locatives. This leads Zwarts and Winter (2000) to the claim that "[a] general compositional treatment of PP modification is not forthcoming if locative prepositions are taken as relations between sets of points." (Zwarts and Winter, 2000)

But what then should the denotation of locatives be, as it would require the phrases *behind the curtain, outside the house, 10 meters* and *diagonally* to all denote the same type of entities? Zwarts (1995) note that the properties of distance and direction, that the PP modifiers modify, are in fact the properties expressed by the mathematical concept of vectors. Vectors are directed line segments between points in space, and assuming vectors as the primitive spatial entity in models of natural language, we can in fact give all the phrases above the same type of denotation: *vectors*.

I will now give an introduction to vector space semantics, as presented in Zwarts and Winter (2000). The underlying idea is that a locative phrase denotes a set of vectors, where a vector is a directed line segment, originating in the PP complement. *Above the table* denotes a set of vectors, which have startpoints in the location of the table and endpoints that define the region we normally think of as *above the table*. In addition to the ground, we also need a notion of verticality, expressed by vertical and horizontal axes related to the ground, as in Fig 3.1.

Zwarts and Winter (2000) defines a *vector space* V, which is identified with the domain of spatial points D_p. Members of this domain are spatial points or *vectors* of type p, and are intuitively viewed such that each vector in V uniquely determines its end-point. Furthermore, we have the domain of *located vectors* of type v. This domain of located vectors, D_v, is defined as the Cartesian product $V \times V$. We now have two types of spatial entities:

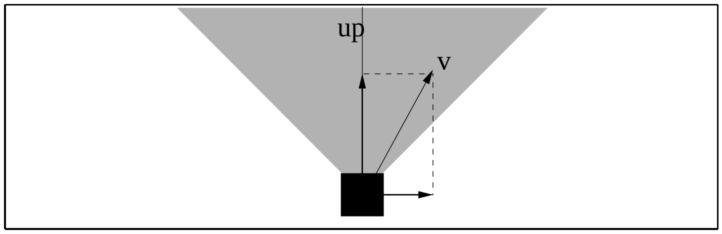
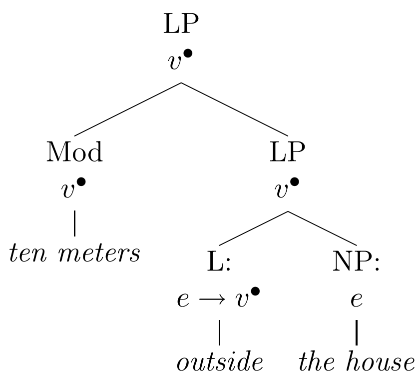


Figure 3.1: Vector representation of *above*

Spatial points or *vectors* of type p, and *located vectors* of type v. In addition, we make the same ontological assumptions as in Kracht (2002), where we find the domains of *objects, events, truth values* and *time points*.

If we consider adnominal locatives, we receive the compositional structure with the logical types in (30), rewriting Zwarts and Winter (2000) slightly and using the terms L and LP from Kracht (2002) and incorporating the loc' function into the localiser semantics rather than introducing this in the compositional rules. We will also adopt the notational variant of Kracht, writing $\alpha^\bullet$ rather than $\alpha \to t$ to denote the groups over type α, letting $v^\bullet$ denote a set of vectors (see Zwarts and Winter (2000), p. 6-7, for a comparison).

(30)



The sets of vectors mentioned above are defined through different functions on vectors. Below, I will go through the definitions from Zwarts and Winter (2000) necessary for interpreting the locative *in*.

First, we need the notion of *start-points* and *end-points* of vectors, *boundaries* of objects, and vectors originating from these boundaries, so-called *boundary vectors*.

- Let $\mathbf{v} \in D_v$ be a vector. If $\mathbf{v} = < u, w >$, then $s\text{-}point(\mathbf{v}) \stackrel{def}{=} w \in V$ is the start-point of $\mathbf{v}$, and $e\text{-}point(\mathbf{v}) \stackrel{def}{=} w + v \in V$ is its end-point.

- Let $\mathbf{v} \in D_v$ be a vector and $A \subseteq D_p$ a set of points. We call $\mathbf{v}$ a boundary vector of A, and denote $boundary(\mathbf{v}, A)$ iff $s\text{-}point(\mathbf{v})$ is in $b(A)$, the boundary of A.
- Let $\mathbf{v} \in D_v$ be a boundary vector of a set of points $A \subseteq D_p$. We say that $\mathbf{v}$ is *a closest vector to* A and denote $closest(\mathbf{v}, A)$ iff for every vector $\mathbf{w} \in D_v$ that is a boundary vector of A s.t. $e\text{-}point(\mathbf{v}) = e\text{-}point(\mathbf{w}) : |\mathbf{v}| \leq |\mathbf{w}|$. In case $e\text{-}point(\mathbf{v}) \in A$ we call $\mathbf{v}$ *internally closest* to A and denote $int(\mathbf{v}, A)$. Otherwise, we call $\mathbf{v}$ externally closest to A and denote $ext(\mathbf{v}, A)$.

The function $|\ |$ in the definition above determines the length of the vector, called a *norm*. This is a function of type $V \rightarrow \mathbf{R}^+$.

We can now define *in* by stating that the denotation of *in* is the set of shortest vectors pointing inward from the boundary of the ground, or the set of *internally closest vectors* to the ground. This is defined in (31).

(31) $\text{in}' = \lambda x\{\mathbf{v} : \ int(\mathbf{v}, x)\}$

A similar definition is given for *outside* in (32).

(32) $\text{outside}' = \lambda x\{\mathbf{v} : \ ext(\mathbf{v}, x)\}$

For measure phrases, we need the definition of a measure set.

> "Definition 2 **(measure set)** A set of located vectors $M \subseteq V \times V$ is called a *measure set* iff for all $v_1, w_1, v_2, w_2 \in V$: if $\langle v_1, w_1 \rangle \in M$ and $|w_1| = |w_2|$ then $\langle v_2, w_2 \rangle \in M$
> (Zwarts and Winter, 2000, p. 6)

Intuitively, a measure set is a set of vectors with the same length, because it is the second vector in a located vector which determines its length (and direction). The first vector only locates the located vector. We can now interpret *10 meters* as a measure set, as in (33).

(33) $[\text{10 meters}]' = \{\mathbf{v} : \ |\mathbf{v}| = 10m\}$

Now, an LP modified by a measure phrase denotes the intersection between the set of vectors denoted by the LP and the measure set denoted by the measure phrase, just as *the blue car*, where *blue* modifies *car*, is the intersection of the set of cars and the set of blue objects.

Apart from overcoming compositional difficulties with respect to PP modifiers, Vector Space Semantics reveal some interesting properties of locatives.

In the spirit of Generalized Quantifier Theory, Zwarts and Winter (2000) define the notions of *point monotonicity* and *vector monotonicity*.

Intuitively, point monotonicity corresponds to truth preservation under enlargement (upward) and diminution (downward) of the ground. According to Zwarts and Winter (2000), the only point-monotone prepositions we find are *in/inside* and *outside*, given in (34).

(34) a. The house is in Paris ⇒ The house is in France *(upward point monotone)*
b. The house is outside France ⇒ The house is outside Paris *(downward point monotone)*

Vector monotonicity corresponds to truth preservation when the figure gets further from or closer to the ground. Zwarts and Winter (2000) note that all simple locative prepositions in natural language are downward vector monotone. A possible counterexample is *far from*, but this preposition is neither a simple preposition nor evidently locative, Zwarts and Winter (2000) claim. With respect to upward vector monotonicity, examples are found in Table 3.1.

Upward vector monotone	Not *upward vector monotone*
in front of	near
behind	on
above, over	at
below, under	inside, in
beside	between
outside	

Table 3.1: Upward vector monotonicity (Zwarts and Winter, 2000, p. 17)

Vector monotonicity is relevant for the modification of locatives, as only upward vector-monotone locatives are modifiable. There are exceptions and border line cases, however. *Inside* is modifiable, but Zwarts and Winter (2000) claim that the ground is conceived as unbounded, and that this partially constitutes the difference between *inside* and *in*. But even *in* can be modified by some adjectives, e.g. in *deep in the forest*.

Zwarts and Winter (2000) conclude that their theory has two main advantages over other proposals. *Compositionally*, we saw that PP modifiers can be treated as intersective modifiers by exploiting the properties of the denotations of PPs (vectors), namely direction and distance. *Ontologically*, we receive a uniform treatment of locatives.

"The *ontological primitives* in the proposed system are taken to

> be standard structures from mathematical theories of space. This step has the advantage of uniformity: all spatial expressions of a certain linguistic category (e.g. locative prepositions) are treated as having the same type of functions over the underlying space ontology. Unlike previous model-theoretic approaches [...] notions like spatial inclusion, betweenness or relative distance are not hard-coded in the ontology."
> (Zwarts and Winter, 2000, p. 33)

Even though spatial inclusion, betweenness or relative distance are not hard-coded in the ontology, other properties of locatives still may have to be hard coded. Gärdenfors (2000) cites personal communication with Annette Herskovits, where Herskovits argues against the position that the meaning of a PP is fully reducible to a region. Herskovits provides the following arguments.

1. Many spatial prepositions, such as "on", "against", "upon" and "on top of" require *contiguity* between objects. This notion is not reducible to a region.
2. The region is *context-dependent* This context-dependence also involves environmental characteristics beyond a frame of reference (Herskovits (1986)).
3. Such a region can be defined; inclusion in it is necessary but not sufficient. Examples:
    - "On": requires support also.
    - "Throughout", "about", "over" (covering): besides being included in the region, the target must be distributed over or extended all over it.
    - "Alongside": a flower bed alongside the fence must have its length parallel to the fence.
    - The static senses of the motion prepositions all present problems; a cable over the yard must extend beyond the yard's edges; a path along the ocean must be be approximately parallel to it; and so forth.
    - "Among": the target must be commensurable with the objects in the landmark.
4. Such a region is definable, but applicability is not uniform within it - there is context dependence involving more than a frame of reference here, too.

(Gärdenfors, 2000, p. 172-173)

Properties like distribution, extension and parallelism can be formulated by lifting the denotation of LPs from regions to sets of regions, as done in Kracht (2002) for the locative *on*. But there are still many challenges to describing the spatial relationships expressed by locatives, and the interpretation must depend on the context in several ways. These problems will not be addressed in this thesis.

3.3 The Semantics of Mode Phrases

Even though parameterized neighborhoods and vector spaces differ in a number of ways, Kracht writes

> "The semantics of localisers has recently been studied by Zwarts and Winter (2000). What we will have to say is quite compatible with their analysis. Two basic differences stand out, however. Our analysis is centered around what they call *non-projective* localisers. Non-projective localisers need only the landmark to determine the location, while projective ones need something else, typically the deictic center or pivot [. . .] On the other hand, we also include time dependency in the semantics of localisers, which is necessary, as we will show below."
> (Kracht, 2002, p. 195)

In this section, I will therefore first describe Kracht's denotation for the modalisers, and then see how this works with LPs denoting vector spaces.

3.3.1 Adverbial Modalisers

As partially described above, Kracht proposes an ontology where we find the following basic types: e (objects), i (time points), p (spatial points), v (events), t (truth-values), r (regions) and j (intervals). Events is a sub-type of objects, regions of type $p \to t$ and intervals of type $i \to t$. Furthermore, we have the time' function, which returns for each event its time, and the loc' function, which returns for each object its location. Moreover, we have that for functor argument pairs, the compositional rule is functional application, where for modifier/modifee pairs, the rule is intersection. In order to define the different modes, we first need to look at some functions which are integral parts of the modes.

- **properly begins**, pbeg': Take two intervals I and J. We say that J properly begins I, in symbols $\mathsf{pbeg}'(J, I)$, if $J \neq I$ and if for all $s \in I$ there is a $t \in J$ such that $t \leq s$.

- **properly ends**, pend': Take two intervals I and J. We say that J properly ends I, in symbols $\mathsf{pend}'(J, I)$, if $J \neq I$ and if for all $s \in I$ there is a $t \in J$ such that $s \leq t$.

- **in**, $\mathbb{I}$: Let x be an object and L a parameterized neighborhood. Then put

  $$\mathbb{I}(x, L) \stackrel{def}{=} \{t : \mathsf{loc}'(x)(t) \in L(t)\}$$

  This is the set of time points such that the location of x is a member of $L(t)$.

- **mover**, μ: Kracht assumes a function μ to pick out the mover of the event type denoted by the verb in question. In the event type denoted by *run*, the mover is the subject (x_1), and in the event type denoted by *throw*, the mover is the object (x_2), as in (35).

  (35) a. $\mu(\text{run}'(y, x_1)) = x_1$
  b. $\mu(\text{throw}'(y, x_1, x_2)) = x_2$

Kracht identifies five different modes for locatives in adverbial position; static, coinitial, cofinal, transitory and approximative.

All these, except the static, are defined through identifying the mover of the event. Verbs denote event types, and the four directional modes, defined in (38), denote functions from localisers to sets of events, where it is predicated over the location of the mover of the event during the time of the event. We say, for instance, that x is **cofinally in** L **during** I if $\mathbb{I}(x,\ L)$ properly ends I. The definitions of **cofinally in**, **coinitially in** and **transitorily in** are given in (36).

(36) $\mathsf{cf}^*(x,\ L,\ I) \Leftrightarrow \mathsf{pend}'(\mathbb{I}(x,\ L) \cap I,\ I)$
$\mathsf{ci}^*(x,\ L,\ I) \Leftrightarrow \mathsf{pbeg}'(\mathbb{I}(x,\ L) \cap I,\ I)$
$\mathsf{tr}^*(x,\ L,\ I) \Leftrightarrow \mathbb{I}(x,\ L) \cap I \neq I \wedge \mathbb{I}(x, L) \cap I \neq \emptyset$

Using the metric function d, which measures the distance between two regions, we can also define **approximatively in** as in (37).

(37) $\mathsf{ap}^*(x,\ L,\ I) \Leftrightarrow \lambda s.d(\mathsf{loc}'(x)(s),\ L(s))$
is monotone decreasing and nonconstant on I.

Using these definitions again, we can define the modes in (38). These modes denote functions from parameterized neighborhoods to sets of events.

(38) coinitial: $\mathsf{ci}' = \lambda L.\{\mathcal{E} : \mathsf{ci}^*(\mu(\mathcal{E}), L, \mathsf{time}'(\mathcal{E}))\}$
cofinal: $\mathsf{cf}' = \lambda L.\{\mathcal{E} : \mathsf{cf}^*(\mu(\mathcal{E}), L, \mathsf{time}'(\mathcal{E}))\}$
transitory: $\mathsf{tr}' = \lambda L.\{\mathcal{E} : \mathsf{tr}^*(\mu(\mathcal{E}), L, \mathsf{time}'(\mathcal{E}))\}$
approximative: $\mathsf{ap}' = \lambda L.\{\mathcal{E} : \mathsf{ap}^*(\mu(\mathcal{E}), L, \mathsf{time}'(\mathcal{E}))\}$

The static mode is a function which returns the set of events where the event is located within the region L throughout the whole event.

(39) $\mathsf{st}' = \lambda L.\{(\mathcal{E} : \forall t \in \mathsf{time}'(\mathcal{E}))(\mathsf{loc}'(\mathcal{E}) \subseteq L(t))\}$

As we see, all these modes are defined in such a way that they are independent of how L is defined, as long as the loc' function is of type $e \rightarrow (i \rightarrow r)$, r being a region of some sort.

3.3.2 Adnominal Modalisers

For adnominal use of locatives, Kracht proposes an empty operator $\mathsf{a} - \mathsf{loc}'$, which semantically turns an LP into an adnominal adjunct, by turning a parameterized neighborhood into a parameterized property of individuals. (in fact, in Kracht's semantics, all properties are parameterized).

(40) $\mathsf{a} - \mathsf{loc}' = \lambda L \lambda t\{x : \mathsf{loc}'(x)(t) \in L(t)\}$

3.4 Combining the Two Analyses

Up to this point, I have used the term adverbial adjunct in the meaning 'event modifying adjunct'. But there are two alternatives; (i) the locative may denote a set of events; or (ii) the locative may have the same denotation as the VP; functions from entities to sets of events. Kracht's semantics assume the former. We adopt Kracht's ontology, and extend it with the type vectors, as defined in Zwarts and Winter (2000). Changing the type of events from v to s, we end up with an ontology which includes the following basic types:

- e, the type of *objects*
- i, the type of *time points*
- p, the type of *spatial points*
- v, the type of *vectors*

- s, the type of *events*
- t, the type of *truth-values*

Vectors in vector space V (i.e. end-points of located vectors) are of type p. Located vectors (which have both a start-point and an end point) are of type v. According to Kracht, LPs denote sets of located vectors. Following Kracht, we let LPs denote sets of sets of vectors rather than only sets of vectors, and parameterize these. Furthermore, the loc' function also need to be parameterized, so that it is a function from entities to the four-dimensional space-time rather than three-dimensional space.

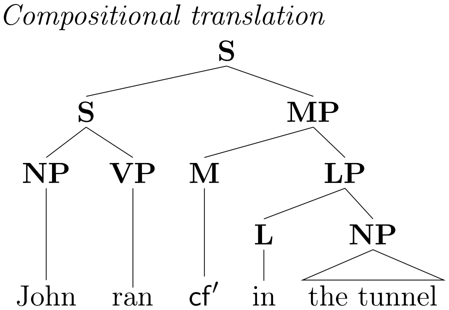
A full analysis

Let us then try to analyze the sentence in (41), step by step.

(41) John ran into the tunnel.

We assume that *into* expresses a cofinal motion, where the moving entity ends up in the region denoted by *in the tunnel*. This is in line with Jackendoff's decompositional analysis of *into* as TO + IN (Jackendoff, 1990, p. 45), and means that *into* is decomposed into $\mathsf{cf}' + in$. We let *in* denote a localiser equivalent to *in*, as defined in (31). We end up with the analysis in (42).

(42) *Compositional translation*



The NP

We simply assume that NPs denote objects of type e, and leave quantifiers unanalyzed. Later, we will analyze quantifiers as generalized quantifiers, and use the MRS[2] formalism to handle quantifier scope ambiguities. But for now, the NPs are analyzed as in (43).

[2]Minimal Recursion Semantics, see Section 4.4

(43)

NP
j
type: e
|
John

NP
tunnel′
type: e
the tunnel

The LP

The LPs denote a parameterized neighborhood in Kracht (2002), i.e. a parameterized set of sets of spatial points. Following Zwarts and Winter (2000), we exchange points with the notion of vectors. Therefore, we want a parameterized set of sets of located vectors to be the denotation of LPs. But keep in mind that these sets of spatial points, or now, sets of vectors, must be path-connected subsets of three-dimensional space. This can be achieved by letting r, the type of regions, be sets of vectors which end-points are path-connected subsets of three-dimensional space. r is a subtype of $v^{\bullet}$. These new parameterized neighborhoods receive the type $i \rightarrow r^{\bullet}$. This means that localisers must be of type $e \rightarrow (i \rightarrow r^{\bullet})$.

Zwarts and Winter proposal for the definition of *in* is given in (44). It is here assumed that the loc' function is applied to the complement NP. We must incorporate this into the localiser semantics, and also let the LP denote a set of sets of vector rather a set of vectors. Kracht's proposal is given in (45) and (46). We can modify (45) by stating that r is a set of located vectors rather than a region (a set of points) as in (47), and define i as in (48). We then see that Zwarts and Winter definition of *in* in (44) corresponds to spelling out the ι function of Kracht in (46).

(44) $\mathsf{in}' \stackrel{def}{=} \lambda A \lambda \mathbf{v}[int(\mathbf{v}, A)]$

(45) $\mathsf{in}' \stackrel{def}{=} \lambda x \lambda t\{r : i(\mathsf{loc}'(x)(t), r), r \text{ a region}\}$

(46) $i(s, r) \stackrel{def}{=} r \subseteq \iota(s)$, where $\iota(s)$ denotes the convex hull of s minus the region s itself.

(47) $\mathsf{in}' \stackrel{def}{=} \lambda x \lambda t\{r : i(\mathsf{loc}'(x)(t), r), r \text{ a set of located vectors}\}$

(48) $i(A, r) \stackrel{def}{=} r \subseteq \{\mathbf{v} : \ int(\mathbf{v}, A)\}$

Kracht gives all localisers a uniform treatment, by letting an arbitrary localiser *local* be defined as in (49).

(49) $\mathsf{local}' = l^{\spadesuit\heartsuit}$
$= \lambda x \lambda t[l^{\spadesuit}(\mathsf{loc}'(x)(t))]$
$= \lambda x \lambda t\{r : l(\mathsf{loc}'(x)(t), r), r$ a region (or now, a set of located vectors)$\}$

All we in fact have to do to adopt Zwarts to Kracht is to exchange the vector space definitions of localisers for l in the general definition of localisers given in (49).

The LP subtree in the analysis of (41) will now look like (50).

(50)

LP
$\lambda t\{r : r \subseteq \{\mathbf{v} : int(\mathbf{v}, \mathsf{loc}'(\text{tunnel}')(t))\}\}$
type: $i \to r^{\bullet}$

L
$\text{in}' =$
$i^{\spadesuit\heartsuit} =$
$\lambda x \lambda t[i^{\spadesuit}(\mathsf{loc}'(x)(t))] =$
$\lambda x \lambda t\{r : i(\mathsf{loc}'(x)(t), r)\} =$
$\lambda x \lambda t\{r : r \subseteq \{\mathbf{v} : int(\mathbf{v}, \mathsf{loc}'(x)(t))\}\}$
type: $e \to (i \to r^{\bullet})$
in

NP
tunnel′
type: e
the tunnel

The MP

Kracht (2002) assumes that VPs denote sets of events, and MPs as intersective modifiers receive the same type of denotation. We receive the interpretation in (51), if we spell out all the definitions given in Kracht (2002).

(51)

MP

$$\mathsf{cf}'(LP') =$$
$$\lambda L[\{\mathcal{E} : \mathsf{cf}{*}(\mu(\mathcal{E}),\ L,\ \mathsf{time}'(\mathcal{E}))\}](LP') =$$
$$\lambda L[\{\mathcal{E} : \mathsf{pend}'(\mathbb{I}(\mu(\mathcal{E}),\ L)\ \cap\ \mathsf{time}'(\mathcal{E}),\ \mathsf{time}'(\mathcal{E}))\}](LP') =$$
$$\lambda L[\{\mathcal{E} : \mathsf{pend}'(\{t : \mathsf{loc}'(\mu(\mathcal{E}))(t) \in L(t)\}$$
$$\cap\ \mathsf{time}'(\mathcal{E}),\ \mathsf{time}'(\mathcal{E}))\}](LP') =$$
$$\{\mathcal{E} : \mathsf{pend}'(\{t : \mathsf{loc}'(\mu(\mathcal{E}))(t) \in \lambda t'[\{r : r \subseteq \{\mathbf{v} :\ int(\mathbf{v}, \mathsf{loc}'(\mathrm{tunnel}')(t'))\}\}](t)\}$$
$$\cap\ \mathsf{time}'(\mathcal{E}),\ \mathsf{time}'(\mathcal{E}))\} =$$

$$\{\mathcal{E} : \mathsf{pend}'(\{t : \mathsf{loc}'(\mu(\mathcal{E}))(t) \in \{r : r \subseteq \{\mathbf{v} :\ int(\mathbf{v}, \mathsf{loc}'(\mathrm{tunnel}')(t))\}\}(\mathsf{time}'(\mathcal{E}))\}$$
$$\cap\ \mathsf{time}'(\mathcal{E}),\ \mathsf{time}'(\mathcal{E}))\}$$

type: $s \rightarrow t$

M
cf'
type: $(i \rightarrow r^{\bullet}) \rightarrow (s \rightarrow t)$
cf'

LP
$LP' = \lambda t\{r : r \subseteq \{\mathbf{v} :\ int(\mathbf{v}, \mathsf{loc}'(\mathrm{tunnel}')(t))\}\}$
type: $i \rightarrow r^{\bullet}$
in the tunnel

Keep in mind that time' applied on an event returns its time, and that pend' means properly ends, and yields true if the set of time points corresponding to the first argument is the end of the interval corresponding the second argument.

The VP and S

We assume that VPs denote functions from individuals to sets of events, and that Ss denote sets of events.

(52)

S
$\lambda x[\{\mathcal{E} : \mathrm{run}'(x)(\mathcal{E})\}](j)$
$\{\mathcal{E} : \mathrm{run}'(j)(\mathcal{E})\}$
type: $s \rightarrow t$

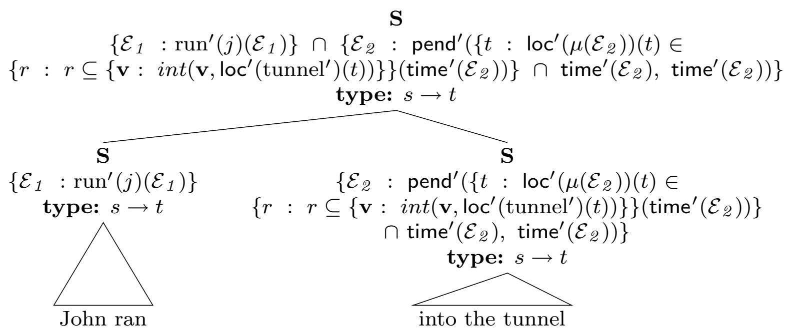
NP
j
type: e
John

VP
$\lambda x[\{\mathcal{E} : \mathrm{run}'(x)(\mathcal{E})\}]$
type: $e \rightarrow (s \rightarrow t)$
ran

MP as intersecting modifier

If we now assume that the MP modifies the S[3], we intersect the two sets of events, as in (53), to receive the denotation of the whole sentence.

(53)



3.5 Summary

In this Chapter, we have looked at the details of how a two-layered analysis of locatives can be defined, and what the denotations of the two layers may look like. We have observed the following:

1. It is possible to give a denotation to both MPs and LPs.
2. LPs denoting regions as *sets of located vectors* may be modified by what we have called PP modifiers.
3. LPs denoting neighborhoods as *sets of* regions can express the notion of contact, as expressed by the preposition *on*.
4. LPs denoting *parameterized* neighborhoods can handle the semantics of locatives with a moving ground.
5. MPs denoting sets of events can modify events, such that all adverbial MPs can be treated as intersective modifiers.
6. MPs denoting properties can modify objects, such that (at least) static adnominal locatives can be treated as intersective modifiers.

We now turn to the frameworks that our implementation of locatives will be encoded in.

[3] In the current representation, the denotation of MPs and Ss are both sets of events, and we let the MP modify the S. The semantic theory we investigate in Chapter 5 is less strict with respect to the order in which the composition takes place, and the MP shall then modify VPs instead.

Chapter 4

MT and the LOGON Project

In this chapter, I will start out with a brief outline of Machine Translation architectures, and describe the Norwegian MT project LOGON with respect to MT architectures in general. Then I will give an introduction to one of the linguistic theories LOGON uses, Head-driven Phrase Structure Grammar (HPSG), and the theory used for semantic descriptions in LOGON, Minimal Recursion Semantics (MRS). The descriptions of HPSG and MRS in this chapter will be used in the following chapters when exploring the semantics and grammar of locatives.

4.1 Machine Translation Architectures

Machine translation (MT) systems are usually considered to consist of 3 parts; (i) analyzing the source language (SL) strings, (ii) mapping SL representations to target language (TL) representations, and (iii) generating TL strings from the TL representation. The representation of the information mapped between the language pairs, and the basis of TL generation, will of course depend on the type of analysis employed. The deeper the analysis, and the further one abstracts away from the SL, the nearer one gets to the TL. This is usually described as an MT triangle. On the basis of this, machine translation architectures may roughly be organized into three classes; Direct, Transfer and Interlingua. The vertical distance represents the distance between the language and the representation of this, and the horizontal distance represents the distance between the SL and TL representation.

Direct MT systems translate SL strings to TL strings without assigning any structure to the strings translated. The order the words appear in may be rearranged in the TL text, but there is no grammar responsible for the rearrangement. Rather, probabilistic or heuristic methods may be used for

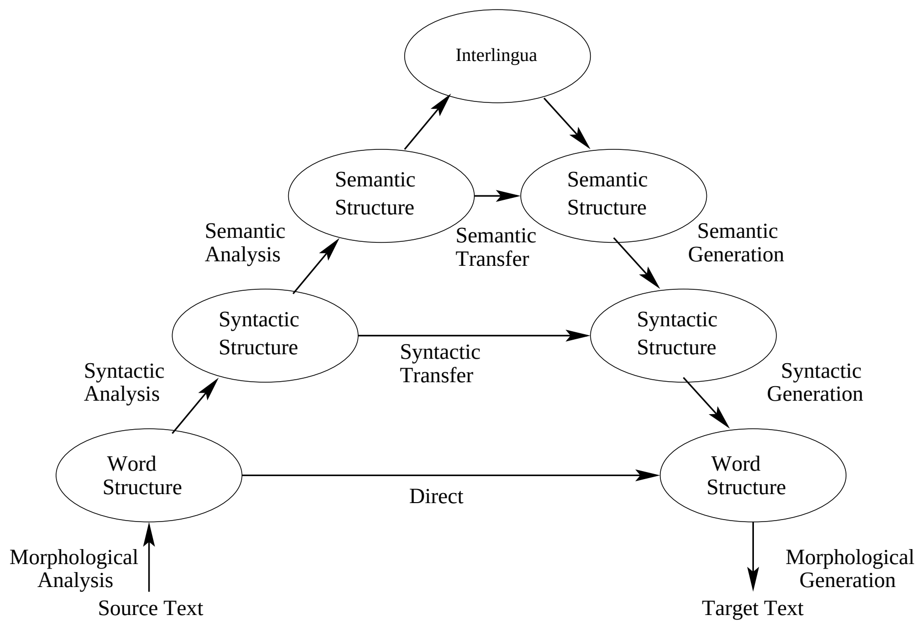


Figure 4.1: Types of MT Systems (from Dorr et al. (1999, p. 13))

TL word ordering.

Interlingua translation systems analyze the SL into a language-independent representation. Interlingua translation has the very appealing feature of being completely independent of the SL-TL language pair. This means that all we need is linking rules between each language and the interlingual form, rather than a set of rules for each language pair in the system. This strategy is based on the assumption that there exists a set of underlying concepts which are common to all the languages of the world, an assumption which has proven to be problematic.

Transfer systems place themselves somewhere in between direct MT and interlingua MT. Transfer can be made on the basis of syntactic analysis alone, or on a deeper semantic analysis. Ideally, the analyses chosen are monolingually motivated. But as the set of transfer rules are based on each language pair, one can also exploit similarities between languages, in only analyzing the syntactic and semantic divergences between the particular language pair and disregard any superfluous analyses.

When settling on how deep one should analyze the languages, lexical decomposition is one area where the line is hard to draw. One example of this is noun compounds. As Norwegian noun compounds are written as one word, while English noun compounds are written as separate words, it is natural to decompose Norwegian compounds in the analysis stage. Compounds are productive, so a 'one-predicate-per-word' semantic analysis would require an infinite list of compounds.

When it comes to locatives, we find complex locatives with the same semantic effects as simple locatives, and the question of how to analyze different constructions with similar semantic effects arise. These constructions will be further investigated in Chapter 5 and 6. Let us now turn to a semantic transfer-based system, the LOGON Project.

4.2 The LOGON Project

The LOGON Project ('Leksikon, ordsemantikk, grammatikk og oversettelse for norsk') is a collaboration between the computational linguistics communities of the Norwegian Universities of Oslo, Bergen and Trondheim. The project is founded by KUNSTI ('Kunnskapsutvikling for norsk språkteknologi'), a program under the Research Council of Norway ('Norges Forskningsråd').

The aim of the project is to employ different parts of computational linguistics in designing a semantic transfer-based MT system, translating texts in a tourist/hiking domain from Norwegian to English, and to develop reusable resources (e.g. grammar and lexicon) for Norwegian language technology.

As the LOGON system is a semantic transfer MT system, there are three main components to the system: semantic analysis of Norwegian, transfer from Norwegian to English semantic representation, and generation of English strings based on the English semantic representations.

4.2.1 Analysis

The syntactic analysis of Norwegian is based on NorGram, a Lexical-Functional Grammar (LFG) for Norwegian. NorGram assigns a c-structure (phrase structure tree) and an f-structure (Attribute-Value Matrix of grammatical functions) to Norwegian sentences. The f-structure is derived from c-structure rules and lexical entries. The f-structure is in turn mapped into a semantic representation (MRS-structure, to be explained in Section 4.4), which is the chosen format for semantic transfer in LOGON (see Oepen et al. (2004) for details).

4.2.2 Transfer

As already mentioned, transfer is based on semantic representations. A SL (i.e. Norwegian) representation is transfered to a TL (i.e. English) representation in a step-wise procedure. Transfer and MRS as semantic representations in transfer will be further explained in Section 4.4.4.

4.2.3 Generation

Generation in the LOGON MT system is provided externally, in the sense that the 'LinGO English Resource Grammar' (Copestake and Flickinger, 2000), a large-scale HPSG grammar, produces the English sentences based on the MRSs delivered from transfer. This component was developed independently of LOGON, but has also been developed further in collaboration with the LOGON Project.

4.3 HPSG

'Head-driven Phrase Structure Grammar' (Pollard and Sag, 1994), or HPSG, is a theory of grammar, which has evolved from 'Generalized Phrase Structure Grammar', or GPSG, a non-transformational approach to syntax. Transformational grammars are based on the assumption that prior to the strings we write or speak, there is stage of generation (a deep structure or an underlying form) which is subject to transformational rules. These rules rearrange the structure of the sentence (i.e. transforms the structure) to a surface structure. Non-transformational grammars try to do syntax without postulating deep structures and transformations on these. Instead, the parts of a linguistic structure constrain each other mutually, through *co-occurrence restrictions*.

The following section is based on Sag et al. (2003), which is closely related to current HPSG.

The inadequacy of context-free grammars

When describing natural languages, context-free grammar (CFG) is a natural start-point. A CFG is defined by a set of atomic symbols (each associated with a meta symbol), a set of production rules and a start symbol S. The production rules are on the form $\alpha \rightarrow \beta$, where α is a meta symbol, and β a string of meta symbols. These rules mean roughly that 'α can consist of β'.

In natural language description, the meta symbols are interpreted as phrase types or grammatical categories, and the production rules are interpreted as phrase structure rules, describing the structure of a syntactic phrase. The atomic symbols constitute the lexicon. This is shown in Table 4.1.

Artificial language:

Production rules:	Atomic Symbols:	Some accepted strings:
S → A B	A: a ,b	
S → B A	B: c ,d	*ac, db*

Natural language:

Phrase Structure rules:	Lexicon:	Some accepted strings:
S → NP VP	D: a, the	*A rabbit ran*
NP → D A* N	N: rabbit, boy	*The hungry boy ran*
VP → V (NP)	A: hungry	*The boy ate a rabbit*
	V: ran, ate	

Table 4.1: Context-free Grammars

There has been much discussion about the formal properties of CFG, and CFG as a theory of natural language grammar. Since the 1960s, the common view has been that natural languages exhibit properties beyond the descriptive capacity of CFGs. The arguments for this view was attacked by the developers of GPSG, Pullum and Gazdar. And in fact, GPSG grammars can be expanded to CFG grammars. But there is now a general agreement that some languages cannot be described adequately by CFGs, and that a more powerful formalism is required.

Independently of whether CFGs can describe natural language, it remains a fact that CFG cannot express significant linguistic generalizations with the necessary clarity. For instance, the subcategorization of verbs, the fact that we find intransitive, transitive and ditransitive verbs, must be modeled by positing three different lexical categories, rather than one category with variation with respect to transitivity. Also, subject-verb-agreement in English would require us to posit different lexical categories for nouns and verbs with respect to number.

In transformational grammars, subject-verb agreement in some constructions, e.g. the passive construction and long-distance dependencies, make use of transformation rules. Transformational grammars make a claim that the surface structure is a transformation of a deeper structure, and the agreement is based on this deep structure.

HPSG, as a non-transformational grammar, makes use of mutual constraints or co-occurrence restrictions instead. Parts of a sentence restrict the context they can appear in, by describing relevant features of the context. I will now describe the central mechanisms for formulating co-occurrence restrictions, namely features and feature structures, types and multiple inheritance type hierarchies, and finally I will describe HPSG, as a theory of natural language grammars.

4.3.1 Feature Structures and Unification

Feature Structures

A feature structure is a way of representing grammatical information. A feature structure is a set of features, each of which is paired with a particular value. Feature structures may also be conceived as functions, specifying a value for each of a set of features. Feature structures are conventionally written as attribute-value matrices (AVMs) in a square bracket notation. The value of a feature may be an atomic value or a feature structure itself, as shown in Figure 4.2.

$$\begin{bmatrix} \textsc{feature1} & \textit{value1} \\ \textsc{feature2} & \textit{value2} \\ \textsc{feature3} & \begin{bmatrix} \textsc{feature4} & \textit{value4} \\ \textsc{feature5} & \textit{value5} \end{bmatrix} \end{bmatrix}$$

Figure 4.2: Feature Structure

Co-occurrence restrictions are encoded through *re-entrancies*, and re-entrancy is a property of feature structures. A feature structure that is re-entrant is a feature structure containing feature paths that share the same value[1]. Re-entrancy is represented by a 'tag' (a boxed integer). In Figure 4.3, the values of the feature paths FEATURE1 and FEATURE2|FEATURE3 are re-entrant, represented by the sharing of the tag $\boxed{1}$.

$$\begin{bmatrix} \textsc{feature1} & \boxed{1} \\ \textsc{feature2} & \begin{bmatrix} \textsc{feature3} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Figure 4.3: Re-entrancy

[1] Note that the re-entrant values are not just 'type identical', but 'token identical', i.e. they are in fact the same value. If we view feature structures as directed acyclic graphs instead of functions, then the arcs representing two features with re-entrant value point to the *same* node, and not just two identical nodes.

Unification

Unification is a process where two feature structures are merged into one feature structure containing all the information in the two feature structures. The effect of unification is equivalent to the conjunction of constraints. Unification (in symbols $\sqcup$) can be defined through a partial ordering of feature structures, *subsumption* (in symbols $\sqsubseteq$).

Subsumption of FSs A more general feature structure $\mathcal{FS}_1$ subsumes a less general feature structure $\mathcal{FS}_2$ iff (i) all features in $\mathcal{FS}_2$ that have *atomic values* have identical values in $\mathcal{FS}_1$, or are unspecified in $\mathcal{FS}_1$, (ii) all features in $\mathcal{FS}_2$ that have *feature structures* as values, have values that are subsumed by the values in $\mathcal{FS}_1$, and (iii) all re-entrant values in $\mathcal{FS}_2$ are also re-entrant in $\mathcal{FS}_1$

Unification of FSs The result of unification of two feature structures $\mathcal{FS}_1$ and $\mathcal{FS}_2$ is the most general feature structure $\mathcal{FS}_3$ which is subsumed by both $\mathcal{FS}_1$ and $\mathcal{FS}_2$

$$\begin{bmatrix} \textsc{feature1} & \textit{value1} \\ \textsc{feature2} & \begin{bmatrix} \textsc{feature3} & \textit{value3} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{feature2} & \begin{bmatrix} \textsc{feature4} & \textit{value4} \end{bmatrix} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \textsc{feature1} & \textit{value1} \\ \textsc{feature2} & \begin{bmatrix} \textsc{feature3} & \textit{value3} \\ \textsc{feature4} & \textit{value4} \end{bmatrix} \end{bmatrix}$$

Figure 4.4: Unification of feature structures

$$\begin{bmatrix} \textsc{feature1} & \begin{bmatrix} \textsc{feature2} & \textit{value2} \end{bmatrix} \\ \textsc{feature3} & \begin{bmatrix} \textsc{feature4} & \textit{value4} \\ \textsc{feature5} & \textit{value5} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{feature1} & \boxed{1} \\ \textsc{feature3} & \boxed{1} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \textsc{feature1} & \boxed{1} \begin{bmatrix} \textsc{feature2} & \textit{value2} \\ \textsc{feature4} & \textit{value4} \\ \textsc{feature5} & \textit{value5} \end{bmatrix} \\ \textsc{feature3} & \boxed{1} \begin{bmatrix} \textsc{feature2} & \textit{value2} \\ \textsc{feature4} & \textit{value4} \\ \textsc{feature5} & \textit{value5} \end{bmatrix} \end{bmatrix}$$

Figure 4.5: Unification of re-entrant features structures

4.3.2 Types

Types and multiple inheritance hierarchies

So far, we have seen instances of untyped feature structures, meaning that the values of features have not been assigned any type. Types constrain the possible values of a feature and capture generalizations across feature structures. Typing is basically the same mechanism as we saw in typed lambda calculus in Chapter 3.1. Feature structures in HPSG can be seen as functions, and as functions in typed lambda calculus, they require arguments and assign values of certain types.

Types are ordered in multiple inheritance type hierarchies. Multiple inheritance type hierarchies have the following three properties: there is a unique top node in the hierarchy; there are no cycles; for implementation purposes, multiple inheritance hierarchies in the LKB[2] require that there is a unique greatest lower bounds. This last property, unique greatest lower bounds, means that two types either don't share any descendants, or they have a unique highest common descendant. Figure 4.6 shows a valid multiple inheritance type hierarchy. And finally, Figure 4.7 shows an invalid multiple inheritance type hierarchy. The invalidity is due to a violation of the greatest unique lower bounds condition. In Figure 4.7, the two types *type2* and *type3* have two common highest descendants, namely *type4* and *type5*.

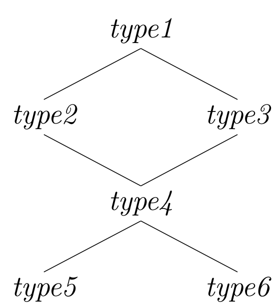


Figure 4.6: A valid type hierarchy.

Typed feature structures

In typed feature structure grammars, not only atomic values, but feature structures as well, are typed. This means that the constraint on a type t is

[2]'Linguistic Knowledge Builder', a grammar engineering platform for writing HPSG grammars, documented in Copestake (2002).

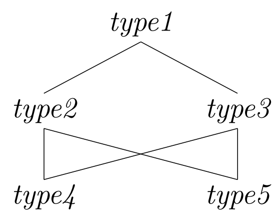


Figure 4.7: A invalid type hierarchy: Multiple greatest lower bounds.

itself a typed feature structure (TFS) of type *t*. Thus, TFSs are also ordered in multiple inheritance hierarchies. Features in typed multiple inheritance hierarchies are inherited. This means that types are constrained by the constraints on its supertypes along with the additional constraints introduced on the type itself.

For instance, the feature GEND(ER) requires its value to be of type *gender*. The feature GEND(ER) may furthermore be part of a feature structure of type *png* (person-number-gender), along with the features PER(SON) and NUM(BER). We see in Figure 4.8 that the constraints on the type *png* define a TFS. The type *png* may in turn serve as the value for a feature AGR , specifying the agreement features for a part of speech category.

$$\begin{bmatrix} \textit{png} \\ \textsc{per}\ \textit{person} \\ \textsc{num}\ \textit{number} \\ \textsc{gend}\ \textit{gender} \end{bmatrix}$$

Figure 4.8: Typed Feature Structure.

Furthermore, the type AGR-POS (agreement part of speech) may be a subtype of *pos* (part of speech), and may be a supertype of e.g. *noun*, a part of speech category with agreement features. The type *agr-pos* is subject to the same constraints as its supertype *pos*. In addition, the constraint AGR is introduced. The value of AGR is of type *png*, which means that the value of of AGR must be a feature structure as defined in Figure 4.8. The position of these types in a multiple inheritance type hierarchy is shown in Figure 4.9. In the definition of the type *agr-pos*, we add the feature AGR, and constrain the value of AGR to be of type *png*. In this way, we capture linguistic generalizations over what the value of agreement features may be.

Unification of Typed Feature Structures

To describe unification of TFSs, we must alter the definition of *subsumption* to accommodate types. The definition of *unification* remains unaltered, except the substitution of FSs with TFSs.

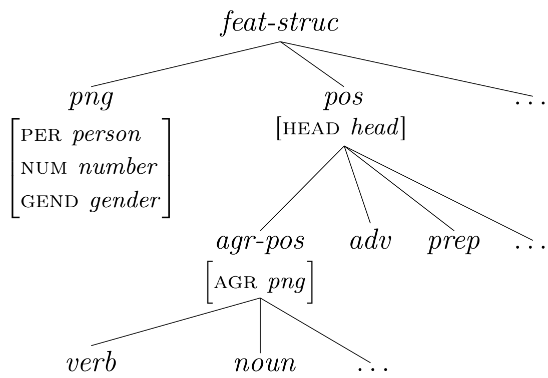


Figure 4.9: Type Hierarchy

Subsumption of TFSs A more general typed feature structure (TFS) $\mathcal{TFS}_1$ subsumes a less general TFS $\mathcal{TFS}_2$ iff all values of features in $\mathcal{TFS}_2$ are the same type as or subtype of the value of the features in $\mathcal{TFS}_1$

Unification of TFSs The result of unification of two typed feature structures $\mathcal{TFS}_1$ and $\mathcal{TFS}_2$ is the most general typed feature structure $\mathcal{TFS}_3$ which is subsumed by both $\mathcal{TFS}_1$ and $\mathcal{TFS}_2$

We see an example of subsumption in Figure 4.10 ($\mathcal{TFS}_1 \sqsubseteq \mathcal{TFS}_2$ is read " $\mathcal{TFS}_1$ subsumes $\mathcal{TFS}_2$"). We see that all information present in the left

$$\begin{bmatrix} \textit{png} \\ \textsc{per}\ \textit{1st} \\ \textsc{gend}\ \textit{fem} \end{bmatrix} \sqsubseteq \begin{bmatrix} \textit{png} \\ \textsc{per}\ \textit{number} \\ \textsc{number}\ \textit{sg} \\ \textsc{gend}\ \textit{fem} \end{bmatrix}$$

Figure 4.10: Subsumption

hand TFS is also present in the right hand side TFS. In the right hand side TFS, we find additional information regarding number. As NUMB is a constraint on the type *png*, NUMB is implicit in the left hand side TFS. We assume that the value of NUMB is type *number*, and that *1st* is a subtype of *number*. It then follows that the left hand side TFS subsumes the right hand side TFS.

4.3.3 Natural Language Grammars with Typed Feature Structures

Using the Type System to Express Linguistic Generalizations

Grammars of natural language using typed feature structures can use the type system to express linguistic generalizations. We saw that agreement is one such generalization, and that the type system expressed the parts of speech categories for which agreement features are appropriate (i.e. *agr-pos*). For the other parts of speech, agreement features are irrelevant and therefore inappropriate. Also, we could want to generalize over parts of a sentence, claiming that words and phrases are the parts of a sentence, or *expressions*, and that all expressions have a HEAD feature. This is expressed in the type hierarchy in Figure 4.11.

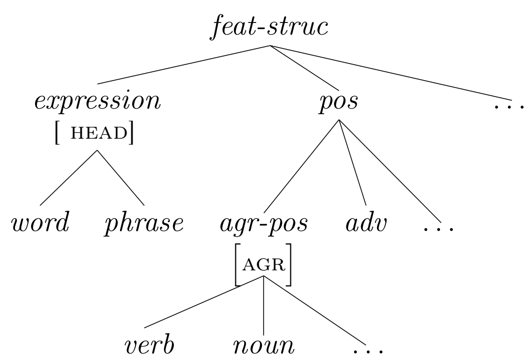


Figure 4.11: Type Hierarchy

Production Rules

In the CFG grammar in Table 4.1, phrase structure rules were written on the form given in (54a). Expanding the symbols on each side with features would yield (54b). We have yet no notion of what the arrow in the production rule means, or how re-entrancy outside a feature structure should be interpreted. The answer to these questions give themselves if we reformulate the production rules as feature structures, as in (54c). We introduce the right hand side of the rule as a list of arguments of the left hand side.

(54) a. *Production rule:*
A → B, C

b. *Production rule with features:*

$$\overset{\text{A}}{\begin{bmatrix}\textsc{feature1}\ \boxed{1}\\ \textsc{feature2}\ \boxed{2}\end{bmatrix}} \rightarrow \overset{\text{B}}{\begin{bmatrix}\textsc{feature1}\ \boxed{1}\end{bmatrix}}\ ,\ \overset{\text{C}}{\begin{bmatrix}\textsc{feature2}\ \boxed{2}\end{bmatrix}}$$

c. *Production rule as a feature structure:*

$$\begin{bmatrix} \textit{production\_rule} \\ \textsc{symbol}\ \text{A} \\ \textsc{feature1}\ \boxed{1} \\ \textsc{feature2}\ \boxed{2} \\ \textsc{arguments}\ \left\langle \begin{bmatrix}\textsc{symbol}\ \text{B}\\ \textsc{feature1}\ \boxed{1}\end{bmatrix}, \begin{bmatrix}\textsc{symbol}\ \text{C}\\ \textsc{feature2}\ \boxed{2}\end{bmatrix} \right\rangle \end{bmatrix}$$

We now turn to HPSG and its central principles.

4.3.4 HPSG as a Theory on Natural Language

HPSG formulates several principles which are basically claims about human language, language universals and language variation. In this section, I present the central principles of HPSG, with a short explanation of the features used.

Head Feature Principle

> "In any headed phrase, the HEAD value of the mother and the HEAD value of the head daughter must be identical."
> (Sag et al., 2003, p. 73)

The name 'Head-driven Phrase Structure Grammars' indicates that the features of the head of a phrase (e.g. the V of a VP) are prominent in the description of a phrase. Features of the head of a phrase are: part of speech, agreement features, case and other properties. The prominence of the head of a phrase is formulated in the 'Head Feature Principle' above. Using feature structures and re-entrancy, we can formalize this principle. The rule in Figure 4.12 can be viewed as a production rule written as a TFS. The feature HD-DTR refers to the head daughter of the phrase. The rule then states what is expressed by the Head Feature Principle. I also assume the feature HEAD as a top feature on expressions (i.e. phrases and words).

$$\begin{bmatrix} \textit{headed-phrase} \\ \textsc{head}\ \boxed{1} \\ \textsc{hd-dtr}\ \begin{bmatrix} \textsc{head}\ \boxed{1} \end{bmatrix} \end{bmatrix}$$

Figure 4.12: Head Feature Principle as a TFS

Valence Principle

> "Unless the rule says otherwise, the mother's value for the VAL features (SPR and COMPS) are identical to those of the head daughter."
> (Sag et al., 2003, p. 106)

The valence of words or phrases is the combinatorial potential of words and phrases, and is used to describe e.g. the syntactic arguments selected by the lexical head. The values of the features SPR and COMPS are lists with the feature structures for the specifier and the complement(s), if there are such. The feature structures constrain the possible specifiers and complements through re-entrancies, as explained above. If the lists are empty, it means that the expression (word or phrase) cannot take a specifier or a complement. The principle says that the values of these features are passed up in the syntactic tree, unless the rule is one regarding specifiers or complements (i.e. "the rule says otherwise"). In Figure 4.13, we see the Head-Complement Rule, which is a rule that explicitly regards valence, and thus, the VAL features are *not* passed up to the mother. Instead, we remove the first element on the COMPS list, and pass the rest of the COMPS list and the value of SPR up to the mother. Other rules regarding valence are Head-Specifier Rule and Head-Modifier Rule.

$$\begin{bmatrix} \textit{head-complement-phrase} \\ \textsc{val} \begin{bmatrix} \textsc{comps}\ \boxed{2} \\ \textsc{spr}\ \boxed{3} \end{bmatrix} \\ \textsc{hd-dtr} \begin{bmatrix} \textsc{val} \begin{bmatrix} \textsc{comps} \begin{bmatrix} \textsc{first}\ \boxed{1} \\ \textsc{rest}\ \boxed{2} \end{bmatrix} \\ \textsc{spr}\ \boxed{3} \end{bmatrix} \end{bmatrix} \\ \textsc{nh-dtr}\ \boxed{1} \end{bmatrix}$$

Figure 4.13: Head-Complement Rule

Specifier-Head Agreement Constraint (SHAC)

> "Verbs and common nouns must be specified as:
>
> $$\begin{bmatrix} \text{HEAD} & \left[\text{AGR } \boxed{1}\right] \\ \text{VAL} & \left[\text{SPR } \left\langle \left[\text{AGR } \boxed{1}\right] \right\rangle \right] \end{bmatrix}\text{''}$$
>
> (Sag et al. (2003, p. 107))

This constraint simply states that verbs and common nouns have a specifier position to be filled, and that the AGR(EEMENT) features of the head and the specifier must be identical.

Semantic Compositionality Principle

> "In any well-formed phrase structure, the mother's RESTR value is the sum of the RESTR values of the daughters."
> (Sag et al. (2003, p. 143)

The value of RESTR[3] is a list containing the semantic predicates that correspond to the expression the feature structure describes. This principle defines HPSG as a compositional theory with respect to semantics. This means that HPSG differs from Lexical Functional Grammar, both in its traditional treatment of semantics (s-structures), and the way semantic representations are produced in LOGON, where the semantics are mapped off a different structure (f-structure). In HPSG, however, the semantics are composed parallel to the composition of the phrases.

4.3.5 The Grammar Matrix

The Grammar Matrix, described in Bender et al. (2002), is an effort of "distilling the wisdom of existing [broad coverage HPSG] grammars and codifying and documenting it in a form that can be used as the basis for new grammars". (Bender et al., 2002, p. 1) It is basically a 'starter-kit' for HSPG grammar engineering in the LKB system, supplying a basic inventory of types needed for writing HPSG grammars. There are basically four components in the Matrix (this list is from Bender et al. (2002, p. 2)):

1. Types defining the basic feature geometry and technical devices (e.g. , for list manipulation).

[3] We shall refer to RESTR as RELS in rest of the thesis, in accordance with the Matrix.

2. Types associated with Minimal Recursion Semantics [...] This portion of the grammar matrix includes a hierarchy of relation types, types and constraints for the propagation of semantic information through the phrase structure tree, a representation of illocutionary force, and provisions for grammar rules which make semantic contribution.

3. General classes of rules, including derivational and inflectional (lexical) rules, unary and binary phrase structure rules, headed and non-headed rules, and head-initial and head-final rules. These rule classes include implementations of general principles of HPSG, like, for example, the Head Feature and Non-Local Feature Principles.

4. Types for basic constructions, such as head-complement, head-specifier, head-subject, head-filler and head-modifier-rules, coordination, as well as more specialized classes of constructions, such as relative clauses and noun-noun compounding. Unlike in specific grammars, these types do not impose any ordering on their daughters in the grammar matrix.

The grammar I will develop in the next chapters are built on the Matrix type hierarchy, with all the generalizations and restrictions encoded in the Matrix. The Matrix is too large to be described here, but I will explain the central types used in the grammar I develop, and the divergences, when there are such, between our grammar and the types in the Matrix.

4.4 Minimal Recursion Semantics

Minimal recursion semantics (MRS) is a framework for computational semantics, developed by Copestake et al. (2003). Part of the motivation for MRS was to develop a formalism with properties (e.g. flat semantics and underspecification) especially suited for parsing and generation in typed feature structure formalisms, such as HPSG. MRS is not a semantic theory in itself, but "a meta-level language for describing semantic structures in some underlying object language" (Copestake et al., 2003, p. 2). The typical object language is predicate calculus with generalized quantifiers. The basic idea is that MRS representations are underspecified, flat semantic representations, corresponding to a set of object language (e.g. predicate calculus) expressions.

Underspecified semantics have several advantages in natural language applications. For instance, parsing a sentence with scope ambiguities, MRS may produce one single analysis, underspecifying the scope ambiguities. The MRS structure produced by such an analysis, can be thought of as a set of building parts, and a specification of how to put the parts together (in one or several ways), such that the scope ambiguities come out correctly. Semantic predicates are represented as *elementary predications* (EPs), constituting the building parts of the MRS structures. These building parts are assembled according to the building instructions, given by the handle constraints, or *hcons*, and the variable bindings.

4.4.1 The Formal Properties of MRS

MRS structures, as defined in Copestake et al. (2003), build upon the notion of Elementary Predications:

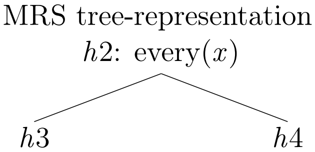
> "*An elementary predication contains exactly four components:*
>
> 1. *a handle which is the label of the* EP
> 2. *a relation*
> 3. *a list of zero or more ordinary variables arguments of the relation*
> 4. *a list of zero or more handles corresponding to scopal arguments of the relation*"
>
> (Copestake et al., 2003, p. 10-11)

A *handle* is a tag identifying the EP. EP s are written in the form in (55a), examples in (55b) and (55c), the EPs for the predicate *dog* and the generalized quantifier *every*, respectively.

(55) a. handle: relation($\text{arg}_1 \ldots \text{arg}_n$, $\text{sc-arg}_1 \ldots \text{sc-arg}_m$)
b. $h1$: dog(y)
c. $h2$: every(x, $h3$, $h4$)

The MRS structure for a sentence can be conceived as a tree. In (56), we see a tree representation of *every*, where the two branches correspond to the restriction and scope (or body) of the quantifier. The MRS structure for a sentence, conceived as a tree structure, is shown in (57), with its predicate logic formula equivalence.

(56) MRS tree-representation of *every*:



(57) a. every white horse sleeps

b. $h2$: every(x)

$h3$ — $h4$

$h3$:white(x), $h3$:horse(x) — $h4$:sleeps(x)

c. $\forall(\text{white}'(x) \wedge \text{horse}'(x) \rightarrow \text{sleep}'(x))$

Multiple EPs that have the same label, are on the same node, and must therefore be conjoined:

> "*An* EP *conjunction is a bag of* EP*s that have the same label*"
> (Copestake et al., 2003, p. 11)

Ordering is semantically irrelevant, but repetition of elements is not, according to Copestake et al. (2003) (e.g. *a big, big horse*), therefore a *bag*[4] of EP s. An EP E *immediately outscopes* another EP E' within an MRS if the value of one of the scopal arguments of E is the label of E'. The *outscopes* relation is the transitive closure of the *immediately outscopes* relation. The *top handle* of the MRS corresponds to a handle which will label the highest EP conjunction in all scope-resolved MRSs which can be derived from this MRS. And finally, *handle constraints* or *hcons* contains a (possibly empty) bag[5]of constraints on the outscopes partial order. The constraints on the outscopes partial order are referred to as *qeq* constraints ($=_q$), which stands for equality modulo quantifiers. If some handle h is qeq to some label l, then the two variables are identical (h=l), or one or more quantifiers 'float in' between h and l.

We can now define an MRS structure.

> "*An MRS structure is a tuple* $\langle GT, R, C\rangle$ *where* GT *is the top handle,* R *is a bag of* EP*s and* C *is a bag of handle constraints, such that:*
>
> **Top:** *There is no handle h that outscopes GT*
>
> **Handle Constraints:** *The outscopes order between the* EP *s in R respects any constraint in C*"
> (Copestake et al., 2003, p. 11)

[4]A bag is equivalent to a multiset.

[5]That we have a *bag* of EPs is interpretable in the sense that the repetition of a predicate (e.g. *big, big horse*) could be said to have an emphatic or amplifying effect. For the handle constraints, which are technical devices for the construction of scoped MRSs, I believe this should be a *set* rather than a *bag*.

These definitions capture some of the main characteristics of Minimal Recursion Semantics. Below, we shall see how MRS treats generalized quantifiers, which is what makes MRS an *underspecified* semantics. And we shall see how MRS treats semantic composition, focusing on two types of modification, *intersective* and *scopal*.

4.4.2 Generalized Quantifiers and qeq Relations

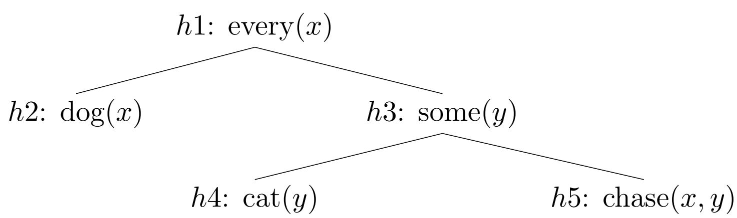
Generalized quantifiers are introduced as a special kind of relation in MRS. They take one ordinary argument (the bound variable), and two scopal arguments, the labels of the *restriction* and *body* of the quantifier. This is shown in (57), where the two branches correspond to *restriction* and *body*. A sentence with more than one quantifier results in scopal ambiguity, as seen in (58). Here, the two readings are represented in (59) and (60) in both conventional notation, MRS bag notation and tree notation.

(58) Every dog chases some cat

(59) a. every(x,dog(x),some(y,cat(y),chase(x,y)))

b. $h1$: every($x, h2, h3$), $h2$: dog(x), $h3$: some($y, h4, h5$), $h4$: cat(y), $h5$: chase(x, y)

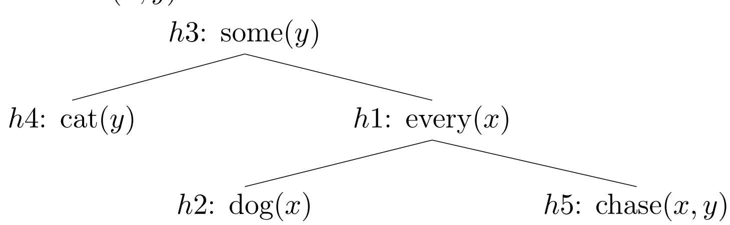
c.



(60) a. some(y,cat(y),every(x, dog(x),chase(x,y)))

b. $h1$: every($x, h2, h5$), $h2$: dog(x), $h3$: some($y, h4, h1$), $h4$: cat(y), $h5$: chase(x, y)

c.



In the MRS structures corresponding to the two readings, the difference lies in whether the label of verbal predicate or the label of other quantifier is the value of the body argument of the quantifier. We can generalize over these structures by replacing the handles in the body argument position with new handles which are not labels of other EPs, as shown in (61).

(61) $h1$: every$(x, h2, hA)$, $h2$: dog(x), $h3$: some$(y, h4, hB)$, $h4$: cat(y), $h5$: chase(x, y)

We can furthermore *link* the structures by adding equalities between the handles. A maximally linked structure is called a scope-resolved MRS, corresponding to a single expression in the object language. A scope-resolved MRS structure must form a tree of EP conjunctions, and the top handle and all handle arguments must be identified with the label of an EP. Scope resolving that $hA = h3$ and $hB = h5$ results in the scope-resolved MRS structure in (59).

When linking the structures to form object language expressions, we need to express constraints on the partial outscopes order, to avoid that e.g. the main verbal predicate take scope over the quantifiers. This is done by the handle constraints or *qeq* constraints between handle variables. These qeq relations basically express that the left side handle must take scope over the right side handle. These qeqs are introduced between the restriction handle of quantifiers and the handle of the noun predicate, between the top handle and the main verb, and in other scopal relationships. The scopal ambiguities in (58) can now be represented in a single MRS structure, assuming that an MRS structure is defined as a tuple $\langle GT, R, C \rangle$. This is shown in (62).

(62) $\langle h0$, {h1: *every*(x, h2, h3), h4: *dog*(x), h5: *some*(y, h6, h7), h8: *cat*(y), $h9$: *chase*(x, y)},
$\{h0 =_q h9,\ h2 =_q h4,\ h6 =_q h8\}\rangle$

If we now take a look at the handle constraints, we see that $h0 =_q h9$ rules out the the verbal predicate as the top predicate in the formula. The two other constraints ensure that the noun predicate is outscoped, but not necessarily immediately outscoped, by the quantifier binding its variable. Linking the structures, i.e. adding equalities between the handles, we end up with only two possible linkings, corresponding to the formulas in (59) and (60).

4.4.3 Modification in MRS

In (57), we saw an example of intersective modification, where the adjective *white* was treated as modifying the noun *horse* intersectively. This is expressed through variable sharing. But for scopal modification, the matter is different.

(63) An alleged killer
 a. $\exists x(alleged(x) \ \wedge \ killer(x))$
 b. $\exists x(alleged(killer(x)))$

Here, an intersective treatment would result in a statement that the killer itself is alleged, as seen in (63a). To avoid this, the adjective *alleged* must be treated as a scopal modifier, taking scope over the noun *killer*, as in (63b). Scopal relations are constrained by qeq relations, just as quantifier scope is. The adjective *alleged* is analyzed as a predicate taking a handle argument instead of an ordinary variable argument, and furthermore, that this handle argument is qeq to the predicate of the noun it modifies, in this case *killer*.

To be able to formulate rules involving handles, we have to extend MRS with a local top handle, LT, which is the topmost label in an MRS which is not the label of a floating EP. MRSs are now tuples on the form $< GT, LT, R, C >$, where GT is the global top, LT is the local top, R is a bag of EPs, and C a set of handle constraints.

The composition of *alleged killer* is shown in Figure 4.14.

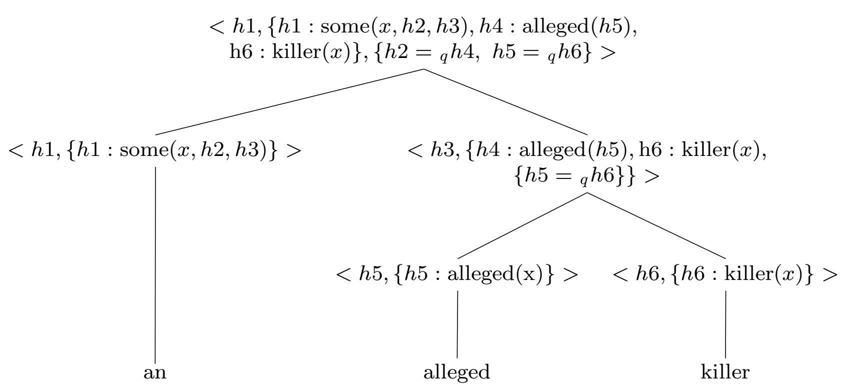


Figure 4.14: Example of a scopal modifier. The global top and the empty sets of handle constraints are omitted in the representation.

4.4.4 MRS as Transfer Representation in LOGON

The semantic representation chosen for transfer in LOGON is MRS. Output from analysis and input to generation are MRSs, and the actual translation takes place on these MRSs. This is done in a step-wise rewriting procedure, through MRS Transfer Rules (MTRs) on the form

[CONTEXT:] INPUT [!FILTER] $\rightarrow$ OUTPUT where each of the four components of the MTR is a partial MRS. Components in brackets are optional. The CONTEXT component restricts the MRSs for which the rule *can* be applied, while the FILTER component specify which MRSs the rule *can not* be applied to. The INPUT component is the partial MRS to be changed (i.e. partial MRS to be deleted), and the OUTPUT component is the partial INPUT MRS after the change (i.e. partial MRS to be added). All other parts of the MRS are unchanged. This means that if only the predicate name is changed, the argument structure of the predicates remains unaltered.

The transfer component also makes use of type hierarchies to allow generalizations about transfer regularities. For instance, temporal prepositions seem to be language-specific. Prepositional relations (e.g. 'i_p') with an internal argument of a temporal sort, are translated as 'temp_loc'. The prepositions corresponding to the relation 'temp_loc' in the TL are then determined by the TL grammar, based on the internal argument. The translation from 'i_p' to 'temp_loc' formulated as an MTR is shown in (64). Remember that only the specified parts of the MRS are rewritten, such that variables etc. are left unchanged unless otherwise stated.

(64) arg12_mtr $\wedge$ {temp_abstr(x_0)} : {på_p(_, x_0)} $\rightarrow$ {temp_loc}

Rewriting halts when all predicates have been transferred. The transferred MRS will now be the basis for the TL generation process.

4.5 Summary

In this Chapter, we have looked at machine translation architectures, and the LOGON project as an instance of a semantic transfer-based MT system. I have described the central features of Head-driven Phrase Structure Grammar, which requires the semantic construction to take place as a compositional process. Furthermore, I have described a theory that meets this requirement, Minimal Recursion Semantics, a meta-level language for describing semantic representations.

In the next chapter, I will make an implementation of the semantic theory of locatives from Chapter 3, using the theories described in this Chapter as the formal frameworks for the syntactic and semantic representation.

Chapter 5

Implementing Locatives

We will now see how the semantics described in Kracht (2002) can be implemented in the Matrix (Bender et al., 2002), an HPSG framework with MRS as the formalism for semantic representation. The grammar is implemented in the LKB system (Copestake, 2002). We shall investigate the relationship between the semantic representations of locatives in lambda calculus and Minimal Recursion Semantics. We will look at how directionals are expressed in English and Norwegian, and how locatives may be ambiguous with respect to the type of motion they express (i.e. mode). Furthermore, we will take a look at a particular class of verbs expressing motion, and how this class interacts with directional prepositional phrases. And finally, we shall compare our implementation of Kracht (2002) with the treatment prepositions are given in the 'LinGO English Resource Grammar', ERG (Copestake and Flickinger, 2000).

5.1 The Features of Lexical Signs in the Matrix

The feature structures of lexical signs in the Matrix differ somewhat from Pollard and Sag (1994). I will give a brief outline of the main features of the Matrix below.

SYNSEM

The Matrix is a *sign-based* grammar, i.e. lexical items and phrases are both considered signs with a surface form (STEM) and associated syntactic and semantic information (SYNSEM). In the SYNSEM feature, under the LOCAL[1] feature, we find features representing syntactic and semantic information of the sign. The most important ones in this context are CAT and CONT.

CAT

Under the feature CAT, we find syntactic information. The feature HEAD has as its value a part-of-speech category, with the relevant features, e.g. person, number, gender, case etc. The feature MOD is also found on the HEAD of a sign, containing descriptions of signs which may be modified by the sign. The feature VAL describes the valence of the sign, as described in Section 4.3.4.

CONT

Under the feature CONT, we find the semantic information carried by the sign. As CONT describes the semantics of the sign, the value of CONT is a description of an MRS structure, of type *mrs*, with the following features: RELS, HCONS, HOOK and MSG. The value of RELS is a list of the semantic relations of the sign (feature structure with predicate, arguments and label, corresponding to the EP). The value of HCONS are the handle constraints of the MRS structure, as described in Section 4.4. The HOOK feature contains the parts of the MRS available to other signs in the composition of the semantics. These are the prominent argument variable (INDEX), the external argument (XARG) for e.g. raising and control constructions, and the handle of the top EP of the sign (LTOP). The motivation for only making some of the variables available in the composition is found in Copestake et al. (2001).

> "We have constrained accessibility by enumerating the possible labels for holes and by stipulating the content of the hooks. We believe that the handle, index, external argument triple constitutes all the semantic information that a sign should make accessible to its functor. The fact that only these pieces of information are visible means, for instance, that it is impossible to define a verb that controls the object of its complement. Although obviously changes to the syntactic valence features would necessitate modification of the hole labels, we think it unlikely that we will need to increase the inventory further."
> Copestake et al. (2001, p. 7)

[1]The Matrix distinguishes between local and non-local features of the sign. I will not discuss this distinction here.

And finally, MSG contains information to construct a message relation, a representation of the illocutionary force of the sentence (interrogative, declarative, imperative, etc.). The grammar we construct will not represent illocutionary force.

Figure 5.1 shows an attribute-value matrix containing the features described above only. The type *sign* in the Matrix contains several other features, but these are not relevant to our grammar, and are therefore omitted in the figure[2].

$$\begin{bmatrix} \textit{sign} \\ \text{STEM}\ \textit{form} \\ \text{SYNSEM.LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \textit{syntactic information} \\ \text{HEAD} \begin{bmatrix} \textit{pos} \\ \text{MOD}\ \textit{potential modifee} \\ \left(\textit{agreement-features}\right) \\ \vdots \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \textit{valence information} \\ \text{SUBJ}\ \textit{list of subject} \\ \text{COMPS}\ \textit{list of complements} \\ \vdots \end{bmatrix} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{semantic information} \\ \text{RELS}\ \textit{list of}\ \text{EP}\textit{s} \\ \text{HCONS}\ \textit{handle constraints} \\ \text{HOOK}\ \textit{variables available to the composition} \\ \text{MSG}\ \textit{information for representing illocutionary force} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 5.1: A sign in the Matrix.

5.2 Implementing Modalisers

In Chapter 3, we saw that the modalisers in Kracht (2002) were defined for both adverbial and adnominal use of locatives. To limit the scope of the thesis, I have only implemented the adverbial modalisers in the grammar fragment which will be described in this and the following chapters.

[2]The italicized text in this particular AVM is meant as an informal description of the value of the feature. Normally, as we saw in Chapter 4, the italicized text is the (formal) type of the value.

Event variables in the Matrix

The Matrix provides the types for implementing a neo-Davidsonian analysis of verbs. A neo-Davidsonian analysis requires that all verbs introduce an extra event argument, and an existential quantifier binding this event, as shown in (65).

(65) $[\![\text{John runs}]\!] = \exists e(\text{run}'(e,\text{John}))$

This analysis is based on the fact that one can make reference to event entities ('He did *it* on purpose') and that intersective adverbial modifiers do not invalidate entailments of the type shown in (66).

(66) a. John runs on the track ⇒ John runs
b. $\exists e(\text{run}'(e,\text{John}) \wedge (\text{on}'(e,\text{track}'))) \Rightarrow \exists e(\text{run}'(e,\text{John}))$

Quantifiers and variable binding were described in Section 4.4.2. It is, however, only referential variables which are explicitly bound in MRS. Event variables are implicitly bound by a wide-scope quantifier. Copestake et al. (2003, p. 27) note that "[t]his is not entirely adequate, because there are some examples in which the scope of events could plausibly be claimed to interact with the explicit quantifiers, but we will not pursue that issue further here".

A suggestion for how this implicit binding can be performed by a higher-order function when translating from an MRS representation into lambda calculus is shown for the logical translation S' of a sentence S in (67).

(67) $\lambda X[\exists e(X(e))]\ (S')\ =\ \exists e(S'(e))$

If we assume such a treatment of implicit quantifier binding of events, then the sentence must denote functions from events to truth values. This means that the intransitive verb *run* and the sentence 'John runs' are translated into lambda calculus as shown in (68) (assuming 'John' denotes an object).

(68) a. $[_{VP}\ \text{run}\ ]'\ =\ \lambda x \lambda e[\text{run}'(e,x)]$
b. $[_{S}\ \text{John runs}\ ]'\ =\ \lambda e[\text{run}'(e,john)]$

The feature structure description of the MRS representation[3] of the neo-Davidsonian analysis of verbs is shown in Figure 5.2. The ARG0 of the verb

[3]The LKB system provides a function to translate feature structures of type *mrs* into MRS representations. Some feature types, e.g. the *relation* type of the EP, are not present in the MRS structure. The function also maps variables of type *event* to e1, e2..., variables of type *ref-ind* to x1, x2... and variables of type *handle* to h1, h2..., according to the MRS conventions of variable naming. We therefore distinguish the feature structure description of an MRS (of type *mrs*) from the MRS structure itself.

relation is the event introduced by the verb, and the ARG1 of the verb is its first argument, corresponding to the index of the subject.

$$\begin{bmatrix} \textit{verb-lex-item} & \\ \text{SYNSEM.LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{VAL.SUBJ} & \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK.INDEX} & \boxed{x} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{mrs} & \\ \text{HOOK} & \begin{bmatrix} \textit{hook} & \\ \text{LTOP} & \boxed{h} \\ \text{INDEX} & \boxed{e} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{arg1-ev-relation} & \\ \text{PRED} & \textit{verb\_rel} \\ \text{LBL} & \boxed{h} \\ \text{ARG0} & \boxed{e}\ \textit{event} \\ \text{ARG1} & \boxed{x} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 5.2: Verbs introduce an event argument.

Modalisers as Semantic Relations

As mentioned earlier, the MRS formalism makes use of semantic relations, whereas lambda calculus makes use of functions. And furthermore, Kracht mixes functions and set theoretic notions in order to improve readability. We will reformulate the definitions of the semantics of modes found in Kracht (2002) as relations to fit the MRS framework. The modes were originally defined as in (69).

(69)

coinitial:	$ci' = \lambda L.\{\mathcal{E} \ : \ ci^*(\mu(\mathcal{E}),\ L,\ time'(\mathcal{E}))\}$
cofinal:	$cf' = \lambda L.\{\mathcal{E} \ : \ cf^*(\mu(\mathcal{E}),\ L,\ time'(\mathcal{E}))\}$
transitory:	$tr' = \lambda L.\{\mathcal{E} \ : \ tr^*(\mu(\mathcal{E}),\ L,\ time'(\mathcal{E}))\}$
approximative:	$ap' = \lambda L.\{\mathcal{E} \ : \ ap^*(\mu(\mathcal{E}),\ L,\ time'(\mathcal{E}))\}$
static:	$st' = \lambda L.\{\mathcal{E} \ : \ \forall t \in time'(\mathcal{E}))(loc'(\mathcal{E}) \subseteq L(t))\}$

First of all, let us reformulate these as pure lambda functions, not mixing set theoretic terms with functions (the event variable $\mathcal{E}$ is replaced by the variable e in the following expressions).

(70) coinitial: $ci' = \lambda L \lambda e.[ci^*(\mu(e),\ L,\ time'(e))]$

$\vdots$

We see now that ci' is a function from a parameterized set of regions to a set of events, or a lambda expression of type $(s \rightarrow t)$. When acting as an intersective modifier of verbs, a modaliser must be of the same type as verb phrases, $(s \rightarrow t)$ (i.e. denote the same type of objects, a set of events). The modaliser is a function from a parameterized set of regions to an event, which again can be formulated as a relation between an event e and a set of regions L, as in (71).

(71) coinitial: $ci' = \lambda L \lambda e.[ci'(e, L)] = \lambda L \lambda e.[ci^*(\mu(e),\ L,\ time'(e))]$
$\vdots$

The natural meaning of the expression in (71) is "e is a coinitial event with respect to the parameterized set of regions L iff the mover of e is coinitially in L in the time of the event e". If we want to spell out the natural meaning of ci^* as well, we can read the expression as "e is a coinitial event with respect to the parameterized set of regions L iff the mover of e is in L at the start time of event e, and not in L at the end time of event e".

Reformulating the modaliser according to MRS conventions, we let modalisers introduce semantic relations as in Figure 5.3, where L corresponds to $\boxed{l}$ and e corresponds to $\boxed{e}$.

$$\begin{bmatrix} \textit{prep-lex-item} \\ \text{SYNSEM.LOCAL.CONT} \begin{bmatrix} \textit{mrs} \\ \text{HOOK} \begin{bmatrix} \textit{hook} \\ \text{LTOP } \boxed{h} \\ \text{INDEX } \boxed{e} \\ \text{XARG } \boxed{l} \end{bmatrix} \\ \text{RELS} \left\langle \begin{bmatrix} \text{PRED } \textit{mode\_rel} \\ \text{LBL } \boxed{h} \\ \text{ARG0 } \boxed{e} \\ \text{ARG1 } \boxed{l} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Figure 5.3: The semantics of modalisers.

The event variable and location variable are present in the HOOK as values of the INDEX and XARG, respectively. The principle I have chosen to govern this assignment of values in the HOOK feature is that the denotation, when interpreted in a model, is the value of INDEX. In this manner, we have the same relationship between the denotation of EPs interpreted into first-order logic and the value of INDEX for verbs and modaliser phrases. They both denote sets of events when interpreted, and they both have the event variable as the value of INDEX. This representation also fits well with the types for transitive lexical items in the Matrix, *transitive-lex-item*, which basically assigns the INDEX value of its syntactic complement to the proper argument slot in the transitive item's EP. For instance, a transitive verb with a nominal complement, will, if it is an instance of *transitive-lex-item*, have the INDEX value of the nominal complement as the value of ARG2 in the verb's EP.

It could be argued that it would be more correct to take the event variable as the external argument, in that the event argument is in a sense proved externally by the verb. Also the location variable is the prominent argument of the EP. I have no objection to this, but see it as two different ways of interpreting the features in the HOOK. And since my solution works well with the encoding of transitivity in the Matrix, I chose to implement the former mapping between the arguments of the EP and the slots in the HOOK.

The representation in Figure 5.3 corresponds to the left hand side of the equation in (71), and contains all the information needed in a grammar, as the truth conditions of the relation depend on the event argument and the set of regions. If we for some reason would need to spell out the full semantics of a modaliser relation, we could translate the MRS relation further, from the list of the single relation (for the coinitial mode) $\langle h0 : ci'(e, L)\rangle$ to a list of three relations $\langle h1 : ci^{*}(h2, L, 1h3), h2 : \mu(e), h3 : time'(e)\rangle$. This is is shown in Figure 5.4.

$$\left\langle \begin{bmatrix} \text{PRED } ci' \\ \text{LBL } \boxed{h1} \\ \text{ARG0 } \boxed{e} \\ \text{ARG1 } \boxed{l} \end{bmatrix} \right\rangle \Rightarrow \left\langle \begin{bmatrix} \text{PRED } ci^{*} \\ \text{LBL } \boxed{h1} \\ \text{ARG1 } \boxed{h2} \\ \text{ARG2 } \boxed{l} \\ \text{ARG3 } \boxed{h3} \end{bmatrix}, \begin{bmatrix} \text{PRED } \mu \\ \text{LBL } \boxed{h2} \\ \text{ARG1 } \boxed{e} \end{bmatrix}, \begin{bmatrix} \text{PRED } time' \\ \text{LBL } \boxed{h3} \\ \text{ARG1 } \boxed{e} \end{bmatrix} \right\rangle$$

Figure 5.4: Spelling out the semantics of a modaliser in terms of MRS relations.

The Mode as a Grammatical Feature on Locatives

Bierwisch (1988) argues that locative prepositions have a binary feature distinguishing directional from static prepositions:

> Focusing on local prepositions, I will primarily be concerned with the feature [± Dir(ectional)] distinguishing directional or path prepositions from place prepositions. [± Dir] must be considered as a grammatical feature on a par with features of Case, Number, Gender, Tense etc. Grammatical features may, but need not,

> correlate with semantic properties, and even if they do, they define grammatical properties which must be kept separate from semantic ones. (Bierwisch, 1988, p. 2)

The semantic mode of a locative corresponds with the grammatical mode, and we will thus represent the grammatical mode of a locative as a re-entrancy of the semantic relation in the HEAD feature of locative prepositions, even though Bierwisch claims that semantic and grammatical properties must be kept separate. This re-entrancy is chosen mainly because the Matrix already has encoded a feature in HEAD called KEYS, where the key semantic predicates of the lexical item, KEY and ALTKEY, may be re-entered. If we were to follow Bierwisch in separating grammatical and semantic properties, we could introduce a new feature on the HEAD feature of prepositions instead.

In the case of modes, there is a correspondence between mode as a *grammatical feature* and as a *semantic predicate*. Therefore, I will make use of the KEYS.KEY feature for representing the mode as a feature on the syntactic head of prepositions, by a re-entrancy of the key relation predicate, i.e. LKEYS.KEYREL.PRED. The key relation is in turn re-entered with the most prominent relation on the RELS list. The KEYS grammatical feature is eligible for selection by e.g. verbs. We also construct a hierarchy of modaliser relation predicates, *mode_rel*s, in the type hierarchy. The re-entrancies are shown in Figure 5.5.

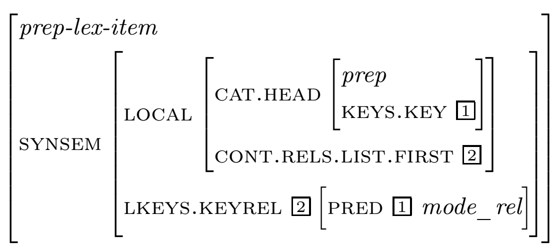


Figure 5.5: Re-entrancy of *mode_rel* as a HEAD feature.

The hierarchy of modaliser relation predicates, or *mode_rel*s, is shown in Figure 5.6 (this will be extended in Chapter 6).

Intersective Modifiers in MRS

The modalisers in Kracht (2002, p. 166) are treated as intersective modifiers. From an MRS point of view, intersective modifiers differ from scopal modifiers in that intersective modifiers share the LTOP value with the modified item, while scopal modifiers do not. This is described in Copestake et al.

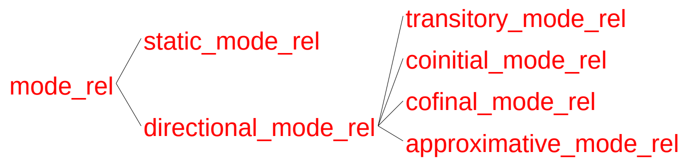


Figure 5.6: *LKB printout:* The hierarchy of modes.

(2003, p. 30), and Bender et al. (2002, p. 10). We see how this can be done for intersective modifiers (e.g. a modaliser) in (72) and for scopal modifiers (e.g. the scopal adverb probably *probably*) in (73). Intersection is represented as unification of arguments, i.e. LTOP values.

(72)
$$\begin{bmatrix} \textit{intersective-modifier} \\ \text{SYNSEM.LOCAL} \begin{bmatrix} \text{CAT.HEAD.MOD} \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK} \begin{bmatrix} \text{LTOP } \boxed{h} \\ \text{INDEX } \boxed{e} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{CONT} \begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{LTOP } \boxed{h} \\ \text{INDEX } \boxed{e} \end{bmatrix} \\ \text{RELS} \left\langle \begin{bmatrix} \textit{modaliser-relation} \\ \text{PRED } \textit{mode\_rel} \\ \text{LBL } \boxed{h} \\ \text{ARG0 } \boxed{e} \\ \text{ARG1 } \boxed{l} \end{bmatrix} \right\rangle \\ \text{HCONS } \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(73)
$$\begin{bmatrix} \textit{scopal-modifier} \\ \text{SYNSEM.LOCAL} \begin{bmatrix} \text{CAT.HEAD.MOD} \left\langle \begin{bmatrix} \text{LOCAL.CONT.HOOK} \begin{bmatrix} \text{LTOP } \boxed{h_3} \\ \text{INDEX } \boxed{e} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{CONT} \begin{bmatrix} \text{HOOK} \begin{bmatrix} \text{LTOP } \boxed{h_1} \\ \text{INDEX } \boxed{e} \end{bmatrix} \\ \text{RELS} \left\langle \begin{bmatrix} \textit{scopal-modifier-relation} \\ \text{PRED } \textit{probably\_adv\_rel} \\ \text{LBL } \boxed{h_1} \\ \text{ARG1 } \boxed{h_2} \end{bmatrix} \right\rangle \\ \text{HCONS } \left\langle \boxed{h_2} =_q \boxed{h_3} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

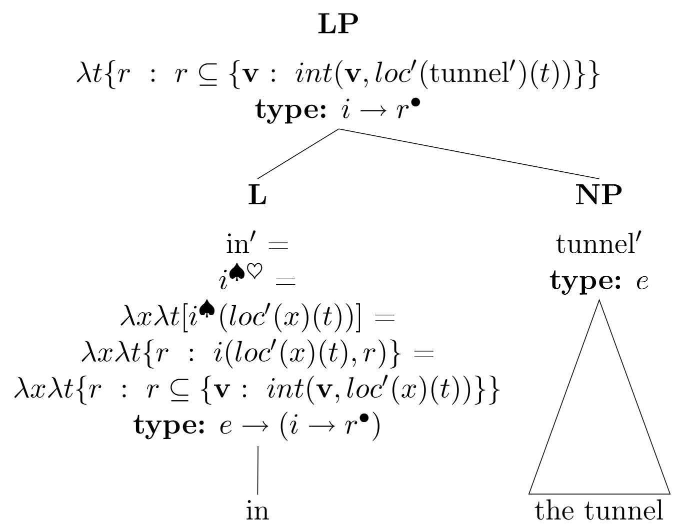
5.3 Implementing Localisers

Extending the Ontology with *loc-ind*

Basically, there are two ontological types encoded as indices in the Matrix: events and referentials (or objects)[4]. We shall extend the ontology with a location index, which is the type of the entities denoted by localiser phrases.

As we recall from Chapter 3, LPs denote parameterized sets of sets of vectors. The analysis of the LP *in the tunnel* is repeated in (74)

(74)



We construct a complex (typed lambda calculus) type location type l, corresponding to the type of LPs, $i \rightarrow r^{\bullet}$. As MRS predicates are relations, we redefine a localiser as a relation, namely a relation between a location of type l and an individual of type e (e stands here for the type of entities, as in lambda calculus. Not to be confused with an event variable in LKB). A localiser relation is true for the (unique) pair $\langle a, b \rangle$, for which the corresponding localiser function applied on b yields a. This is expressed in (75).

(75) $localiser_{func}(x) = l \ \Leftrightarrow \ localiser_{rel}(l, x) = 1$

We have now made the location entity explicit in the grammar we shall construct. The location entity is passed up as the argument to the modaliser relation in the grammar. In a typed lambda calculus, the localiser relation

[4]In addition, we find types for conjoined indices and expletives, which can be regarded as types for implementation purposes rather than ontological types.

will be of type $e \rightarrow (l \rightarrow t)$, and the localiser phrase of type $l \rightarrow t$, as shown for the localiser *in* in (76).

(76) a. $[_L \text{ in}]' = \lambda x \lambda l.[in'(l,x)]$
b. $[_{LP} \text{ in Oslo}]' = \lambda l.[in'(l, oslo')]$

We see the consequence this has on the analysis of a sentence with a locative PP in (77a), where (77b) is the analysis with a localiser *function*, and (77c) the analysis with a localiser *relation*.

(77) a. John walks in Oslo
b. $[_S \text{ John walks in Oslo}]' = \exists e(walk'(e, john) \wedge st'(e,\ in'_{func}(oslo))$
c. $[_S \text{ John walks in Oslo}]' = \exists e(walk'(e, john)\ \wedge\ st'(e,\ l)$
$\wedge\ in'_{rel}(l,\ oslo))$

The location variable l is unbound in this expression. Localisers are functions from an object to a parameterized set of regions, i.e. they return *one* parameterized set of regions for each object. As with event variables in MRS, we assume an implicit existential quantifier for each location variable, with fixed narrow scope (i.e. scoping over the modaliser and localiser relations, and outscoped by all quantifiers binding referential indices, including the quantifier introduced by the complement of the PP). This quantifier states that there exists *at least one*, but not necessarily *exactly* one parameterized set of regions. We could either replace the existential quantifier with a definite quantifier, introduce an axiom (or meaning postulate) in the logic, or construe the model such that there is only one pair of a parameterized set of regions and an object for each localiser relation.

I have not made the existential quantifier explicit in the MRS representations, but it could be made explicit when transferring the meta-level MRS representations to object-level representations in e.g. lambda calculus, by letting an existential quantifier binding the location variable take scope over all EPs with the same handle value as the modaliser, i.e. at least the modaliser relation, the localiser relation and the verb modified.

The scope ambiguity in (78) is due to the ordering of the quantifiers binding the referential indices, whereas the order of the quantifier binding the location variable is fixed. The fact that there is *one* location [$_{LP}$ under a tree] in (78b) and (potentially) *several* locations in (78c) is due to the fact that there is a mapping from *one* object (tree) to a location in (78b) and (potentially) *several* objects (trees) to a location in (78b). (The expressions are deliberately not on prenex normal form, to make it easier to see what is the restriction and body of the generalized quantifiers the expressions are derived from.)

(78) a. Every man slept under a tree.

b. $\exists e \exists y (tree'(y) \wedge \forall x (man'(x) \rightarrow \exists l (sleep'(e,\ x) \wedge st'(e,\ l) \wedge under'(l,\ y))))$

c. $\exists e \forall x (man'(x) \rightarrow \exists y (tree'(y) \wedge \exists l (sleep'(e,\ x) \wedge st'(e,\ l) \wedge under'(l,\ y))))$

The Description of Localisers in the Grammar

The ARG0 of a localiser relation is interpreted as a location individual of logic type l, which is typed in our grammar as *loc-ind*, and represented in the feature structure descriptions as $\boxed{l}$. The type *loc-ind* is introduced as a subtype of *individual* in the type hierarchy, in the hierarchy of *semargs* (semantic arguments of relations). The type hierarchy of *semargs* from the Matrix extended with the type *loc-ind* is given in Figure 5.7, and the semantics for a localiser is shown in Figure 5.8.

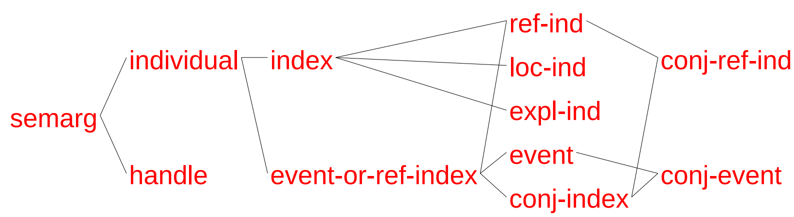


Figure 5.7: *LKB printout:* The type hierarchy of semantic arguments.

$$\begin{bmatrix} \textit{prep-lex-item} & \\ \text{SYNSEM.LOCAL.CONT} & \begin{bmatrix} \textit{mrs} & \\ \text{HOOK} & \begin{bmatrix} \textit{hook} \\ \text{LTOP } \boxed{h} \\ \text{INDEX } \boxed{l} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{localiser-relation} \\ \text{PRED } \textit{loc\_rel} \\ \text{LBL } \boxed{h} \\ \text{ARG0 } \boxed{l}\ \textit{loc-ind} \\ \text{ARG1 } \boxed{x}\ \textit{ref-ind} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Figure 5.8: The semantics of localisers.

The Context of Localisers

As we shall see in the next section, localiser relations may either be part of the denotation of a complex locative preposition denoting both a modaliser and a localiser, or as the denotation of a word denoting a localiser only. Words that only denote localiser relations, must be syntactically selected:

> "LPs seem to occur in rather restricted environments.
>
> 1. LPs can be complements of a modaliser.
> 2. LPs can occur in a copular construction.
> 3. LPs can occur sentence initially."
>
> (Kracht, 2002, p. 201-202)

Examples of 'sentence-initial' LPs are given in (79).

(79) a. In this restaurant, nobody is allowed to smoke.
b. In Berlin, Claver was running faster than ever.
(Kracht, 2002, p. 202)

In addition, LPs may be selected by verbs.

> "We shall argue that the verb has three possibilities for entering a relationship with a locative. It can (a) enter a relationship with the entire complex [M [L DP]], or (b) with only the [L DP] or, finally, (c) it can enter a relationship only with the DP. This means syntactically that it either takes a locative adverbial as an adjunct (Case (a)), or it selects an LP (Case (b)) or it selects a DP as its complement (Case (c))."
> (Kracht, 2002, p. 202-203)

What Kracht calls LPs that occur 'sentence initially' seems to me to be topicalized MPs rather than a distinct syntactic structure for LPs. I will not give topicalization any treatment in the following grammar.

As LPs are syntactically selected, we need to distinguish localisers from modalisers on the HEAD feature. But, as we shall see, we can exploit the fact that LPs are selected by treating localisers as a specialization of modalisers, and let the sign selecting the LP 'force' it from a MP to a LP. For this, we need an implementation of what Kracht calls 'the Emptiness Principle', which will be discussed in Chapter 6.

5.4 MPs in Norwegian and English

The Transparency of the Two-Layering

According to Kracht (2002), we saw that, semantically, locatives had the structure [M [L DP]]. But the morphological segmentation may be different:

> "It is untypical for *L* and *DP* to form a unit excluding *M*. (In fact, as far as we know only Chinese forms an exception to this.) All other combinations however are frequently encountered. Hence, unless all three elements are morphologically free or on opposing sides of the *NP*, we find that *M* + *L* is a unit, which is either an adposition or a case."
> (Kracht, 2002, p. 160)

Kracht claims that there is a certain degree of transparency in the morphology in the analysis of Finnish locatives The morphological pattern for Finnish locative cases is shown in Table 5.1 and 5.2[5].

	Mode →		
Configuration ↓	Stative	Cofinal	Coinitial
in	talossa	taloon	talosta
on	talolla	talolle	talolta

Table 5.1: The local cases of Finnish: *talo* (*house.*) (Kracht, 2002, p. 174)

	Mode →		
Configuration ↓	Stative	Cofinal	Coinitial
∅	-∅-na	-∅-ne	-∅-ta
in	-s-sa (<-*s-na)	-s-se (<-*s-ne)	-s-ta
on	-l-la (<-*l-na)	-l-le (<-*l-ne)	-l-ta

Table 5.2: Analysis of the Finnish local cases¨. (Kracht, 2002, p. 175)

In Norwegian and English, however, this segmentation is rarely encountered. In Norwegian and English, locatives are realized as prepositions. Normally, the locative is realized as a single preposition (e.g. *i* ('in'), *bak* ('behind'), *gjennom* ('through'), *ved* ('at') etc.), but we also find complex prepositions, both idiomatic (e.g. *ved siden av* ('next to')) and productive (*fra* + PP

[5]Kracht (2002) notes that the cofinal mode of *in the house* forms an exception, as the analysis predicts this to be *talosse*, and not *taloon*. But otherwise, the morphology is predicted by the two-layering, and the localiser is closer to the stem, as is also expected according to Kracht.

('from' + PP))[6]. The latter case seems to be a good candidate for where this segmentation is transparent. Consider (80) and (81) below.

(80) Per gikk av teppet.
Per walked off carpet.DEF.

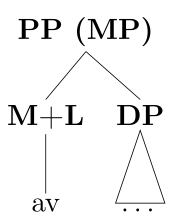
(81) Musa løp ut fra under teppet.
Mouse.DEF ran out from under carpet.DEF.

If we accept that *av teppet* in (80) and *fra under teppet* in (81) express movement from the locations *på teppet* and *under teppet*, respectively, then we must accept that *av* and *fra under* have the same type of semantic effect. If we also accept that *fra under* is the result of a compositional process, then it must be given a compositional treatment in the analysis, rather than being listed in the lexicon as an idiomatic complex preposition. This can only be done by letting *fra* and *under* introduce one predicate each (modaliser and localiser, respectively), and hence, if we want that locative PPs should be given a uniform treatment with respect to the semantic predicates they introduce, then *av* must introduce two predicates as well.

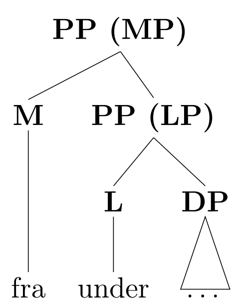
One could of course argue that the difference between a modaliser and a combined modaliser+localiser lies in the logic type of the semantic predicate they denote, and not the number of predicates they introduce. One has to decide which principle to give primacy: That lexical items denote one single relation each, or that predicates in the semantics denote the natural meaning of the expressions in a consequent manner. We will see in Chapter 7 that it is to some degree arbitrary which locatives are modaliser/localiser only, and which are a combination, depending on the lexicalization of concepts in the specific language we analyze.

We therefore assume that all modifying locative PPs introduce both a modaliser relation and a localiser relation. The modaliser and localiser relation may be introduced by one word (82), or by two distinct words (83).

(82)



(83)



The segmentation in (82) is by far the most common in Norwegian and

[6] The preposition *fra* naturally also occur with nominal complements

English, but the segmentation in (83) does occur for *fra/from* and possibly *til/to*, in (84).

(84) a. ?The mouse ran to under the table.
b. ?Musa løp til under bordet.

Trujillo (1995) claims that many languages have locative prepositions corresponding to the coinitial and cofinal modalisers:

> "For example, Japanese has *e* (to) and *kara* (from). It is interesting to note also that in many languages these prepositions may combine with other types of prepositions to give more refined meanings (e.g. to under, from in front of, etc.). [. . .] Other examples include Malay [. . .] and Turkish[. . .]." (Trujillo, 1995, p. 172)

This seems to correspond well with the picture of locatives as two-layered, as it is quite possible that the Japanese prepositions *e* and *kara* here act as modalisers.

M+L Prepositions

In Figure 5.9 we see how the lexical semantics for a simple, coinitial locative may be defined to reflect this two-layering.

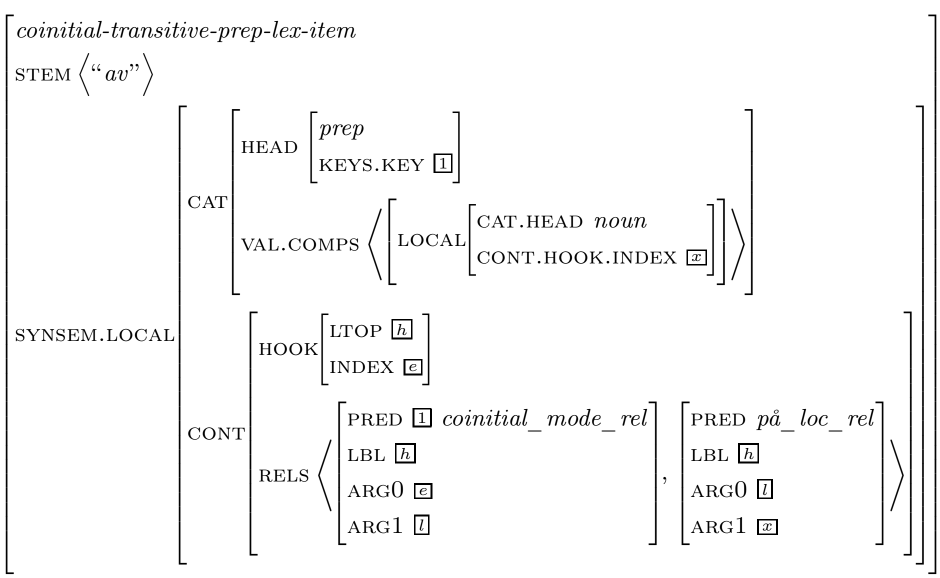


Figure 5.9: Partial TFS of a complex M+L preposition.

We assume that idiomatic complex locatives, as *ved siden av*, are treated as multi-word expressions, which introduce semantics in the same manner as one-word expressions do. This means that *ved siden av* is not given a compositional analysis[7].

M Prepositions (Modalisers)

For the productive construction, *fra* + PP, we state that it selects a prepositional complement, and more specifically, a localiser. The exact description of the head of localisers will be given in Chapter 6, but for now we note that the modaliser selects a PP complement with HOOK.INDEX value of type *loc-ind*, and that the HOOK.LTOP value of the PP complement is unified with the HOOK.LTOP of the modaliser. The description of *fra* is given in Figure 5.10.

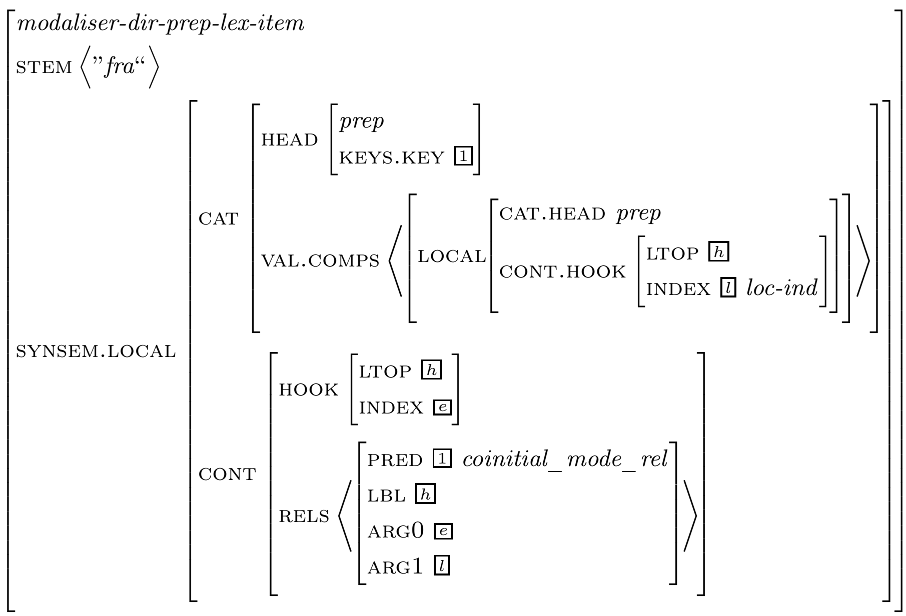


Figure 5.10: Partial TFS of a modaliser.

L Prepositions (Localisers)

Localisers denote location entities, and take nominal complements. Grammars should be written such that localisers only appear in a syntactic context when selected. The representation for a localiser will be given in Section 6.2.

[7] Idiomatic complex locatives could of course still be lexically decomposed

5.5 Motion Verbs

Adverbial directional locatives only combine with certain types of verbs. The examples we have seen above all contain motion verbs, but this class is not easily defined. Kracht proposes that motion verbs are verbs which assign to one of its arguments the semantic role of a mover. A mover is in turn something that necessarily moves in the event denoted by the verb.

> "...whether or not something is an eligible mover only depends on the question whether it is logically necessary for it to move if the concrete event has that type."
> (Kracht, 2002, p. 220)

But this only moves the burden to the expression "logically necessary for it to move", which is not at all clearly defined.

Jackendoff (1990) separates motion verbs from what he calls Verbs of Manner of Motion. These verbs focus on the internal motion, the manner of motion, rather than the path of motion. But they also combine with directionals:

(85) a. Willy wiggled out of the hole.
b. Debbie danced into the room.
(Jackendoff, 1990, p. 89)

Since Jackendoff treats directionals (paths) as arguments to motion verbs, he has to assume that these verbs introduce two predicates: one which describes the internal motion, and the other describing the external motion. The latter takes the directional (path) as an argument. This is not necessary in Kracht's semantics, as the directional introduced the motion predicate. But we still need to decide whether or not *wiggle* and *dance* are motion verbs.

My intuition is that these verbs are not treated as motion verbs, unless the locative adjugated is explicitly directional. Then, the directional predicates over some potential mover of the verb.

(86) Debbie went in the other room. *cofinal*

(87) Debbie danced in the other room. *static*

(88) Debbie danced into the other room. *cofinal*

So, we need to separate potential from necessary movers. This may not solely depend on the verb, as it also seems to be a matter of aspect. And the same verb can differ with regard to the aspect it is assigned.

Also, as Jackendoff points out, directionals combine with other classes of verbs.

(89) The weather-vane points to the north. *orientation*

(90) The road goes from here to Bergen. *extension*

In these cases, we would need to define a different semantics, as it is not anything that physically *moves* to the north or to Bergen in these cases.

Determining which verbs are verbs of motion, is far beyond the scope of this thesis. Some verbs are definitely verbs of motion (*go, run*) and some definitely are not (*live, think*). But a number of verbs may occur as motion verbs, as we saw with *wiggle* and *dance*. In our grammar fragment, I will postulate that some verbs are motion verbs and some verbs are not.

5.5.1 Directional PPs: Adjuncts or Complements?

Mai Ellin Tungseth (Forthcoming) claims that static and directional PPs have different syntactic properties. She claims that static PPs are verbal adjuncts, while directional PPs are verbal complements. By emphatic fronting of the VP, a static interpretation is forced on the PP, as seen in (91). Only phrases that do not appear internally to the fronted constituent can be stranded. This indicates that directional PPs are complements to motion verbs, while static PPs are adjuncts.

(91) a. Hårek rulla tønna i kjelleren. *static or cofinal*
Hårek rolled keg.DEF in basement.DEF.

b. Rulle tønna gjorde Hårek i kjelleren. *only static*
Roll keg.DEF did Hårek in basement.DEF.

Furthermore, Tungseth claims that directional PPs only permit reflexives to be co-referent with the direct object, while locatives also admit pronouns to be co-referent with the direct object, as the sentences in (92) supposedly show. According to the principles of Binding Theory, Tungseth argues, a reflexive is required to be c-commanded by an element which is coindexed with it, while a pronoun cannot be c-commanded by an element coindexed with it.

(92) a. Jeg kastet Per_i i svømmebassenget $sitt_i$.
I threw Per in swimming pool.DEF his.REFL

b. Jeg kastet Per_i i svømmebassenget $hans_i$.
I threw Per in swimming pool.DEF his.PRON

According to Tungseth, the sentence in (92a) must be given a directional interpretation, while the sentence in (92b) must be be given a static interpretation. But according to my intuitions, (92a) must be cofinal, whereas (92b) can be either cofinal or static. If we assume that (92b) can be given a cofinal interpretation, then the pronoun in the cofinal PP is c-commanded by its co-referent, even though the principles of Binding Theory predict the contrary. So even if tests for constituent stranding indicate that directionals are VP internal, the binding of anaphora seems to some degree to contradict this view[8].

Kracht (2002) claims that directional PPs are adjuncts, but he does not give any syntactic data to support this view. Rather, he bases his conclusion on the nature of arguments and the semantic properties of directionals:

> "Further, we disagree with Creary et al. that locatives are always arguments. There are three reasons for us to disagree.
>
> 1. Locatives can - with some exceptions - be freely omitted. As a rule, arguments cannot be omitted.
> 2. Different types of locatives can be used with the same verb. However, if as Creary et al. (1989) claim, locatives fill semantic argument positions, some verbs must make room for several such positions.
> 3. The semantics of locatives is basically intersective. This is a strong indication that they are basically adjuncts. However, as we shall see, there are a few verbs that take locatives as arguments, in which case their semantics is not necessarily intersective."
>
> (Kracht, 2002, p. 163-164)

As a general agreement on the syntactic status of directional PPs seem to be lacking, I do not want to make any strong claim on this matter. But when implementing a grammar, we have to make decisions. I have chosen to implement directional PPs as syntactic complements of motion verbs, but with the semantics of intersective modifiers. This means that the lexical entries of motion verbs have an optional directional PP complement. Optional complements are marked by the feature-value pair 'OPT +'. Directional PPs

[8]This last point may prove not to be an argument against the view that directional PPs are complements of verbs, but rather the result of a flaw in Binding Theory. Pollard and Sag (1994, p. 238-281) argue extensively that the principles in Binding Theory based on c-command make numerous of incorrect predictions when it comes to binding. Pollard and Sag (1994) propose a different concept, *o-command* (obliqueness-command), a relation based not on tree configurations, but rather on the relative obliqueness of grammatical functions.

differ from static PPs in that directional PPs have an empty list as the value of MOD (i.e. they may not serve as syntactic adjuncts), whereas static PPs have on their MOD list a description of the sign they may modify. The description of a motion verb is shown in Figure 5.11, and the syntactic tree of a motion verb with a modaliser phrase (MP) complement is shown in Figure 5.12.

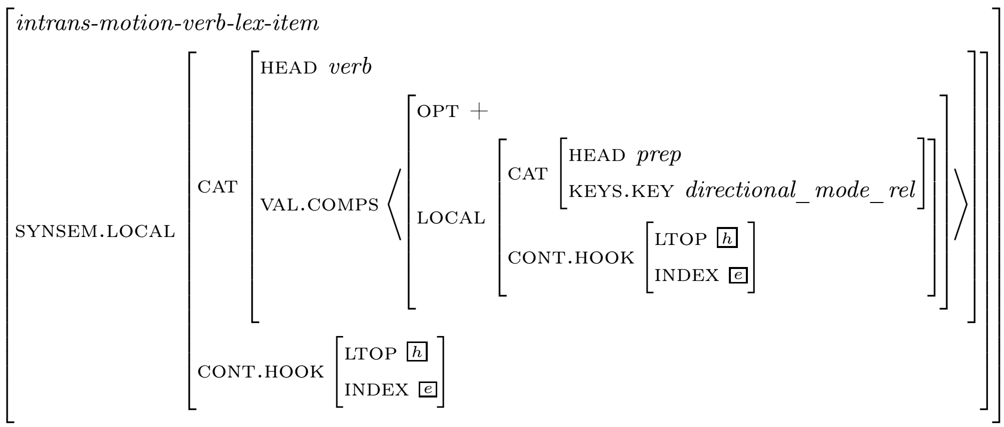


Figure 5.11: Motion verb with optional PP complement.

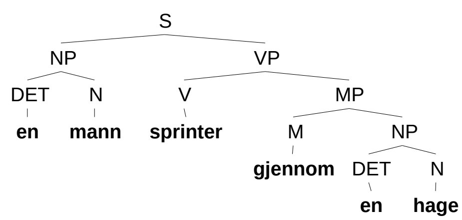


Figure 5.12: *LKB printout:* Motion verb with complement.

5.5.2 PP Clusters

One problem that needs to be addressed as a direct result of the complement analysis of directionals, is the fact that several directional PPs can occur together, as shown in (93).

(93) a. En mann løp fra en hage til en garasje.
A man run.PST from a garden to a garage
'A man ran from a garden to a garage.'

b. Mari klatrer inn i huset gjennom vinduet.
Mary climbs into house.DEF through window.DEF
'Mary climbs into the house through the window.'

To make the grammar cope with PP clusters as a single complement of motion verbs, we need to construct these PP clusters as one PP. I have chosen to implement PP clusters as modificational adjuncts, such that directional PPs modify other directional PPs. The representation of directional prepositions as PP modifiers is given in Figure 5.13, and the syntactic structure of a motion verb with a PP cluster complement is shown in Figure 5.14[9].

5.5.3 More on the Internal Structure of PPs

Apparently, there are constraints on how directional PPs combine with respect to mode. Normally, we only find one coinitial and one cofinal PP combined, as in (94a)-(94e). (All instances of *til* ('to') are interpreted as cofinal, and not possessive or benefactive).

(94) a. Per løp fra hagen til garasjen.
Peter ran from garden.DEF to garage.DEF

b. Per løp til garasjen fra hagen.
Peter ran to garage.DEF from garden.DEF

c. Per løp inn i garasjen fra hagen.
Peter ran into garage.DEF from garden.DEF

d. *Per løp til hagen til garasjen.
Peter ran to garden.DEF to garage.DEF

e. ?Per løp inn i hagen til garasjen.
Peter ran into garden.DEF to garage.DEF

We have the tools at hand to distinguish between the different directional modes in the HEAD feature of locatives, and can define negative types (e.g. *not_coinitial_mode_rel*) or disjunctive types (e.g. *cofinal_or_transitory_or_approximative_mode_rel*). I have implemented one such negative type, *not_cofinal_mode_rel*, for illustration[10], and con-

[9]Our grammar fragment does not work properly on PP cluster constructions. We need the modifier to precede the head, and not the opposite, in order to specify the correct type of the semantic arguments in the EPs, as they are underspecified (cf. Ch 6). The relationship between the PPs in a PP cluster seem to be perfectly symmetric, the one or the other ordering of the head-modifier relationship should be equally correct in describing this symmetric relationship. Describing PP clusters as conjoined phrases would preserve this symmetry even better. I have not pursued this issue further.

[10]See the Appendix or the accompanying CD-ROM with the implementation of the grammar

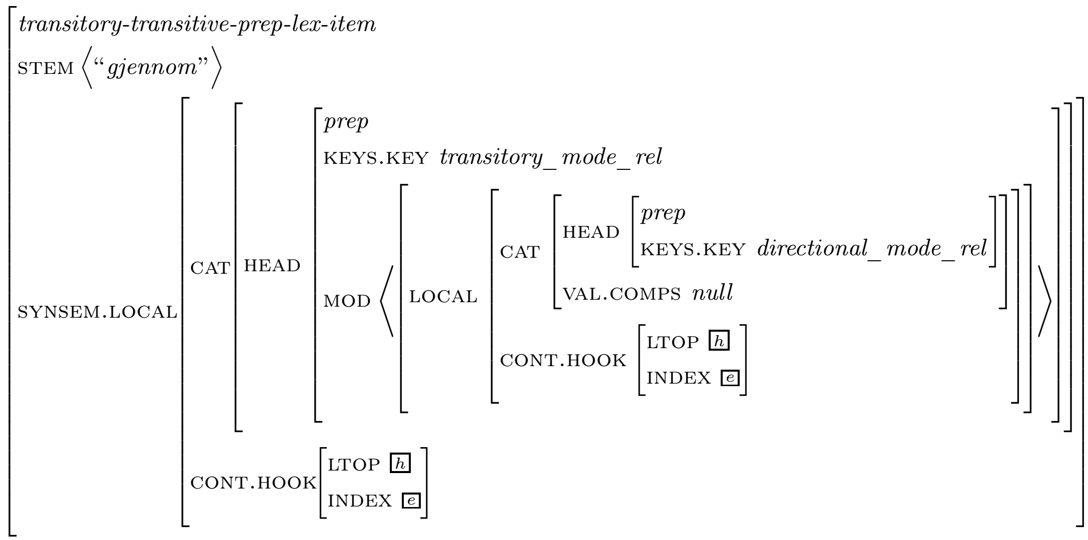


Figure 5.13: Construction of PP clusters through modification.

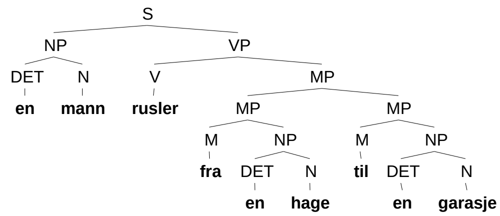


Figure 5.14: *LKB printout:* PP cluster.

strained cofinal PPs in the grammar to modify other directional PPs which are not cofinal. This constraint is based on the assumption that we find only one cofinal PP in a PP cluster. (But we may find complex cofinal PPs, which will be described in Chapter 7. In the analysis of complex PPs, the constraint described above also has the effect that it removes some spurious ambiguity.)

5.6 Mode Ambiguity

5.6.1 Types of Ambiguity

Static-Cofinal Ambiguity

Mai Ellin Tungseth (Forthcoming) argues that "[i]n Norwegian, and to a lesser extent also in English, a sentence containing a combination of a verb of motion and a stative preposition like *i* ('in') or *på* ('on') can be seen to be ambiguous between a telic reading of directed motion and an atelic reading of located motion" (Tungseth, Forthcoming, p. 1).

(95) a. Jon syklet i grøfta.
Jon biked in ditch.DEF.

b. Hans kastet ballen i stua.
Hans threw ball.DEF in living room.DEF.

c. Spionen gikk på taket.
Spy.DEF walked on roof.DEF.

(Example sentences from Tungseth (Forthcoming)).

I agree with this claim. Most static locative prepositions may be given a cofinal interpretation in combination with a motion verb. But some exceptions do occur. *Ved* ('at') cannot be used in cofinal mode, and it doesn't seem to occur as a complement of the modalisers *fra* ('from') or *til* ('to'), either.

(96) a. *Per løp ved treet på ti sekunder.
Per ran at tree.DEF in ten seconds.

b. *Per løp fra ved treet til ved huset.
Per ran from at tree.DEF to at house.DEF.

One way of explaining this is that *til* can be viewed as the cofinal lexicalization of *ved*, as noted many times for the English prepositions *at/to, in/into, on/onto*, based on valid inferences as seen in (97).

(97) a. Per løp til X ⇒ Per er ved X

b. Peter runs to/into/onto X ⇒ Peter is at/in/on X

If we assume that *til* ('to') is the cofinal variant of *ved* ('at'), there is no need for two cofinal lexicalizations of *ved*, and a cofinal interpretation of *ved* ('at') is blocked. *Til* ('to') is the only case of an unambiguous transitive cofinal locative that I can come to think of in Norwegian. There seem to be no simple Norwegian locatives corresponding to the English locatives *into* and *onto*.

Inni ('inside'), which can be seen as a lexical variant of the complex preposition *inne i*, cannot be used as a cofinal preposition either. But it may be used as the complement of a modaliser. I believe the same applies to *nær* ('near'), although my intuition is somewhat unclear.

(98) a. ?Per løp inni/nær huset på ti sekunder.
Per ran inside/near house.DEF in ten seconds.

b. ?Per løp fra inni/nær huset til under parasollen.
Per ran from inside/near house.DEF to under parasol.DEF.

Blant ('among'), on the other hand, appears to occur as a cofinal locative when combined with words like *inn*. This does not seem to apply from the static prepositions mentioned above, as seen in the sentences in (99).

(99) a. Per løp inn blant trærne.
Per ran in among trees.DEF.

b. *Per løp inn inni trærne. (with cofinal *inni*)
Per ran in inside trees.DEF.

c. *Per løp inn nær trærne. (with cofinal *nær*)
Per ran in near trees.DEF.

d. *Per løp inn ved trærne. (with cofinal *ved*)
Per ran in at/by trees.DEF.

(The sentences in (99c) and (99d) locate the event rather than specifying the goal of motion, i.e. static mode and not cofinal mode).

Thus, it seems as if most static prepositions are also cofinal, but that there are some exceptions to this picture.

Static-Cofinal-Transitory Ambiguity

As we see in (100), from Jackendoff (1990), locatives can also be three-ways ambiguous.

(100) a. The mouse ran (around) under the table. *static mode*
b. The mouse ran under the table (and stayed there). *cofinal mode*
c. The mouse ran under the table (to the other side of the room). *transitory mode*

Static-Coinitial Ambiguity

A third possibility of mode ambiguity is found in locatives which are ambiguous between coinitial and static readings, as is the case for the preposition *off*.

(101) John is driving off the road.

5.6.2 Ambiguity in the Type Hierarchy

As we have seen, prepositions may be ambiguous with respect to mode in several ways. Expressing this in the type hierarchy by underspecifying the mode of mode ambiguous locatives, we would capture significant generalizations. An underspecified locative would need to be further specified in the linguistic context, such that e.g. the mode in the final MRS come out right. But when trying to underspecify locatives, two significant problems arise.

For instance, the locative *i* ('in') is ambiguous between a static and a cofinal reading. If we want to collapse these two readings into a single lexical entry, we would have to deal with their different values of MOD, i.e. the items they modify. We have defined directional locatives to modify other directional PPs, in order to form PP clusters. Static locatives, on the other hand, act as syntactic adjuncts of VPs. As the MOD list only may contain *one* item (at least the way the rules for adjuncts in the Matrix currently are set up), we would need to specify a supertype which subsumes all VPs and directional

PPs. This is not a trivial task. And the generalization of VPs and directional PPs to a supertype seem to me to be (at least to some degree) arbitrary.

The second problem is top underspecify the PRED value of locatives, of type *mod_rel*. Forcing an underspecified *mod_rel* for directional PPs is unproblematic, as this can be executed by the verb selecting the PP as its complement. But getting the *mod_rel* of static PPs right is more difficult. There are several alternatives. One could define a separate rule for (static) PP adjuncts. This seem to be in conflict with the 'head-drivenness' of HPSG. One could also make the static modaliser relation a supertype of directional modaliser relations in the *mode_rel* hierarchy. I am of the opinion that this violates the gist of such a hierarchy, as the cost of underspecifying lexical entries is blurring the interpretation of the hierarchy of semantic relations. The third, and probably best alternative, is to introduce a new type into the type hierarchy, e.g. *adjunct_rel*, and constrain the adjunct rule to only allow signs with PRED value of type *adjunct_rel* to act as adjuncts. This also seems to be in conflict with interpreting the hierarchy of relations (the *predsort* hierarchy) as a semantic hierarchy, when types are introduced in this hierarchy for purely syntactic reasons. But at least it is better than making the directional modaliser relations subtypes of the static modaliser relation. I have not dared to follow this path, as underspecifying the lexical entries for mode ambiguous locatives in our current grammar may prove to be a time-consuming enterprise. But as this discussion shows, I cannot rule out the possibility that an underspecification is possible in principle.

Ambiguity between directional modes can be collapsed into a single lexical entry without running into these problems, however, as they share MOD value. In our grammar fragment, directional prepositions are given the same syntactic analysis, regardless of which of the directional modes they express. Therefore, a preposition like *under* ('under'), which we found to be static/cofinal/transitory ambiguous, will have two lexical entries, one static and one cofinal/transitory. If this preposition is the complement of a verb that selects cofinal complements, it will be forced down to the cofinal subtype.

5.7 Comparison with the ERG

This section contains a short comparison of our implementation of Kracht's (2002) two-layering with the representations prepositions have been assigned in the English Resource Grammar. ERG, also mentioned in section 4.2.3, is a working large-scale grammar for English, which gives the representations chosen an empirical weight, in that they function well in a grammar coping with numerous syntactic phenomena. ERG uses HPSG and produces MRS representations, and is also used for generation in LOGON. For these reasons, I will compare the representation chosen for prepositions in the ERG with the representations of locatives we have arrived at on basis of Kracht (2002).

Adjuncts or Complements?

Adverbial locative prepositions in ERG act as intersective modifiers and introduce relations with three arguments, as shown in (102), where $e1$ is an event introduced by the preposition, and $e2$ is the event modified.

(102)
$$\left[\text{SYNSEM.LOCAL}\ \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \textit{prep}\\ \text{VAL.COMPS} & \left\langle\left[\text{LOCAL.CONT.HOOK.INDEX}\ \boxed{x1}\right]\right\rangle\end{array}\right]\\ \text{CONT} & \left[\begin{array}{ll}\text{HOOK.LTOP} & \boxed{h1}\\ \text{RELS} & \left\langle\left[\begin{array}{ll}\text{LBL} & \boxed{h1}\\ \text{ARG0} & \boxed{e1}\\ \text{ARG1} & \boxed{e2}\\ \text{ARG2} & \boxed{x1}\end{array}\right]\right\rangle\end{array}\right]\end{array}\right]\right]$$

Static locatives are analyzed as syntactic adjuncts. Directional prepositions are treated as syntactic complements to motion verbs, but with intersective semantics. I have chosen to follow the ERG in this analysis[11]. Note that nothing in the semantic interpretation of directional PPs constrains us to choose either the adjunct or the complement analysis of directionals.

Event Argument Introduced by the Preposition

Prepositions introduce an event argument in the ERG analysis. This is used in copular constructions. Copulars in the ERG do not introduce events. Instead, they are 'parasitic' on the extra event argument, which is the top index of the MRS in copular constructions involving PPs[12]. Adjectives, e.g. *red*, also introduce an event argument available in copular constructions, in the same manner as prepositions. In our representations, we have chosen to implement modaliser relations as having two arguments. Even though copulars are not included in our grammar fragment, the PP does not have to

[11] With respect to the semantics, this is in line with Kracht's (2002) view, and in conflict with Jackendoff's (1990) view. But when it comes to syntax, the tables are turned, as Kracht (2002) argued for directional PPs as syntactic adjuncts, whereas Jackendoff (1990) regarded them as syntactic complements.

[12] Based on personal communication with Dan Flickinger, the main developer of the ERG

introduce the event argument in such constructions. Several other solutions are possible. For instance, Kracht argues that copulars introduce relations between an object and a region, and hence the copular verb would be the sign that introduced the event argument in copular constructions.

Relations Between Event and an Object

The ARG2 in the prepositional relation in ERG is an object, more specifically the referential index introduced by the nominal complement of the preposition. In our implementation, to reflect the two-layering, modalisers introduce a relation between an event and a location, and the localiser introduces a relation between a location and an object. This difference reflects the fact that the two analyses treat locatives as one-layered and two-layered.

Lexical Entries for Ambiguous Prepositions

In the ERG, mode-ambiguous prepositions have one lexical entry, and the motion verb selecting the PP forces the type of relation introduced from a general or static relation to a directional relation. This has one disadvantage, as it makes directional relations daughters of, rather than sisters of, static relations in the type hierarchy. The cost of avoiding this in our current grammar, is lexical ambiguity, i.e. two entries for ambiguous locatives. But as the static and directional locatives only appear in complementary syntactic contexts, the number of analysis is not expected to grow as a result of this lexical ambiguity in our grammar fragment.

5.8 Summary

We have now become familiar with the Matrix, and how MRS representations are constructed. We have seen the relationship between MRS representations and lambda calculus representations of modalisers and localisers. We have also seen some examples of the modaliser/localiser segmentation in Norwegian and English, in prepositions (modalisers) selecting PP complements. We have looked at one prominent syntactic context of directional PPs, motion verbs, and how static and directional PPs interact with motion verbs. We have also seen a proposal for how PPs form PP clusters acting as the complements of motion verbs. And finally, we have compared our analysis with the analysis chosen in the ERG. Our analysis avoids defining directional relations as subtypes of static relations, but at the cost of having separate lexical entries for directional and static locatives.

In the next chapter, we shall implement an important principle in Kracht (2002), which is in fact what enables us to access the different layers in the two-layering of Kracht's semantic theory on locatives. We shall arrive at the representation of localiser phrases, which were not fully described in the current chapter. And we shall see how LPs are selected by modalisers and certain verbs, and how this enables us to express a more accurate semantics for the verbs which select LP complements.

Chapter 6

The Emptiness Principle

(103) a. Tore bor i Bergen
Tore lives in Bergen
'Tore lives in Bergen'

b. Jon så Mari på damedo
Jon saw Mari on ladies'-room.DEF
'John saw Mary in the ladies' room

c. Jon la smøret inne i kjøleskapet
Jon put butter.DEF in$_{static}$ in refrigerator.DEF
'John put the butter in the refrigerator'

The sentences in (103) show three different types of verbs that, when combined with static PPs, exhibit semantic properties which make the event-locating interpretation of PPs difficult. In treating locatives as intersective modifiers, we have assumed that directional MPs have an orientation toward the mover of the event, while static MPs are oriented toward the event itself, with its participants. But while the sentences in (103) all contain static MPs, they do not entail that the event itself is located in the region expressed by the PP.

In (103a), Tore does not have to be located in Bergen to live there. It may be argued that *å bo* ('to live') does not denote an event at all, but a property of the subject or a more general 'eventuality' of some sort. But if it does denote an event, then the event participants are not located by the PP.

Neither does sentence (103b) necessarily locate John and Mary in the same room. Rather, John may have seen Mary from outside the ladies' room. Nor is the PP likely to attach to the NP, as the examples from Verspoor (1997) in (104) suggest.

(104) a. John saw Mary with the red hair in the ladies' room.
b. Mary with the red hair, John saw in the ladies' room.
c. ?Mary with the red hair in the ladies' room, John saw.

In (103c), John does not have to fit inside the refrigerator to make the sentence true, even though the preposition *inne* ('inne) is unambiguously static. For the verbs in (103a) and (103c), we shall claim that the PP is an argument of the verb, and furthermore that the PP denotes an LP rather than an MP.

But before assigning a description of the TFSs of these verbs, we shall take a look at the principle that governs the relationship between the syntactic and semantic structure of different types of PPs.

In Section 5.3, we quoted Kracht on the observation that the verb can enter a relationship with the PP in three different ways, i.e. with the MP, LP or NP of the PP. Investigating the consequences of this claim, Kracht writes:

> "[...] we shall advance here the thesis that if an element is fixed regardless of the meaning of the entire sentence, then it has no interpretive impact. Since this is a very important and general observation, we shall work out the details of this principle, which we call the *emptiness principle*.
>
> **Emptiness Principle** Suppose that X is a syntactic marker in the constituent C. Suppose further that the presence of X in C is determined purely by non-semantic rules (for example selection, agreement, Sandhi). Then the meaning of X is empty, namely the identity function."
> (Kracht, 2002, p. 203-204)

This principle is perhaps not as controversial as it may seem at first glance. One possible instantiation of it is when a preposition (as opposed to the word order) marks the indirect object of dative alternating verbs. The preposition in sentences like (105a) does not contribute to the semantics, and the sentences (105a) and (105b) are generally regarded identical with respect to their semantics.

(105) a. John gave the book to Mary.
b. John gave Mary the book.

If we assume that the selected preposition *to* in (105a) is determined purely by non-semantic rules, we can implement it as the identity function. The identity function will then (indirectly) pass the referential index of the NP of the PP up to to the argument slot of the verb, and the two sentences will yield two identical semantic representations.

Applying the Emptiness Principle on the MP and LP layers of PPs, new possibilities arise in analyzing the verbs with PP complements in the sentences in (103).

6.1 Implementing the Emptiness Principle

In the following section, we shall investigate how the Emptiness Principle can be implemented in the Matrix. The Matrix requires the process of semantic composition to be strictly monotonic, i.e. a process where nothing is deleted from the semantics once it has been added in the composition.

Kracht claims that all syntactic markers that have no interpretive impact have empty meaning. This means that any preposition marking a grammatical role (object, indirect object, etc.) denotes the identity function. There are two ways to implement this principle. Either, one could let each preposition have two separate entries, one semantically full and one semantically empty. Or, one could claim that for any preposition that may fill the role of a syntactic marker with no interpretative impact, the identity function is a subtype of the predicate introduced by the preposition. I have chosen to implement the latter.

In the Matrix, this can be implemented as introducing an identity relation instead of an identity function. Where the identity function returns the argument it is given, the identity relation is a relation yielding true for identical arguments. We introduce *id_rel* as a subtype of preposition predicates, and let each preposition that is selected for syntactic reasons, written as *(prep_)id_rel*, denote a predicate which is a subtype of both the specific preposition's predicate and *id_rel*.

As verbs may select either MPs, LPs or NPs, we allow both localiser predicates (*loc_rel*s) and modaliser predicates (*mode_rel*s) to denote the identity function.

The semantics of the identity relation corresponds to unifying two variables in a feature structure. But this cannot be done directly in the feature structure description, as it would violate the algebra for semantic construction described in Copestake et al. (2001). And it could be said to be more in the spirit of Kracht (2002) to make the identity relation explicit. This means that all subtypes of *id_rel* are identity relations, which are interpreted as the identity relation, i.e. '=' of type $\tau \rightarrow (\tau \rightarrow t)$ (where τ is an arbitrary type, and t is the type for truth values), when translated into typed lambda calculus. The MRSs containing identity relations can also be made subject to a truth-preserving post-processing, where the arguments of identity relations are unified, and the identity relations deleted. This post-processing can be performed by the same mechanism as the transfer process in the LOGON system, described in Chapter 4. The form of MRS transfer rules are repeated in (106). A post-processing rule for exchanging a variable x with a variable y, where the identity between the two variables is expressed in the same MRS, will in this formalism look as in (107). The rule in (107a) expresses that if an *id_rel* is present in the MRS, then every variable appearing as the first argument of an *id_rel* is replaced by the second argument of that same *id_rel* in all relations in the MRS (arguments are in our grammar always 'passed up', and never 'down', in *id_rel*s. The rule in (107b) states that an *id_rel* can be deleted except when that relation's first argument appears in other predicates in the MRS. This rule only applies for one place-predicates, and similar rules would have to be specified for each position the variable to be replaced can occur in.

(106) [CONTEXT:] INPUT [!FILTER] → OUTPUT

(107) a. {id_rel(x, y)} : {predsort(x)} → {predsort(y)}
b. {id_rel(x, _)} ! {predsort(x)} → { }

6.2 LP Selection

Having the *id_rel*s at hand, we can now fully specify the head of localisers. By constraining the value of the head feature KEYS.KEY to be of type *id_rel*, the semantic representation will only have a contentful representation of the localiser layer of the locative, whereas the modaliser layer is a semantically empty identity relation, as the Emptiness Principle states.

This means that a localiser is never specified as a lexical item, but is subsumed by a modaliser, with an identity relation instead of the modaliser predicate. And furthermore, that localisers only appear when selected. I believe this hierarchic organization of localiser-denoting signs as a subtype of combined modaliser/localiser-denoting signs gives an interpretable type hierarchy with respect to the generalizations we want to capture. This hierarchy expresses that location denoting signs are a special type of signs that in an unconstrained context denote an event, but when selected for syntactic reasons denote a location instead.

Part of the description of *fra* ('from') is shown in Figure 6.1. The description of a localiser preposition as a subtype of a full M+L preposition, with an *id_rel* as the first item on the RELS list is shown in Figure 6.2. The semantics of the PP *fra under et bord* ('from under a table') is given in Figure 6.3. I have assumed that only static PPs appear as LP complements, as none of the sentences in (108) seem well-formed.

(108) a. *Jon kom fra gjennom et vindu.
*John came from through a window.

b. *Jon kom fra av et teppe.
*John came from off a carpet.

c. *Jon kom fra mot et tre.
*John came from toward a tree.

As the ARG0 of modaliser EPs may be either an event (when acting as an intersective modifier) or a location, in LPs, the type must be underspecified to accommodate this. The value of ARG0 of modaliser EPs is therefore defined to be of type *individual*, and the context constrains the type of ARG0 further. For static MPs, which are syntactic adjuncts, this is done by unifying the INDEX of the *modifee*, which is an event, with the INDEX of the *modifier*. For directional MPs, which are syntactic complements, the type is constrained by the verb selecting it.

After a translation into lambda calculus, the PP *fra under et teppe* ('from under a carpet') will yield the semantics given in (109), where the *id_rel* is represented as '=', and ignoring the quantifier binding x.

(109) $[_{PP}\text{ from under a carpet}]' = \lambda e[(ci'(e, l1) \wedge = (l1, l2) \wedge under'(l2, x) \wedge carpet'(x)$

6.3 Verbs with PP Complements

6.3.1 The Verb *å bo* ('to live')

Bierwisch (1988) claims that some verbs, e.g. *wohnen* ('live') and *liegen* ('lie') take mandatory locative arguments. But they differ with respect to locating the subject. If we consider a sentence like (110), the subject needs not be located in any of the locations for the sentence to be true.

(110) Tore lives in Bergen, but works in Oslo.

This seems to imply that at least for some verbs, e.g. *to live* and *to work*, the locative cannot be an MP, as it would locate the event with its participants. But if we construe the verb to express a relationship between the subject and a location, we do not get these incorrect inferences. The relationship

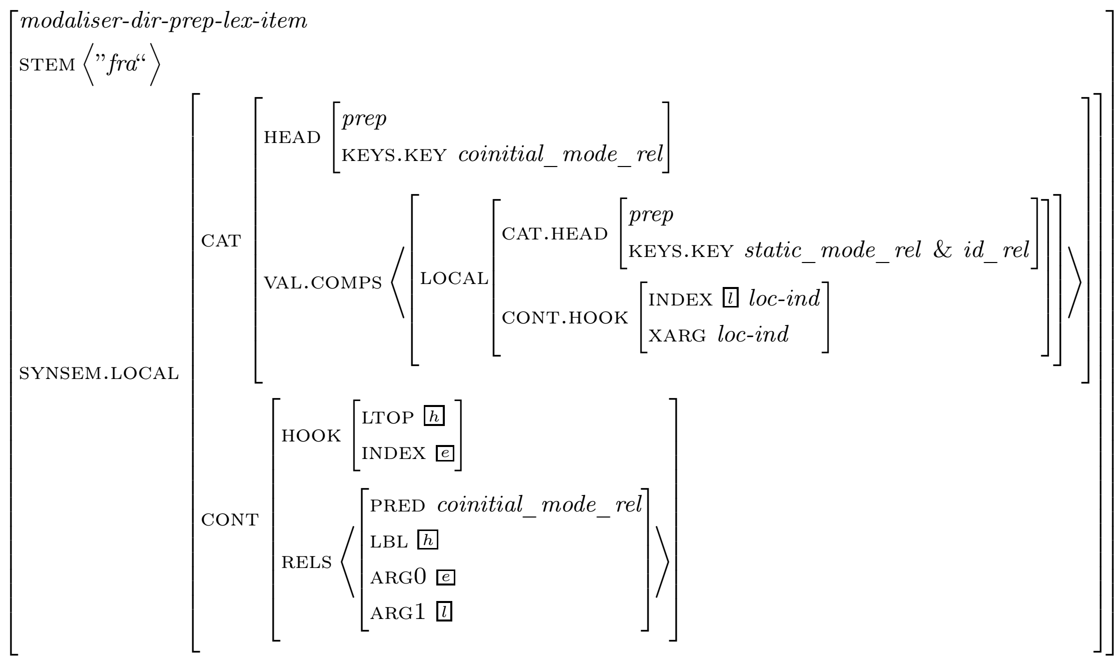


Figure 6.1: The modaliser *fra* ('from') with an LP on the COMP list

```
[ static-transitive-prep-lex-item
  STEM ⟨ "under" ⟩
  SYNSEM.LOCAL [ CAT  [ HEAD prep
                        VAL.COMPS ⟨ [ LOCAL [ CAT.HEAD noun
                                              CONT.HOOK.INDEX [x] ] ] ⟩ ]
                 CONT [ HOOK [ LTOP [h]
                               INDEX [l1] loc-ind
                               XARG [l2] loc-ind ]
                        RELS ⟨ [ LBL [h]
                                 PRED static_id_rel
                                 ARG0 [l1]
                                 ARG1 [l2] ],
                               [ PRED under_loc_rel
                                 LBL [h]
                                 ARG0 [l2]
                                 ARG1 [x] ] ⟩ ] ] ]
```

Figure 6.2: The preposition *under* ('under') as a localiser.

```
[ head-initial-comp-phrase
  LOCAL [ CAT  [ HEAD [ prep
                        KEYS.KEY [ m-relation
                                   PRED coinitial_mode_rel ] ]
                 COMPS null ]
          CONT [ HOOK [ LTOP [h1]
                        INDEX [e] ]
                 RELS ⟨ [ BL [h1]
                          PRED coinitial_mode_rel
                          ARG0 [e]
                          ARG1 [l1] ],
                        [ LBL [h1]
                          PRED static_id_rel
                          ARG0 [l1]
                          ARG1 [l2] ],
                        [ LBL [h1]
                          PRED under_loc_rel
                          ARG0 [l2]
                          ARG1 [x] ],
                        [ LBL [h2]
                          PRED some_q_rel
                          ARG0 [x]
                          RSTR [h3]
                          BODY [h4] ],
                        [ noun-relation
                          LBL [h5]
                          PRED bord_n_rel
                          ARG0 [x] ] ⟩
                 HCONS ⟨ [h3] =q [h5] ⟩ ] ] ]
```

Figure 6.3: The semantics of *fra under et bord* ('from under a table')

expressed by the verbs can be paraphrased as e.g. *the place of Tore's residence is in Bergen* and *the place of Tore's work/employment is in Oslo*.

We construe the feature structure description of the verb to select an LP, as shown for *å bo* ('to live') in Figure 6.4. The complement must have HEAD.KEYS.KEY value *id_rel*, and the INDEX and XARG of the complement are constrained to be of type *loc-ind*, as the identity relation takes two arguments of the same type. The INDEX value of the complement is unified with the ARG2 value of the verbal EP, to model that the verb expresses a relationship between an event, an entity and a location.

6.3.2 The Verb *å se* ('to see')

The verb *to see* seems to select a different type of complement. We saw that for syntactic reasons, we do not want the locative PP to attach to the NP in (104). And treating the PP as a static locative adjunct gives us the wrong semantics, as the subject may well be located outside the region denoted by the PP.

This construction resembles constructions discussed in Hellan (2003), examples given below.

(111) a. Han hørte henne *synge*
He heard her sing

b. Vi anså ham *som uegnet*
We considered him as non-suited

Hellan (2003) names these verbs (non-resultative) secondary predicatives, and claim that these take a propositional ARG2 (direct objects). This claim seems to hold for the sentence in (104), based on the paraphrase in (112b) and the observation that the expletive in (112c) can be promoted, which Hellan mentions as criterion for identifying secondary predicatives.

(112) a. Jon så Mari på damedo
Jon saw Mari in ladies'-room.DEF
'John saw Mary in the ladies' room'

b. Jon så at Mari var på damedo
Jon saw that Mary was in ladies'-room.DEF
'John saw that Mary was in the ladies' room'

c. Jon så det stå en kvinne på jentedo.
Jon saw there stand a woman in ladies'-room.DEF
'John saw a woman standing in the ladies' room

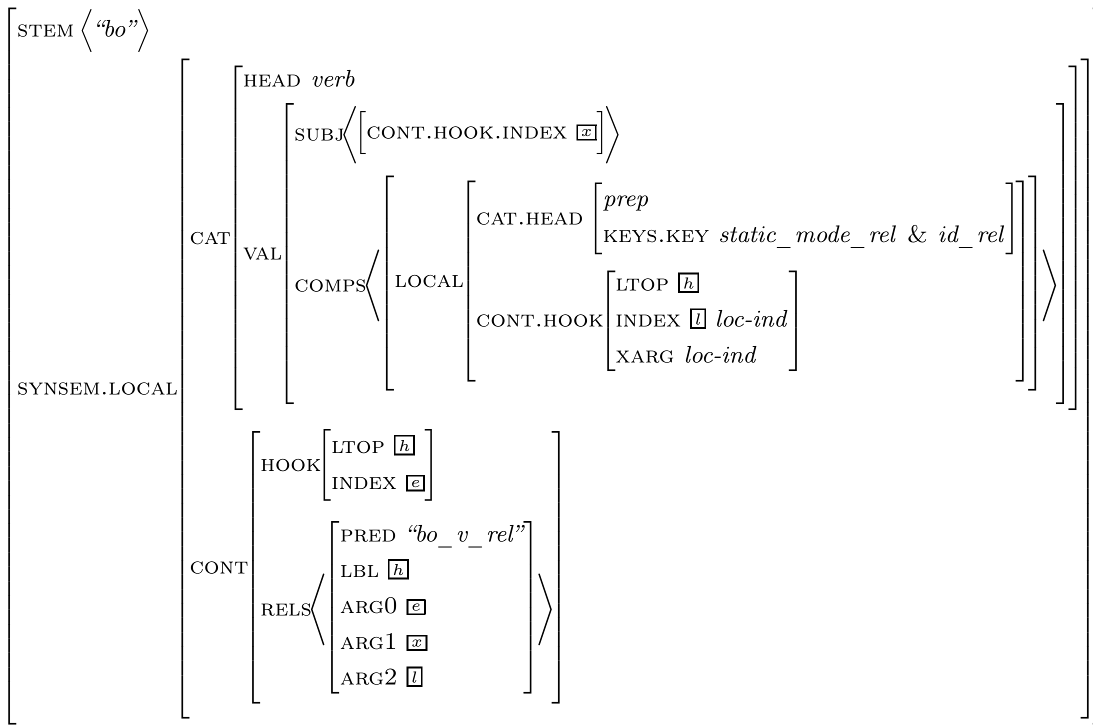


Figure 6.4: The verb *å bo* ('to live')

I will not analyze this use of perception as verbs here, but refer to Hellan (2003) for the treatment of non-resultatives in NorSource. NorSource is a grammar based on the Matrix, but extended with some features, which also makes it cope with e.g. different type of predicative constructions.

Note that the predicative use of perception verbs with object orientation differs from the use of *å se* ('to see') in combination with directional PPs, seen in (113). In these constructions, the mover of the directional events seems to be the observation, moving from the observer to the object of observation (i.e. the opposite direction of how the visual impression, or the light, actually moves). And in (113d), we even find an ambiguity in whether John or Mary is located in the ladies' room, i.e. whether the source of the sound or the place of observation is the ladies' room.

(113) a. Jon så Mari (inn) i øynene. *cofinal*
Jon saw Mari (in$_{cofinal}$) in eyes.

b. Jon så Mari gjennom vinduet. *transitory*
Jon saw Mari through window.DEF.

c. Jon så Mari fra damedo. *coinitial*
Jon saw Mari from ladies'-room.

d. Jon hørte Mari fra damedo. *coinitial/cofinal*
Jon heard Mari from ladies'-room.

6.3.3 The Verb *å legge* ('to lay/put')

Verbs of putting have the property that they seem to combine with both static and directional locatives.

(114) a. Jon la smøret inne i kjøleskapet.
Jon put butter.DEF in$_{static}$ in fridge.DEF
'John put the butter inside the fridge.'

b. Jon la smøret inn i kjøleskapet.
Jon put butter.DEF in$_{cofinal}$ in fridge.DEF
'John put the butter into the fridge.'

It may be argued that the focus is more on the end-state in (114a) and on the motion itself in (114b). But truth-conditionally, the semantics of the two sentences seem to be the same. My claim is that this verb alternate between taking a directional and a static PP complement, but yield the same of semantics. The two uses (i.e. with static or directional complement) can be treated uniformly with the MP/LP distinction, by letting the LP be the complement semantically. Verbs of putting then select for PP objects in static or cofinal mode, with an *m_ id_ rel* as the key relation of the PP. Part of the lexical entry for *legge* ('lay'/'put') is shown in Figure 6.5.

The hierarchy of *mode_ rel*s is altered from Chapter 5 in order to underspecify for three reasons: (i) to underspecify between the directional modes (e.g. *under* as ambiguous between cofinal and transitory mode), (ii) to underspecify for selectional purposes (e.g. verbs of putting select cofinal or static PPs) and (iii) to define negative or complementary modes (e.g. *not_ cofinal_ mode_ rel*). The extended *mode_ rel* hierarchy is shown in Figure 6.6.

Verbs of Putting in the ERG

Verbs of putting in the ERG are analyzed as taking PP complements. But, differing from our analysis, the semantics of the PP complement is a message relation, as described in Chapter 5.1. This resembles the analysis I described above predicative constructions in Hellan (2003), in that the message relation is meant to describe a proposition, i.e. the proposition expressed by the subordinate clause of the paraphrase for non-resultatives, as in sentence (115b), and the result or end-state for resultatives, as in sentence (115c).

(115) a. Jon hørte Mari synge.
Jon heard Mari sing.
'Jon heard Mari sing.'

b. Jon hørte at Mari sang.
Jon heard that Mari sang.
'Jon heard that Mari was singing.'

c. Han malte huset rødt.
He painted house.DEF red
'He painted the house red.'

But the semantics of directional PPs have an inherent motion, and the proposition expressed by a directional PP complement of e.g. *put* is then a proposition that the end state of the event is a *state* of directed motion. This seem contradictory, as states are inherently atelic events, while some directionals, e.g. *into* are telic motion events. Our analysis avoids this, by letting the verb select a location as the semantic argument both for the static and the directional (cofinal) PPs.

The MRS produced by the ERG for *put* with a directional PP complement is shown in Figure 6.7, while the MRS for an equivalent construction in our grammar is shown in 6.8. The MRSs are shown both as a feature structure

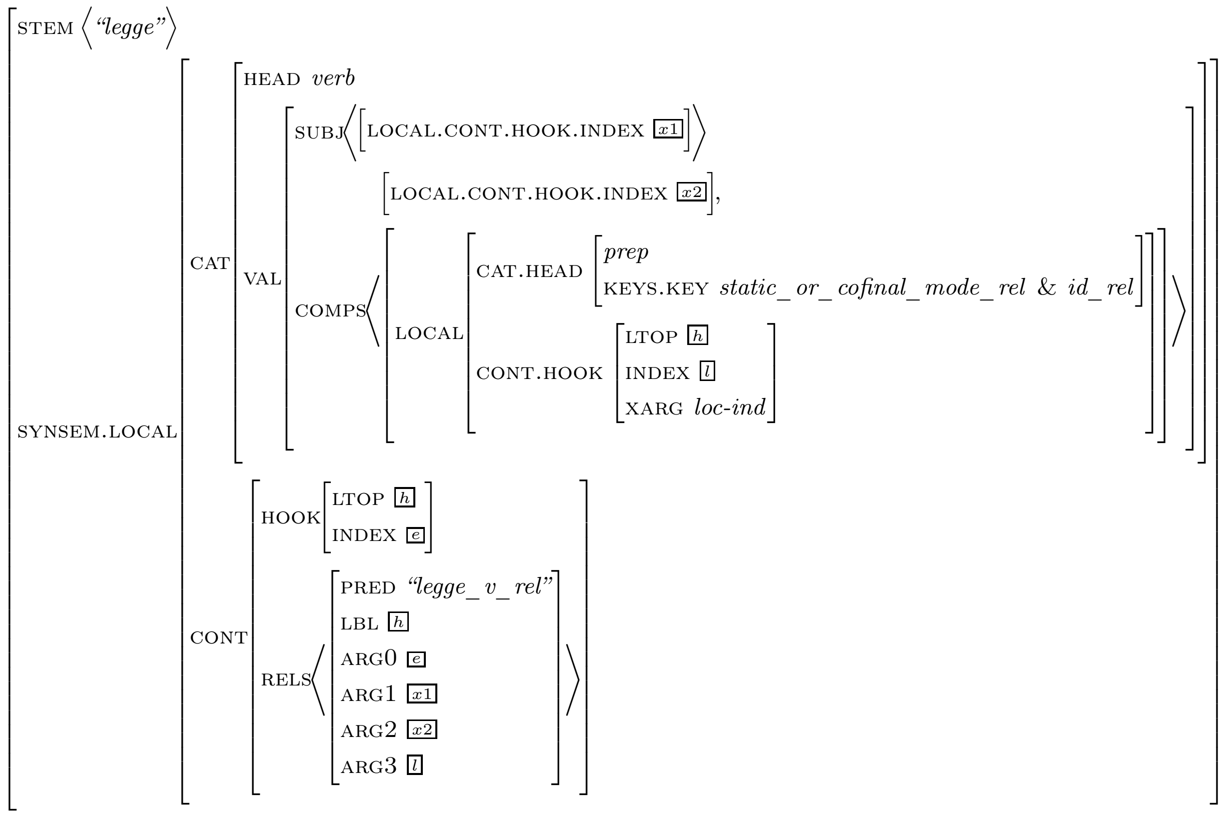


Figure 6.5: The verb *å legge* ('to put/lay')

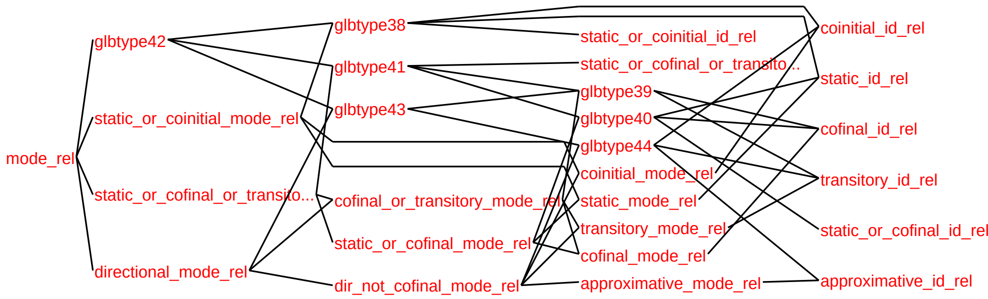


Figure 6.6: *LKB printout:* The subtypes of *mode_rel*

description and as an indexed MRS[1].

The Verb *å gi* ('to give')

In the beginning of this Chapter, we noted that for dative alternating verbs, there is a general agreement in that the role of the preposition is to mark the indirect object, without contributing semantically. We have so far seen verbs which select PP complements with a specific mode, but where the LP is the semantic argument. By implementing dative alternating verbs in a similar fashion, I want to show how the use of the emptiness principle can be extended. For this class of verbs, both the modaliser and the localiser of the PP complement are semantically vacuous, i.e. they both denote the identity function. The verb *å gi* ('to give') then selects a PP complement where both the HEAD features KEYS.KEY and KEYS.ALTKEY are *id_rel*s. This is shown in Figure 6.9. Note that *til* ('to') is decomposed into the modaliser *cofinal_mode_rel* and the localiser *ved_loc_rel*. This will be discussed in the next section.

Just as the ARG0 of the modaliser EP must be underspecified to accommodate both intersective modification and selected LPs, the ARG0 of the localiser EP must be underspecified to accommodate both *loc-ind*, for LPs, or *ref-ind*, when both layers are selected, as we saw for dative alternating verbs. The ARG0 of the *loc_rel* is unified with the ARG1 of the *mod_rel* and the HOOK.XARG of the lexical item. Constraining the XARG of the PP complement, whether it is a verb or a preposition selecting a PP, ensures that the identity relations come out to be identity statements between variables of same type, i.e. of type *loc-ind* for LP selection and of type *ref-ind* for dative alternating verbs.

[1] An indexed MRS is a tuple $\langle GT, MI, R, C\rangle$ where GT is the global top, MI is the main index, R is a bag of EPs and C is a bag of handle constraints.

Feature Structure Description:

$$
\begin{bmatrix}
\textit{subjh\_rule\_decl} \\
\textsc{local} \begin{bmatrix}
\textsc{cat.head}\ \textit{verb} \\
\textsc{cont} \begin{bmatrix}
\textsc{hook} \begin{bmatrix} \textsc{ltop}\ \boxed{h1} \\ \textsc{index}\ \boxed{e2} \end{bmatrix} \\
\textsc{rels} \left\langle
\begin{array}{l}
\begin{bmatrix} \textsc{pred}\ \textit{prpstn\_m\_rel} \\ \textsc{lbl}\ \boxed{h1} \\ \textsc{marg}\ \boxed{h5} \\ \textsc{psv}\ \boxed{u3} \\ \textsc{tpc}\ \boxed{u4} \end{bmatrix},
\begin{bmatrix} \textsc{pred}\ \textit{\_a\_q\_rel} \\ \textsc{lbl}\ \boxed{h6} \\ \textsc{arg0}\ \boxed{x7} \\ \textsc{body}\ \boxed{h9} \\ \textsc{rstr}\ \boxed{h8} \end{bmatrix},
\begin{bmatrix} \textsc{pred}\ \textit{\_man\_n\_rel} \\ \textsc{lbl}\ \boxed{h10} \\ \textsc{arg0}\ \boxed{x7} \end{bmatrix}, \\
\begin{bmatrix} \textsc{pred}\ \textit{\_put\_v\_1\_rel} \\ \textsc{lbl}\ \boxed{h11} \\ \textsc{arg0}\ \boxed{e2} \\ \textsc{arg1}\ \boxed{x7} \\ \textsc{arg2}\ \boxed{x12} \\ \textsc{arg3}\ \boxed{h13} \end{bmatrix},
\begin{bmatrix} \textsc{pred}\ \textit{prpstn\_m\_rel} \\ \textsc{lbl}\ \boxed{h13} \\ \textsc{marg}\ \boxed{h14} \\ \textsc{psv}\ \boxed{u16} \\ \textsc{tpc}\ \boxed{u15} \end{bmatrix},
\begin{bmatrix} \textsc{pred}\ \textit{\_a\_q\_rel} \\ \textsc{lbl}\ \boxed{h17} \\ \textsc{arg0}\ \boxed{x12} \\ \textsc{body}\ \boxed{h19} \\ \textsc{rstr}\ \boxed{h18} \end{bmatrix}, \\
\begin{bmatrix} \textsc{pred}\ \textit{\_ball\_n\_rel} \\ \textsc{lbl}\ \boxed{h10} \\ \textsc{arg0}\ \boxed{x7} \\ \textsc{arg1}\ \boxed{u21} \end{bmatrix},
\begin{bmatrix} \textsc{pred}\ \textit{\_into\_p\_rel} \\ \textsc{lbl}\ \boxed{h22} \\ \textsc{arg0}\ \boxed{e24} \\ \textsc{arg1}\ \boxed{x12} \\ \textsc{arg2}\ \boxed{x23} \end{bmatrix}
\begin{bmatrix} \textsc{pred}\ \textit{\_a\_q\_rel} \\ \textsc{lbl}\ \boxed{h25} \\ \textsc{arg0}\ \boxed{x23} \\ \textsc{body}\ \boxed{h27} \\ \textsc{rstr}\ \boxed{h26} \end{bmatrix}, \\
\begin{bmatrix} \textsc{pred}\ \textit{\_garden\_n\_rel} \\ \textsc{lbl}\ \boxed{h28} \\ \textsc{arg0}\ \boxed{x23} \end{bmatrix}
\end{array}
\right\rangle \\
\textsc{hcons} \left\langle
\begin{array}{l}
\boxed{h5} =_q \boxed{h11}\ ,\ \boxed{h8} =_q \boxed{h10}\ ,\ \boxed{h14} =_q \boxed{h22}\ , \\
\boxed{h18} =_q \boxed{h20}\ ,\ \boxed{h26} =_q \boxed{h28}
\end{array}
\right\rangle
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Indexed MRS:

$\langle\ h1, e2,$
$\{h1$: prpstn_m($h5$, u3, u4),
$h6$: _a_q($x7$, $h9$, $h8$),
$h10$: _man_n($x7$),
$h11$: _put_v_1($e2$, $x7$, $x12$, $h13$),
$h13$: prpstn_m($h14$, u16, u15),
$h17$: _a_q($x12$, $h19$, $h18$),
$h20$: _ball_n($x12$, i21),
$h22$: _into_p($e24$, $x12$, $x23$),
$h25$: _a_q($x23$, $h27$, $h26$),
$h28$: _garden_n($x23$)},
$\{h5 =_q h11,\ h8 =_q h10,\ h14 =_q h22,\ h18 =_q h20,\ h26 =_q h28\}\ \rangle$

Figure 6.7: ERG analysis of *A man puts a ball into a garden*

Feature Structure Description:

```
[head-final-subj-phrase
 LOCAL [CAT.HEAD verb
        CONT [HOOK [LTOP  [h1]
                    INDEX [e2]]
              RELS ⟨ [PRED some_q_rel
                      LBL  [h3]
                      ARG0 [x4]
                      BODY [h6]
                      RSTR [h5]],
                     [PRED mann_n_rel
                      LBL  [h7]
                      ARG0 [x4]],
                     [PRED legge_v_rel
                      LBL  [h1]
                      ARG0 [e2]
                      ARG1 [x4]
                      ARG2 [x8]
                      ARG3 [l9]],
                     [PRED some_q_rel
                      LBL  [h10]
                      ARG0 [x8]
                      BODY [h12]
                      RSTR [h11]],
                     [PRED ball_n_rel
                      LBL  [h13]
                      ARG0 [x8]],
                     [PRED cofinal_id_rel
                      LBL  [h14]
                      ARG0 [l9]
                      ARG1 [l14]],
                     [PRED i_loc_rel
                      LBL  [h15]
                      ARG0 [l8]
                      ARG1 [x16]],
                     [PRED some_q_rel
                      LBL  [h16]
                      ARG0 [x15]
                      BODY [h18]
                      RSTR [h17]],
                     [PRED hage_n_rel
                      LBL  [h19]
                      ARG0 [x16]] ⟩
              HCONS ⟨ [h5] =q [h7] , [h11] =q [h13] , [h17] =q [h19] ⟩ ]]]
```

Indexed MRS:

<h1,e2,
{h3: some_q_rel(x4, h6, h5),
h7: mann_n_rel(x4),
h1: legge_v_rel(e2, x4, x8, l9),
h10: some_q_rel(x8, h12, h11),
h13: ball_n_rel(x8),
h1: static_id_rel(l9, l14),
h1: i_loc_rel(l14, x15),
h16: some_q_rel(x15, h18, h17),
h19: hage_n_rel(x15)},
{$h5 =_q h7$,
$h11 =_q h13$,
$h17 =_q h19$}>

Figure 6.8: Our analysis of *En mann legger en ball i en hage* (‘A man puts/places a ball in a garden’).

```
[ STEM ⟨ "gi" ⟩
  SYNSEM.LOCAL [ CAT [ HEAD verb
                       VAL [ SUBJ ⟨ [ LOCAL.CONT.HOOK.INDEX [x1] ] ⟩
                             COMPS ⟨ [ LOCAL [ CAT.HEAD noun
                                               CONT.HOOK.INDEX [x2] ] ],
                                     [ LOCAL [ CAT.HEAD [ prep
                                                          KEYS [ KEY cofinal_mode_rel & id_rel
                                                                 ALTKEY ved_loc_rel & id_rel ] ]
                                               CONT.HOOK [ INDEX [x3] ref-ind
                                                           XARG ref-ind ] ] ] ⟩ ] ]
                 CONT [ HOOK [ LTOP [h]
                               INDEX [e] ]
                        RELS ⟨ [ PRED "gi_v_rel"
                                 LBL [h]
                                 ARG0 [e]
                                 ARG1 [x1]
                                 ARG2 [x2]
                                 ARG3 [x3] ] ⟩ ] ] ]
```

Figure 6.9: The verb *å gi* ('to give')

6.4 The Analysis of *til*

Herskovits (1986): The Three Basic Topological Prepositions

Herskovits (1986) groups the three prepositions *at, on* and *in* under the label "The three basic topological prepositions". These three prepositions have in common that they express "cognitively basic, essentially topological, relations" (Herskovits, 1986, p. 127). Herskovits note that in contexts where one of the lexicalized prepositions is deleted, one can often induce the correct preposition.

> "In this process, the distinctions between *at, on* and *in* are neutralized; yet, inasmuch as the reference object is identified with a privileged space (the space enclosed, or above and adjacent, etc.), no misunderstanding will arise. The same deletions would not be acceptable with other prepositions"
> (Herskovits, 1986, p. 34).

In (116), we see a few examples of deleted prepositions.

(116) a. The bedroom is a pleasant place to work.
b. The worst place for a store is the street corner.
c. The bed is the best place to put the blanket.
d. *The best place to hide is (under) the bed.

As we see from the examples above, the reference object does not always single out a preposition to express this "privileged space", as a blanket may well be put both *in* and *on* a bed, and a shop may well be placed *on* and *at* a corner. The semantic difference between these uses seem marginal, however.

Trujillo (1995): Lexicalized Prepositions

Given the arguments above that objects can be identified with a certain privileged space, Trujillo (1995), in his MT approach to analyzing prepositions, claims that these prepositions are best treated as a part of the lexical entry of the noun. Trujillo therefore labels these prepositions *lexicalized prepositions*. Frequency counts of co-occurrences for some nouns and the lexicalized prepositions, gathered from the LOB corpus, are given in Table 6.1.

Trujillo (1995) makes several observations regarding lexicalized prepositions. First, one must separate *lexicalized* prepositions from *literal*, such that we can account for both the interpretations in (117).

	bus	coach	car	building	house	station	table	chair	seat	window
at	0	0	0	1	8	10	22	1	1	7
in	3	3	30	8	52	1	0	17	7	8
on	2	2	2	0	2	0	28	4	9	1

Table 6.1: Co-occurrences of lexicalized prepositions and some nouns (Trujillo, 1995, p .180)

(117) John is on the bus.

1. John is *inside* the bus.
2. John is *on (top of)* the bus.

This could be done by checking if the NP is preceded by it's lexicalized preposition. If it is, the preposition is ambiguous between the lexicalized and literal preposition.

In the translation of lexicalized prepositions, we then get that lexicalized prepositions of the source language translate into lexicalized prepositions of the target language, and that it is up to their respective grammars to select correct preposition. A proposal for representing lexicalized prepositions is given in Trujillo (1995, p. 209-212).

If we had wanted to implement a treatment of lexicalized prepositions in our grammar, we could let lexicalized preposition introduce an abstract type of localiser relations, e.g. *lex_ loc_ rel*, in the MRS.

The special type of localiser relation above, could give a more correct analysis of *til* ('to'), based on the fact that the inferences in (118) seem to differ with respect to the lexicalized prepositions.

(118) a. John løp til treet. ⇒ John er ved treet.
John ran to tree.DEF. ⇒ John is at tree.DEF.

b. John løp til skolen. ⇒ John er på skolen.
John ran to school.DEF. ⇒ John is on school.DEF.

c. John løp til sentrum. ⇒ ?John er i sentrum.
John ran to center ⇒ ?John is in center

6.5 Summary

In this Chapter, we have seen how LPs are implemented. This was done by introducing identity relations between arguments of the same type. We have seen how modalisers and some verbs select LP complements and how they semantically select a location as the argument. This was implemented without violating constraints on the semantic composition, by indirectly passing the location index through the modaliser layer with the identity relation. We have argued that the semantics of some classes of verbs are given a more precise treatment by giving them a location as a semantic argument. And finally, we discussed a more general treatment for a small class of lexicalized locative prepositions.

In the following chapter, we shall use the modaliser and localiser dimensions in order to classify locatives. We shall also investigate the syntactic and semantic properties of a small class of prepositions which marks the proceeding PP as directional. We shall also explore the possibilities of using our decompositional approach to locatives in a semantic transfer based translation system, where the decomposition enables us to make more accurate predictions and more principled translation of locatives.

Chapter 7

Decomposition and Translation

The modaliser/localiser distinction provides us with two dimensions along which locatives can be classified. Kracht (2002) argues that the number of modes is limited, while the number of localisers is in principle unlimited. The localiser dimension describes the location that motion occurs with respect to, and the modaliser dimension the type of motion occurring, which in turn has implications for the type of syntactic contexts the different PPs occur in, telicity, etc. In (120), we see how the different locatives in (119) express different types of motion with respect to the same locations.

(119) a. Per løp til skolen.
Per ran to school.DEF.
'Per ran to (the) school.'

b. Per løp hjem.
Per ran (to) home.
'Per ran home.'

c. Per løp hit.
Per ran (to) here.
'Per ran here.'

d. Per løp mot skolen.
Per ran toward school.DEF.
'Per ran toward (the) school.'

e. Per løp hjem.
Per ran (toward) home.
'Per ran homeward.'

f. Per løp hitover.
Per ran (toward) here
'Per ran hitherward.'

(120) a. Per løp til L ('Per ran to L')
where $L =$ *(på/ved) skolen* ('at (the) school') in (119a)
$L =$ *hjemme* ('at home') in (119b)
$L =$ *her* ('here') in (119c)

b. Per løp mot L ('Per ran toward L')
where $L =$ *(på/ved) skolen* ('at (the) school') in (119d)
$L =$ *hjemme* ('at home') in (119e)
$L =$ *her* ('here') in (119f)

It is also a well known fact that PP modifiers have an impact on the telicity of events. This can be seen by introducing temporal PPs to test whether an event is telic or atelic, dividing events into the so-called Vendler classes. The temporal PP *på en time* ('in an hour') modifies telic events only, while the PP *i en time* ('for an hour') modifies atelic events only. In terms of Vendler classes, these tests separate 'activity' motion verbs, which are inherently atelic, from 'accomplishment' motion verbs, which are inherently telic.

We see in (121) that static locatives modifying motion verbs denote atelic events, while cofinal locatives modifying motion verbs denote telic events. In (121c), the preposition is ambiguous between a cofinal (telic) and a static (atelic) reading, as the English translation also shows, and the event may therefore be modified by both types of temporal PPs.

(121) a. Per løp til butikken på en time/*i en time. *(cofinal)*
Per ran to shop.DEF on an hour/*in an hour.
'Per ran to the shop in an hour/*for an hour.'

b. Per løp ved butikken *på en time/i en time. *(static)*
Per ran at shop.DEF *on an hour/in an hour.
'Per ran at the shop *in an hour/?for an hour.'

c. Per løp på butikken på en time/i en time. *(ambiguous)*
Per ran on shop.DEF on an hour/in an hour.
'Per ran to/in the shop in an hour/for an hour.'

However, when the verb is used in isolation[1], only atelic readings are acceptable, as seen in sentence (122).

(122) Per løp *på en time/i en time.
Per ran *on an hour/in an hour.
'Per ran *in an hour/for an hour.'

[1]It should be noted that the transitive *løpe* ('run') as in *Per løp en kilometer* ('Per ran a kilometer') is telic, while the intransitive verb is atelic.

From this, we conclude that cofinal locatives introduce telicity. We will not incorporate telicity in our grammar, but this discussion sheds some light on how different modes are related to telicity, and the grammar could be extended to model telicity based on the modes of adverbial locative modifiers.

We decomposed locatives semantically according to the modaliser and localiser dimensions in the previous chapter. In this chapter, we will look at how different locatives can be classified and analyzed, with respect to these two dimensions. We will also see how this can be used in translation systems based on semantic transfer.

7.1 Transitive Locatives

In Chapter 5, we suggested that the locative *av* ('off') could be decomposed into a combination of the modaliser *coinitial* and the localiser *på* ('on') (cf. Section 5.4). We can define a number of locatives as a combination of a modaliser and a localiser. This can be tested on basis of the definitions of the modalisers, and conceptualizing the modaliser and localiser as separate prepositions. For instance, *av* ('off') can be thought of as the coinitial modaliser applied on the localiser *på* ('on') or the complex (but not so well-formed) PP *fra på* ('from on'), where the modaliser and localiser are lexicalized separately. Similarly, *over* ('over/across') be thought of as the transitory mode of *på*('on') or the complex PP *gjennom/inn-og-ut-av på* ('through/in-and-out-of on'). Generally, the static mode of a localiser seems to express the concept of the localiser function. We can therefore test the different modes with respect to a localiser, by inserting the static preposition as a conceptualization of the localiser.

Let P_L be a static preposition conceptualizing the localiser function L, where L takes an object as its argument and denotes a parameterized neighborhood. We can then test:

- **static:** What is the static locative P_{st} with respect to the localiser L? Or, for which preposition P_{st} is the following true:
  $[P_{st}]' = \lambda x.\{(\mathcal{E} : \forall t \in time'(\mathcal{E}))(loc'(\mathcal{E}) \subseteq L(x)(t))\}$?
- **coinitial:** What is the coinitial locative P_{ci} with respect to L? Or, for which preposition is the following true:
  $[P_{ci}]' = \lambda x.\{\mathcal{E} : ci^*(\mu(\mathcal{E}),\ L(x),\ time'(\mathcal{E}))\}$?
- **cofinal:** What is the cofinal locative P_{cf} with respect to L? Or, for which preposition is the following true:
  $[P_{cf}]' = \lambda x.\{\mathcal{E} : cf^*(\mu(\mathcal{E}),\ L(x),\ time'(\mathcal{E}))\}$?

- **transitory:** What is the transitory locative P_{tr} with respect to L?
  Or, for which preposition is the following true:
  $[P_{tr}]' = \lambda x.\{\mathcal{E} \ : \ tr^*(\mu(\mathcal{E}),\ L(x),\ time'(\mathcal{E}))\}$?

- **approximative:** What is the approximative locative P_{ap} with respect to L?
  Or, for which preposition is the following true:
  $[P_{ap}]' = \lambda x.\{\mathcal{E} \ : \ ap^*(\mu(\mathcal{E}),\ L(x),\ time'(\mathcal{E}))\}$?

Testing this for the localiser *på* ('on'), we receive the results shown in Table 7.1. It is hard to find a good candidate for the approximative mode, and I

L	**static**	**coinitial**	**cofinal**	**transitory**	**approximative**
på	på	av	på	over	-

Table 7.1: The different modes of *på*

believe that very few transitive prepositions express this mode in Norwegian. I expect languages to vary with respect to which combinations of mode and localiser are lexicalized, as we find case marking languages with a case corresponding to the approximative mode.

We can now proceed in a similar fashion for other locative prepositions, as shown in Table 7.2.

L	**static**	**coinitial**	**cofinal**	**transitory**	**approximative**
på	på	av	på	over	-
i	i	ut/opp fra/av	i	gjennom	-
ved	ved	fra	til	om/via/forbi	mot
under	under	fra under	under	under	-
over	over	fra over	inn/ut over	over	-
mellom	mellom	fra mellom	inn/... mellom	mellom	-

Table 7.2: The different modes of localisers

7.1.1 Translation

There are some interesting contrasts between Norwegian and English with respect to lexicalizations of modes. For instance, *into* and *onto* express the cofinal modes of *in* and *on*, respectively, as seen in the translated sentences from OMC in (123). And there is reason to believe that the locatives *above* and *below* express static mode only, as we see in the sentences in (124).

(123) a. *Title: Under the Evening Sky, Author: Finn Carling, Translator: Louis A. Muinzer*

'Det virket som om gjestens ord måtte synke gjennom ham, slik skjell skjener glimtende mot bunnen når man kaster dem *i* havet.'
'It seemed as if his guest's words had to sink through him like shells that shine gleaming towards the bottom when one casts them *into* the sea.'

b. *Title: The Joker, Author: Lars Saabye Christensen, Translator: Michael Nordby*
'Jeg hoppet inn *på* fortauet og så en rad med hvite ansikter forsvinne nedover mot Bislet og byen.'
'I hopped *onto* the sidewalk and saw a row of white faces disappear down toward Bislett Stadium and the city.'

(124) a. Musa hoppet over$_{tr}$ bordet på et sekund ⇒ The mouse jumped over/across/?above the table in one second

b. Musa løp under$_{cf}$ bordet og gjemte seg på et sekund ⇒ The mouse ran under/?below the table to hide in one second

As we saw in Chapter 5.6, many locatives are mode ambiguous, and disambiguation is a difficult task. We shall see how the different modes can be associated with different syntactic contexts later, and how this may help to disambiguate the mode.

7.2 Intransitive Locatives

In Faarlund et al. (1997), it is argued that all locatives are prepositions, regardless of whether they take complements or not. In earlier descriptions of Norwegian, intransitive locatives were classified as adverbs. In (125), from Faarlund et al. (1997, p. 412), the italicized words are therefore considered intransitive prepositions[2].

(125) a. Jeg satte sykkelen *inn.*
I placed bicycle.DEF inside.

b. Han måtte vente *utenfor.*
He must wait outside

c. Gi meg pengene *tilbake.*
Give me money.DEF back.

[2]I name these prepositions 'intransitive prepositions', as they often occur without a complement. But, as we shall see, they may occur with both NP and PP complements. The term 'intransitive preposition' is therefore somewhat misleading.

The same goes for the pro-words in (126) below.

(126) a. Jeg satte sykkelen *dit*.
I placed bicycle.DEF there.

b. Han måtte vente *her*.
He must wait here.

This classification is based on the common syntactic distribution of locatives, abandoning the former criterion that prepositions relate the prepositional object to other parts of the sentence. In addition to the syntactic distribution, the semantics of intransitive and transitive prepositions are strikingly similar. But as the transitive prepositions receive their denotation by mapping the prepositional complement to a parameterized set of regions by a localiser function, intransitive prepositions generally map from a variable induced from the context to a parameterized set of regions.

Take for instance the pro-word *her* ('here'). When used deictically, it means something like "proximal to the speaker". Similarly, *der* ('there') means "distal to the speaker"[3]. As a localiser, we let *her* be interpreted as a parameterized neighborhood, building on a function *proximal*:

> Let r be a region. Denote by $proximal(r)$ the region close to r minus r itself.

We furthermore base one region on the location of the speaker (or whatever is the contextually relevant object). We therefore must assume a function *speaker*, which for a given context return the speaker in the context.

The localiser *her* can now be defined in (127).

(127) $\text{her}' = \lambda t\{r : r \subseteq proximal(loc(speaker(c), t)), r \text{ a region}\}$

We can try to instantiate these words in the test described in the previous section. We have to modify the test, as *her* is intransitive, and denotes a full localiser phrase, not only a localiser, as described above. This means that the locatives (modaliser and localiser combined) building on the localiser *her* denote sets of events rather than functions from objects to sets of events. We omit the lambda operator from the test we constructed for identifying different locatives with respect to a localiser, assuming that the argument of the localiser function is given from the real-world context.

[3] These pro-words may be used anaphorically as well. When used anaphorically, it takes as an argument the relevant object from the linguistic context rather than the real-world context. The notion of context in this section could therefore be generalized, to reflect this.

Let *her* be a localiser phrase consisting of the localiser *her*, where *her* denotes a location
$L = \lambda t.\{r \ : \ r \subseteq \mathit{proximal}(\mathit{loc}(\mathit{speaker}(c), t)), r \text{ a region}\}$. We can then test:

- **static:** What is the static locative P_{st} with respect to the localiser L?
  Or, for which preposition P_{st} is the following true:
  $[P_{st}]' = \{(\mathcal{E} \ : \ \forall t \ \in \ \mathit{time}'(\mathcal{E}))(\mathit{loc}'(\mathcal{E}) \subseteq L(t))\}$?
  $P_{st} = \mathit{her}$
- **coinitial:** What is the coinitial locative P_{ci} with respect to L?
  Or, for which preposition is the following true:
  $[P_{ci}]' = \{\mathcal{E} \ : \ ci^{*}(\mu(\mathcal{E}), \ L, \ \mathit{time}'(\mathcal{E}))\}$?
  $P_{ci} = \mathit{herfra}$
- **cofinal:** What is the cofinal locative P_{cf} with respect to L?
  Or, for which preposition is the following true:
  $[P_{cf}]' = \{\mathcal{E} \ : \ cf^{*}(\mu(\mathcal{E}), \ L, \ \mathit{time}'(\mathcal{E}))\}$?
  $P_{cf} = \mathit{hit}$
- **transitory:** What is the transitory locative P_{tr} with respect to L?
  Or, for which preposition is the following true:
  $[P_{tr}]' = \{\mathcal{E} \ : \ tr^{*}(\mu(\mathcal{E}), \ L, \ \mathit{time}'(\mathcal{E}))\}$?
  $P_{tr} = \mathit{gjennom/via/forbi\ her?}$
- **approximative:** What is the approximative locative P_{ap} with respect to L?
  Or, for which preposition is the following true:
  $[P_{ap}]' = \{\mathcal{E} \ : \ ap^{*}(\mu(\mathcal{E}), \ L, \ \mathit{time}'(\mathcal{E}))\}$?
  $P_{ap} = \mathit{hitover}$

We find that all the words above are related to the localiser *her* in a systematic way, and that this pattern is captured by the set of modes. The only mode which has no lexicalization of *her* ('hit') is the transitory mode.

The pattern for *der* is identical, as well as for the intransitive locatives, such as *inne* ('in'/'inside'), *oppe* ('up'/'upstairs'), and *hjemme* ('at home'). Again, the semantics of the localiser is induced from the context, i.e. an implicit function on the context of the utterance. And for these words, we also find instances of the transitory mode. By instantiating the intransitive locatives in the test above, we get the results shown in Table 7.3. The morphological pattern is shown in the bottom part of the table[4].

[4]Note that there are several exceptions to this pattern in the table.

L	**static**	**coinitial**	**cofinal**	**transitory**	**approximative**
her	her	herfra	hit	-	hitover
der	der	derfra	dit	-	ditover
inne	inne	innenfra	inn	innom?	innover
oppe	oppe	ovenfra	opp	oppom?	oppover
borte	borte	bortenfra	bort	bortom?	bortover
hjemme	hjemme	hjemmefra	hjem	hjemom?	hjemover
⋮					
her/der	(-er)	-erfra	-it	∅	-itover
inne, oppe...	-e	-enfra	-	-om?	-over

Table 7.3: The different modes of contextually inferred localisers

7.2.1 Translation

When translating the pro-words, the situation is opposite to the one for the prepositions *in* and *on*, namely that Norwegian has lexicalized the cofinal modes (*hit* and *dit*), whereas the English pro-words (*here* and *there*) are ambiguous with respect to static/cofinal mode. This would result in ambiguity translating from English to Norwegian, but does not cause ambiguity problems translating in the opposite direction. Another interesting case is the coinitial mode of *her*. In Norwegian, it is lexicalized as *herfra*. In English, on the other hand, this mode is expressed as the complex preposition *from here*. This is now accurately described by the decomposition into modaliser and localiser relations.

We can view the translation process after the analysis stage, as a process where (i) in the transfer component interlingua predicates pass through unaltered (as is the case for the *message relation*s in LOGON), and all other predicates are transferred according to the source and target language, and (ii) in the generation component, the relevant semantic relations are grouped to form words, i.e. lexicalizations of the concepts expressed by the semantics. We could then pass all the modaliser relations (*mode_rel*s) through the transfer unchanged, and transfer the localiser relations (*loc_rel*s)[5]. This is shown in Figure 7.1. As there is no lexicalization of the combination of the two concepts *coinitial_mode_rel* and *here_loc_rel*, they are lexicalized individually, as *from* and *here*, respectively.

The same holds for the prepositions *ovenfra* ('from above'), *nedenfra* ('from below') and *innenfra* ('from inside').

[5]It could be argued that in this particular case, the *proximal_loc_rel* and *distal_loc_rel* are interlingua predicates as well.

Input	After analysis	After transfer	Output
"herfra"	[PRED *coinitial_mode_rel* ARG0 [e] ARG1 [l]]	[PRED *coinitial_mode_rel* ARG0 [e] ARG1 [l]]	"from"
	[PRED *her_loc_rel* ARG0 [l]]	[PRED *here_loc_rel* ARG0 [l]]	"here"

Figure 7.1: Translating *herfra* ('from here')

7.3 Complex Locatives

In Table 7.2 we saw several instances of complex prepositions. Some of these are repeated here in Table 7.4. Apart from the combination of *fra* and a

L	**static**	**coinitial**	**cofinal**	**transitory**	**approximative**
i		ut/opp fra/av			
ved					
under		fra under			
over		fra over	inn/ut over		
mellom		inn/. . . mellom			

Table 7.4: Complex locatives

localiser, which we examined in the previous chapter, we find a combination of what we called intransitive locatives (*inn, ut, ned* etc.) and other locative prepositions. These words were formerly classified as adverbs, but according to Faarlund et al. (1997), they are now considered prepositions, based on the fact that they share syntactic distribution with locative PPs. The class consists of the words *inn, ut, opp, ned, bort, hjem, frem/fram* and possibly some more, and have the following characteristics:

1. They can occur as intransitive or transitive prepositions, with the subcategorization frame in (128).

(128) a. Per løp [$_{PP}$ inn]. *(intransitive)*
Per ran in.
'Per ran inside.'

b. Per løp [$_{PP}$ inn døra]. *(NP complement)*
Per ran in door.DEF.
'Per ran in through the door.'

c. Per løp [$_{PP}$ inn i huset]. *(PP complement)*
Per ran in in house.DEF
'Per ran into the house.'

2. When occurring as intransitive prepositions, as in (128a), they express motion into a contextually salient location. (cf. Beerman and Hellan (Forthcoming, p. 5))

3. They disambiguate between static and cofinal mode on the succeeding PP, in favor of cofinal mode.

> In many cases however, these kinds of prepositional phrases must be a complement of the goal preposition (e.g. *opp* ('up'), *ned* ('down'), *ut* ('out'), *inn* ('in'), *fram* ('forward'), etc.) to make explicit the goal sense/interpretation, cf. the differences between [the ambiguous prepositions on the left side and the unambiguous cofinal prepositions on the right side][6]:
>
> | | |
> |---|---|
> | Han gikk på taket | - Han gikk opp på$_{dir}$ taket |
> | De løp i tunnelen | - De løp inn i$_{dir}$ tunnelen |
> | Hunden sprang foran bilen | - Hunden sprang fram foran$_{dir}$ bilen" |
>
> (Faarlund et al., 1997, p. 426)

4. They all have non-directional counterparts, with the additional suffix *-e*, which in the same contexts disambiguate in favor of static readings of the succeeding PP.

(129) a. Han gikk oppe på$_{stat}$ taket.
He walked up on roof.DEF.
'He walked/was walking on$_{stat}$ the roof.'

b. De løp inne i$_{stat}$ tunnelen.
They ran in in tunnel.DEF.
'They ran inside$_{stat}$ the tunnel.'

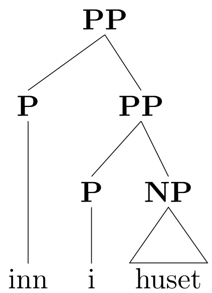
The intransitive uses of these locatives were treated in the previous section. We will now take a look at some contexts where these preposition combine with other prepositions to form complex locatives.

7.3.1 Intransitive Prepositions with PP Complements

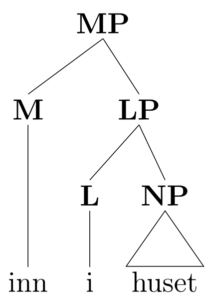
According to Faarlund et al. (1997), the PP headed by *inn* in (130) is assigned a structure as in (131).

[6] Translation from Norwegian by the author. The original text reads: "I en del tilfeller må imidlertid slike preposisjonsfraser stå som utfylling til tilstedspreposisjoner (typen *opp, ned, ut, inn, fram* osv.) for at slik tilstedsbetydning skal bli tydeliggjort, jf. forskjellene mellom:"

(130) Per løp inn i huset.
Per ran in in house.DEF.
'Per ran into the house.'

(131)


With the MP/LP distinction introduced, we can now interpret this structure in two ways, as the PP is ambiguous between an MP and an LP, on our view. The semantics we want, is that Per ends up *i huset* ('in the house'), i.e. a cofinal modaliser, cf', with the LP complement *i huset*. This can be done in two ways; either by letting *inn* introduce cf' and take an LP complement, or let *i huset* be cofinal mode, and state that *inn* select cofinal MPs. I propose the former, namely that the PP in (131) is assigned the structure in (132).

(132)


If *inn* selects a cofinal PP (MP) complement, then the complement PP must be able to occur as cofinal in contexts where it is not selected, as well. For the prepositions *over* ('over'), *mellom* ('between) and *blant* ('among') in (133), this static/cofinal ambiguity is less than obvious. I may be colored by pragmatic factors, but I really find it hard to get a cofinal reading of the prepositions in these sentences, when they occur without the intransitive preposition *inn* ('in').

(133) a. Helikopteret fløy inn over byen. *(cofinal)*
The helicopter flew in over city.DEF.

b. Helikopteret fløy over byen. *(static/transitory)*
The helicopter flew over city.DEF.

c. Per løp inn blant trærne. *(cofinal)*
Per ran in among trees.DEF.

d. Per løp blant trærne. *(static)*
Per ran among trees.DEF.

e. Per løp inn mellom trærne. *(cofinal)*
Per ran in between trees.DEF.

f. Per løp mellom trærne. *(static/transitory)*
Per ran between trees.DEF.

This interpretation of intransitive prepositions does not predict that complex cofinal PPs consisting of two intransitive prepositions and a transitive preposition with an NP complement group together to form one cofinal constituent, as in the sentence in (134).

(134) Per kom hit inn i stuen.
Per came (to) here (to) in in living-room.DEF.
'Per came here into the living room.'[7]

I have not dealt with this problem in my grammar fragment, but one solution is to construct a complex intransitive preposition which then in turn takes an LP complement. Thus, in our grammar, the lexical entry for *inn* ('(to) in') is shown in Figure 7.2, taking LP complements of the same type as modalisers in Chapter 6, i.e. static PPs denoting a location.

Faarlund et al. (1997) claim that the cofinal intransitive preposition disambiguates the subsequent PP with respect to mode (or directionality). This was implemented by letting the intransitive prepositions select optional LP complements. But there is nothing in the grammar that prevents the cofinal intransitive prepositions *not* to select the following LP as its complement, and the PP will act as a static adjunct instead. The syntactic structures for the two analyses are given in Figure 7.3. In the first tree, the PP (LP) *i hagen* ('in the garden') acts as a complement of *inn*, and in the second, the PP (MP) acts as an adverbial adjunct[8].

(135) a. En mann rusler inn i en hage.
b. En mann rusler i en hage.

I am of the opinion that these intransitive prepositions are used in many cases mainly for disambiguation purposes, and that the semantics expressed by the intransitive preposition is less significant in these contexts. It seems most natural to interpret the sentence in (135a) to express cofinal motion

[7]This sentence is rather difficult to translate, and I'm not certain my translation of it is correct.

[8]The LP in the second tree is actually an MP, but I couldn't get the labeling of tree nodes right in the LKB.

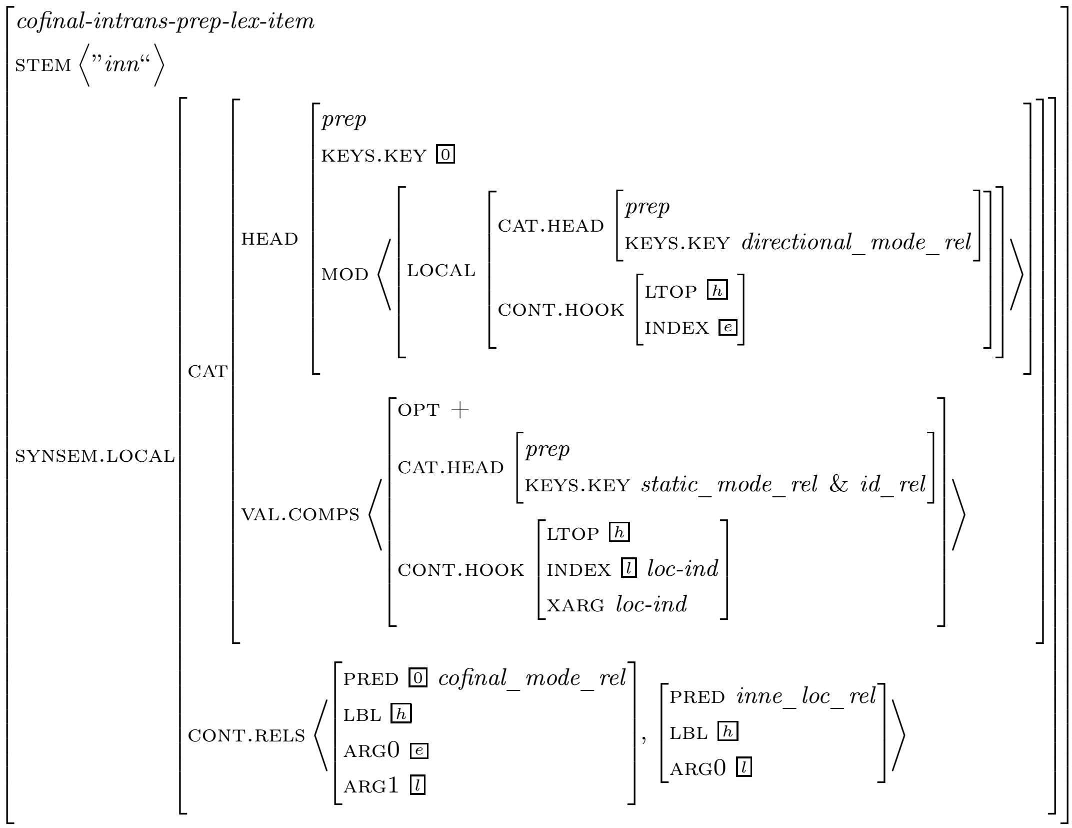


Figure 7.2: The intransitive preposition *inn* with an optional LP complement

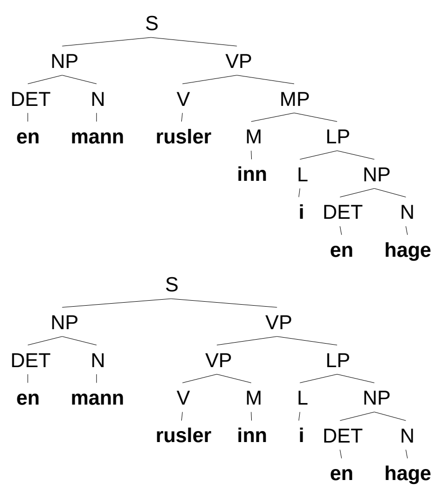


Figure 7.3: The two analyses of *En mann rusler inn i en hage* (‘A man strolls into/(in in) a garden’)

with respect to the LP *i en hage* (‘in a garden’), and to take the PP in (135b) as a static event modifier. But despite this, the other interpretations do not seem totally inconceivable (i.e. that *i en hage* (‘in a garden’) is a static MP in (135a) and an cofinal MP in (135b)). The claim of Faarlund et al. (1997) that these intransitive prepositions disambiguate the preceding ambiguous PP, may prove to be too strong. But the presence of the intransitive prepositions in these contexts certainly favors the directional reading, just as the absence of the intransitive preposition favors the static reading[9].

In (136a), the ‘inward’ motion is not salient. Neither the ‘outward’ motion in (136c). In these contexts, the presence of the intransitive preposition favor the cofinal reading, but it is less likely to contribute to the semantics in other respects. Thus, it could be argued that the ´inne′(l)’ relation in (137b) could be deleted from the semantic representation, and likewise with the ‘ute′(l)’ relation in (136d).

(136) a. Per løp inn i en hage
Per ran in in a garden
‘Per ran into a garden’
b. $\exists e \exists x(\text{løpe}'(e,\ per)\ \wedge\ cf'(e,\ l) \wedge \text{inne}'(l) \wedge \text{i}'(l,x) \wedge \text{hage}'(x))$
c. Per hoppet ut i et basseng
Per jumped out in a pool
‘Per jumped into a pool’
d. $\exists e \exists x(\text{hoppe}'(e,\ per) \wedge cf'(e,\ l) \wedge \text{ute}'(l) \wedge \text{i}'(l,x) \wedge \text{basseng}'(x))$

In other contexts, the intransitive preposition is more likely to contribute to the semantics, as in (137), as the ‘outward’ motion occur to be more salient in this particular context.

(137) a. Per løp ut i en hage
Per ran out in a garden
’Per ran out into a garden’
b. $\exists e \exists x(\text{løpe}'(e,\ per)\ \wedge\ cf'(e,\ l) \wedge \text{ute}'(l) \wedge \text{i}'(l,x) \wedge \text{hage}'(x))$

Exactly in which contexts intransitive prepositions contribute to the semantics and where they don’t, is very hard to ascertain (and I am not too sure about the judgments above). It depends partly on the relationship between the two regions motion occurs with respect to, and whether the motion is marked with respect to this relationship. For instance, you are ‘outside’ in the garden with respect to the house, but ‘inside’ the garden with respect

[9]In an MT system, the favored reading may be marked for optimality prior to the selection of which analyses should be passed to the transfer component.

to the surroundings (the street, the city etc.). Also, the verb and the preposition seem to share the semantic labor of expressing directionality. For a verb like *å falle* ('to fall'), the inherent direction of motion is downward. In this case, one could claim that the intransitive preposition is less likely to carry over in a translation from Norwegian to English. But for *å hoppe* ('to jump'), which does not have the same degree of an inherent direction, the particle is more likely to carry over in the translation. A quick check in the Oslo Multilingual Corpus shows that none of the 13 occurrences of *falle ned* are translations from *fall down*, only from *fall* (or in some cases other verbs with similar meaning).[10] Sample translations from the Oslo Multilingual Corpus are given in (138)

(138) a. From *The Women of Brewster Place* by *Gloria Naylor*:
'Her mother screamed, "For the love of Jesus, Sam!" and jumped on his back and tried to wrestle the stick from him.'
'Moren hennes skrek: "I Jesu navn, Sam!" og hoppet opp på ryggen hans og forsøkte å vriste stokken fra ham.'

b. From *Arcadia* by *Jim Crace*:
'The laurel branches fell amongst the booty at his feet.'
'Laurbærgreinene falt ned blant byttet ved føttene hans.'

As we have seen above, one place where the analysis of complex locatives could be put to use, is in the translation of what Trujillo (1995) named 'lexicalized prepositions'. In particular, *inn* ('in') seems to appear frequently with the lexicalized prepositions *i* ('in') and *på* ('on'). A quick check in the Oslo Corpus of Tagged Norwegian Text seems to support this, represented in Table 7.5. (These figures include the uses of *inn* ('in') in other contexts as well, e.g. as verb particles, part of idiomatic expressions, or just in non-locative interpretations.)

	i ('in')	på ('on')	under ('under')	over ('over/above')	blant ('among')
inn ('in')+ ...	8290	2389	1265	166	23
ut ('out')+ ...	3049	2025	32	5	30
ned ('down')+ ...	1382	889	25	0	5
opp ('up')+ ...	3081	1464	154	74	24

Table 7.5: Occurrences of complex PPs in the Oslo Corpus of Tagged Norwegian Text

If there is anything in the claim that *inn* ('in') often function as a disambiguator of the following ambiguous preposition, as in e.g. *inn i* ('in in')

[10] I looked at the translations from English to Norwegian, as I assume translations in the opposite direction are more likely to be colored by the way these states of affairs are expressed in Norwegian (i.e. with an intransitive preposition)

and *inn på* ('in on'), and we want to translate these to 'into' and 'onto', respectively, then our analysis of complex PPs enables us to do exactly that. The transfer rule (written on the format from Chapter 4) for this is given in (139a), under the assumption that the right hand side corresponds to the English grammar's MRS representation of *into*. The transfer rule for an English grammar with a single MRS representation of *into* is shown in (139b). If we implement lexicalized prepositions, as described in Chapter 6, we could make a more general rule, substituting i_loc_rel and $\text{in\_loc}_r\text{el}$ with lex_loc_rel ('lexicalized localiser relation').

(139) a. $\{\text{cofinal\_mode\_rel}(e, l) \wedge \text{inne\_loc\_rel(l)} \wedge \text{i\_loc\_rel}(l, x)\}$
$\rightarrow \{\text{cofinal\_mode\_rel}(e, l) \wedge \text{in\_loc\_rel}(l, x)\}$
b. $\{\text{cofinal\_mode\_rel}(e, l) \wedge \text{inne\_loc\_rel(l)} \wedge \text{i\_loc\_rel}(l, x)\}$
$\rightarrow \{\text{into\_p\_rel}(e, x)\}$

7.3.2 Intransitive Prepositions with NP complements

Many of the intransitive prepositions occur with both nominal complements, as well. All nominal complements I've found denote paths or apertures, as seen in (140) from the Oslo Corpus.

(140) a. Hun pleide å snike seg inn bakdøren her i annen
She used to sneak herself in back-door.DEF here in second
etasje.
floor.
'She used to sneak in through the back door here on the first floor.'
b. Hun satt og stirret på et skip som kom inn fjorden.
She sat and stared on a ship which came in fjord.DEF
'She sat staring on a ship coming in the fjord.'
c. Jeg går ned trappene, sier ikke noe; bare forsvinner.
I walk down stairs.DEF, say nothing; just disappear.
'I walk down the stairs, say nothing; just disappear'
d. Og alle tre sprang ned skråningen, over brua og
And all three ran down slope.DEF, over bridge.DEF and
inn i landsbyen.
in in village.DEF
'And all three ran down the slope, over the bridge and into the village.'

Characteristic for these uses is that they seem to alternate with PPs headed by *av* (or *ad*), with the same type of nouns as complements, seen in (141).

(141) a. Kastanjeduften driver i bølger inn av døren, ...
Chestnut-smell.DEF drift in waves in of door.DEF, ...
'The smell of chestnuts drift in through the door, ...'

b. ... kom han inn av døren belesset med pakker og frukt.
... came he in of door loaded with packages and fruit.
'... he came in through the door, loaded with packages and fruit.'

According to Nynorskordboka[11], this use of *av* is derived from *ad* (Danish) or *at* (Old Norse), which means *via* ('via') or *gjennom* ('through'). It is here also interesting to see that *inn* in these contexts probably translates to the English expression *in through*.

(142) Jon gikk inn (av) døra ⇒ Jon went in through the door.

The different uses of *inn* with a nominal complement, seem to express two types of motion. Transitory motion with respect to its nominal complement, and a cofinal motion with respect to the contextually induced location (as for the intransitive prepositions). We therefore let *inn* introduce the same semantics as the intransitive *inn* and *gjennom NP* combined. The optional preposition *av* can be treated as a selected (i.e. semantically empty, see Section 2.1) preposition. In this way, we receive the same semantics for all the three sentences in (143).

(143) a. Per gikk inn døra.
Per walked in door.DEF.

b. Per gikk inn av døra.
Per walked in of door.DEF.

c. Per gikk inn gjennom døra.
Per walked in through door.DEF.

Part of the lexical entry for *inn* with an NP complement will now look like Figure 7.4, with the semantics given in Figure 7.5.

When translating, we assume that the generation of the TL string proceeds as described before, where the relevant relations form words, as shown in Figure 7.6.

7.4 Summary

In this chapter, we have seen some uses of the two-layer analysis. Locatives can be classified along the two dimensions corresponding to the two

[11] Search interface on <http://www.dokpro.uio.no/>

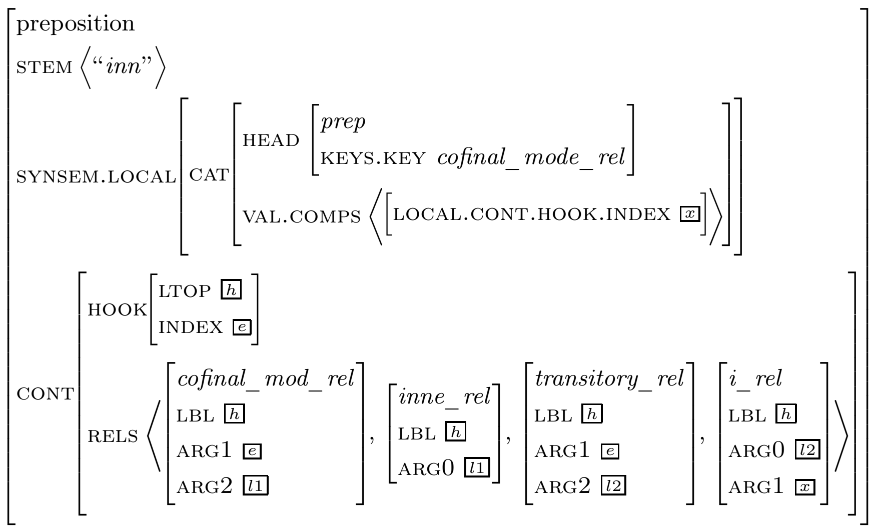


Figure 7.4: *Inn* ('in') with NP complement

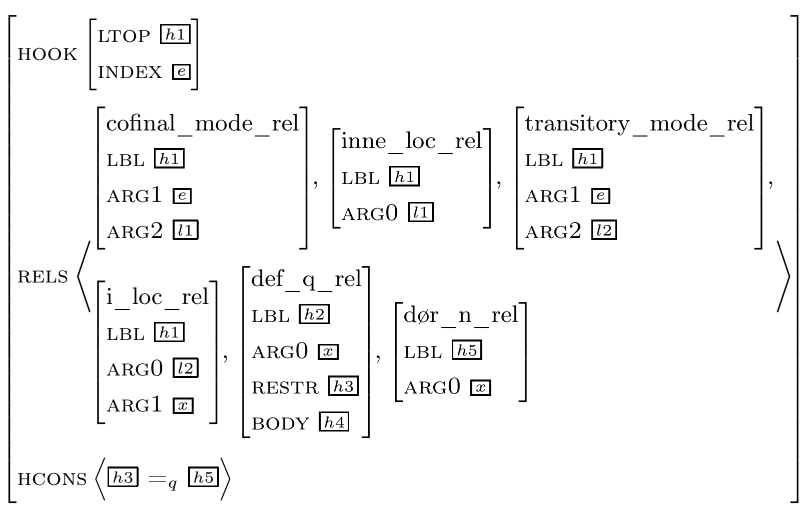


Figure 7.5: The semantics of the PP *inn av døra* ('in through the door')

Input	After analysis	After transfer	Output
"inn"	[PRED *cofinal_mode_rel* ARG0 [e] ARG1 [l1]] [PRED *inne_loc_rel* ARG0 [l1]]	[PRED *cofinal_mode_rel* ARG0 [e] ARG1 [l1]] [PRED *inside_loc_rel* ARG0 [l1]]	"in"
	[PRED *transitory_mode_rel* ARG0 [e] ARG1 [l2]] [PRED *i_loc_rel* ARG0 [l2] ARG1 [x]]	[PRED *transitory_mode_rel* ARG0 [e] ARG1 [l2]] [PRED *in_loc_rel* ARG0 [l1] ARG1 [x]]	"through"

Figure 7.6: Translating *inn* ('in') with an NP complement

layers, and these dimensions capture generalizations both of the semantics and the syntactic behavior of locatives. We have also seen how our analysis can be used in semantic transfer based machine translation, and how some differences between Norwegian and English expressions can be predicted and explained by decomposing locatives.

The next chapter, which is also the last, will take a step back to analyze the achievements of this thesis. I will discuss some particular places where the two-layer is of interest, and how this approach positions itself with respect to semantic transfer-based translation. I will also briefly discuss the challenges ahead, with respect to analyzing and translating locative prepositional phrases.

Chapter 8

Conclusion

I have in the present thesis implemented a fine-grained semantic theory of locatives in the computational grammar writing environment LKB. I have implemented locatives as two-layered, and shown the correspondence between the MRS representations and the underlying object-language lambda calculus expressions.

Interpreting locative preposition as consisting of two layers has given several interesting results. In assuming that locatives express a limited number of motion types with respect to a location, we have found a way to classify Norwegian locatives along two dimensions. This classification enables us to generalize over both semantic and syntactic properties of classes of locatives. We have also seen the range of locatives which fit into this scheme: Intransitive locatives, transitive locatives, as well as two types of complex locatives that share central semantic properties.

The two-layer analysis is of interest both monolingually and bilingually:

- **Explicit segmentation:** In both Norwegian and English, we find a lexicalization of the coinitial modaliser, i.e. *fra* and *from*, respectively. The complement of the modaliser serves semantically as a localiser. Thus, we find instances of both modalisers proper and localisers proper in both languages. This gives us reasons monolingually to decompose locatives.

- **Different segmentation:** The translation of *herfra* ('from here') gave us an example where the Norwegian locative is a combined modaliser/localiser, whereas the English expression is built up separately from a modaliser and a localiser. The decomposition gives us an accurate prediction of this translation, as there is no English lexicalization of the two concepts combined (i.e. coinitial and proximal), and they are lexicalized separately.

- **Differences with respect to mode ambiguity:** We saw that the Norwegian locatives *i* ('in') and *på* ('on') are ambiguous between static and cofinal mode. The English locatives *into* and *onto* are unambiguously cofinal mode. In the contexts where the Norwegian locatives are disambiguated in favor of the cofinal mode, a translation into the cofinal English locatives gives a more accurate translation.

In addition to these considerations, which concern Norwegian and English isolated, and this particular language pair, the claim of Kracht (2002) is that there is evidence for five different modes, independently of how these modes are expressed (by prepositions, postpositions or case). If this claim is correct, it gives the decomposition into modalisers and localisers an 'interlingua edge'.

The particular MT system described in Chapter 4 of the thesis, LOGON, is a system based on semantic transfer. But the term 'semantic transfer' does not make a claim about what kind of semantics, or how deep semantic analysis one applies before the transfer. How concepts are lexicalized in different languages is to some extent arbitrary, and arriving at partially language-independent analyses in isolated conceptual domains (here modes or motion with respect to a region) must be considered attractive in semantic transfer based translation. As the complexity of the semantic expressions rise, the burden on the transfer component should be expected to diminish in the cases where decomposition results in arriving at more or less interlingua representations.

The decompositional approach described in this thesis may prove hard to scale. In a larger grammar, e.g. the ERG, our analysis of locatives might be expected to cause a substantial growth in the number of analyses, both due to the separation of locatives from other types of prepositions, and due to the decomposition of locatives. One approach for coping with high numbers of parses is to underspecify semantic predicates as much as possible. This certainly has attractive practical consequences, and I do not want to undermine the importance of practical considerations in natural language applications. But when doing semantics, one often has to resolve these underspecifications at a later stage.

My view is that semantic transfer-based machine based on ambiguous semantic predicates makes it very difficult to predict the outcome. I believe one will have to compensate for the underspecification in the transfer module, where all underspecified predicates must be further specified (or at least the predicates causing incorrect translations). To me it seems as translation on basis of underspecified semantic predicates, where exactly one semantic predicate corresponds to exactly one word, is more of a lexical approach than a semantic approach to machine translation.

Just consider the preposition *til*, which may serve as a locative preposition (*løpe til*/'run to'), case marker (*gi til*/'give to'), beneficiary preposition (*bake en kake til*/'bake a cake for'), possessive preposition (*sykkelen til John*/'John's bike'), etc. When doing syntactical analysis, these ambiguities need not be resolved, given that the syntactic behavior of *til* does not vary. But when doing semantic analysis, I am of the opinion that disambiguation is necessary *unless we are certain disambiguation has no effect*. Otherwise, we cannot predict the outcome of underspecified semantics. One instance where semantic underspecification (at least in general) does not have impact on translation, is the ordering of NP quantifiers. But for language in general, we know that one lexical item (e.g. *til*) may represent several semantic predicates, and that, in principle, all these may translate to different lexical items (or morphemes, for possessive *til*) in the target language.

Both the syntax and semantics for prepositional phrases are difficult fields, caused very much by the ambiguity both of syntactic structure, i.e. PP attachment site, and the ambiguity of this class of words. Disambiguation is extremely difficult, and it is probably here the most important challenges lie in order to translate prepositions correctly. The topic of disambiguating locative prepositions from other types of preposition has not been discussed in this thesis, as I have assumed all prepositions occurring in the grammar to be locative. I suspect this field will benefit much from both pragmatic theories and more elaborated semantic theories, when it comes to modeling the interpretation process of PPs. But still I believe the implementation of a decompositional approach to locatives, presented in this thesis, has proven successful in shedding light on some important aspects of the nature of locative prepositional phrases.

Bibliography

Dorothee Beerman and Lars Hellan. Semantic decomposition in a computational HPSG grammar - a treatment of context-dependent directionals. Unpublished, Forthcoming.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. The grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, 2002.

Manfred Bierwisch. On the grammar of local prepositions. In M. Bierwisch, M. Motsch, and W. Zimmermann, editors, *Syntax, Semantix und Lexicon*, pages 1–65. Akademie Verlag, 1988.

Ann Copestake. *Implementing Typed Feature Structure Grammars.* CSLI Publications, 2002.

Ann Copestake and Dan Flickinger. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, 2000.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. Minimal Recursion Semantics: An introduction. Unpublished, 2003.

Ann Copestake, Alex Lascarides, and Dan Flickinger. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001), Toulouse, France*, 2001.

Bonnie J. Dorr, Pamela W. Jordan, and John W. Benoit. A survey of current research in machine translation. *Advances in Computers*, 49:1–68, 1999.

Jan Terje Faarlund, Svein Lie, and Kjell Ivar Vannebo. *Norsk Refereansegrammatikk.* Universitetsforlaget, 1997.

L. T. F. Gamut. *Logic, Language and Meaning.* University of Chicago Press, 1991.

Peter Gärdenfors. *Conceptual spaces: The Geometry of Thought.* The MIT Press, 2000.

Lars Hellan. The Norwegian computational HPSG grammar *NorSource*. Unpublished manuscript, 2003.

Annette Herskovits. *Language and Spatial Cognition.* Cambridge University Press, 1986.

Ray Jackendoff. *Consciousness and the Computational Mind.* The MIT Press, 1987.

Ray Jackendoff. *Semantic Structures.* The MIT Press, 1990.

Markus Kracht. On the semantics of locatives. *Linguistics and Philosophy*, 25:157–232, 2002.

Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. Som å kappete med trollet - towards high quality Norwegian - English machine translation. Unpublished, 2004.

Carl Pollard and Ivan Sag. *Head-Driven Phrase Structure Grammar.* Chicago: University of Chicago Press and Stanford: CSLI Publications, 1994.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. *Syntactic Theory.* CSLI Publications, 2nd edition, 2003.

Arturo Trujillo. *Lexicalist Machine Translation of Spatial Prepositions.* PhD thesis, University of Cambridge, 1995.

Mai Ellin Tungseth. PP, FP and the telic/atelic distinction in Norwegian motion constructions. To appear in the proceedings from 'Workshop on Aspect at the University of Iowa', in a volume to be published by Kluwer, Forthcoming.

Cornelia Maria Verspoor. *Contextually-Dependent Lexical Semantics.* PhD thesis, The University of Edinburgh, 1997.

Dieter Wunderlich. How do prepositional phrases fit into compositional syntax and semantics? *Linguistics*, 29:591–621, 1991.

Joost Zwarts. The semantics of relative position. In Mandy Simons and Teresa Galloway, editors, *SALT V: Proceedings from Semantics and Linguistic Theory*, pages 405–421. CLC Publications, 1995.

Joost Zwarts and Yoad Winter. Vector space semantics: a modeltheoretic analysis of locative prepositions. *Journal of Logic, Language and Information*, 9:171–213, 2000.

Appendix A

LKB implementation

A.1 norsk.tdl

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; GRAMMAR MODELLING THE SEMANTICS OF          ;;
;; LOCATIVE PREPOSITIONS BASED ON THE          ;;
;; SEMANTICAL ANALYSIS IN KRACHT (2002)        ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; HEAD types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

noun := head &
[ MOD null ].

det := head &
[ MOD null ].

verb := head  &
[ MOD null ].

;; PREP head type underspecifed for MOD

prep := head.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; LEX-ITEM types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

noun-lex-item := basic-noun-lex &
[SYNSEM.LOCAL.CAT [ HEAD noun,
                    VAL [ COMPS null,
                          SPR < [ LOCAL.CAT.HEAD det ] >,
                          SUBJ null] ] ].
```

```
det-lex-item := basic-determiner-lex &
[ SYNSEM.LOCAL.CAT [ HEAD det,
                     VAL [ COMPS null,
                           SPR null,
                           SUBJ null] ] ].

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; VERB lex-item types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

verb-lex-item := basic-verb-lex &
[ SYNSEM [ LOCAL [ CAT [ HEAD verb,
                         VAL [ SUBJ < synsem & #subj >,
                               COMPS #comps,
                               SPR null ] ],
                   ARG-S < #subj . #comps > ],
           LKEYS.KEYREL event-relation ] ].

;; MOTION VERBS have an optional dir-PP argument

intrans-motion-verb-lex-item := verb-lex-item &
[ SYNSEM [ LOCAL [ CAT.VAL
                   [ SUBJ < [ LOCAL.CONT.HOOK.INDEX #subj-ind ] >,
                     COMPS < [ OPT +,
                               LOCAL [ CAT [  HEAD prep &
                                                    [ KEYS.KEY directional_mode_rel ],
                                              VAL.COMPS olist ],
                                       CONT.HOOK [ LTOP #lbl,
                                                   INDEX event & #mod-event,
                                                   XARG loc-ind ] ] ] > ],
                   CONT.HOOK [ LTOP #lbl,
                               INDEX #mod-event ] ],
           LKEYS.KEYREL arg1-ev-relation &
                 [ ARG0 #mod-event,
                   ARG1 #subj-ind ] ] ].

;; STATIC VERB

static-verb-lex-item := verb-lex-item.

intrans-static-verb-lex-item :=  static-verb-lex-item & intransitive-lex-item.
```

```
;; VERBS WITH PP COMPLEMENTS

;; LP-TRANS-VERB-LEX-ITEM select LPs
;; ie. static mode PPs with KEYS.KEY id_rel

;; e.g. 'bo' ('to live')

lp-trans-verb-lex-item := verb-lex-item & transitive-lex-item &
[ SYNSEM.LOCAL [ CAT.VAL.COMPS < [ LOCAL
                                   [ CAT.HEAD prep &
                                              [ KEYS.KEY static_mode_rel & id_rel ],
                                     CONT.HOOK [ LTOP #lbl,
                                                 XARG loc-ind,
                                                 INDEX loc-ind ] ],
                                   OPT - ] >,
                 CONT.HOOK.LTOP #lbl ] ].

;; NP-LP-DITRANS-VERB-LEX-ITEM select NP + LP
;; (PP comps with different modes possible, here specified for static_or_cofinal PPs)

;; e.g. 'legge' ('to lay/put/place')

np-lp-ditrans-verb-lex-item := ditransitive-lex-item & verb-lex-item &
[ SYNSEM.LOCAL [ CAT.VAL.COMPS < [ LOCAL.CAT.HEAD noun,
                                   OPT - ],
                                 [ LOCAL
                                   [ CAT.HEAD prep &
                                              [ KEYS.KEY id_rel &
                                                         static_or_cofinal_mode_rel ],
                                     CONT.HOOK [ LTOP #lbl,
                                                 XARG loc-ind,
                                                 INDEX loc-ind ] ],

                                 OPT - ] >,
                 CONT.HOOK.LTOP #lbl ] ].

;; NP-PP-DITRANS-VERB-LEX-ITEM select NP + NP (PP with INDEX 'ref-ind')
;; i.e. PP with KEYS.KEY and KEYS.ALTKEY id_rel
;; headed by 'til' (cofinal + ved)

;; e.g. 'å gi' ('to give')

np-pp-ditrans-verb-lex-item := ditransitive-lex-item & static-verb-lex-item &
[ SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL.CAT.HEAD noun,
                                 OPT - ],
                               [ LOCAL [ CAT [ HEAD prep &
                                                    [ KEYS [ KEY id_rel &
                                                                 cofinal_mode_rel,
                                                             ALTKEY id_rel &
                                                                 ved_loc_rel ] ],
                                               VAL.COMPS olist ],
                                         CONT.HOOK [ INDEX ref-ind,
                                                     XARG ref-ind ] ],
                                 OPT - ] > ].
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; PREP lex-item types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

prep-lex-item := no-hcons-lex-item &
[ SYNSEM [ LOCAL [ CAT [ HEAD prep &
                              [ KEYS.KEY #keypred ],
                         VAL [ SUBJ  null,
                               COMPS list,
                               SPR null,
                               SPEC null ] ],
                   CONT [ HOOK [ INDEX #arg0,
                                 XARG #arg1,
                                 LTOP #lbl ],
                          RELS.LIST.FIRST #keyrel ] ],
           LKEYS [ KEYREL #keyrel &
                          [ PRED #keypred & mode_rel,
                            LBL #lbl,
                            ARG0 #arg0,
                            ARG1 #arg1 ] ] ] ].

;; STATIC locative prepositions which aren't selected
;; are adjuncts and intersective modifiers

static-prep-lex-item := prep-lex-item &
[ SYNSEM [ LOCAL.CAT.HEAD.MOD   < [ LOCAL [ CAT [ HEAD verb,
                                                  VAL [ SUBJ < synsem >,
                                                        COMPS olist ] ],
                                            CONT.HOOK [ LTOP #lbl,
                                                        INDEX #mod-event ] ] ] >,
           LKEYS.KEYREL [ PRED static_mode_rel,
                          LBL #lbl,
                          ARG0 #mod-event,
                          ARG1 loc-ind ] ] ].

;; DIRECTIONAL locative prepositions are complements of motion verbs,
;; but have intersective semantics

;; directional PPs can modify other directional PPs to form PP clusters

dir-prep-lex-item := prep-lex-item &
[ SYNSEM [ LOCAL.CAT.HEAD.MOD < [ LOCAL [ CAT [ HEAD prep &
                                                     [ KEYS.KEY directional_mode_rel ],
                                                VAL.COMPS olist ],
                                          CONT.HOOK [ LTOP #lbl,
                                                      INDEX #mod-event,
                                                      XARG loc-ind ] ] ] >,
           LKEYS.KEYREL [ LBL #lbl,
                          ARG0 #mod-event ] ] ].
```

```
;; Subtypes of DIR-PREP-LEX-ITEM

cofinal-prep-lex-item := cofinal-or-transitory-prep-lex-item &
[ SYNSEM.LOCAL.CAT.HEAD
  [ KEYS.KEY cofinal_mode_rel,
    MOD.FIRST.LOCAL.CAT.HEAD.KEYS.KEY dir_not_cofinal_mode_rel ] ].

coinitial-prep-lex-item := dir-prep-lex-item &
[ SYNSEM.LOCAL.CAT.HEAD.KEYS.KEY coinitial_mode_rel ].

transitory-prep-lex-item := cofinal-or-transitory-prep-lex-item &
[ SYNSEM.LOCAL.CAT.HEAD.KEYS.KEY transitory_mode_rel ].

approximative-prep-lex-item := dir-prep-lex-item &
[ SYNSEM.LOCAL.CAT.HEAD.KEYS.KEY approximative_mode_rel ].

;; 'under'
cofinal-or-transitory-prep-lex-item := dir-prep-lex-item &
[ SYNSEM.LOCAL.CAT.HEAD.KEYS.KEY cofinal_or_transitory_mode_rel ].

;; MODALISER (cf. Kracht (2002) denotation
;; e.g. 'fra' ('from')

single-rel-prep-lex-item := prep-lex-item & single-rel-lex-item &
[ SYNSEM [ LOCAL.CAT.VAL.COMPS < [ LOCAL
                                   [ CAT [ HEAD prep &
                                                [ KEYS.KEY id_rel & static_mode_rel ],
                                           VAL.COMPS olist ],
                                     CONT.HOOK [ LTOP #lbl,
                                                 XARG loc-ind,
                                                 INDEX #loc & loc-ind ] ],
                                   OPT - ] >,
           LKEYS.KEYREL [ LBL #lbl,
                          ARG1 #loc ] ] ].


modaliser-dir-prep-lex-item := dir-prep-lex-item & single-rel-prep-lex-item &
[ SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL.CAT.HEAD.KEYS.KEY id_rel ] > ].

;; MODALISER+LOCALISER (cf. Kracht (2002)), lexicalized as
;; a single preposition
;; most prepositions, e.g. 'i' ('in'), 'på' ('on'), 'gjennom' ('through')
```

```
;; TWO-REL-PREP-LEX-ITEM introduce two semantic relations

two-rel-prep-lex-item := prep-lex-item &
[ SYNSEM [ LOCAL [ CAT.HEAD.KEYS.ALTKEY #altkeypred,
                   CONT.RELS [ LIST.REST [ FIRST #altkeyrel,
                                           REST #rest ],
                               LAST #rest ] ],
           LKEYS [ KEYREL [ LBL #lbl,
                            ARG1 #loc ],
                   ALTKEYREL #altkeyrel &
                          [ PRED loc_rel & #altkeypred,
                            LBL #lbl,
                            ARG0 #loc ] ] ] ].

;; TRANSITIVE-PREP-LEX-ITEM takes an NP complement

transitive-prep-lex-item := two-rel-prep-lex-item &
[ SYNSEM [ LOCAL [ CAT.VAL.COMPS < [ LOCAL [ CAT [  HEAD noun,
                                                    VAL.SPEC null ],
                                             CONT.HOOK.INDEX #index ],
                                     OPT - ] > ],
           LKEYS.ALTKEYREL arg1-relation &
                 [ ARG1 #index ] ] ].


;; OPT-COMP-INTRANS-PREP-LEX-ITEM select an optional
;; LP complement with static mode

;; e.g. her ('here) 'hit', ('(to) here'), 'inn' ('(to) inside'), inne ('inside')

opt-comp-intrans-prep-lex-item := two-rel-prep-lex-item &
[ SYNSEM [ LOCAL.CAT.VAL.COMPS < [ LOCAL [ CAT [ HEAD prep &
                                                      [ KEYS.KEY
                                                      static_mode_rel & id_rel ],
                                                 VAL.COMPS olist ],
                                           CONT.HOOK [ LTOP #lbl,
                                                       XARG  loc-ind,
                                                       INDEX #loc & loc-ind ] ],
                                   OPT + ] >,
           LKEYS.KEYREL [ LBL #lbl,
                          ARG1 #loc ] ] ].


no-comp-intrans-prep-lex-item := two-rel-prep-lex-item &
[ SYNSEM.LOCAL.CAT.VAL.COMPS null ].

;; PREP-LEX-ITEMS leaf types

static-intrans-prep-lex-item := opt-comp-intrans-prep-lex-item &
                                static-prep-lex-item.
cofinal-intrans-prep-lex-item := opt-comp-intrans-prep-lex-item &
                                 cofinal-prep-lex-item.
coinitial-intrans-prep-lex-item := no-comp-intrans-prep-lex-item &
                                   coinitial-prep-lex-item.
```

```
transitory-intrans-prep-lex-item := no-comp-intrans-prep-lex-item &
                                    transitory-prep-lex-item.
approximative-intrans-prep-lex-item := no-comp-intrans-prep-lex-item &
                                       approximative-prep-lex-item.

static-transitive-prep-lex-item := static-prep-lex-item &
                                   transitive-prep-lex-item.
cofinal-transitive-prep-lex-item := transitive-prep-lex-item &
                                    cofinal-prep-lex-item.
coinitial-transitive-prep-lex-item := transitive-prep-lex-item &
                                      coinitial-prep-lex-item.
transitory-transitive-prep-lex-item := transitive-prep-lex-item &
                                       transitory-prep-lex-item.
approximative-transitive-prep-lex-item := transitive-prep-lex-item &
                                          approximative-prep-lex-item.

cofinal-or-transitory-transitive-prep-lex-item := transitive-prep-lex-item &
                                                  cofinal-or-transitory-prep-lex-item.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; MODE PREDs
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

p_rel := predsort.
mode_rel := p_rel.
loc_rel := p_rel.
id_rel := p_rel.

;; modaliser predicates the hierarchy of mode_rels is built to underspecify
;; for selection (static/directional) and underspecify directional mode
;; predicates (cofinal/transitory)


directional_mode_rel := mode_rel.

static_or_cofinal_or_transitory_mode_rel := mode_rel.
static_or_cofinal_or_transitory_id_rel := static_or_cofinal_or_transitory_mode_rel &
                                          id_rel.

static_or_cofinal_mode_rel := static_or_cofinal_or_transitory_mode_rel.
static_or_cofinal_id_rel := static_or_cofinal_mode_rel & id_rel.

cofinal_or_transitory_mode_rel := static_or_cofinal_or_transitory_mode_rel &
                                  directional_mode_rel.

static_or_coinitial_mode_rel := mode_rel.
static_or_coinitial_id_rel := static_or_coinitial_mode_rel & id_rel.

static_mode_rel := static_or_cofinal_mode_rel & static_or_coinitial_mode_rel.
static_id_rel := static_mode_rel & id_rel.

coinitial_mode_rel := static_or_coinitial_mode_rel & dir_not_cofinal_mode_rel.
coinitial_id_rel := coinitial_mode_rel & id_rel.
```

```
transitory_mode_rel := cofinal_or_transitory_mode_rel & dir_not_cofinal_mode_rel.
transitory_id_rel := transitory_mode_rel & id_rel.

cofinal_mode_rel := cofinal_or_transitory_mode_rel & static_or_cofinal_mode_rel.
cofinal_id_rel := cofinal_mode_rel & id_rel.

approximative_mode_rel := dir_not_cofinal_mode_rel.
approximative_id_rel := approximative_mode_rel & id_rel.

;; one complementary mode, to constrain PP modifiation

dir_not_cofinal_mode_rel := directional_mode_rel.

;; localiser predicates

i_loc_rel := loc_rel.
på_loc_rel := loc_rel.
ved_loc_rel := loc_rel.
ved_id_rel := ved_loc_rel & id_rel.
under_loc_rel := loc_rel.
over_loc_rel := loc_rel.
mellom_loc_rel := loc_rel.
ute_loc_rel := loc_rel.
inne_loc_rel := loc_rel.
oppe_loc_rel := loc_rel.
her_loc_rel := loc_rel.
der_loc_rel := loc_rel.
hjemme_loc_rel := loc_rel.
borte_loc_rel := loc_rel.


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ONOTLOGICAL types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; LOC-IND is a location-denoting index, as opposed
;; to REF-IND, which denote proper objects

loc-ind := index.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; PHRASE types
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

head-final-subj-phrase := basic-head-subj-phrase & head-final & head-compositional &
[ SYNSEM.LOCAL.CAT [ HEAD verb,
                     VAL.COMPS null ],
  ARGS.REST.FIRST.SYNSEM.LOCAL.CAT.VAL.COMPS olist ].

head-final-spec-phrase := basic-head-spec-phrase & head-final.

head-initial-comp-phrase := basic-head-comp-phrase & head-initial.

head-modifier-int-phrase := head-adj-int-phrase.
```

A.2 lexicon.tdl

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; NOUNS and DETERMINERS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mann := noun-lex-item &
  [ STEM < "mann" >,
    SYNSEM.LKEYS.KEYREL.PRED "mann_n_rel" ].

dame := noun-lex-item &
  [ STEM < "dame" >,
    SYNSEM.LKEYS.KEYREL.PRED "dame_n_rel" ].

hage := noun-lex-item &
  [ STEM < "hage" >,
    SYNSEM.LKEYS.KEYREL.PRED "hage_n_rel" ].

garasje := noun-lex-item &
  [ STEM < "garasje" >,
    SYNSEM.LKEYS.KEYREL.PRED "garasje_n_rel" ].

ball := noun-lex-item &
  [ STEM < "ball" >,
    SYNSEM.LKEYS.KEYREL.PRED "ball_n_rel" ].

bord := noun-lex-item &
  [ STEM < "bord" >,
    SYNSEM.LKEYS.KEYREL.PRED "bord_n_rel" ].

en := det-lex-item &
  [ STEM < "en" >,
    SYNSEM.LKEYS.KEYREL.PRED "some_q_rel" ].

et := det-lex-item &
  [ STEM < "et" >,
    SYNSEM.LKEYS.KEYREL.PRED "some_q_rel" ].

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; TRANSITIVE PREPOSITIONS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; MODALISER
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

fra := modaliser-dir-prep-lex-item &
  [ STEM < "fra" >,
    SYNSEM.LKEYS.KEYREL.PRED coinitial_mode_rel ].
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; TRANSITIVE LOCATIVES (according to Table 7.2)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;----
;; på
;;----

;; - static
på_p_st := static-transitive-prep-lex-item &
  [ STEM < "på" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED på_loc_rel].

;; - coinitial
på_p_ci := coinitial-transitive-prep-lex-item &
  [ STEM < "av" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED på_loc_rel].

;; - cofinal
på_p_cf := cofinal-transitive-prep-lex-item &
  [ STEM < "på" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED på_loc_rel].

;; - transitory
på_p_tr := transitory-transitive-prep-lex-item &
  [ STEM < "over" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED på_loc_rel].

;; approximative: not realized

;;---
;; i
;;---

;; -static
i_p_st := static-transitive-prep-lex-item &
  [ STEM < "i" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED i_loc_rel ].

;; -coinitial: 'ut av', no representation

;; -cofinal
i_p_cf := cofinal-transitive-prep-lex-item &
  [ STEM < "i" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED i_loc_rel ].

;; -transitory
i_p_tr := transitory-transitive-prep-lex-item &
  [ STEM < "gjennom" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED i_loc_rel ].

;; -approximative: not realized
```

```
;;-----
;; ved
;;-----

;; -static
ved_p_st := static-transitive-prep-lex-item &
  [ STEM < "ved" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ved_loc_rel].

;; -coinitial: ut av???
ved_p_ci := coinitial-transitive-prep-lex-item &
  [ STEM < "fra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ved_loc_rel].

;; -cofinal
ved_p_cf := cofinal-transitive-prep-lex-item &
  [ STEM < "til" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ved_loc_rel].

;; -transitory
ved_p_tr := transitory-transitive-prep-lex-item &
  [ STEM < "forbi" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ved_loc_rel].

;; -approximative:

ved_p_ap := approximative-transitive-prep-lex-item &
  [ STEM < "mot" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ved_loc_rel].

;;-------
;; under
;;-------

;; -static:
under_p_st := static-transitive-prep-lex-item &
  [ STEM < "under" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED under_loc_rel].

;; -cofinal-or-transitory:
under_p_cf_or_tr := cofinal-or-transitory-transitive-prep-lex-item &
  [ STEM < "under" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED under_loc_rel].

;;------
;; over
;;------

;; -static
over_p_st := static-transitive-prep-lex-item &
  [ STEM < "over" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED over_loc_rel].

;; -cointial: complement of fra, 'fra over'
```

```
;; -cofinal: complement of inn, 'inn over'

;; -transitory
over_p_tr := transitory-transitive-prep-lex-item &
  [ STEM < "over" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED over_loc_rel].

;;------
;; mellom
;;------

;; -static
mellom_p_st := static-transitive-prep-lex-item &
  [ STEM < "mellom" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED mellom_loc_rel].


;; -cointial: complement of fra, 'fra mellom

;; -cofinal: complement of inn, 'inn mellom

;; -transitory
mellom_p_tr := transitory-transitive-prep-lex-item &
  [ STEM < "mellom" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED mellom_loc_rel].

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; INTRANSITIVE LOCATIVES (according to Table 7.3)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;-----
;; her
;;-----

;; -static:
her_p_st := static-intrans-prep-lex-item &
  [ STEM < "her" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED her_loc_rel].

;; -cofinal:
her_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "hit" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED her_loc_rel].

;; -coinitial:
her_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "herfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED her_loc_rel].

;; -approxomative:
her_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "hitover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED her_loc_rel].
```

```
;;-----
;; der
;;-----

;; -static:
der_p_st := static-intrans-prep-lex-item &
  [ STEM < "der" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED der_loc_rel].

;; -cofinal:
der_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "dit" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED der_loc_rel].

;; -coinitial:
der_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "derfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED der_loc_rel].

;; -approxomative:
der_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "ditover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED der_loc_rel].

;;----
;; inne
;;----

;; -static:
inne_p_st := static-intrans-prep-lex-item &
  [ STEM < "inne" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED inne_loc_rel].

;; -cofinal:
inne_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "inn" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED inne_loc_rel].

;; -coinitial:
inne_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "innenfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED inne_loc_rel].


;; -approxomative:
inne_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "innom" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED inne_loc_rel].
```

```
;;----
;; oppe
;;----

;; -static:
oppe_p_st := static-intrans-prep-lex-item &
  [ STEM < "oppe" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED oppe_loc_rel].

;; -cofinal:
oppe_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "opp" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED oppe_loc_rel].

;; -coinitial:
oppe_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "ovenfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED oppe_loc_rel].

;; -approxomative:
oppe_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "oppover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED oppe_loc_rel].

;;-----
;; borte
;;-----

;; -static:
borte_p_st := static-intrans-prep-lex-item &
  [ STEM < "borte" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED borte_loc_rel].

;; -cofinal:
borte_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "bort" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED borte_loc_rel].

;; -coinitial:
borte_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "bortenfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED borte_loc_rel].

;; -approxomative:
borte_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "bortover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED borte_loc_rel].
```

```
;;-----
;; hjemme
;;-----

;; -static:
hjemme_p_st := static-intrans-prep-lex-item &
  [ STEM < "hjemme" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED hjemme_loc_rel].

;; -cofinal:
hjemme_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "hjem" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED hjemme_loc_rel].

;; -coinitial:
hjemme_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "hjemmefra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED hjemme_loc_rel].

;; -approxomative:
hjemme_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "hjemover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED hjemme_loc_rel].

;;----
;; ut
;;----

;; -static:
ute_p_st := static-intrans-prep-lex-item &
  [ STEM < "ute" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ute_loc_rel].

;; -cofinal:
ute_p_cf := cofinal-intrans-prep-lex-item &
  [ STEM < "ut" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ute_loc_rel].

;; -coinitial:
ute_p_ci := coinitial-intrans-prep-lex-item &
  [ STEM < "utenfra" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ute_loc_rel].

;; -approxomative:
ute_p_ap := approximative-intrans-prep-lex-item &
  [ STEM < "utover" >,
    SYNSEM.LKEYS.ALTKEYREL.PRED ute_loc_rel].
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; VERBS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; motion verbs

rusler := intrans-motion-verb-lex-item &
  [ STEM < "rusler" >,
    SYNSEM.LKEYS.KEYREL.PRED "rusle_v_re" ].

sprinter := intrans-motion-verb-lex-item &
  [ STEM < "sprinter" >,
    SYNSEM.LKEYS.KEYREL.PRED "sprint_v_re" ].

;; static verbs

sitter := intrans-static-verb-lex-item &
  [ STEM < "sitter" >,
    SYNSEM.LKEYS.KEYREL.PRED "sitte_v_re" ].

;; verbs with lp-complements

bo_v := lp-trans-verb-lex-item &
  [ STEM < "bor" >,
    SYNSEM.LKEYS.KEYREL.PRED "bo_v_rel"].

;; dative alternating verbs

gi_v := np-pp-ditrans-verb-lex-item &
  [ STEM < "gir" >,
    SYNSEM.LKEYS.KEYREL.PRED "gi_v_rel"].

;; verbs of putting

legge_v := np-lp-ditrans-verb-lex-item &
  [ STEM < "legger" >,
    SYNSEM.LKEYS.KEYREL.PRED "legge_v_rel"].
```

A.3 rules.tdl

```
head-comp-rule := head-initial-comp-phrase.

head-final-subj-rule := head-final-subj-phrase.

head-final-spec-rule := head-final-spec-phrase.

head-modifier-int-rule := head-modifier-int-phrase.
```

A.4 labels.tdl

```
s-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD verb,
                       VAL [ SUBJ <anti-synsem>,
                             COMPS <> ] ] ],
  LABEL-NAME "S" ].

n-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD noun,
                       VAL [ SPR < synsem > ] ] ],

  LABEL-NAME "N" ].

np-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD noun,
                       VAL [ SPR null ] ] ],

  LABEL-NAME "NP" ].

v-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD verb,
                       VAL [ SUBJ < synsem >,
                             COMPS cons ] ] ],
  LABEL-NAME "V" ].

vp-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD verb,
                       VAL [ SUBJ < synsem >,
                             COMPS olist ] ] ],
  LABEL-NAME "VP" ].

det-label := label &
[ SYNSEM [ LOCAL.CAT [ HEAD det,
                       VAL [ SUBJ null,
                             COMPS null ] ] ],
  LABEL-NAME "DET" ].

mp-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep,
                     VAL.COMPS olist ],
                 CONT.HOOK.INDEX event ],
  LABEL-NAME "MP" ].

m-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep,
                     VAL.COMPS cons ],
                 CONT.HOOK.INDEX event ],
  LABEL-NAME "M" ].
```

```
lp-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep,
                     VAL.COMPS olist ],
                 CONT.HOOK.INDEX loc-ind ],
  LABEL-NAME "LP" ].

l-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep,
                     VAL.COMPS cons ],
                 CONT.HOOK.INDEX loc-ind ],
  LABEL-NAME "L" ].

pp-sel-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep & [ KEYS.ALTKEY id_rel ],
                       VAL.COMPS olist],
                 CONT.HOOK.INDEX ref-ind ],
  LABEL-NAME "PP-SEL" ].

p-sel-label := label &
[ SYNSEM.LOCAL [ CAT [ HEAD prep & [ KEYS.ALTKEY id_rel ],
                       VAL.COMPS cons ],
                 CONT.HOOK.INDEX ref-ind ],
  LABEL-NAME "P-SEL" ].
```