Type theory and language
From perception to linguistic communication

Robin Cooper

Draft, December 9, 2014
PLEASE QUOTE WITH CARE

Contents

Acknowledgements

I am grateful to many people for discussion which has led to significant changes in this material. Among them are: Simon Dobnik, Tim Fernando, Jonathan Ginzburg, Staffan Larsson, Bengt Nordström, Aarne Ranta. None of these people is responsible for what I have done with their ideas and suggestions.

This research was supported in part by the following projects: Records, types and computational dialogue semantics, Vetenskapsrådet, 2002-4879, Library-based grammar engineering, Vetenskapsrådet, 2005-4211, and Semantic analysis of interaction and coordination in dialogue (SAICD), Vetenskapsrådet, 2009-1569.

Part I

From perception and action to grammar

Part II

Towards a dialogical view of semantics

Chapter 4

Proper names, salience and accommodation

4.1 Montague's PTQ as a semantic benchmark

In this chapter and the following chapters we will extend the linguistic coverage of the toy grammar we presented in Chapter 3. We will take Montague's PTQ (Montague, 1973, 1974) as providing a benchmark of linguistic phenomena that need to be covered and try to cover a sizeable part of what Montague covered, although we will add a few things which are obviously closely related to Montague's original benchmark and which have been treated subsequently in the literature.

For many of the phenomena we discuss we will first present a treatment which is as close as possible to Montague's original treatment and then present a treatment which exploits the advantages of the approach we are proposing in this book as well as more recent developments since Montague's original work. Our aim is to show that we have something to say about all these phenomena in an overall consistent framework, that is, to show that we can cover a significant part of the benchmark using the tools we are proposing and in many cases say something new concerning a dialogical approach to these phenomena. In doing this within the space of a single book we will not be able to cover all the aspects of these phenomena which have been studied in the literature following after Montague. We hope, however, to show that it is a fruitful line of research to add a rich type theoretic perspective and a dialogical approach to current work in linguistic semantics.

4.2 Montague's treatment of proper names and a sign-based approach

The treatment of proper names that we presented in Chapter 3, encapsulated in the definition of SemPropName and $\text{Lex}_{\text{PropName}}$ in Appendix B, is an adaptation of Montague's original treatment in that it has the content of a proper name utterance as a quantifier generated from an individual. The essence of Montague's treatment was that if we have a proper name *Sam* whose denotation is based on an individual 'sam', then the denotation of *Sam* is the characteristic function of the set of properties possessed by the individual concept of 'sam'. Montague modelled individual concepts as functions from possible worlds to individuals. Using more or less Montague's logical notation, the denotation of *Sam* would be represented by (1).

(1) $\lambda P.P\{[\hat{}\mathrm{sam}]\}$

Here $[\hat{}\mathrm{sam}]$ represents the individual concept of 'sam', that is, that function, f, on the set of possible worlds such that for any world w, $f(w) = \mathrm{sam}$. The reason that Montague used the individual concept (and the associated special notion of application involved in applying a property to an individual concept represented by the '$\{\}$'-brackets) was to treat what is known as the Partee-puzzle concerning temperature and price which we will discuss in Chapter 5. Many subsequent researchers came to the conclusion that Montague's treatment of this puzzle was not the correct one and that the individual concept was not necessary in the treatment of proper names. Thus (1) could be simplified to (2).

(2) $\lambda P.P(\mathrm{sam})$

The content that we assigned to an utterance of *Sam* in Chapter 3 is represented in (3).

(3) $\lambda P{:}\mathit{Ppty}.P(\left[\text{x=sam}\right])$

The reason that we have chosen to characterize properties as having records as their domain rather than individuals, has to do with our treatment of the Partee puzzle as we will explain in Chapter 5. Thus the reason that we have the record $\left[\text{x=sam}\right]$ as the argument to the property rather than an individual as in (4) is for the same reason as Montague introduced an individual concept.

(4) $\lambda P{:}\mathit{Ppty}.P(\text{sam})$

The treatment of proper names we presented in Chapter 3 has an important advantage over Montague's original. For Montague, (1) is the result of applying an interpretation function to the linguistic expression *Sam* and a number of indices for the interpretation, $\mathfrak{A}$, a possible world, i, a time, j, and an assignment to variables, g. This is represented in (5).

(5) $[\![\text{ Sam }]\!]^{\mathfrak{A},i,j,g} = \lambda P.P\{[\hat{}\text{sam}]\}$

This requires that the English expression *Sam* is always associated with the same individual 'sam' with respect to $\mathfrak{A}$ and any i, j, g related to $\mathfrak{A}$. This seems to go against the obvious fact that more than one individual can have the name *Sam*. It does not work to say that a different individual can be associated with *Sam* when it is evaluated with respect to different parameters. g is irrelevant since it is defined as an assignment to variables and the English expression *Sam* is not (associated with) a variable — it cannot be bound by a quantifier.[1] A strategy which involves varying the possible world and time to get a different individual associated with *Sam* would be defeated by the fact that there are many people called Sam in the actual world right now as well as having the unintuitive consequence that *Sam might be Sam* would be true if it is true that Sam might be somebody else called Sam and *Sam will be Sam* could be true if somebody called Sam now is somebody else called Sam at a future time. We might try saying that associating a different individual with Sam involves a different interpretation, $\mathfrak{A}'$, of the language. This has some intuitive appeal and we will discuss a variant of it in Section 4.5 in relation to a recent proposal by Ludlow (2014). But it will come to grief when we need to talk about two people named Sam in the same sentence unless we allow a switch in interpretation mid-sentence. While allowing interpretation to change mid-sentence may be an attractive option for other reasons it is not an option that is available on Montague's account of meaning. The normal assumption is that in cases where two individuals have the same name the language contains two expressions which are pronounced the same, for example, *Sam*$_1$ and *Sam*$_2$. This would make the treatment of proper names somewhat like Montague's treatment of pronouns in that they have silent numerical subscripts attached to them. How many *Sam*$_i$ should the language contain? One for each person named Sam, now, in the past and future and who could be named Sam in some non-actual world? If we follow the strategy with variables we would introduce countably many *Sam*$_i$ so that we would always have enough. But with assignments to variables we can always assign individuals to more that one variable without this causing a problem. But the consequence of doing this with proper names would be to say that an individual can have many names that are pronounced the same. (Sam says, "My name is Sam", not "My names are Sam".) Similarly no two individuals would have the same name, although they would be able to have distinct names which are pronounced the same. This would mean that the interpretation of *have the same name* would have to mean "have names which are pronounced the same". This might cause difficulties distinguishing between a case where we have two people named Sam and a case where people really do have distinct names which are pronounced the same such as *Ann* and *Anne* (unless you want to count this as a case of spelling the same name differently).

[1]This claim has been called into question by more recent research. See Maier (2009) for discussion.

In contrast the analysis of proper names we presented in Chapter 3 is sign-based. It allows several sign types to share the same phonology but be associated with different contents. Treating the language in terms of signs eliminates the need for arbitrary indexing of proper names. It also allows us to individuate names in a sensible way. One way to individuate names is by the phonologies occurring in proper name sign types. Thus if we have two proper name sign types with the same phonology but contents associated with different individuals, then we have two individuals with the same name. Note that this proposal would make *Ann* and *Anne* different spellings of the same name since they are both associated with the same phonological type. How we individuate names can be different in different contexts if we follow the kind of proposal for counting discussed by Cooper (2011). We could, for example, introduce a field into lexical sign types for an orthographical type and allow the individuation of names by either phonology or orthography or a combination of both depending on what is most useful to the purpose at hand.

Using signs in this way seems to give us a clear, if rather simple, advantage over Montague's formal language approach, even though we have so far essentially just transplanted Montague's analysis of proper names into our variant of a sign-based approach. However, there is a remaining question within sign-based approaches which is a kind of correlate to the need on Montague's approach to create many different names *Sam*$_i$. We are tempted to think of a "language" as being defined as a collection of sign types. Thus a person who knows English will know sign types which pair the phonological type "Sam" with various individuals who are called Sam. The problem with this is that different speakers of English will know different people named Sam and thus technically we would have to say that they speak different languages. This may well be a coherent technical notion of language. In the terminology of Chapter 3 we would say that the two agents indeed have different linguistic resources available to them. But there is also a resource which the two agents share, even if they do not have any overlap in the people named Sam that they are aware of. This is the knowledge that *Sam* is a proper name in English and can be used to name individuals. Arguably it is this knowledge which is constitutive of English, rather than the knowledge of who is actually called Sam, important though that might be for performing adequately in linguistic situations. In Chapter 3 we introduced sign type contruction operations and in particular 'Lex$_{\mathrm{PropName}}$' which maps a phonological type and an individual to an appropiate proper name sign type (see Appendix B). We called this a universal resource since it represents the general knowledge that utterances can be used to name individuals. In the English resources we defined there we named sign types such as 'Lex$_{\mathrm{PropName}}$("Sam", sam)', where we specify both the phonological type and the individual associated with it. But, given the power of functional abstraction, we can identify (6) as an English resource where the phonological type is specified but not the particular individual.

(6) λx:*Ind* . Lex$_{\mathrm{PropName}}$("Sam", x)

Saying that an agent has this function available as an English resource could be argued to encode the fact that the agent has the knowledge that *Sam* is a proper name in English. An agent who has this resource has a recipe for constructing an appropriate sign type in their resources whenever they meet somebody called Sam. Knowing that *Sam* is a proper name in English is not a matter of knowing who is called Sam but rather a matter of knowing what to do linguistically when you encounter somebody called Sam. Thus while we have so far just taken over Montague's original analysis of proper names we have given ourselves the opportunity to recast it in terms of a theory which enables agents to update their linguistic resources as they become aware of new facts about the world.

4.3 Proper names and communication

However, what we have done so far tells us little about the communicative processes associated with utterances of proper names. In Cooper (2013b) we pointed out that this kind of analysis does not give us any way of placing the requirement on the interlocutor's gameboard that there already be a person named Sam available in order to integrate the new information onto the gameboard. As Ginzburg (2012) points out, the successful use of a proper name to refer to an individual a requires that the name be publically known as a name for a. We will follow the analysis of Cooper (2013b) in parametrizing the content. A *parametric content* is a function which maps a context to a content. As such it relates to Montague's technical notion of *meaning* in his paper 'Universal Grammar' (Montague, 1970, 1974) where he regarded meaning as a function from possible worlds and contexts of use to denotations. This also corresponds to the notion of *character* in Kaplan (1978).

We will take a context to be a situation modelled as a record. A simple proposal for a parametric content for a proper name might be (7).

(7) $\lambda r{:}\big[\text{x:}\textit{Ind}\big]\ .$
$\qquad \lambda P{:}\textit{Ppty}\ .\ P(r)$

This would allow any record with an individual labelled 'x' to be mapped to a proper name content. Recall that the label 'x' is picked up by the notion of property that we defined in Chapter 3 as being of type ($\big[\text{x:}\textit{Ind}\big]\rightarrow\textit{RecType}$), an example being (8).

(8) $\lambda r{:}\big[\text{x:}\textit{Ind}\big]\ .\ \big[\text{e:run}(r.\text{x})\big]$

Associating the phonological type "Sam" with (7) would essentially be a way of encapsulating in the interpretation of *Sam* what is expressed by (6) — namely, that potentially any individual can be called Sam. We want the parametric content of *Sam* to be more restrictive than this. It is going to be the tool that we use to help us identify an appropriate referent when we are confronted with an utterance of type "Sam". The obvious constraint that we should place is that the referent is indeed named Sam. Thus we can restrict (7) so that it is an appropriate parametric content for *Sam* rather than something that appears to be a parametric content appropriate to proper names in general. The modification is given in (9).

(9) $$\lambda r{:}\begin{bmatrix}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{bmatrix}\,.\;\lambda P{:}\mathit{Ppty}\;.\;P(r)$$

This is closely related to treatments of proper names that were proposed earlier in situation semantics (Gawron and Peters, 1990; Cooper, 1991; Barwise and Cooper, 1993). A more recent close relation is Maier's (2009) proposal for the treatment of proper names in terms of layered discourse representation theory (LDRT). Maier points out in a useful overview of the history of semantic treatments of proper names that this view of proper names is a hybrid of the descriptivist and referential approaches: it uses a description like "named Sam" to provide a presuppositional restriction on the kind of referent which can be assigned to the proper name. (9) maps a context in which there is an individual named Sam to a proper name content based on that individual. Care has to be taken with the predicate 'named' on this kind of analysis. It is important that it not be too restrictive, for example, requiring the legal registering of the name. It may be sufficient that someone at some point has called the individual by the name. The exact conditions under which a situation may be of a type constructed with this predicate will vary depending on the needs associated with the conversation at hand. We may, for example, take a stricter view of what it means to have a certain name if we are talking in a court of law than if we are trying to attract somebody's attention to avoid an accident on a mountainside. This flexibility of meaning "in flux" has been discussed in Cooper and Kempson (2008); Cooper (2012b); Ludlow (2014); Ginzburg and Cooper (2014); Kracht and Klein (2014) among many other places and we will return to it several times in the following chapters.

An alternative to the use of parametric contents is to use parametric signs. This could be formulated as in (10) where $\text{Lex}_{\text{PropName}}$ is the function for associating lexical content with phonological types that was introduced in Chapter 3 and summarized in Appendix B.1.2.

(10) $$\lambda r{:}\begin{bmatrix}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{bmatrix}\,.\;\text{Lex}_{\text{PropName}}(\text{“Sam”}, r.\text{x})$$

Intuitively, (10) says that given a situation in which there is an individual named by the phonological type "Sam" we can construct a sign type in which the phonological type "Sam" is associated with that individual. From the point of view of the formal semantics tradition (10) is a much more radical proposal than (9). The function (9) is a close relative of Montague's *meaning* and Kaplan's *character*. It is a function from contexts to contents, although our theory of what contexts and contents are differs from both Montague's and Kaplan's proposals. The function in (10), however, is something that creates a kind of linguistic resource on the basis of a context. That is, given a context in which 'sam' is named by "Sam" we derive the information that linguistic signs can be used which associate "Sam" with 'sam'. If we did not know this before we are extending the collection of linguistic resources we have available. We suspect that both parametric contents and parametric sign types could be of importance for a theory of linguistic interpretation and learning. For now, we will work with the less radical notion of parametric content.

There are two main questions that need to be answered about parametric contents. One concerns how the compositional semantics works and the other concerns the nature of contexts and how you compute with them. We will take the compositionality issue first. Let us assume that all signs provide us with a parametric content rather than a content. In those cases where there is no constraint on what the context must be we will use a trivial parametric content, that is, one that maps any context (modelled as a record) to the same content. Thus, for example, if we wish to represent a theory in which the intransitive verb *leave* does not place any restrictions on the context, we could represent its parametric content as (11).

(11) $\lambda r_1{:}\mathit{Rec}.\lambda r_2{:}\left[\text{x:}\mathit{Ind}\right].\ \left[\text{e:leave}(r_2.\text{x})\right]$

This will map any context r_1 to the function $\lambda r_2{:}\left[\text{x:}\mathit{Ind}\right].\ \left[\text{e:leave}(r_2.\text{x})\right]$ which does not depend in any way on r_1.

A standard strategy for dealing with compositional semantics when using parametric contents is to use a version of what is known in combinatorial logic as the S-combinator. In its λ-calculus version this is (12).

(12) $\lambda z.\alpha(z)(\beta(z))$

Our version of the S-combinator including different type requirements on the context arising from the function and the argument will be (13).

(13) if $\alpha : (T_1 \to (T_2 \to T_3))$ and $\beta : (T_4 \to T_2)$ then the *combination of α and β based on functional application* is

$$\lambda r{:}\begin{bmatrix}\text{f:}T_1\\ \text{a:}T_4\end{bmatrix}.\ \alpha(r.\text{f})(\beta(r.\text{a}))$$

Note that in the context requirements for the result we have kept the context types for the functor α and the argument β separated in their own fields labelled 'f' ("function") and 'a' ("argument").[2] This means that we avoid an unwanted clash of labels if T_1 and T_4 should happen to share labels. We could used (13) to combine the contents (9) and (11). This is given in (14) where we can show by successive applications of β-reduction that (14a–d) all represent the same function.

(14)

a. $\lambda r_1{:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}$.
$(\lambda r_2{:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix} . \lambda P{:}\textit{Ppty} . P(r_2))(r_1.\text{f})$
$(\lambda r_3{:}\textit{Rec}.\lambda r_4{:}\begin{bmatrix}\text{x:}\textit{Ind}\end{bmatrix}. \begin{bmatrix}\text{e:leave}(r_4.\text{x})\end{bmatrix}(r_1.\text{a}))$

b. $\lambda r_1{:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}$.
$\lambda P{:}\textit{Ppty} . P(r_1.\text{f})$
$(\lambda r_4{:}\begin{bmatrix}\text{x:}\textit{Ind}\end{bmatrix}. \begin{bmatrix}\text{e:leave}(r_4.\text{x})\end{bmatrix})$

c. $\lambda r_1{:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}$.
$\lambda r_4{:}\begin{bmatrix}\text{x:}\textit{Ind}\end{bmatrix}. \begin{bmatrix}\text{e:leave}(r_4.\text{x})\end{bmatrix}(r_1.\text{f})$

d. $\lambda r_1{:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}$.
$\begin{bmatrix}\text{e:leave}(r_1.\text{f.x})\end{bmatrix}$

(14) represents the parametric content of *Sam leaves*. Given a situation containing an individual, a, named by "Sam" it returns a type of situation in which a leaves. As usual this type can play the role of a "proposition". It is, for example, "true" if there is a situation of the type and "false" if there is no situation of the type.

The domain type on the parametric content is to be thought of as placing a constraint on the context. The idea is that you can only get to the non-parametric content if you have an appropriate situation available. The domain type of the parametric content represents a kind of *presupposition*. We shall treat presuppositions as constraints on the resources available to dialogue participants. In Chapter 2 we introduced the notion of a dialogue gameboard as a type of

[2] While textually this statement of the combination will be correct, we need to take account of the fact that the abbreviatory notation for labels in argument positions to predicates now represent path-names in T_1 and T_4 to which the labels 'f' and 'a' have been prefixed respectively. To be precise we could notate this is ${T_1}^{f\cdot}$ and ${T_4}^{a\cdot}$.

dialogue information state. The most obvious place to look for the referent of an utterance of a proper name is in the shared commitments represented on the gameboard representing what has been committed to in the dialogue so far. If an individual named Sam has already been introduced in the dialogue, then a subsequent utterance of *Sam* in that dialogue is most likely to refer to that individual unless there is an explicit indication to the contrary. The shared commitments on an agent's dialogue gameboard represent information that is particularly *salient* to the agent. The notion of salience in semantics was first introduced by Lewis (1979b) in connection with the analysis of definite descriptions. As Lewis says, "There are various ways for something to gain salience. Some have to do with the course of conversation, others do not." We wish to suggest that a way of gaining salience in a conversation is by figuring in the shared commitments on the gameboard. (Ginzburg (2012) argues that being on shared commitments, or FACTS in his terminology, is not always sufficient to indicate salience.)

A reasonable strategy, then, is to look at the shared commitments on the dialogue gameboard first and then look elsewhere if that fails. We will first explore what we need to do to match the domain type of a parametric content against the type which models the shared commitments of the dialogue and then we will discuss what needs to be done if there is not a successful match with the shared commitments. In Chapter 2 we treated the gameboard as a record type. In Chapter 2, example (74), for instance, the shared commitments were represented as the type (15).

(15) $$\left[\begin{array}{l}\text{prev:}\left[\begin{array}{l}\text{prev:}\left[\begin{array}{l}\text{prev:}\mathit{Rec}\\ \text{e:conductor(dudamel)}\end{array}\right]\\ \text{e:composer(beethoven)}\end{array}\right]\\ \text{e:pianist(uchida)}\end{array}\right]$$

Recall that with each successive updating of the shared commitments the record type representing the previous state of shared commitments was embedded under the label 'prev' ("previous"). This prevented label clash and also kept a record of the order in which information was introduced. As Lewis (1979b) observed, information introduced later in the dialogue tends to be more salient than information introduced earlier. Thus keeping track of the order also gives us one measure of relative salience.

In Chapter 2 we were using the Montague treatment of proper names that did not introduce the naming predicate. In this chapter we will work towards shared commitments where the naming associated with proper names is made explicit, as in (16).

$$
(16)\quad \left[\begin{array}{l}
\text{prev:}\left[\begin{array}{l}
\text{prev:}\left[\begin{array}{l}
\text{prev:}\mathit{Rec}\\
\text{bg:}\left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Dudamel”)}\end{array}\right]\\
\text{e:conductor(bg.x)}
\end{array}\right]\\
\text{bg:}\left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Beethoven”)}\end{array}\right]\\
\text{e:composer(bg.x)}
\end{array}\right]\\
\text{bg:}\left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Uchida”)}\end{array}\right]\\
\text{e:pianist(bg.x)}
\end{array}\right]
$$

Here we are using the label ‘bg’ to represent background information in the manner suggested by Larsson (2010). Note that in this version of the shared commitments we have lost the connection with the actual individuals ‘dudamel’, ‘beethoven’ and ‘uchida’. This can be seen as an advantage if we are representing the information state of an agent in the kind of situation described in Chapter 2. If we simply inform an agent with no previous knowledge of Dudamel that Dudamel is a conductor, then the information that this agent will get is that there is somebody named Dudamel who is a conductor. There will be no connection to a particular individual of whom the agent is aware. If this is not the case, we can reinstate the connection to the individuals by using manifest fields to anchor the information as in (17).

$$
(17)\quad \left[\begin{array}{l}
\text{prev:}\left[\begin{array}{l}
\text{prev:}\left[\begin{array}{l}
\text{prev:}\mathit{Rec}\\
\text{bg:}\left[\begin{array}{l}\text{x=dudamel:}\mathit{Ind}\\ \text{e:named(x, “Dudamel”)}\end{array}\right]\\
\text{e:conductor(bg.x)}
\end{array}\right]\\
\text{bg:}\left[\begin{array}{l}\text{x=beethoven:}\mathit{Ind}\\ \text{e:named(x, “Beethoven”)}\end{array}\right]\\
\text{e:composer(bg.x)}
\end{array}\right]\\
\text{bg:}\left[\begin{array}{l}\text{x=uchida:}\mathit{Ind}\\ \text{e:named(x, “Uchida”)}\end{array}\right]\\
\text{e:pianist(bg.x)}
\end{array}\right]
$$

The ‘bg’-fields in (16) can be thought of as corresponding to the internal anchors of Kamp (1990); Kamp *et al.* (2011). The use of manifest fields in (17) would then correspond to the association of what they call external anchors with those internal anchors.

The task we have before us is to try to match the domain type of the function in (14), that is (18), against the types of shared commitments in (16) or (17).

$$
(18)\quad \left[\begin{array}{ll}
\text{f:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{a:}\mathit{Rec}}
\end{array}\right]
$$

Intuitively, this attempt at matching should fail since there is no commitment to an individual named Sam in the shared commitments. Suppose now that we add to (16) as in (19).

$$
(19)\quad \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\text{prev:}\mathit{Rec}}\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Dudamel”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:conductor(bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Beethoven”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:composer(bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Uchida”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:pianist(bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:singer(bg.x)}}
\end{array}\right]
$$

Intuitively, this should enable a match since this does commit to an individual named Sam. However, there is not a direct formal relationship between (18) and (19) corresponding to this intuition. We will use flattening and relabelling of record types in order to capture the relationship. First recall that (19) is an abbreviated form of (20) where we have expanded the paths of the labels which are used as arguments to predicates. (We use ℓ^n for $\ell.\ell.\ldots.\ell$ where the label ℓ occurs n times.)

$$
(20)\quad \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\text{prev:}\mathit{Rec}}\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(prev}^3\text{.bg.x, “Dudamel”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:conductor(prev}^3\text{.bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(prev}^2\text{.bg.x, “Beethoven”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:composer(prev}^2\text{.bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(prev.bg.x, “Uchida”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:pianist(prev.bg.x)}}
\end{array}\right]\\
\text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(bg.x, “Sam”)}\end{array}\right]\\
\multicolumn{2}{l}{\text{e:singer(bg.x)}}
\end{array}\right]
$$

The result of flattening (20) will be a new type (21) where each path has been replaced by a single complex label consisting of the sequence of labels on the path (which we represent using the normal dot-notation for paths).

$$(21)\quad \begin{bmatrix} \text{prev}^4 & : & \mathit{Rec} \\ \text{prev}^3\text{.bg.x} & : & \mathit{Ind} \\ \text{prev}^3\text{.bg.e} & : & \text{named(prev}^3\text{.bg.x, “Dudamel”)} \\ \text{prev}^3\text{.e} & : & \text{conductor(prev}^3\text{.bg.x)} \\ \text{prev}^2\text{.bg.x} & : & \mathit{Ind} \\ \text{prev}^2\text{.bg.e} & : & \text{named(prev}^2\text{.bg.x, “Beethoven”)} \\ \text{prev}^2\text{.e} & : & \text{composer(prev}^2\text{.bg.x)} \\ \text{prev.bg.x} & : & \mathit{Ind} \\ \text{prev.bg.e} & : & \text{named(prev.bg.x, “Uchida”)} \\ \text{prev.e} & : & \text{pianist(prev.bg.x)} \\ \text{bg.x} & : & \mathit{Ind} \\ \text{bg.e} & : & \text{named(bg.x, “Sam”)} \\ \text{e} & : & \text{singer(bg.x)} \end{bmatrix}$$

While (20) and (21) are distinct record types which do not share any witnesses there is nevertheless a strong equivalence between them in that for any record which is of the type (20) there is a multiset extensionally equivalent record (see Appendix A.12) of type (21) and *vice versa*. There is a one-one mapping between the two types which preserves multiset extension. Intuitively, this means that the two types represent the same basic commitments about the world, namely Dudamel is a conducor, Beethoven is a composer, Uchida is a pianist and Sam is a singer. The difference between the two types involves the structure they impose on this world. In the case of (21) we have one big situation in which all of these facts hold and in (20) we have a situation which is made up of several smaller situations for each of the individuals involved. Note, however, that because we have used the complex labels representing the paths we are able to recreate that structure from the flattened type in (21). Note also that we can still read off the relative salience of the various individuals and facts by checking the number of occurrences of ‘prev’ in the label.

We can also flatten the type we are trying to match, that is (18). The result is (22).

$$(22)\quad \begin{bmatrix} \text{f.x} & : & \mathit{Ind} \\ \text{f.e} & : & \text{named(f.x, “Sam”)} \\ \text{a} & : & \mathit{Rec} \end{bmatrix}$$

In order to match (22) against (21) we look for a relabelling of (22) that would make (21) be a subtype of (22). Such a relabelling is given in (23).

(23) $\left[\begin{array}{lll} \text{bg.x} & : & \textit{Ind} \\ \text{bg.e} & : & \text{named(bg.x, “Sam”)} \\ \text{prev}^4 & : & \textit{Rec} \end{array}\right]$

This means, then, that any situation which is of the type required by the shared commitments would, modulo the relabelling, be of the type required for the domain of the parametric content (14). The domain of the parametric content is being used as a presupposition which is being matched against the hearer's current information state.

Given that we have now found a match, how can we go about updating the shared commitments with the new information represented by the parametric content? The technique for doing this will involve the notion of a *fixed point type*. Given a function, f, that returns a type, we can always ask the question: is there an object, a, such that $a : f(a)$. That is, a is of the type which is the result of applying f to a itself. This means that a is a *fixed point* for f. This raises a further question: can we characterize a type T such that any a which is of type T will be a fixed point for f? In this case T will be called a *fixed point type* for f. Consider the example of the parametric content of *Sam left* in (14) repeated as (24a). A fixed point type of this function is given in (24b).

(24)

a. $\lambda r_1{:}\left[\begin{array}{l} \text{f:}\left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Sam”)} \end{array}\right] \\ \text{a:}\textit{Rec} \end{array}\right] . \left[\text{e:leave}(r_1\text{.f.x})\right]$

b. $\left[\begin{array}{l} \text{f:}\left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Sam”)} \end{array}\right] \\ \text{a:}\textit{Rec} \\ \text{e:leave(f.x)} \end{array}\right]$

Any record of type (24b) will be a fixed point for (24a). (24b) can be obtained by applying the fixed point type construction function, $\mathcal{F}$, to (24a). This function is defined explicitly in Appendix A.13, p. 172.

If we are updating (19) with the parametric content (24a) then the result should be (25) where (19) has been embedded under the label ‘prev’ and the new information provided by the parametric content has been added at the top level of the new type, suitably relabelled so as to pick out the individual named Sam which has been previously introduced.

$$
(25)\quad \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\text{prev:} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\text{prev:}\textit{Rec}} \\
\text{bg:} & \left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Dudamel”)} \end{array}\right] \\
\multicolumn{2}{l}{\text{e:conductor(bg.x)}}
\end{array}\right] \\
\text{bg:} & \left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Beethoven”)} \end{array}\right] \\
\multicolumn{2}{l}{\text{e:composer(bg.x)}}
\end{array}\right] \\
\text{bg:} & \left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Uchida”)} \end{array}\right] \\
\multicolumn{2}{l}{\text{e:pianist(bg.x)}}
\end{array}\right] \\
\text{bg:} & \left[\begin{array}{l} \text{x:}\textit{Ind} \\ \text{e:named(x, “Sam”)} \end{array}\right] \\
\multicolumn{2}{l}{\text{e:singer(bg.x)}}
\end{array}\right] \\
\multicolumn{2}{l}{\text{e:leave(prev.bg.x)}}
\end{array}\right]
$$

We can achieve this update by using the tools of flattening and relabelling that we have just introduced. Suppose that T_{comm} is the type representing shared commitments that we wish to update with a function $f : (T_{\text{bg}} \rightarrow \textit{Type})$ corresponding to a parametric content for some utterance. The first thing to do is embed T_{comm} under the label ‘prev’, obtaining $\left[\text{prev:}T_{\text{comm}}\right]$. We need to consider the flattened version of this type, that is, $\varphi(\left[\text{prev:}T_{\text{comm}}\right])$. We need to find a relabelling, η, of the flattened version of T_{bg} such that $\varphi(\left[\text{prev:}T_{\text{comm}}\right]) \sqsubseteq [\varphi(T_{\text{bg}})]_\eta$, that is, the flattened version of $\left[\text{prev:}T_{\text{comm}}\right]$ is a subtype of result of relabelling the flattened version of T_{bg} with η. Having found such an η we use it to relabel the fixed point type, $\mathcal{F}(f)$. η in general will not be defined on all the labels of $\mathcal{F}(f)$ since its set of labels will be a superset of the set of labels of T_{bg}. What we need is a relabelling of $\mathcal{F}(f)$, η' which is an extension of η, that is, the domain of η is a subset of the domain of η' and for any label, ℓ, in the domain of η, $\eta'(\ell) = \eta(\ell)$. We can now compute the merge of the two flattened types we have obtained, that is, $\varphi(\left[\text{prev:}T_{\text{comm}}\right]) \dot{\wedge} [\varphi(\mathcal{F}(f))]_{\eta'}$. Finally, we can unflatten the resulting merge, obtaining the final result of the update, $\varphi^{-}(\varphi(\left[\text{prev:}T_{\text{comm}}\right]) \dot{\wedge} [\varphi(\mathcal{F}(f))]_{\eta'})$.

4.4 Proper names, salience and accommodation

What we have presented so far enables us to find a match for presuppositions introduced by a parametric content when such a match is present in shared commitments. Suppose there is more than one such match. In that case there will be a choice of relabellings η. In this case we may wish to choose the relabelling that corresponds to a match with the most salient match in terms of recency of introduction into the shared commitments. Technically, this means that we choose the relabelling which introduces labels with the least number of occurrences of ‘prev’. This would be an oversimplified notion of saliency in terms of recency since it is based on the first introduction of the referent into the discourse. Obviously any subsequent mention of the referent will raise its salience in terms of recency. Also there may be other factors than recency which contribute to salience. We will leave it to future work to give a more detailed account of saliency.

What happens when there is no match for *Sam* in the shared commitments? Here we need some kind of accommodation in order to use the parametric content to update the gameboard. There are two kinds of accommodation we will consider. The first is where the agent knows of a person named Sam independently of the current conversation. That is, a match for *Sam* can be found in the agent's resources corresponding to long term memory. We will not attempt a detailed account of the stucture of long term memory. We assume that it is complex and constantly in flux not only in terms of new information being added but also in terms of what is salient in the old information, depending on which part of the memory is being focussed on at any particular time. Here we will content ourselves with a simple model of long term memory as a record type of a similar kind to that we have proposed for shared commitments. This means that the techniques we need for matching will be the same as those discussed above. In reality the notion of salience with respect to long term memory will be a good deal more complicated than salience with respect to the shared commitments on the dialogue gameboard. You have to take into account not only recency but also likelihood based on other knowledge that it is this particular Sam that is being referred to. For example, if you believe that your interlocutor could not possibly know of the Sam in your memory who is otherwise the most likely candidate you should not choose that Sam as a match. Choosing an appropriate match involves a great deal of world knowledge and common sense. We will ignore these matters and concentrate our attention on what needs to be done if we find a suitable match. The idea is that if you have failed to find a match in shared commitments on the gameboard but you do find a match in long term memory, then you need to load the item from long term memory into the shared commitments on your gameboard. This is what will constitute accommodation in this case.

We will introduce the notion of a *total information state* (*cf.* Larsson, 2002) which includes a record type corresponding to long term memory, represented by the 'ltm'-field in (26) and a dialogue gameboard, represented by the 'gb'-field in (26). Up until now we have thought of the gameboard as a record type. Now, however, we want to be able to make links from the gameboard to long term memory and we will achieve this by making the gameboard be a dependent type which maps records (situations) of the type representing long term memory to the record type representing the gameboard. Thus a total information state will be of the type (26).

(26) $$\left[\begin{array}{lll} \text{ltm} & : & \textit{RecType} \\ \text{gb} & : & (\text{ltm}{\rightarrow}\textit{GameBoard}) \end{array}\right]$$

Here we use *GameBoard* as the type of types which are a subtype of *InfoState* (as defined in Appendix C.1.1), that is, a gameboard is a type of information states. Formally, this is expressed as in (27).

(27) T : *GameBoard* iff $T \sqsubseteq$ *InfoState*

An example of a type corresponding to long term memory is given in (28).

(28)
$$\begin{bmatrix} \text{id}_0\text{:}\textit{Rec} \\ \text{id}_1\text{:}\begin{bmatrix}\text{x:}\textit{Ind} \\ \text{e:named(x, “Dudamel”)}\end{bmatrix} \\ \text{id}_2\text{:}\begin{bmatrix}\text{e:conductor(id}_1\text{.x)}\end{bmatrix} \\ \text{id}_3\text{:}\begin{bmatrix}\text{x:}\textit{Ind} \\ \text{e:named(x, “Beethoven”)}\end{bmatrix} \\ \text{id}_4\text{:}\begin{bmatrix}\text{e:composer(id}_3\text{.x)}\end{bmatrix} \\ \text{id}_5\text{:}\begin{bmatrix}\text{x:}\textit{Ind} \\ \text{e:named(x, “Uchida”)}\end{bmatrix} \\ \text{id}_6\text{:}\begin{bmatrix}\text{x:pianist(id}_5\text{.x)}\end{bmatrix} \\ \text{id}_7\text{:}\begin{bmatrix}\text{x:}\textit{Ind} \\ \text{e:named(x, “Sam”)}\end{bmatrix} \\ \text{id}_8\text{:}\begin{bmatrix}\text{e:singer(id}_7\text{.x)}\end{bmatrix} \end{bmatrix}$$

(28) is one way of putting the information in shared commitments represented by (20) into a type corresponding to long term memory. We are assuming that in long term memory information is indexed by unique identifiers modelled here by the labels ‘id$_n$’ (of which we assume there is a countably infinite stock, one for each natural number, n). It is important that in long term memory paths are persistent under updating, that is, the old paths do not change when we add information to long term memory. This is in contrast to the kind of updating we proposed for the gameboard, adding the label ‘prev’ to the path for the old gameboard. This meant that all labels within the old gameboard were adjusted by an update. When we link from the gameboard to long term memory we want to make sure that the link uses a persistent path which will still be correct if the long term memory should get updated. When long term memory is updated we prefix the path to the new information with the identifier ‘id$_{i+1}$’, where i is the highest index on an ‘id’-label in the long term memory type we are updating. (This is the same technique we used for ‘e’-labels in our treatment of chart parsing in Chapter 3.) The way of achieving the link is illustrated schematically in (29) where we use M to represent the long term memory (28) and leave out all irrelevant details of the gameboard.

(29)
$$\begin{bmatrix} \text{ltm=}M\text{:}\textit{RecType} \\ \text{gb=}\lambda r\text{:ltm . }\begin{bmatrix} \ldots \\ \text{shared: }\begin{bmatrix}\text{commitments=}\begin{bmatrix}\ldots \\ \text{e:leave(}r\text{.id}_7\text{.x)} \\ \ldots\end{bmatrix}\text{:}\textit{RecType}\end{bmatrix} \\ \ldots \end{bmatrix}\text{:(ltm}\rightarrow\textit{RecType}\text{)} \end{bmatrix}$$

The intuition expressed in (29) is as follows: given a situation, r, of the type represented by our long term memory, that is one in which a particular individual is labelled by 'id_7.x', the gameboard will be a type of information state where the shared commitments require that $r.\text{id}_7.\text{x}$ leaves. Two agents are aligned in their shared commitments to the extent that we can find an equivalence between the two types which represent their respective view of the shared commitments obtained by applying their respective functions labelled 'gb' to a situation of their respective memory types.

The link represented by the dependence on the long term memory type corresponds to what Kamp (1990); Kamp *et al.* (2011) call an internal anchor. We are representing here how individual roles in an agent's view of shared commitments can be anchored in that agent's long term memory. In a more complete treatment we could in addition make the gameboard depend on a type for the current visual scene and also types for other sensory input. Our use of dependent types and Kamp *et al.*'s use of internal anchors allow us to link different components of cognitive structure. Cognitive structure can also be linked to objects in the external world, giving rise to what Kamp *et al.* call external anchors. Our manifest fields can be used to correspond to their external anchors. Suppose, for example, that we have an individual 'sam' who is named Sam. We can use a manifest field to restrict the long term memory type (28) so that any record ("situation") of that type has 'sam' in the 'id_7.x'-field. This is represented in (30) where for convenience we have omitted all but the 'id_7'-field in (28).

(30) $$\left[\begin{array}{l} \ldots \\ \text{id}_7{:}\left[\begin{array}{l}\text{x=sam:}\textit{Ind}\\ \text{e:named(x, ``Sam'')}\end{array}\right] \\ \ldots \end{array}\right]$$

If M in (29) is the type (30) then for any $r : M$, it will be the case that $r.\text{id}_7.\text{x}$ will be 'sam'. Thus the shared commitment is that 'sam' leaves. Given that manifest fields can occur in any record type, this kind of external anchoring is not restricted to long term memory but could also be directly in the gameboard if that is desired.

Let us now consider how the update of a gameboard dependent on long term memory can be carried out when there is a match between the parametric content used for updating and an item in long term memory. Suppose that the current total information state, ι_{curr}, is of the type in (31)

(31) $$\left[\begin{array}{lll} \text{ltm} & : & \textit{RecType} \\ \text{gb=}\lambda r\text{:ltm . } T_{\text{gb}}(r) & : & (\text{ltm}\rightarrow\textit{RecType}) \end{array}\right]$$

and that we wish to update this with the function, a parametric content, f, given in (32) (where $T_{\text{bg}} \sqsubseteq \left[\text{x:}\textit{Ind}\right]$).

(32) $\lambda r{:}T_{\text{bg}}\ .\ T_{\text{upd}}(r)$

In order to find a match between T_{bg} and ι_{curr}.ltm (that is, to ascertain that the presupposition associated with the parametric content is met by the long term memory of the current total information state) we need to find a relabelling, η, of the flattened version of T_{bg}, $\varphi(T_{\text{bg}})$, such that (33) holds.

(33) $\varphi(\iota_{\text{curr}}.\text{ltm}) \sqsubseteq [\varphi(T_{\text{bg}})]_\eta$

Then we can derive (34) as a type of the updated total information state.

(34)
$$\begin{bmatrix} \text{ltm:}\textit{RecType} \\ \text{gb=}\lambda r\text{:ltm . }(T_{\text{gb}}(r)\boxed{\dot\wedge} \\ \quad \begin{bmatrix}\text{shared:}\begin{bmatrix}\text{commitments=}\begin{bmatrix}\text{prev:}\iota_{\text{curr}}\text{.gb.shared.commitments} \\ \text{bg:}T_{\text{bg}}\dot\wedge\left[\text{x=}\iota_{\text{curr}}\text{.ltm.}\eta(\text{x})\text{:}\textit{Ind}\right] \\ \text{e:}f(\text{bg})\end{bmatrix}\text{:}\textit{RecType}\end{bmatrix}\end{bmatrix}) \\ \qquad\qquad\qquad\qquad\text{:(ltm}\rightarrow\textit{RecType}\text{)}\end{bmatrix}$$

We can now put all this together as the update function in (35), which we call **AccLTM**(η) ("accommodate match with long term memory").

(35) **AccLTM**(η) =
$$\lambda r{:}\begin{bmatrix}\text{ltm} & : & \textit{RecType} \\ \text{gb} & : & (\text{ltm}\rightarrow\textit{GameBoard})\end{bmatrix}$$
$$\lambda f{:} \bigvee_{T\sqsubseteq[\text{x:}\textit{Ind}]} (T \rightarrow \textit{RecType})\ .$$
$$\begin{bmatrix}\text{ltm:}\textit{RecType} \\ \text{gb=}\lambda r_1\text{:ltm . }((r\text{.gb})(r_1)\boxed{\dot\wedge} \\ \quad\begin{bmatrix}\text{shared:}\begin{bmatrix}\text{commitments=}\left[\text{prev:}(r\text{.gb})(\Uparrow^3\text{ltm})\text{.shared.commitments}\right]\dot\wedge\mathcal{B}(f)\dot\wedge \\ \qquad\left[\text{bg:}\left[\text{x=}r\text{.ltm.}\eta(\text{x})\text{:}\textit{Ind}\right]\right]\text{:}\textit{RecType}\end{bmatrix}\end{bmatrix}) \\ \qquad\qquad\qquad\qquad\text{:(ltm}\rightarrow\textit{GameBoard}\text{)}\end{bmatrix}$$

Here *GameBoard* is as defined in (27).

We have used accommodation from long term memory to represent the kind of accommodation where the agent has a resource which provides a match. In a more complete treatment we could use this technique for accommodation from other available resources such as the visual scene. We now turn our attention to accommodation where there is no appropriate match with other resources. This corresponds to the case where the hearer does not know any appropriate person named Sam but merely adds that there is a person named Sam to the shared dialogue commitments.

The first step in this update is to create a type from the parametric content under consideration so that we can merge it with $\left[\text{prev:}T\right]$, where T is the type representing the current shared commitments. The type we create is one where the domain type of the function which is the parametric content is embedded under the label 'bg' (for "background") and the result is merged with the type which the function returns, appropriately relabelled. In (36a) we repeat the parametric content given in (24) and in (36b) we give the corresponding backgrounding type.

(36)

a. $$\lambda r_1{:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}\ .\ \begin{bmatrix}\text{e:leave(}r_1\text{.f.x)}\end{bmatrix}$$

b. $$\begin{bmatrix}\text{bg:}\begin{bmatrix}\text{f:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Sam")}\end{bmatrix}\\ \text{a:}\textit{Rec}\end{bmatrix}\\ \text{e:leave(bg.f.x)}\end{bmatrix}$$

An explicit definition of how to obtain the backgrounding type, $\mathcal{B}(f)$, from a function, f, is given in Appendix A.3.2, p. 172. Suppose now that the current shared commitments are given by the type in (37).

(37) $$\begin{bmatrix}\text{prev:}\begin{bmatrix}\text{prev:}\begin{bmatrix}\text{prev:}\textit{Rec}\\ \text{bg:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Dudamel")}\end{bmatrix}\\ \text{e:conductor(bg.x)}\end{bmatrix}\\ \text{bg:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Beethoven")}\end{bmatrix}\\ \text{e:composer(bg.x)}\end{bmatrix}\\ \text{bg:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{e:named(x, "Uchida")}\end{bmatrix}\\ \text{e:pianist(bg.x)}\end{bmatrix}$$

Then the new shared commitments will be (38a) which is (38b).

(38)

a.
$$\left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \multicolumn{2}{l}{\text{prev:}\mathit{Rec}} \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Dudamel”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:conductor(bg.x)}}\end{array}\right] \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Beethoven”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:composer(bg.x)}} \end{array}\right] \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Uchida”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:pianist(bg.x)}} \end{array}\right] \end{array}\right]$$
$$\underset{.}{\wedge} \left[\begin{array}{ll} \text{bg:} & \left[\begin{array}{ll} \text{f:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{a:}\mathit{Rec}} \end{array}\right] \\ \multicolumn{2}{l}{\text{e:leave(bg.f.x)}} \end{array}\right]$$

b.
$$\left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \text{prev:} & \left[\begin{array}{ll} \multicolumn{2}{l}{\text{prev:}\mathit{Rec}} \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Dudamel”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:conductor(bg.x)}}\end{array}\right] \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Beethoven”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:composer(bg.x)}} \end{array}\right] \\ \text{bg:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Uchida”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{e:pianist(bg.x)}} \end{array}\right] \\ \text{bg:} & \left[\begin{array}{ll} \text{f:} & \left[\begin{array}{l}\text{x:}\mathit{Ind}\\ \text{e:named(x, “Sam”)}\end{array}\right] \\ \multicolumn{2}{l}{\text{a:}\mathit{Rec}} \end{array}\right] \\ \multicolumn{2}{l}{\text{e:leave(bg.f.x)}} \end{array}\right]$$

We can now put this together as the update function in (39), which we call **AccNM** (“accommodate no match”).

(39) **AccNM** =

$$\lambda r{:}\begin{bmatrix} \text{ltm} & : & \textit{RecType} \\ \text{gb} & : & (\text{ltm}\rightarrow\textit{GameBoard}) \end{bmatrix}$$

$$\lambda f{:} \bigvee_{T\sqsubseteq[\text{x}:\mathit{Ind}]} (T \rightarrow \mathit{RecType})\ .$$

$$\begin{bmatrix} \text{ltm}=r.\text{ltm}{:}\textit{RecType} \\ \text{gb}=\lambda r_1{:}\text{ltm}\ .\ ((r.\text{gb})(r_1)\boxed{\dot\wedge} \\ \quad \begin{bmatrix}\text{shared:}\begin{bmatrix}\text{commitments}=\begin{bmatrix}\text{prev:}(r.\text{gb})(\Uparrow^3\text{ltm}).\text{shared.commitments}\end{bmatrix}\dot\wedge\mathcal{B}(f){:}\textit{RecType}\end{bmatrix}\end{bmatrix}) \\ \qquad{:}(\text{ltm}\rightarrow\textit{GameBoard}) \end{bmatrix}$$

We can now accommodate the first case we discussed where there was a match on the gameboard into the general format of update functions for total information states. Assuming that we have relabellings η and η' as defined on p. 130 we can formulate the update function in (40), which we call **AccGB**(η') ("accommodate match on gameboard").

(40) **AccGB**(η') =

$$\lambda r{:}\begin{bmatrix} \text{ltm} & : & \textit{RecType} \\ \text{gb} & : & (\text{ltm}\rightarrow\textit{GameBoard}) \end{bmatrix}$$

$$\lambda f{:} \bigvee_{T\sqsubseteq[\text{x}:\mathit{Ind}]} (T \rightarrow \mathit{RecType})\ .$$

$$\begin{bmatrix} \text{ltm}=r.\text{ltm}{:}\textit{RecType} \\ \text{gb}=\lambda r_1{:}\text{ltm}\ .\ ((r.\text{gb})(r_1)\boxed{\dot\wedge} \\ \quad \begin{bmatrix}\text{shared:}\begin{bmatrix}\text{commitments}=\varphi^{-}(\varphi(\begin{bmatrix}\text{prev:}(r.\text{gb})(\Uparrow^3\text{ltm}).\text{shared.commitments}\end{bmatrix})\dot\wedge[\varphi(\mathcal{F}(f))]_{\eta'}) \\ {:}\textit{RecType}\end{bmatrix}\end{bmatrix}) \\ \qquad{:}(\text{ltm}\rightarrow\textit{GameBoard}) \end{bmatrix}$$

The three update functions for accommodation that we have defined are governed by the single licensing condition given in (41).

(41) If A is an agent, s_i is A's current information state, f is a parametric content of type $(T_1 \rightarrow T_2)$ and $s_i :_A T_i$ for some T_i such that

$$T_i \sqsubseteq \begin{bmatrix} \text{ltm:}\textit{RecType} \\ \text{gb:}\begin{bmatrix}\text{shared:}\begin{bmatrix}\text{commitments:}\textit{RecType} \\ \text{latest-move:}\begin{bmatrix}\text{cont=}f\text{:}(T_1 \rightarrow T_2)\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

then

if there is some η which is a labelling for T_1 such that

s_i.gb.shared.commitments $\sqsubseteq [T_1]_\eta$

and there is some η' which is a labelling of $\mathcal{F}(f)$ such that η' is an extension of η,
then $s_{i+1} :_A T_i \boxed{\dot{\wedge}}$ **AccGB**$(\eta')(s_i)(f)$ is licensed

else if there is some η which is a labelling for T_1, s_i.ltm $\sqsubseteq [T_1]_\eta$
then $s_{i+1} :_A T_i \boxed{\dot{\wedge}}$ **AccLTM**$(\eta)(s_i)(f)$ is licensed

else $s_{i+1} :_A T_i \boxed{\dot{\wedge}}$ **AccNM**$(s_i)(f)$ is licensed

This account of accommodation for proper names where a new item is allowed to be created in memory when attempts at matching have failed is similar to a proposal by de Groote and Lebedeva (2010) to treat accommodation as error handling when a match has failed to be found. Our information states can be thought of as corresponding to their environment which they consider to be not simply a list of individuals but individuals with their properties, thus providing objects similar to those like the record types which can be found in our information states. One difference between the two proposals, apart from the obvious fact that our aim here has been to embed the theory in a more general theory of dialogue, is that de Groote and Lebedeva use a selection function to select the matches thus apparently assuming an algorithm which yields a unique result. We, on the other hand, talk in terms of matches being licensed and thereby allow for the possibility of non-deterministic selection. What we have in common, though, is that in order to account for the way accommodation is carried out we both add an additional layer to a type theory based semantics and talk in procedural terms of actions to be carried out: we with our licensing conditions for type acts and de Groote and Lebedeva with their error handling mechanism.

4.5 Paderewski

Kripke (1979) discusses the case of Peter who hears about a pianist called Paderewski. Later, in a different context, he learns of a Polish national leader and Prime Minister called Paderewski. In reality there was a single (remarkable) man called Paderewski who was both a famous concert pianist and a distinguished statesman. But Peter does not realize this and thinks that he has learned about two distinct people, both named Paderewski. Thus, in our terms, Peter's long term memory might be a subtype of (42) for some natural numbers i, j, k and l.

(42)
$$\left[\begin{array}{l} \mathrm{id}_i : \left[\begin{array}{l} \mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_j : \left[\mathrm{e}:\mathrm{pianist}(\mathrm{id}_i.\mathrm{x})\right] \\ \mathrm{id}_k : \left[\begin{array}{l} \mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_l : \left[\mathrm{e}:\mathrm{statesman}(\mathrm{id}_k.\mathrm{x})\right] \end{array}\right]$$

(42) technically allows for the two Paderewskis to be the same individual but if there is nothing in Peter's long term memory that requires them to be the same individual we will count that as corresponding to his view of them as distinct. If Peter were in this state and asked whether the pianist Paderewski and the statesman Paderewski were the same person Peter might reply, “Well, I wouldn't have thought so, but I suppose they could be the same person. I don't know.” On being told that the two Paderewskis are in fact the same person he might update his long term memory by carrying out the merge in (43a), that is, his long term memory would now be (43b).

(43)

a. $$\left[\begin{array}{l} \mathrm{id}_i : \left[\begin{array}{l} \mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_j : \left[\mathrm{e}:\mathrm{pianist}(\mathrm{id}_i.\mathrm{x})\right] \\ \mathrm{id}_k : \left[\begin{array}{l} \mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_l : \left[\mathrm{e}:\mathrm{statesman}(\mathrm{id}_k.\mathrm{x})\right] \end{array}\right] \underset{.}{\wedge} \left[\begin{array}{l} \mathrm{id}_i : \left[\mathrm{x}:\mathit{Ind}\right] \\ \mathrm{id}_k : \left[\mathrm{x}{=}\mathrm{id}_i.\mathrm{x}:\mathit{Ind}\right] \end{array}\right]$$

b. $$\left[\begin{array}{l} \mathrm{id}_i : \left[\begin{array}{l} \mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_j : \left[\mathrm{e}:\mathrm{pianist}(\mathrm{id}_i.\mathrm{x})\right] \\ \mathrm{id}_k : \left[\begin{array}{l} \mathrm{x}{=}\mathrm{id}_i.\mathrm{x}:\mathit{Ind} \\ \mathrm{e}:\mathrm{named}(\mathrm{x}, \text{“Paderewski”}) \end{array}\right] \\ \mathrm{id}_l : \left[\mathrm{e}:\mathrm{statesman}(\mathrm{id}_k.\mathrm{x})\right] \end{array}\right]$$

Eventually, his long term memory may be restructured to the type in (44) which is set equivalent to that in (43), though not multiset equivalent to it since in any record of this type the individual named Paderewski will only occur once, not twice as in (43).

(44) $$\begin{bmatrix} \mathrm{id}_i : \begin{bmatrix} \mathrm{x}{:}\mathit{Ind} \\ \mathrm{e{:}named(x,\ \text{“Paderewski”})} \end{bmatrix} \\ \mathrm{id}_j : \begin{bmatrix} \mathrm{e{:}pianist(id}_i\mathrm{.x)} \end{bmatrix} \\ \mathrm{id}_l : \begin{bmatrix} \mathrm{e{:}statesman(id}_i\mathrm{.x)} \end{bmatrix} \end{bmatrix}$$

We might think of the two types (43b) and (44) as representing two subtly different states of mind which Peter could be in. In (43b) he has two concepts of Paderewski, one concept associated with him being a pianist and perhaps other associated properties, such as practicing hard, wearing tails when he is performing, and so on and the other concept where he is a statesman, and perhaps associated with other properties such as being a dynamic national leader, a driver of hard political bargains or whatever. In (44) he has a single concept of Paderewski including all he knows about him. The first state is perhaps a natural one to be in after just learning that the two Paderewskis are in fact the same, before you have fully assimilated the identity. It is harder to discover contradictions between the two concepts here since it will only be the manifest field linking the two concepts which will reveal the contradiction. Suppose, for example, Peter's concept of the statesman Paderewski has him always late for appointments and pressed for time whereas his concept of the pianist Paderewski has him never late for appointments and not pressed for time. There is no contradiction in the state when Peter believes there to be two Paderewskis. Checking for the inconsistency in the two concept state involves reasoning about the identity expressed by the manifest field. One could imagine a simple consistency checker that does not do this – logically inadequate, of course, but human perhaps. The single concept state could however involve a direct conflict between type and its negation which, one imagines, even the simplest of consistency checkers would find. Thus if Peter finds himself in such a state he might need to refine the properties that he was ascribing to the two Paderewskis in order to make the unified concept of the single Paderewski consistent, for example, by modifying the properties to be always late for political meetings and pressed for time in his political life but never late to a musical event and not pressed for time in concerts.

Note that the link that we have expressed between the two concepts in (43b) does not involve anything like an external anchor. An alternative offered us by the type theory to represent that the two Paderewskis are identical is (45), where we are using p to represent the individual Paderewski.

(45) $$\begin{bmatrix} \mathrm{id}_i : \begin{bmatrix} \mathrm{x}{=}p{:}\mathit{Ind} \\ \mathrm{e{:}named(x,\ \text{“Paderewski”})} \end{bmatrix} \\ \mathrm{id}_j : \begin{bmatrix} \mathrm{e{:}pianist(id}_i\mathrm{.x)} \end{bmatrix} \\ \mathrm{id}_k : \begin{bmatrix} \mathrm{x}{=}p{:}\mathit{Ind} \\ \mathrm{e{:}named(x,\ \text{“Paderewski”})} \end{bmatrix} \\ \mathrm{id}_l : \begin{bmatrix} \mathrm{e{:}statesman(id}_k\mathrm{.x)} \end{bmatrix} \end{bmatrix}$$

Here the link between Peter's two concepts goes through the world since both his Paderewski concepts are linked to the individual p. If an agent's long term memory is a subtype of (45), then Ind_p figures in the long term memory type (recall that the manifest field $\left[\text{x}{=}p{:}\mathit{Ind}\right]$ is a notation for $\left[\text{x}{:}\mathit{Ind}_p\right]$, where Ind_p is a type whose only witness is p (see Appendix A.7)). We take this to mean that the agent has a direct way of identifying Paderewski but that he has not in this case become conscious of the identity of the object involved in different perceptions of Paderewski.[3] The situation could be that Peter observes Paderewski on the concert platform in tails and then sees him later in the parliament building. His observations are connected to the same individual although without him realizing that he has observed the same Paderewski twice. Thus the situation is similar to that decribed for *Hesperus* and *Phosphorus* in Frege (1892). In Frege's case the agent was visually aware of the planet Venus on different occasions, conceived of as the Evening Star (Hesperus) and the Morning Star (Phosphorus) without being aware that the same heavenly body was being observed in the morning as in the evening. The difference between Frege's example and that represented by (45) is that in Frege's case two different proper names were associated with the different observations of the same individual whereas here the same proper name is being used for the same individual, though without awareness that the proper name is being associated with the same individual on both occasions.

Ludlow (2014) has discussed Kripke's Paderewski recently and argues that the reason that proper names can be used to refer to different individuals can be due to the fact that our lexicons are dynamic and that we use different microlanguages on different occasions. In this discussion he is building on previous work by Larson and Ludlow (1993) although in that work the emphasis is on interpreted logical forms (pairs of abstract syntactic representations and semantic values such as truth values for sentences) rather than on local microlanguages constructed for use in a particular situation as argued for on the basis of a number of different kinds of examples in Ludlow (2014). In general the idea of local microlanguages being constructed on the fly during the course of dialogues and for the purposes at hand is something for which I have a great deal of sympathy and have argued for in the past (Cooper and Ranta, 2008; Larsson and Cooper, 2009; Cooper, 2010, 2012b). And indeed Ludlow (2014) is right to argue that proper names provide support for this view of language. The argument is straightforward in the case of proper names and does not involve the kinds of subtleties of meaning variation which can lead some people to suspicion of this view in the case of other words. If somebody says to me at a party, "I'd like to introduce you to my friend Sam" and indeed I have never met Sam before, I can, as a competent speaker of English, immediately form an association between the phonological type "Sam" and the individual to whom I have been introduced. It is obviously not part of my competence as a speaker of English to know all of the individuals in the universe named Sam. Our competence lies rather in our ability to make the connection between the phonological type (a name) and an individual as the need arises. The competence involves a *dynamic* process of acquiring a linguistic coupling of a speech event type with another part of the world and not a *static* knowledge of all the available couplings. Once I have added this pairing, modelled in

[3]One could choose to interpret such types differently in cognitive terms.

our terms as a sign type, to my resources, I have in a technical sense modified my language.[4] An advantage of sign-based approaches of the kind we are proposing is that you do not have to resort to subscripts in some logical language in order to distinguish between pairings of the same phonological type with different individuals. This is a trap which Larson and Ludlow (1993) fall into when they claim that there are two (or more) names in such cases distinguished by subscripts in logical form. A disadvantage of this analysis is that no two individuals could have the same name in logical form and thus we would have to use something else to analyze sentences like (46).

(46) My wife's sister, one of my graduate students and our neighbour all have the same name: Karin

(46) describes a confusing situation which I have to contend with on a daily basis. If the logical form theory with subscripts were correct this sentence would be necessarily false and one might have expected that the natural way to describe this situation would rather have been (47).

(47) My wife's sister, one of my graduate students and our neighbour all have similar names in that they are pronounced "Karin"

(47), according to my intuitions, is not a natural way of describing the situation. This suggests to me that one would need something in addition to, or in place of, a logical form with subscripts to explain how speakers of natural languages individuate names.

One interpretation of Ludlow's proposal is that when a proper name is used to refer to different individuals, different microlanguages are used for the references to the different individuals. Thus when Elisabet says *Karin* and means her sister, she is using a slightly different language than when she says *Karin* and means our neighbour. While I am much in sympathy with the idea of different microlanguages in general it seems to me that such a proposal could not be quite right. Consider dialogues like (48), a kind of dialogue which is not infrequent in our house.

(48) Elisabet: Karin called
Robin: Karin?
Elisabet: My sister

My utterance in (48) is an example of what is called a clarification request in the dialogue literature (Ginzburg and Cooper, 2004; Ginzburg, 2012, and much other literature). According to that

[4] In my case the resource is quite likely to disappear again shortly afterwards. People vary in their ability to remember names.

literature one of the uses of a clarification request such as *Karin?* is to ask for further identification of the referent of the use of the proper name in the previous dialogue turn. It might initially seem tempting to regard such a request as being in effect a request for (partial) identification of the microlanguage Elisabet is talking. But if we take that route then we have to ask ourselves what language the clarification request itself is in. Assuming that we have three variants of microlanguages available, one where *Karin* refers to Elisabet's sister, one where it refers to our neighbour and one where it refers to my graduate student, then if the request is in any of those languages the answer to the question is selfevident and it is hard to see why I would ask it. And in particular if I was thinking of Karin, my graduate student, I might be justified in saying that Elisabet's answer was wrong. This, of course, is not at all what is going on. It seems that the clarification request is part of a microlanguage in which *Karin* can be used to refer to any of the three and I am interested in finding out which was meant here. This is the kind of option that might be offered by our sign-based approach where a single (micro)language can contain several different signs with the same phonology but with different contents. The exact treatment of this needs, of course, an account of questions and clarification questions in particular which we will not undertake here.

One can understand, however, why the idea of a single referent for a proper name in a single microlanguage might seem attractive. When Kripke (1979) introduces the puzzle about Peter and Paderewski he is careful to point out the circumstances under which Peter came to the conclusion that there were two Paderewskis. Peter first learns the name Paderewski in connection with the famous pianist. Then: "Later, in a different circle, Peter learns of someone called 'Paderewski' who was a Polish nationalist leader and prime minister." Kripke's example would not have been at all as convincing if Peter had learned about Paderewski, the pianist and Paderewski, the statesman from the same person in the same conversation. Ludlow (2014) makes a similar point in criticising Kripke's construction of the apparent contradiction that Peter believes, namely that Paderewski both is and is not a pianist. "The fallacy involves the conjunction of two sentences that have the appearance of contradicting each other... but they do not contradict because they come from different microlanguages." (p. 148). The fact of the matter is that we do tend to use proper names to refer uniquely within the same dialogue, all other things being equal. Suppose we are involved in a conversation about pianists and have been, say, comparing the relative merits of Paderewski and Ashkenazy, and at some point I say (49)

(49) Paderewski was a leading statesman in Poland

You would naturally infer that I was talking about the same Paderewski, unless I explicitly point out that I intend to refer to a different person with the same name. It is, of course, possible to refer to two different people with the same name within the same dialogue and even within the same sentence, even though it may lead to confusion. The assumption is normally, though, that within the space of a dialogue a name will refer to a unique individual unless it is explicitly stated otherwise. One way of being explicit is to say something like (50)

(50) I know another person named Paderewski

If both dialogue participants are aware of the two people with the same name it is possible to use the names together in a construction which normally requires different intended referents as in (51).[5]

(51) Churchland and Churchland think that replacement of symbol manipulation computer-like devices... with connectionist machines hold (*sic*) great promise

(Globus, 1995, p. 21)

Two people named John engaged in conversation with a third person can refer to each other with the name *John* when addressing the third person without risk of confusion as in (52)

(52) John E: I remember John as an inspiring professor when I was a student
John P: Well, I remember John as an extremely bright student
Third person: I didn't realize you'd known each other that long

When addressing a person you can always use their name as a vocative even if the message you wish to convey involves a person with the same name as in (53).

(53) A: John, I'd like to introduce you to my good friend John
B: Glad to meet you. Another John, eh?

It is conceivable that somebody would want to argue that all of these cases where the same name is used twice to refer to different people are examples of code-switching between microlanguages within the space of a dialogue or sentence. Since code-switching does take place even between different languages like English and Portuguese within single dialogues and sentences it is hard to say that such an analysis is impossible. However, given that a sign-based analysis of proper names does not require these examples to be cases of code-switching perhaps the onus is on the proponent of the code-switching analysis to motivate this more complex analysis.

Puzzles about proper names and reference such as the Paderewski puzzle and Frege's (1892) original puzzle about *Hesperus* and *Phosphorus* are standardly presented as puzzles about belief reports. Indeed the matters we have discussed in this section do give rise to puzzles in belief

[5] I am grateful to Anders Tolland and Stellan Petersson for calling my attention to the fact that the Churchlands are often referred to as "Churchland and Churchland".

reports and we will return to this later. However, we would like to claim that the discussion here shows that the basis of these puzzles does not lie in the analysis of belief reports *per se* but in the nature of communication in dialogue and the resulting organization of memory. While these phenomena seem puzzling from a Fregean or Montagovian formal language perspective, from the point of view of a dialogic approach employing a sign-based analysis they seem to be a natural consequence of the way that communication takes place and knowledge gets stored.

4.6 Summary

In this chapter we have looked at the analysis of proper names. We started by showing how Montague's analysis of proper names could be recast in TTR and we showed that there was an advantage in the sign-based approach that we have adopted in accounting for the fact that different individuals can have the same name. Montague's original analysis did not say anything about the presupposition-like nature of proper names in that they seem to require interlocutors to be able to identify appropriate referents for the use of a proper name from among a number of potential referents which might be available. We showed how this could be treated by introducing parametric contents for proper names and we showed how accommodation phenomena could be accounted for including a simple-minded analysis of salience analyzed in terms of the information states of agents. Finally, we discussed Kripke's puzzle concerning Paderewski and its possible relation to a theory of microlanguages as discussed recently by Ludlow. While in general we find the idea of microlanguages appealing we suggested that it plays a role in the analysis of proper names in a rather different way to that suggested by Ludlow.

Bibliography

Artstein, Ron, Mark Core, David DeVault, Kallirroi Georgila, Elsi Kaiser and Amanda Stent, eds. (2011) SemDial 2011 (Los Angelogue): Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue.

Austin, J. (1962) *How to Do Things with Words*, Oxford University Press, ed. by J. O. Urmson.

Austin, J. L. (1961) Truth, in J. O. Urmson and G. J. Warnock (eds.), *J. L. Austin: Philosophical Papers*, Oxford University Press, Oxford.

Barwise, Jon (1989) *The Situation in Logic*, CSLI Publications, Stanford.

Barwise, Jon and Robin Cooper (1981) Generalized quantifiers and natural language, *Linguistics and Philosophy*, Vol. 4, No. 2, pp. 159–219.

Barwise, Jon and Robin Cooper (1993) Extended Kamp Notation: a Graphical Notation for Situation Theory, in P. Aczel, D. Israel, Y. Katagiri and S. Peters (eds.), *Situation Theory and its Applications*, Vol. 3, CSLI, Stanford.

Barwise, Jon and John Perry (1983) *Situations and Attitudes*, Bradford Books, MIT Press, Cambridge, Mass.

Blackburn, Patrick, Maarten de Rijke and Ydes Venema (2001) *Modal logic* (*Cambridge Tracts in Theoretical Computer Science* 53), Cambridge University Press.

Boas, Hans C. and Ivan A. Sag, eds. (2012) *Sign-Based Construction Grammar*, CSLI Publications.

Botvinick, Matthew M. (2008) Hierarchical models of behavior and prefrontal function, *Trends in Cognitive Sciences*, Vol. 12, No. 5, pp. 201 – 208.

Botvinick, Matthew M., Yael Niv and Andrew C. Barto (2009) Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective, *Cognition*, Vol. 113, No. 3, pp. 262 – 280. Reinforcement learning and higher cognition.

Breitholtz, Ellen (2010) Clarification Requests as Enthymeme Elicitors, in *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue ,*.

Breitholtz, Ellen and Robin Cooper (2011) Enthymemes as Rhetorical Resources, in Artstein *et al.* (2011).

Breitholtz, Ellen and Jessica Villing (2008) Can Aristotelian Enthymemes Decrease the Cognitive Load of a Dialogue System User?, in *Proceedings of LonDial 2008, the 12th SEMDIAL workshop*.

Bullock, Barbara E. and Almeida Jacqueline Toribio, eds. (2009) *The Cambridge Handbook of Linguistic Code-Switching*, Cambridge University Press.

Carnap, Rudolf (1956) *Meaning and Necessity: A Study in Semantics and Modal Logic*, second edition, University of Chicago Press.

Chierchia, Gennaro and Raymond Turner (1988) Semantics and property theory, *Linguistics and Philosophy*, Vol. 11, No. 3, pp. 261–302.

Cooper, Robin (1982) Binding in wholewheat* syntax (*unenriched with inaudibilia), in P. Jacobson and G. K. Pullum (eds.), *The Nature of Syntactic Representation* (*Synthese Language Library* 15), Reidel Publishing Company.

Cooper, Robin (1991) Three lectures on situation theoretic grammar, in M. Filgueiras, L. Damas, N. Moreira and A. P. Tomás (eds.), *Natural Language Processing, EAIA 90, Proceedings*, Lecture Notes in Artificial Intelligence *476*, pp. 101–140, Springer Verlag, Berlin.

Cooper, Robin (2005) Records and Record Types in Semantic Theory, *Journal of Logic and Computation*, Vol. 15, No. 2, pp. 99–112.

Cooper, Robin (2010) Frames in formal semantics, in H. Loftsson, E. Rögnvaldsson and S. Helgadóttir (eds.), *IceTAL 2010*, Springer Verlag.

Cooper, Robin (2011) Copredication, Quantfication and Frames, in S. Pogodalla and J.-P. Prost (eds.), *Logical Aspects of Computational Linguistics: 6th International Conference, LACL 2011*, pp. 64–79, Springer.

Cooper, Robin (2012a) Intensional quantifiers, in T. Graf, D. Paperno, A. Szabolcsi and J. Tellings (eds.), *Theories of Everything: In Honor of Ed Keenan*, UCLA Working Papers in Linguistics *17*, pp. 69–71, Department of Linguistics, UCLA.

Cooper, Robin (2012b) Type Theory and Semantics in Flux, in R. Kempson, N. Asher and T. Fernando (eds.), *Handbook of the Philosophy of Science*, Vol. 14: Philosophy of Linguistics, pp. 271–323, Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Cooper, Robin (2013a) Clarification and Generalized Quantifiers, *Dialogue and Discourse*, Vol. 4, No. 1, pp. 1–25.

Cooper, Robin (2013b) Update conditions and intensionality in a type-theoretic approach to dialogue semantics, in R. Fernández and A. Isard (eds.), *Proceedings of the 17th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 15–24, University of Amsterdam.

Cooper, Robin (fthc) Type Theory and Semantics in Flux, in R. Kempson, N. Asher and T. Fernando (eds.), *Handbook of the Philosophy of Science*, Vol. 14: Philosophy of Linguistics, Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Cooper, Robin and Jonathan Ginzburg (2011a) Negation in Dialogue, in Artstein *et al.* (2011), pp. 130–139.

Cooper, Robin and Jonathan Ginzburg (2011b) Negative inquisitiveness and alternatives-based negation, in *Proceedings of the Amsterdam Colloquium, 2011*.

Cooper, Robin and Ruth Kempson, eds. (2008) *Language in Flux: Dialogue Coordination, Language Variation, Change and Evolution* (*Communication, Mind and Language* 1), College Publications, London.

Cooper, Robin and Aarne Ranta (2008) Natural Languages as Collections of Resources, in Cooper and Kempson (2008), pp. 109–120.

Cresswell, M.J. (1985) *Structured Meanings: The Semantics of Propositional Attitudes*, MIT Press.

Davidson, Donald (1967) The Logical Form of Action Sentences, in N. Rescher (ed.), *The Logic of Decision and Action*, University of Pittsburgh Press. Reprinted in Davidson (1980).

Davidson, Donald (1980) *Essays on Actions and Events*, Oxford University Press, New edition 2001.

Dowty, David (1989) On the semantic content of the notion of 'Thematic Role', in G. Chierchia, B. H. Partee and R. Turner (eds.), *Properties, Types and Meanings*, Vol. II: Semantic Issues, pp. 69–130, Kluwer, Dordrecht.

Dowty, David, Robert Wall and Stanley Peters (1981) *Introduction to Montague Semantics*, Reidel (Springer).

Fernando, Tim (2004) A finite-state approach to events in natural language semantics, *Journal of Logic and Computation*, Vol. 14, No. 1, pp. 79–92.

Fernando, Tim (2006) Situations as Strings, *Electronic Notes in Theoretical Computer Science*, Vol. 165, pp. 23–36.

Fernando, Tim (2008) Finite-state descriptions for temporal semantics, in H. Bunt and R. Muskens (eds.), *Computing Meaning, Volume 3* (*Studies in Linguistics and Philosophy* 83), pp. 347–368, Springer.

Fernando, Tim (2009) Situations in LTL as strings, *Information and Computation*, Vol. 207, No. 10, pp. 980–999.

Fernando, Tim (2011) Constructing Situations and Time, *Journal of Philosophical Logic*, Vol. 40, pp. 371–396.

Fillmore, Charles J. (1982) Frame semantics, in *Linguistics in the Morning Calm*, pp. 111–137, Hanshin Publishing Co., Seoul.

Fillmore, Charles J. (1985) Frames and the semantics of understanding, *Quaderni di Semantica*, Vol. 6, No. 2, pp. 222–254.

Fox, Chris and Shalom Lappin (2005) *Foundations of Intensional Semantics*, Blackwell Publishing.

Frege, Gottlob (1892) Über Sinn und Bedeutung, *Zeitschrift für Philosophie und philosophische Kritik*, Vol. 100, pp. 25–50. Translated in Geach and Black (1980).

Gawron, Jean Mark and Stanley Peters (1990) *Anaphora and Quantfication in Situation Semantics*, CSLI Publications.

Geach, P. and M. Black, eds. (1980) *Translations from the Philosophical Writings of Gottlob Frege*, third edition, Blackwell, Oxford.

Gibson, James J. (1986) *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates.

Gil, David (2000) Syntactic categories, cross-linguistic variation and universal grammar, in P. M. Vogel and B. Comrie (eds.), *Approaches to the typology of word classes* (*Empirical approaches to language typology* 23), Mouton de Gruyter, Berlin.

Ginzburg, Jonathan (1994) An update semantics for dialogue, in H. Bunt (ed.), *Proceedings of the 1st International Workshop on Computational Semantics*, Tilburg University.

Ginzburg, Jonathan (2010) Relevance for Dialogue, in Łupkowski and Purver (2010), pp. 121–129, Polish Society for Cognitive Science.

Ginzburg, Jonathan (2012) *The Interactive Stance: Meaning for Conversation*, Oxford University Press, Oxford.

Ginzburg, Jonathan and Robin Cooper (2004) Clarification, ellipsis, and the nature of contextual updates in dialogue, *Linguistics and Philosophy*, Vol. 27, No. 3, pp. 297–365.

Ginzburg, Jonathan and Robin Cooper (2014) Quotation via Dialogical Interaction, *Journal of Logic, Language and Information*, Vol. 23, No. 3, pp. 287–311.

Ginzburg, Jonathan and Raquel Fernández (2010) Computational Models of Dialogue, in A. Clark, C. Fox and S. Lappin (eds.), *The Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell.

Ginzburg, Jonathan and Ivan A. Sag (2000) *Interrogative Investigations: The Form, Meaning, and Use of English Interrogatives*, CSLI Lecture Notes *123*, CSLI Publications, Stanford, California.

Globus, Gordon G. (1995) *The Postmodern Brain* (*Advances in Concsiousness Research* 1), John Benjamins Publishing Company.

Groenendijk, Jeroen and Floris Roelofsen (2012) Course Notes on Inquisitive Semantics, NASSLLI 2012. Available at `https://sites.google.com/site/inquisitivesemantics/documents/NASSLLI-2012-inquisitive-semantics-lecture-notes.pdf`.

de Groote, Philippe and Ekaterina Lebedeva (2010) Presupposition Accommodation as Exception Handling, in *Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 71–74.

Heim, Irene and Angelika Kratzer (1998) *Semantics in Generative Grammar*, Blackwell Publishing.

Hughes, G.E. and M.J. Cresswell (1968) *Introduction to modal logic*, Methuen and Co., Ltd.

Jackendoff, Ray (2002) *Foundations of Language: Brain, Meaning, Grammar, Evolution*, Oxford University Press.

Jurafsky, Daniel and James H. Martin (2009) *Speech and Language Processing*, second edition, Pearson Education.

Kamp, Hans (1990) Prolegomena to a Structural Theory of Belief and other Attitudes, in C. A. Anderson and J. Owens (eds.), *Propositional Attitudes: the Role of Content in Logic, Language and Mind*, CSLI Publications, Stanford.

Kamp, Hans, Josef van Genabith and Uwe Reyle (2011) Discourse Representation Theory, in D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, Vol. 15, Springer Science+Business Media B.V. .

Kamp, Hans and Uwe Reyle (1993) *From Discourse to Logic*, Kluwer, Dordrecht.

Kant, Immanuel (1781) *Critik der reinen Vernunft (Critique of Pure Reason)*, Johann Friedrich Hartknoch, Riga, second edition 1787.

Kaplan, David (1978) On the Logic of Demonstratives, *Journal of Philosophical Logic*, Vol. 8, pp. 81–98.

Keenan, E. L. and J. Stavi (1986) Natural Language Determiners, *Linguistics and Philosophy*, Vol. 9, pp. 253–326.

Kracht, Marcus and Udo Klein (2014) The Grammar of Code Switching, *Journal of Logic, Language and Information*, Vol. 23, pp. 313–329.

Kripke, Saul (1979) A Puzzle about Belief, in A. Margalit (ed.), *Meaning and Use*, Reidel.

Larson, Richard K. and Peter Ludlow (1993) Interpreted Logical Forms, *Synthese*, Vol. 96, pp. 305–55.

Larsson, Staffan (2002) *Issue-based Dialogue Management*, PhD dissertation, University of Gothenburg.

Larsson, Staffan (2010) Accommodating innovative meaning in dialogue, in Łupkowski and Purver (2010), pp. 83–90, Polish Society for Cognitive Science.

Larsson, Staffan (2011) The TTR perceptron: Dynamic perceptual meanings and semantic coordination., in Artstein *et al.* (2011).

Larsson, Staffan and Robin Cooper (2009) Towards a formal view of corrective feedback, in A. Alishahi, T. Poibeau and A. Villavicencio (eds.), *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pp. 1–9.

Larsson, Staffan and David R. Traum (2001) Information state and dialogue management in the TRINDI dialogue move engine toolkit, *Natural Language Engineering*, Vol. 6, No. 3&4, pp. 323–340.

Lewis, David (1972) General Semantics, in D. Davidson and G. Harman (eds.), *Semantics of Natural Language*, pp. 169–218, Reidel Publishing Company.

Lewis, David (1979a) Attitudes de dicto and de se, *Philosophical Review*, Vol. 88, pp. 513–543. Reprinted in Lewis (1983).

Lewis, David (1979b) Scorekeeping in a Language Game, *Journal of Philosophical Logic*, Vol. 8, pp. 339–359.

Lewis, David (1983) *Philosophical Papers, Volume 1*, Oxford University Press.

Ludlow, Peter (2014) *Living Words: Meaning Underdetermination and the Dynamic Lexicon*, Oxford University Press.

Łupkowski, Paweł and Matthew Purver, eds. (2010) Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue. Poznań: Polish Society for Cognitive Science.

Maier, Emar (2009) Proper names and indexicals trigger rigid presuppositions, *Journal of Semantics*, Vol. 26, pp. 253–315.

Martin-Löf, Per (1984) *Intuitionistic Type Theory*, Bibliopolis, Naples.

McCarthy, J. and P. J. Hayes (1969) Some philosophical problems from the standpoint of artificial intelligence, *Machine Intelligence*, Vol. 4, pp. 463–502.

Montague, Richard (1970) Universal Grammar, *Theoria*, Vol. 36, pp. 373–398.

Montague, Richard (1973) The Proper Treatment of Quantification in Ordinary English, in J. Hintikka, J. Moravcsik and P. Suppes (eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, pp. 247–270, D. Reidel Publishing Company, Dordrecht.

Montague, Richard (1974) *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, ed. and with an introduction by Richmond H. Thomason.

Ninan, Dilip (2010) *De Se* Attitudes: Ascription and Communication, *Philosophy Compass*, Vol. 5, No. 7, pp. 551–567.

Nordström, Bengt, Kent Petersson and Jan M. Smith (1990) *Programming in Martin-Löf's Type Theory* (*International Series of Monographs on Computer Science* 7), Clarendon Press, Oxford.

Partee, B.H., A.G.B. ter Meulen and R.E. Wall (1990) *Mathematical Methods in Linguistics*, Springer.

Partee, Barbara H. (1986) Noun Phrase Interpretation and Type-Shifting Principles, in J. Groenendijk, D. de Jongh and M. Stokhof (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, Foris Publications.

Partee, Barbara H. and Vladimir Borschev (2012) Sortal, Relational, and Functional Interpretations of Nouns and Russian Container Constructions, *Journal of Semantics*, Vol. 29, No. 4, pp. 445–486.

Perry, John (1979) The Problem of the Essential Indexical, *Noûs*, Vol. 13, No. 1, pp. 3–21. Reprinted in Perry (1993).

Perry, John (1993) *The Problem of the Essential Indexical and Other Essays*, Oxford University Press.

Peters, Stanley and Dag Westerståhl (2006) *Quantifiers in Language and Logics*, Oxford University Press.

Prior, Arthur N. (1957) *Time and modality*, Oxford University Press.

Prior, Arthur N. (1967) *Past, present and future*, Oxford University Press.

Prior, Arthur N. (2003) *Papers on Time and Tense*, Oxford University Press, new edition, edited by Per Hasle, Peter Øhrstrøm, Torbren Braüner and Jack Copeland.

Purver, Matthew, Eleni Gregoromichelaki, Wilfried Meyer-Viol and Ronnie Cann (2010) Splitting the *I*s and Crossing the *You*s: Context, Speech Acts and Grammar, in Łupkowski and Purver (2010), pp. 43–50, Polish Society for Cognitive Science.

Pustejovsky, James (1995) *The Generative Lexicon*, MIT Press, Cambridge, Mass.

Pustejovsky, James (2006) Type theory and lexical decomposition, *Journal of Cognitive Science*, Vol. 6, pp. 39–76.

Ranta, Aarne (1994) *Type-Theoretical Grammar*, Clarendon Press, Oxford.

Reichenbach, Hans (1947) *Elements of Symbolic Logic*, University of California Press.

Ribas-Fernandes, José J.F., Alec Solway, Carlos Diuk, Joseph T. McGuire, Andrew G. Barto, Yael Niv and Matthew M. Botvinick (2011) A Neural Signature of Hierarchical Reinforcement Learning, *Neuron*, Vol. 71, No. 2, pp. 370 – 379.

Ruppenhofer, Josef, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson and Jan Scheffczyk (2006) FrameNet II: Extended Theory and Practice. Available from the FrameNet website.

Russell, Betrand (1903) *Principles of Mathematics*, Cambridge University Press.

Sacks, H., E.A. Schegloff and G. Jefferson (1974) A simplest systematics for the organization of turn-taking for conversation, *Language*, Vol. 50, pp. 696–735.

Sag, Ivan A., Thomas Wasow and Emily M. Bender (2003) *Syntactic Theory: A Formal Introduction*, 2nd edition, CSLI Publications, Stanford.

de Saussure, Ferdinand (1916) *Cours de linguistique générale*, Payot, Lausanne and Paris, edited by Charles Bally and Albert Séchehaye.

Schlenker, Philippe (2011) Indexicality and *De Se* Reports, in C. Maienborn, K. v. Heusinger and P. Portner (eds.), *Semantics: an international handbook of natural language meaning*, pp. 1561–1604, de Gruyter.

Searle, John R. (1969) *Speech Acts: an Essay in the Philosophy of Language*, Cambridge University Press.

Seligman, Jerry and Larry Moss (1997) Situation Theory, in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*, North Holland and MIT Press.

Shanahan, Murray (2009) The Frame Problem, in E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, winter 2009 edition, `http://plato.stanford.edu/archives/win2009/entries/frame-problem/`.

Shieber, Stuart (1986) *An Introduction to Unification-Based Approaches to Grammar*, CSLI Publications, Stanford.

Suppes, Patrick (1960) *Axiomatic Set Theory*, The University Series in Undergraduate Mathematics, D. van Nostrand Company, Inc.

Thomason, Richmond (1980) A model theory for propositional attitudes, *Linguistics and Philosophy*, Vol. 4, pp. 47–70.

Traum, David R. (1994) *A Computational Theory of Grounding in Natural Language Conversation*, PhD dissertation, University of Rochester, Department of Computer Science.

Turner, Raymond (2005) Semantics and Stratification, *Journal of Logic and Computation*, Vol. 15, No. 2, pp. 145–158.