Journal of Semantics, 31, 2014: 457–511
doi:10.1093/jos/fft008
Advance Access publication August 24, 2013

Partial Dynamic Semantics for Anaphora: Compositionality without Syntactic Coindexation

DAG TRYGVE TRUSLEW HAUG

Abstract

This article points out problems in current dynamic treatments of anaphora and provides a new account that solves these by grafting Muskens' Compositional Discourse Representation Theory onto a partial theory of types. Partiality is exploited to keep track of which discourse referents have been introduced in the text (thus avoiding the overwrite problem) and to account for cases of anaphoric failure. Another key assumption is that the set of discourse referents is well-ordered, so that we can keep track of the order in which they have been introduced, allowing a semantic characterization of anaphoric accessibility across stretches of discourse. Unlike other dynamic approaches, the system defines semantic values for unresolved anaphors. This leads to a clear separation of monotonic and non-monotonic content (in this case anaphoric resolution) and arguably provides a sound basis for a non-monotonic theory of anaphoric resolution.

1 INTRODUCTION

The realization that language meaning is sensitive to the discourse context led to the development of dynamic semantic frameworks such as Discourse Representation Theory (DRT; Kamp 1981) and File Change Semantics (FCS; Heim 1982). The main motivation was anaphoric expressions, for example pronouns such as *he, she*, but also tense and aspect in discourse (Kamp & Rohrer 1983). The idea is to have sentences denote not truth values, as they do in static semantics,[1] but 'updates' or context change potentials (CCP), that is, their ability to transform the context in which they are uttered.

This is spelled out formally in various ways in different dynamic frameworks, but the general principle is the same. Individual-type expressions introduce discourse referents (Karttunen 1976). Contexts are assignments of individuals in the model to these discourse referents.[2]

[1] Throughout the article I consider only an extensional semantics.

[2] The approach is easily generalized to cover events and times by having different types of discourse referents.

A CCP, then, naturally comes out as a relation between input and output assignments (or alternatively a function from sets of input assignments to sets of output assignments).

Dynamic semantics nicely captures some facts about anaphoric expressions. For example, expressions such as negation, quantifiers and conditionals can be given meanings that introduce embedded contexts which are not available in the further discourse. This accounts for the non-felicity of the discourse in (1), where the indexation shows intended coreference.

(1) No_1 girl walks. *She_1 is sad.

Roughly speaking, the CCP of the first sentence will be a relation between input contexts *i* and identical output contexts *o*, such that *o* cannot be extended with a discourse referent 1 which is a girl and walks. Thus the fact that *no girl* does not introduce a discourse referent in the global context falls out of the semantics of *no*.

While the treatment of anaphora is hailed as one of the big achievements of dynamic theories of semantics, quite a number of problems are left unsolved in the 'standard' frameworks FCS, DRT, compositional DRT (CDRT; Muskens 1996) and Dynamic Predicate Logic (DPL; see Groenendijk & Stokhof 1991).[3]

First, the *semantic* account sketched above for the accessibility violation in (1) only holds in DRT and arguably in FCS; other dynamic frameworks rather capture the effect through a *syntactic* familiarity constraint requiring anaphoric expressions to be coindexed according to stipulated rules. Second, dynamic frameworks suffer from the so-called 'overwrite problem': how to avoid the same discourse referent being used twice, either overwriting the old value or causing unwanted coreference (van Eijck & Visser 2012, section 6.6). This is again solved syntactically, through a novelty constraint.

The most serious problem with dynamic treatments of anaphora, however, is that none of them offers a model-theoretic interpretation of unresolved anaphora, that is words like *he, she* do not have meanings at all until they are assigned a referent. Instead, it is either assumed that they have already been assigned referents in the input to semantics; or the meanings of anaphors are characterized procedurally, as 'invitations to pick a reference from the current context' (van Eijck 2001: 349). On both approaches, it is impossible to construct a meaning for a sentence

[3] For expository purposes, I discuss only these theories in the introduction. There are other approaches that have been developed in reaction to perceived failures of the 'standard' frameworks. I discuss a number of these in Section 5.1.

with unresolved anaphors. But the sentence meaning (and even the meaning of sentences *following* the anaphor) is an important input to anaphor resolution, so we seem to have gotten things the wrong way around. While a procedural account of anaphor resolution is desirable at some stage, the procedure should start from the semantic representations, and not be a prerequisite for constructing them.

The use of syntactic and procedural solutions makes the treatment of anaphora *representational* in the sense that it is not reducible to semantic values alone (Muskens 1996: 172)—the representations themselves are essential.[4] This clashes with Montague's original programme for formal semantics, but several proponents of dynamic semantics, notably Hans Kamp (see e.g. Kamp *et al.* 2011, chapter 5), have argued that essential use of representations is necessary and desirable. However, Kamp's defence of representationalism is tied to his theory of propositional attitudes and logical omniscience. As far as I know, no one has argued that it is desirable to have a representational account of anaphora and the goal of this article is to show that a fully semantic account is both possible and preferable.

The plan of the article is as follows. In Section 2, I look closer at the problematic aspects of anaphora in dynamic semantics and show how existing frameworks fail. In Section 3, I give an introduction to DRT and its compositional version CDRT. In my view, it is an advantage to build on this tradition, which is well-known and has been used extensively in research on many languages. In Section 4, I introduce a partial theory of types which will be the logical underpinning of my theory. Readers who are not interested in the formal details can go lightly on this. Section 5 is the core of the article: I reconstruct CDRT in a partial setting and show how we can construct model-theoretic meanings for anaphors that can serve as input to a pragmatic theory of anaphor resolution. Anaphor resolution itself, however, is beyond the scope of this article. Section 6 offers conclusions and outlook. Some formal details are relegated to the appendices.

2 ASPECTS OF ANAPHORA IN DYNAMIC SEMANTICS

I will now discuss some problematic aspects of anaphora in dynamic semanatics. First, in sections 2.1–2.3, we will see how unresolved anaphora, anaphoric accessibility and the overwrite problem are dealt with in the above-mentioned 'standard' versions of dynamic semantics. As far

[4] Notice that on this definition, the question is not whether a framework itself is representational or not, but whether its treatment of a particular phenomenon is representational.

as possible, the discussion will be kept neutral with respect to particular frameworks, but on some occasions there are important differences that must be noted.

The problems discussed in sections 2.1–2.3 are all tied to the representationalism of the standard frameworks. In section 2.4, I argue that to solve them properly, we need a theory with a clear separation of monotonic and non-monotonic meaning, where all monotonic meaning can be constructed without relying on non-monotonic meaning. Later on, in section 5.1, we will see that more recent approaches to dynamic semantics than the 'standard ones' also do not achieve this, even though they solve the overwrite problem and the problem of giving a semantic characterization of accessibility.

2.1 *Unresolved anaphors*

Why would one want a model theoretic treatment of unresolved anaphors? Intuitively, I think the answer is that it is part of the *meaning* of an anaphoric expression that it should refer to some item in the preceding context. A syntactic account fails to capture this and that leads to at least two serious problems.

First, Beaver (2002) observes that if the semantics deals only with resolved anaphors, then resolution must happen in a pre-semantic component of the grammar. But it is unclear which component is resonsible for this. Syntax is unlikely to do the job, since anaphoric resolution works across sentences. Lexical ambiguity seems far-fetched, since there appear to be no languages that actually distinguish such words as he_1 and he_{12}. Indeed, anaphoric resolution is commonly thought of as a pragmatic process, but if anaphors are already indexed in the input to semantics, this would entail pragmatics operating *before* semantics.

Second, a fully pre-semantic account of anaphoric resolution is unrealistic because it is clear that the semantic content of a sentence contributes to the resolution of any anaphor in it. Consider (2)

(2) It mooed.

It is likely to be resolved to an animal that makes the appropriate sound, because whatever *it* denotes, its denotation must be in the denotation of the predicate *mooed*. If we want to model this in a formal characterization of the semantics–pragmatics interface, we need to have a semantic representation of the sentence *with unresolved anaphors*, which can serve as input to such reasoning.[5]

[5] A reviewer suggests an alternative picture in which we first get a syntactic tree with unresolved anaphora which is then expanded to a set of syntactic trees with all possible anaphoric resolutions.

DRT offers a representation of unresolved anaphors in the form of preliminary discourse representation structures. However, these representations are uninterpreted, which is unsatisfactory and means that they cannot serve as a basis for reasoning about anaphoric resolution.

FCS, DPL and CDRT, on the other hand, assume that the input to semantics bears referential indices spelling out any coreference relationships. Obviously, such theories cannot offer a model theoretic treatment of unresolved anaphors: it is not possible to represent unresolved anaphors in the first place, let alone provide an interpretation of the representation.

Muskens (2011) offers a solution to this problem within CDRT. In his approach, anaphoric expressions are modelled as free variables over discourse referents. So (1) will look as in (3).

(3) No_1 girl walks. $\text{She}_{?2}$ is sad.

?2 stands for a variable index, and in the next step, a constant is substituted for this variable. However, nothing in the interpretation of this altered CDRT language prevents the substitution of ?2 by 1, leading to the same situation as in (1). In fact, nothing in the semantics prevents us from substituting the unused discourse referent 3 for ?2, which will let the anaphoric expression introduce a new discourse referent. So although the approach of Muskens (2011) gives us a model-theoretic denotation for unresolved anaphors, it is not the correct one. Anaphoric expressions are treated as variables over discourse referents, but they are semantically *unconstrained* variables. Constraints on anaphora must be stated in purely syntactic terms, which brings us to the next section.

2.2 *Constraints on anaphora*

A syntactic account of accessibility is non-explanatory in the sense that it just stipulates constraints on coindexation. On a *semantic* account of accessibility, these constraints would ideally follow from the semantics

Each of these are then processed to give a semantic representation, with some filtering happening in this component (e.g. resolutions of *it* in *it mooed* to non-animate entities are ruled out as semantically non-felicitous). Finally, we get pragmatic reasononing over the output semantic representations. Observe that on this view, if constraints on anaphora are semantic, the presemantic generation of all anaphoric resolution would have to include inadmissible indexations such as in (1), which would then be filtered out by the semantics. The approach seems computationally unattractive since it requires the computation of several syntactic representations that are identical up to the representation of anaphors, with consequent computation of semantic representations that only differ in their assignments of values to anaphors.

of words such as negation, quantifiers and conditionals that put constraints on accessibility.[6]

DRT has such a semantic account of constraints on anaphora, as can be seen when we consider discourses with resolved anaphors. Consider the indexed discourse in (4).[7]

(4) No_1 girl walks. *If she_1 talks, she_1 talks.

The denotation of the first sentence will be a relation between input assignments *i* and identical output assignments *o*, such that *o* cannot be extended to an assignment *k* that assigns an individual to discourse referent 1 such that that individual is in the denotation of *walk*. Observe that the semantic representation of the negated sentence is just a test on output assignments and no discourse referent 1 is added to the global context. DRT uses *partial* assignments to model contexts, so the global context will not map 1 in the second sentence to any individual and the sentence is semantically infelicitous. In FCS, (4) will be ruled out syntactically by the Novelty-Familiarity condition because the second sentence has an anaphoric discourse referent which is not coindexed with a discourse referent in the context. But on a partial setup of FCS, which is what Heim (1982) intends, (4) will also be semantically malformed for precisely the same reasons as in DRT.[8]

CDRT loses this model-theoretic characterization of accessibility. The reason is its classical (i.e. non-partial) setup. There will be more formal detail in section 3.2, so suffice it to say that she_1 is guaranteed to have an interpretation and therefore the tautology in the second sentence will be true. Muskens (1996: 172) solves this through a syntactic constraint on acceptable indexations, not unlike Heim, so there is no model-theoretic treatment of these constraints. Muskens also shows that the same problem arises in DPL and must be given a syntactic solution there too.

2.3 *The overwrite problem*

Using constants (as in CDRT) or free variables (as in DRT, FCS and DPL) for discourse referents provides an account of coreference, but it

[6] There is an interesting parallel here to the discussion of whether the presupposition projection behavior of the logical connectives in dynamic semantics can be made to follow from their semantics or not, see Rothschild (2011) for discussion and references.

[7] This slightly artifical-sounding example is from Muskens (1996: 171, ex. (46)). As a reviewer points out, a more natural example could perhaps arise from the discourse *No two-year old* boy_1 *is an angel. Boys will always be boys.* if we try to replace the tautology with something expressing a similar thought such as *If* he_1*'s a boy,* he_1*'s a boy.*

[8] However, Heim does not give an interpretation for a partial type theory, so her partial framework is arguably not compositional.

can also entail unwanted coreference. This is the so-called 'overwrite problem'. Consider the discourse in (5).

(5) A_1 boy arrived. A_1 girl was sleeping.

If we merge the representations of these two sentences, we will get an unwanted coreference: the discourse referent 1 will refer both to the boy who arrived and to the girl who was sleeping. The question, then, is how to avoid using the discourse referent 1 when we construct the representation of the second sentence. In FCS, this is captured by the Novelty/Familiarity Condition, and there is a similar condition in standard construction algorithms for DRT. Note that a syntactic account is not a priori implausible here. We only need to stipulate that the syntax can tell the two tokens of the same item (in casu the indefinite article) apart and this is in effect what DRT, FCS and DPL does. Still, the fact that the indefinite article introduces a new referent is intuitively part of its meaning, not a syntactic feature.

More damagingly, there are cases where a natural semantic analysis involves reuse of a single syntactic token, as in the distributive reading of (6) from Muskens (1996: 181).

(6) *Bill*$_1$ *and Sue*$_2$ *own a*$_3$ *donkey.*

We somehow want to use *a donkey* twice, but syntax provides only a single index. It is unclear how this can be fixed in a purely syntactic analysis. CDRT, on the other hand, will predict the correct truth conditions for (6), but it will make the wrong predictions about anaphoric accessibility, since the donkey owned by Bill will be 'overwritten' by the donkey owned by Sue and only the latter will be available for anaphoric uptake.

2.4 *Monotonic and non-monotonic content*

Developing a theory of actual anaphora resolution is beyond the scope of this article. But it is important that the framework can be extended with a theory of anaphora resolution. As I will argue here, that requires a proper separation of monotonic and non-monotonic content. Consider the mini-discourse in (7).

(7) a. Pedro_1 is in a_2 bar.
b. Every_3 woman who ever dated a_4 man despises him_5.
c. He_6 is a well-known date crasher.

Let us assume that after (7-b), the preferred interpretation takes *him*$_5$ as coreferent with *a*$_4$, perhaps according to a preference for the closest antecedent. But (7-c) attaches most coherently (in the sense of Segmented Discourse Representation Theory, Asher & Lascarides 2003) to the previous discourse as an *Explanation* of (7-b). This requires he_6 to be coreferential with him_5. But him_5 is not accessible to he_6 if it is coreferent with a_4;[9] in this discourse, him_5 can only be accessible to he_6 if it is coreferent with $Pedro_1$. So we need to change the interpretation of an anaphor in (7-b) based on the contents of the entire discourse (7-a)–(7-c).

The best way to achieve this is to have a clean separation of the monotonic part of the interpretation (7-a)–(7-c), including any accessibility constraints on anaphora, and the non-monotonic part of the interpretation, including anaphoric resolution. We can then model resolution as non-monotonic reasoning over premises that include the monotonic content of the discourse. Besides traditional conditions on individuals (the entities discourse referents refer to), the monotonic content must also deliver input such as the order of discourse referents and perhaps even their topicality and so on, that is conditions on discourse referents themselves, and not just their referents. In other words, discourse referents are not simply superfluous but convenient machinery.

These preliminaries to a theory of anaphora resolution are in fact problematic for some of the 'non-standard' frameworks that have dealt with anaphora in non-representational ways. As an example, let us now look at how Transition Preference Pragmatics (Beaver 2002) will deal with the discourse in (7).

Transition Preference Pragmatics models anaphoric resolution as a preference order over state transitions. Let us say that we are in state s_a after processing (7-a). The meaning of (7-b) is ambiguous between a transition $\langle s_a, s_{b_1}\rangle$ (where him_5 is resolved to a_4 *man*) and $\langle s_a, s_{b_2}\rangle$ (where him_5 is resolved to $Pedro_1$). Let us assume that resolution of him_5 to *a* man_4 is preferred, so the preference order puts us in state s_{b_1}. Now we come to (7-c). This attaches more coherently to the discourse if we assume that he_6 from (7-c) corefers with him_5 from (7-b), which means that him_5 must corefer with $Pedro_1$. But there is no transition $\langle s_{b_1}, s_c\rangle$ such that in s_c him_5 refers to $Pedro_5$, since the underlying Resolution Predicate Logic does not allow destructive updates. Presumably we need to 'downdate' our interpretation, but we cannot simply go back to s_a and reprocess (7-b). Instead, we need to go back to s_a *plus the monotonic content* of (7-b)-(7-c), namely that some discourse referent occurring in (7-a) or (7-b) (or both) has the property of being a well-known date

[9] Unless, that is, the indefinite a_4 *man* takes wide scope. We ignore this reading here.

crasher, but there is no obvious way to do this if monotonic and non-monotonic contents are mixed, as they are in Beaver's states, where unresolved anaphors are treated as ambiguous rather than underspecified.

The issue arises because Transition Preference Pragmatics (like other dynamic frameworks) cannot represent discourse with the proper, monotonic constraints on resolution of anaphors but without the actual resolution. What we need is the ability to merge representations of long streches of discourses *without* resolving anaphors. In fact, anaphors should *never* be resolved in the representation of monotonic semantics, as anaphor resolution is a fundamentally non-monotonic process. This requires a notion of anaphoric accessibility (which is monotonic) *within* our representations of the discourse.

2.5 *Summary*

Table 1 sums up the extent to which the various standard frameworks offer model-theoretic treatments of the problems in sections 2.1–2.3. To do better than these theories, then, we need a model-theoretic interpretation of unresolved (and resolved) anaphors and give a model-theoretic characterization of accessibility. The theory should also be compositional[10] and based on lambda calculus.[11]

My proposal builds on the DRT/CDRT tradition. The standard frameworks—in particular DRT—are more developed and empirically well-tested in studies on numerous languages than the alternatives, so I believe there is an important value in building on a DRT-like language. CDRT gives us that, and moreover it does not go beyond ordinary type logic. As we will see, Muskens shows that if we adopt certain first-order axioms, we can view DRSs as abbreviations for terms in ordinary type logic. This has several formal advantages: everything we know about ordinary type logic carries over to CDRT, we do not need special mechanisms for binding variables etc. There are also some practical advantages that are not negligible: although some of the object-language entities are unfamiliar, they are kept behind the scenes, as it were, and the framework gives the impression of being just DRT with

[10] According to the classification in Geurts & Beaver (2011), DRT is not compositional, FCS is intended to be compositional and the frameworks they lump as 'dynamic semantics' (see below) are fully compositional. Krahmer (1998) is based on the DRT construction algorithm rather than being compositional, but otherwise it is in some respects similar to the framework presented here.

[11] The latter point is essential for theories of the syntax-semantics interface that rely on the Curry-Howard isomorphism, as in for example Categorial Grammar (van Benthem 1986), Dynamic Syntax (Cann *et al.* 2005) or Glue Semantics (Dalrymple 1999).

Theory	Semantics for unresolved anaphors	Semantic account of constraints on anaphora	Semantic solution to overwrite problem
DRT	No	Yes	No
FCS	No	Yes	No
CDRT	No*	No	No
DPL	No	No	No

*As we saw in section 2.1, Muskens (2011) offers an (ultimately unsatisfactory) solution to this with CDRT.

Table 1 Model-theoretic aspects of anaphora in dynamic theories

lambdas. Everything the working linguist knows about DRT carries over to CDRT—everything, that is, except the treatment of anaphora.

The problems in Table 1 all arise from the fact that dynamic theories use constants or free variables as the semantic representations of discourse referents. As pointed out by Jacobson (1999: 127), following Barbara Partee, *John loves* x_1 and *John loves* x_2 never seem to function like different semantic objects. But if x_1 and x_2 are constants or free variables, they *will* be different semantic objects.

We will adopt a two-fold strategy to solve this underlying problem. First, we will graft Muskens' CDRT onto a partial theory of types[12] (see e.g. Lepage 1992, Lapierre 1992 and Muskens 1995, though my particular approach will differ from all of these). This will make sure that used and unused discourse referents have different interpretations. Second, we will assume a total precedence order on the set of discourse referents. This makes sure we can define a function that at any point picks out the *next unused register*. In this way, we will have a variable-free dynamic semantics (as argued for in Jacobson 1999, who develops a static semantics without variables). But before we do this, we will present more details concerning (C)DRT.

3 DYNAMIC THEORIES OF SEMANTICS

3.1 *DRT*

In this section we present the basics of DRT, in the main following the exposition of Two Stage Bottom-Up DRS Construction in Kamp, van

[12] Muskens (1996: 148n4, 172n13) already noted that using partial assignments would give a more realistic theory, which would be easier to generalize to other phenomena.

Genabith, and Reyle (2011: 139–44). In this modern version of DRT, each sentence is first assigned a preliminary representation, which is a pair consisting of the set of presuppositions that the sentence gives rise to, and its assertoric content. Both the presuppositions and the assertoric part are represented as Discourse Representation Structures (DRSs, also known as ‘boxes’).

There are various proposals on how to construct these preliminary representations; see Kamp *et al.* (2011: 315–26) for discussion. All of these use a unification-based algorithm rather than relying on functional application as the means of combining meanings. For our purpose here we need not focus on how the sentence representations are built, but only on the end result.

Consider the mini discourse in (8) from Kamp *et al.* (2011: 138). The first sentence receives the representation in (9).

(8) A delegate arrived. She registered.

(9)
$$\left\langle \{\,\},\ \begin{array}{|l|}\hline x \\ \hline \\ delegate(x) \\ arrive(x) \\ \\ \hline \end{array} \right\rangle$$

(8) carries no presuppositions, so the first member of the paired representation is the empty set. The second member is the DRS that represents the assertoric part of the sentence. The DRS has two parts: a (possibly empty) universe, which declares the discourse referents introduced by that DRS; and a (possibly empty) set of conditions. DRSs are interpreted on models similar to those of first-order logic: $\mathcal{M} = \langle U, \mathfrak{I} \rangle$, where U is a non-empty domain and $\mathfrak{I}$ is a function mapping names to U and n-ary predicates to sets of n-tuples in U^n. Assignments, or in DRT terminology, embeddings, on such models are partial functions mapping discourse referents to U. The notation $i \subset_{\{x\}} o$ means that o extends i with x, that is i and o are (partial) functions with the same value for all arguments in the domain of i, and o is also defined for x.

The interpretation of the assertoric DRS in (9) is as in (10).

(10) $\{\langle i, o\rangle | i \subset_{\{x\}} o \wedge o(x) \in \mathfrak{I}(delegate) \wedge o(x) \in \mathfrak{I}(arrive)\}$

In other words, i is an input assignment and o is an output assignment and o extends i by assigning to the discourse referent x some individual $o(x)$ who is a delegate and arrives.

The second sentence gets the representation in (11).

(11)
$$\underbrace{\begin{array}{|l|}\hline x \\ \hline \mathit{delegate}(x) \\ \mathit{arrive}(x) \\ \hline \end{array}}_{\text{context}} \qquad \underbrace{\left\langle \left\{ \begin{array}{|l|}\hline y \\ \hline \mathit{female}(y) \\ \hline \end{array} \right\}, \begin{array}{|l|}\hline y \\ \hline \mathit{registered}(y) \\ \hline \end{array} \right\rangle}_{\text{representation}}$$

The first member of the pair is a set of presuppositional DRSs. The tuple does *not* get an interpretation: instead, the presuppositional DRSs must be resolved. In this case, there is only one, and resolution happens by accommodating the condition *female*(x) in the context, based on the knowledge that delegates are either male or female, and resolving the anaphor by adding the condition $x = y$ in the non-presuppositional DRS. The result is as in (12).

(12)
$$\underbrace{\begin{array}{|l|}\hline x \\ \hline \mathit{delegate}(x) \\ \mathit{arrive}(x) \\ \mathit{female}(x) \\ \hline \end{array}}_{\text{context}} \qquad \underbrace{\begin{array}{|l|}\hline y \\ \hline \mathit{registered}(y) \\ x = y \\ \hline \end{array}}_{\text{representation}}$$

The representation in (12) is interpreted as the set of assignment pairs $\{\langle i, o\rangle | i \subset_{\{y\}} o \wedge o(y) \in \Im(\mathit{register}) \wedge o(x) = o(y)\}$. As the final step, we can now merge the representation with the preceding context, yielding the DRS in (13).

(13)
$$\begin{array}{|l|}\hline x\ y \\ \hline \mathit{delegate}(x) \\ \mathit{arrive}(x) \\ \mathit{female}(x) \\ \mathit{registered}(y) \\ x = y \\ \hline \end{array}$$

This is interpreted as in (14):

(14) $\{\langle i, o\rangle | i \subset_{\{x,y\}} o \wedge o(x) \in \Im(\mathit{delegate}) \wedge o(x) \in \Im(\mathit{arrive}) \wedge$
$o(x) \in \Im(\mathit{female}) \wedge o(y) \in \Im(\mathit{register}) \wedge o(x) = o(y)\}$

We see that the CCP of the discourse is a relation between assignments i and o such that o extends i with values for the discourse referents x and y such that the conditions in (13) are true.

3.2 *Compositional DRT*

Compositional DRT aims at equipping DRT with lambdas to achieve compositionality and replace the construction algorithm with standard functional application. The idea of CDRT is to inject the part of the metalanguage dealing with assignments into the object language. DRSs can then be viewed as abbreviations for more complex lambda terms that contain variables for states, which as we will see serve as CDRT's reconstruction of assignments/embeddings. The attraction of the language lies not primarily in the full representations, but in the fact that the working linguist can use lambda terms over DRSs/abbreviated terms and be guaranteed that the result has a proper set-theoretic interpretation.

To achieve this, the language is enriched with new types. Apart from ordinary individuals (type e) and truth values (type t), it has registers (type π) and states (type s). We let ν be a non-logical constant, a total function of type $s(\pi e)$, and use the type e term $\nu(i)(\delta)$ to denote the inhabitant of register δ in state i. In other words, each state assigns an inhabitant to each register; it works as an assignment. In Muskens' formulation, there are two types of registers, specific (for names) and non-specific. Here, and throughout the article, we simplify the presentation by only considering non-specific referents and leave for future research how Muskens' treatment of names could be implemented within the theory developed here.

We need a couple of axioms to restrict the models we consider. In stating these we will use the abbreviation $i[\delta_1 \ldots \delta_n]j$ for

(15) $\forall\delta((\delta_1 \neq \delta \wedge \ldots \wedge \delta_n \neq \delta) \rightarrow \nu(i)(\delta) = \nu(j)(\delta))$

that is states i and j differ at most in the inhabitant they assign to registers $\delta_1 \ldots \delta_n$. We can now state the axioms. Muskens' AX2 and AX4 are not needed here as they concern the difference between specific and non-specific referents. $u_1 \ldots u_n$ are register constants.

(16) AX1 $\forall i.\forall\delta.\forall x.\exists j.i[\delta]j \wedge \nu(j)(\delta) = x$
(17) AX3 $u_n \neq u_m$ for each two different u_n and u_m

The first axiom ensures that we have enough states: for each state, each register and each individual, there is another state just like the first one, differing only in having the given register inhabited by the given individual. The third axiom requires that our model does not interpret two different constants as referring to the same register.[13]

We can then view a DRS such as (18) as an abbreviation for the lambda term in (19):

(18)
$$\begin{array}{|l|}\hline u_1 \ldots u_n \\ \hline \\ \Gamma_1 \\ \vdots \\ \Gamma_n \\ \\ \hline \end{array}$$

(19) $\lambda i.\lambda o.i[u_1 \ldots u_n]o \wedge \Gamma_1(o) \wedge \ldots \wedge \Gamma_n(o)$

For reasons of space, we will often use the condensed notation $[u_1 \ldots u_n \mid \Gamma_1, \ldots, \Gamma_n]$ for boxes such as (18).

In stating the conditions $\Gamma_1, \ldots, \Gamma_n$, we use the abbreviations in (20), where K and L are DRSs in complex conditions.

(20)

$R(\delta_1, \ldots, \delta_n)$	$\lambda i.R(\nu(i)(\delta_1), \ldots, \nu(i)(\delta_n))$
δ_1 is δ_2	$\lambda i.\nu(i)(\delta_1) = \nu(i)(\delta_2)$
not K	$\lambda i.\neg\exists j.K(i)(j)$
K or L	$\lambda i.\exists j.K(i)(j) \vee L(i)(j)$
$K \Rightarrow L$	$\lambda i.\forall j.L(i)(j) \rightarrow \exists k.L(j)(k)$

The assertoric DRS in (9) does not get a direct interpretation, but is viewed as an abbreviation for (21), where u_1, though it represents a register, which can be thought of as a variable, is in fact a *constant* of the language.

(21) $\lambda i.\lambda o.i[u_1]o \wedge \mathit{delegate}(\nu(o)(u_1)) \wedge \mathit{arrived}(\nu(o)(u_1))$

This expression is interpreted on a usual type theoretic (Henkin) model, and it is useful to see how this works. Consider a simple model where there are only two individuals, *kim* and *kirsten*, and three discourse referents 1, 2, 3. Formally we have $\mathcal{M} = \langle\{D_e, D_\pi, D_t, D_s\}, \mathfrak{I}\rangle$ where $D_e = \{\text{kim, kirsten}\}$, $D_\pi = \{1, 2, 3\}$, $D_t = \{\mathbf{T}, \mathbf{F}\}$, $D_s = \{s_1, s_2, \ldots, s_8\}$. It follows from AX1 that there must be states assigning each possible

[13] This axiom is often found confusing and/or tautologous. The need for it arises from the fact that CDRT deals with assignments in the object language. In classical DRT, an assignment i can of course assign the same individual to the variables/discourse referents x and y, but x and y will not denote the same individual across all possible assignments. But in CDRT, if we allowed models in which $\mathfrak{I}(u_1) = \mathfrak{I}(u_2)$, then ν would assign the same inhabitant to u_1 and u_2 in all states.

inhabitant to each register, so we cannot have fewer than $2^3 = 8$ states. The function ν, which assigns inhabitants to each register in each state, is as in (22).

(22)

State	register 1	2	3
S_1	kim	kim	kim
S_2	kim	kim	kirsten
S_3	kim	kirsten	kim
S_4	kim	kirsten	kirsten
S_5	kirsten	kim	kim
S_6	kirsten	kim	kirsten
S_7	kirsten	kirsten	kim
S_8	kirsten	kirsten	kirsten

$\mathfrak{I}$ interprets the constants of the language, which are the two predicates, *delegate* and *arrive*, plus the constants naming the three registers. By AX3, two constants cannot refer to the same register. So $\mathfrak{I}$ will be as in (23), in the case where *Kim* is the only individual in the extension of *delegate* and *arrive*.

(23)
$$\begin{array}{lll} \text{delegate} & \mapsto & \{kim\} \\ \text{arrive} & \mapsto & \{kim\} \\ u_1 & \mapsto & 1 \\ u_2 & \mapsto & 2 \\ u_3 & \mapsto & 3 \end{array}$$

As is transparent from the term, (21) is interpreted as a function from states to a function from states to truth values. We can view this as the characteristic function of a two-place relation between states *i* and *o* such that *o* differs from *i* at most in the inhabitant it assigns to 1 and the inhabitant that *o* assigns to 1 is in the extension of the predicates *delegate* and *register*. The pairs of states that satisfy this relation in our model are given in (24).

(24) $\{\langle s_1, s_1\rangle, \langle s_2, s_2\rangle, \langle s_3, s_3\rangle, \langle s_4, s_4\rangle, \langle s_5, s_1\rangle, \langle s_6, s_2\rangle, \langle s_7, s_3\rangle, \langle s_8, s_4\rangle \}$

It is useful to compare this interpretation to the standard DRT interpretation in (10). The assignments that were only part of the metalanguage in DRT are now object-language entities (states), and DRSs are relations between these. But if we consider how ν assigns inhabitants to registers in these states, the similarities between (24) and (10) emerge and we see that the output states in (24) are s_1, s_2, s_3, s_4, which all map u_1 to *kim*, who is in the extension of *delegate* and *register*. The crucial difference is that these states also map arbitrary inhabitants to the registers u_2 and u_3, which do not occur in the text. Whereas in DRT a discourse context is something that grows as the discourse proceeds and new referents get added, CDRT models a discourse context as something that gradually gets fixed. Unfortunately, there is no way to tell whether an inhabitant is arbitrary or fixed. Partializing ν with respect to registers, on the other hand, will give us control over which registers are actually in use and let us keep the model-theoretic characterization of accessibility which is found in classical DRT.

Before we go on, we need a way of combining DRSs. This we do by means of the sequencing operation ; which is defined in (25).

(25) $K \ ; \ L \equiv \lambda i.\lambda o.\exists k.K(i)(k) \wedge L(k)(o)$

That is, the sequencing of K and L introduces an intermediate state k which is the output of K and the input to L.

Let us now see how CDRT deals with anaphoric expressions. Consider the second sentence *She registered*. CDRT does not have rules to expand presuppositional DRSs like the one in (11) to lambda terms. Instead it requires that the syntax deliver discourse referents with indices. So the input is as in (26).

(26) A^1 delegate arrived. She_1 registers.

With these indices, it is possible to define a function *ant* which returns the antecedent of an anaphoric expression, and *dr* which returns the discourse referent introduced by a lexical item. We can therefore give meanings to *a* and *she* as in (27).

(27) a^n $\quad \lambda P.\lambda P'.[u_n|] \ ; \ P(u_n) \ ; \ P'(u_n)$
she_n $\quad \lambda P.P(\delta) \ ; [\,|\,female(\delta)]$ where δ is $dr(ant(\mathrm{she}_n))$

With these meanings in place, *She registered* is as in (28).

(28) $\lambda P.P(u_1) \, ; [\,|\,female(u_1)](\lambda x.[\,|\,register(x)]) \equiv [\,|\,register(u_1)]; [\,|\,female(u_1)]$

The sequencing of $[\,|\,register(u_1)] \, ; [\,|\,female(u_1)]$ results, as we would expect, in (29).

(29) $[\,|\,register(u_1), \, female(u_1)]$

This sequencing is relatively trivial since the universes of the DRSs are empty. Let us now look at what happens when we sequence (21) and (the expansion of the abbreviation in) (29). By the definition in (29), this gives us (30).

(30) $\lambda i.\lambda o.\exists k.i[u_1]k \wedge delegate(\nu(k)(u_1)) \wedge arrive(\nu(k)(u_1)) \wedge k[\,]o \wedge register(\nu(o)(u_1)) \wedge \; female(\nu(o)(u_1))$

We see that since state o does not differ from state k in the value of any register δ which occurs in a term of the type $\nu(k)(\delta)$, we can reduce this by eliminating state k:[14]

(31) $\lambda i.\lambda o.i[u_1]o \wedge delegate(\nu(o)(u_1)) \wedge arrive(\nu(o)(u_1)) \wedge register(\nu(o)(u_1)) \wedge female(\nu(o)(u_1))$

(31), unlike (30), can be abbreviated as a single DRS, namely (32):

(32) $[u_1|delegate(u_1),\ arrive(u_1),\ register(u_1),\ female(u_1)]$

This is clearly a reasonable representation of the discourse, but it presupposes that the discourse referents are coindexed in the input to semantics. As we noted in section 2.1, the modification of CDRT proposed in Muskens (2011) does not really offer a satisfactory solution to this: in our case, it would involve replacing the representation in (32) with (33), where x_1 is a variable.

(33) $[u_1|delegate(u_1),\ arrive(u_1),\ register(x_1),\ female(x_1)]$

Nothing in the semantics prevents us from substituting the constant u_4 for x_1. The declaration of u_1 in the universe means that the input and output states are only allowed to differ in their inhabitant of u_1. But all registers are inhabited in all states, so (33) with u_4 for x_1 is semantically well-formed.

4 A PARTIAL THEORY OF TYPES

Let us now build up a partial theory of types, which will allow us to have partial assignments. We use a semantics based on a special 'undefined object' # (as in Blamey (1995) for first-order logic and Lapierre (1992) and Lepage (1992) for the theory of types). This allows us to give a functional semantics for partial lambda calculus rather than the more unfamiliar relational semantics given in Muskens (1995), although the latter would also be fully adequate.

[14] More formally, this reduction follows from the merging lemma, which is discussed in Muskens (1996: 150, 159), and (for partial CDRT) in Appendix A of this article.

Notice that it is not trivial to build a partial theory of types. Several approaches to anaphora (e.g. FCS and Predicate Logic with Anaphora (PLA), Dekker 1994) make use of a first-order language which is partial in the sense that atomic formulae can fail to have a truth value. These approaches typically assume that a lambda calculus over such a language is automatically available, but this is in general not the case. There are various problems with a partial theory of types, including the failure of currying, observed by Tichý (1982) (see Lapierre (1992: 521f.) for other problems).

There are several ways to solve the problems that arise in partial type theory, and the issues are not only technical. The logics of Blamey (1995), Lapierre (1992) and Lepage (1992) are all based on an underlying philosophy of undefinedness as lack of information, which leads to a focus on *monotonicity*: if the arguments of functions increase in definedness, then so should the definedness of the output. This is the same intuition as that underlying Kleene's strong logic. As Fox (2008: 248) put it, 'at every point of evaluation, w, every instance of # should be thought of as either 1 or 0; it's just that we are not told which one it is. If we can determine the truth value of a sentence in w ignoring all instances of #, the sentence will receive that value'.

But our motivation for partiality is different. What we need is to control which discourse referents have been introduced. Attributing properties to discourse referents that have not been introduced (as in (33) with u_4 for x_1) should be *nonsensical*. This is the intuition behind # in weak Kleene logic. Moreover, we need a non-monotonic identity predicate. Remember that in CDRT, we talk about states (aka assignments) *in the object language*. To compare partial assignments, we need a classical identity predicate, in which # = # is true, rather than undefined. Such a predicate is clearly non-monotonic.

4.1 *Types*

(1) a, b, e are types (times, eventualities, individuals)
(2) π_a, π_b, π_e are types (registers for times, eventualities, individuals)[15]
(3) s is a type (states)
(4) t is a type (truth values)
(5) n is a type (integers)
(6) if α and β are types, then $(\alpha\beta)$ is a type

[15] Throughout the article, we simplify and assume there is only one type π for individuals.

4.2 *Syntax*

Since our focus is not on the properties of the formal language, we do not consider issues such as expressive completeness, but simply define the connectives that we will need.

(1) Every variable and constant of any type is a term of that type. Terms are subscripted with their type unless no confusion can arise.
(2) $\star$ is a term of type t.
(3) $(A_t \wedge B_t)$, $\neg A_t$ and ∂A_t are terms of type t.
(4) $(A_\alpha = B_\alpha)$ is a term of type t.
(5) $A_{\alpha\beta}(B_\alpha)$ is a term of type β.
(6) $\lambda x_\alpha.A_\beta$ is a term of type $(\alpha\beta)$.
(7) $\forall x_\alpha.A_t$ is a term of type t.

∂ is Beaver's unary presupposition operator (Beaver 1992). We use the abbreviations in (34).

(34)	$\top$	for	$\star = \star$
	$\bot$	for	$\star = \top$
	$(A_t \vee B_t)$	for	$\neg(\neg A_t \wedge \neg B_t)$
	$(A_t \rightarrow B_t)$	for	$\neg A_t \vee B_t$
	$(A_\alpha \neq B_\alpha)$	for	$\neg(A_\alpha = B_\alpha)$
	$\exists x.A_t$	for	$\neg\forall x.\neg A_t$

4.3 *Semantics*

We interpret the language on a set of pairs of domains $\langle D, D'\rangle$ for each type such that

(1) For base types α the classical domain D_α is a non-empty set where $\#_\alpha \notin D_\alpha$ (in particular D_t is $\{\mathbf{T}, \mathbf{F}\}$), and the fixed-up domain D'_α is $D_\alpha \cup \{\#_\alpha\}$.
(2) For functional types $\alpha\beta$, the classical domain $D_{\alpha\beta}$ is the set of functions $D_\alpha \mapsto D_\beta$ and the fixed-up domain $D'_{\alpha\beta}$ is the set of functions $D'_\alpha \mapsto D'_\beta$.

Let $\sqsubseteq_\alpha$ be a partial order on the fixed-up domains of base types D'_α such that $x \sqsubseteq_\alpha y$ iff $x = y$ or $x = \#_\alpha$. On the fixed-up domains of functional types $D'_{\alpha\beta}$, $f \sqsubseteq_{\alpha\beta} g$ iff for all $x \in D'_\alpha$, we have $f(x) \sqsubseteq_\beta g(x)$. Clearly, for all types α, $\langle D'_\alpha, \sqsubseteq_\alpha\rangle$ is a meet semi-lattice and we will refer to its bottom element as the *undefined object* of type α. If in a model a term of type α is interpreted as the undefined object of type α, we will say that it is undefined.

Let M be a model consisting of the set of domains as defined above and an interpretation function $\Im(c)$, which maps constants of each type α to D'_α and a be an assignment of variables of each type α to elements of D'_α. As usual, $a_{\xi/d}$ is the assignment that differs from a only in assigning d to the variable ξ. We define an interpretation of our language relative to the model M and the assignment a as in (35).

(35)
$$
\begin{array}{lcl}
\|c\|^{M,a} & = & \Im(c) \text{ if } c \text{ is a constant} \\
\|x\|^{M,a} & = & a(x) \text{ if } x \text{ is a variable} \\
\|*\|^{M,a} & = & \#_t \\
\left\|(A \wedge B)\right\|^{M,a} & = & \mathbf{T} \text{ if } \|A\|^{M,a} \text{ is } \mathbf{T} \text{ and } \|B\|^{M,a} \text{ is } \mathbf{T} \\
 & = & \#_t \text{ if } \|A\|^{M,a} \text{ is } \#_t \text{ or } \|B\|^{M,a} \text{ is } \#_t \\
 & = & \mathbf{F} \text{ otherwise} \\
\left\|\neg(A)\right\|^{M,a} & = & \mathbf{T} \text{ if } \|A\|^{M,a} \text{ is } \mathbf{F} \\
 & = & \mathbf{F} \text{ if } \|A\|^{M,a} \text{ is } \mathbf{T} \\
 & = & \#_t \text{ otherwise} \\
\left\|\partial(A)\right\|^{M,a} & = & \mathbf{T} \text{ if } \|A\|^{M,a} \text{ is } \mathbf{T} \\
 & = & \#_t \text{ otherwise} \\
\left\|(A_\alpha = B_\alpha)\right\|^{M,a} & = & \mathbf{T} \text{ if } \|A\|^{M,a} = \|B\|^{M,a} \\
 & = & \mathbf{F} \text{ otherwise} \\
\left\|\phi_{\alpha\beta}(\psi_\alpha)\right\|^{M,a} & = & \left\|\phi_{\alpha\beta}\right\|^{M,a}(\left\|\psi_\alpha\right\|^{M,a}) \\
\left\|\lambda\xi_\alpha.\phi_\beta\right\|^{M,a} & = & \text{the function } f:\ D'_\alpha \mapsto D'_\beta, \\
 & = & \text{such that for all } d \in D'_\alpha, f(d) \ = \|\phi\|^{M,a_{\xi/d}} \\
\|\forall\xi_\alpha.\phi_t\|^{M,a} & = & \#_t \text{ if } \|\phi\|^{M,a_{\xi/d}} \text{ is } \#_t \text{ for all } d \text{ in } D_\alpha \\
 & = & \mathbf{F} \text{ if } \|\phi\|^{M,a_{\xi/d}} \text{ is } \mathbf{F} \text{ for some } d \text{ in } D_\alpha \\
 & = & \mathbf{T} \text{ otherwise}
\end{array}
$$

This truth definition induces standard weak Kleene semantics for $\wedge$, $\vee$ and $\rightarrow$. Observe that bound variables range only over the classical domains. Our underlying logic is in fact a free logic where (for the base types) the inner and outer domains differ only by # and there are no quantifiers ranging over the outer domains. Also, in line with our interpretation of $\#_t$ as nonsense, quantifiers range over 'sensible' values, that is if $P(x)$ is $\#_t$ for some values of x, $\forall x.P(x)$ can still be true as long as $P(x)$ is true of all values of x that do *not* make $P(x)$ undefined.

So far we have a general, partial theory of types. But for our purposes, we do not want to consider all possible models on these domains.

First, we want to put some restrictions on partiality. Since we are not interested in the formal system per se, we do not attempt to axiomatize these, but some remarks are in order. We want a classical identity predicate: every object, even the undefined one, is identical with itself. Predicates of (n-tuples of) individuals should be classical whenever their arguments are defined, and nonsensical whenever (one of) their arguments are undefined. Notice that this means our system admits existential generalization as in (36) as a theorem.

(36) $P(a_1, \ldots, a_n) \rightarrow \exists x_1 \ldots \exists\ x_n.P(x_1, \ldots, x_n)$

where P is a predicate distinct from identity. This will be of some use in proving the unselective binding lemma (Appendix C). Note also how (36) follows naturally from our view of undefinedness as absurdity.[16]

Besides restricting partiality, we need some axioms that make sure our logic functions as a logic of change, extending Muskens' original proposal but keeping much of the same machinery. This is introduced formally here. It is also explained in a more informal way in section 5.1, so the reader who is not interested in the details can go directly to that section.

For each type of register α, we have a constant ν_α of type $s(\pi_\alpha e)$. To each state s, ν_α assigns a (possibly partial) function from registers to individuals of type α. If $\nu(s)(\delta)$ is undefined, we will say that δ is uninhabited in s. Similarly if $\nu(s)(\delta)$ denotes an individual e, we will say that e is the inhabitant of δ in s. If $\nu(s)$ is undefined, our definition of undefinedness for non-base types means it is interpreted as the bottom of $\langle D'_{\pi e}, \sqsubseteq_{\pi e}\rangle$, which is the function that maps all registers to the undefined individual $\#_e$, that is s is 'the empty state', the state where no register is inhabited.

We require all sets D_{π_α} of registers to be well-ordered under a relation $<$. In other words, we can take D_{π_α} to be $\mathbb{N}$.[17] This will let us pick out the next discourse referent to introduce: we have a successor function s_α on each domain D_{π_α}, and for each register type π_α we have a function $\mathcal{L}_\alpha$ of type $(s\pi_\alpha)$ which gives us the first uninhabited register in the given state.

Notice that if we have three states i, j, k and j differs from i only in assigning an inhabitant to $\mathcal{L}_e(i)$ and k differs from j only in assigning an

[16] In strong Kleene logic, on the other hand, it would not be natural to exclude the undefined object from having properties: for example, if P was true of all defined objects in the model, it should be true of the undefined object as well.

[17] Observe that the fixed-up domain D'_{π_α} is not well-ordered as neither $x < y$ nor $y < x$ holds whenever $\|x\|$ is $\#_{\pi_\alpha}$. This follows from the fact that we do not let $\#_{\pi_\alpha}$ have any properties except self-identity.

inhabitant to $\mathcal{L}_e(j)$, then k differs from i only in assigning inhabitants to $\mathcal{L}_e(i)$ and $s_e(\mathcal{L}_e(i))$ since the order of registers is identical across states. In other words, we can express the difference between i and k without reference to the intermediate state j, something which will be useful when dealing with DRS sequencing. We will use x_{i_1} to abbreviate $\mathcal{L}(i)$. In other words, x_{i_1} picks out the register which is 'next in line' among those not present in i. Writing s^n_α for n applications of s_α, x_{i_2} abbreviates $s(\mathcal{L}(i))$ and x_{i_n} abbreviates $s^{n-1}(\mathcal{L}(i))$.

Our states must satisfy certain axioms. In expressing these, we use the abbreviation $i[\delta_1 \ldots \delta_n]o$ that was defined in (15), but require that $\delta_1 \ldots \delta_n$ is a continuous range of registers according to the order $<$. From the point of view of the definition in (15), o could 'overwrite' inhabitants of $\delta_1, \ldots, \delta_n$ in i, but we will use $\mathcal{L}$ to limit such destructive updates to embedded states where we want to rule out coreference.

We can now state the following three axioms concerning ν.

(1) $\exists s.\forall \delta.\neg\exists e.\nu(s)(\delta) = e$
(2) $\forall s.\forall e.\exists s'.s[x_{s_1}]s' \wedge \nu(s')(x_{s_1}) = e$
(3) $\forall s.\forall \delta.\forall \delta'.(\exists e.\nu(s)(\delta) = e \wedge \delta' < \delta) \rightarrow \exists e'.\nu(s)(\delta') = e'$

1 says that there is an empty state. We will occasionally use $s_\emptyset$ as a constant referring to this state. 2 says that for each state and each individual, there is another state which differs only in assigning the given individual to the first uninhabited register in the given state. This has the same function as Muskens' AX1 (16), ensuring that we have 'enough states' to encode all combinations of registers and individuals. 3 says that there are no gaps among the inhabited registers; if a given register is inhabited in a given state, then all lower registers are inhabited in that state. We do not need an equivalent to Muskens' AX3, as we are simply going to avoid using constants for registers.

5 PARTIAL CDRT

5.1 *Tracking discourse referents: states, stacks and other contexts*

The crucial part of any dynamic system is the tracking of discourse referents. This is what sets dynamic systems apart from static ones: in contrast, the conditions that we put on discourse referents work similarly as in static systems.

A set of discourse referents and their associated values is traditionally called a *context* in dynamic semantics and this is what gave rise to the notion of sentence meanings as context change potentials. CDRT, both in Muskens' version and the present one, uses *states* for a very similar notion. In this section we explain how states work in partial CDRT, and compare them to another concept that has been used to reconstruct the contexts of traditional dynamic semantics, namely *stacks* (Vermeulen 1993; Dekker 1994; Bittner 2001; van Eijck 2001; Nouwen 2003; Schlenker 2005; Bittner 2007; Nouwen 2007; van Eijck & Unger 2010).[18]

We saw in section 3.2 that in classical CDRT, states work as total assignments, that is each state assigns inhabitants to all discourse referents of the language. DRSs are interpreted as state updates, that is functions from states to states. Discourse referents that have been used get fixed inhabitants, that is the update functions are not allowed to change them. Unused discourse referents, on the other hand, have arbitrary inhabitants.

In partial CDRT, states are *partial* assignments. They assign inhabitants only to discourse referents that have been used, and are undefined (in the technical sense that was made precise in section 4.3) for other discourse referents. Moreover, we assume that discourse referents are ordered and that there is a lowest discourse referent according to this order. Except for the empty state, which is undefined for all discourse referents, all states assign an inhabitant to the first discourse referent 1 and possibly to a continuous range of other discourse referents. That is, our axioms admit of states such as $s_\emptyset$ (the empty state), j, k and l in (37), but state m in (38) is malformed, since the range of inhabited discourse referents is not continuous.

(37)

$s_\emptyset$:	1	2	3	4	...
	$\#_e$	$\#_e$	$\#_e$	$\#_e$	...
j:	1	2	3	4	...
	kim	$\#_e$	$\#_e$	$\#_e$	...
k:	1	2	3	4	...
	kirsten	$\#_e$	$\#_e$	$\#_e$	...
l:	1	2	3	4	...
	kim	*kirsten*	$\#_e$	$\#_e$	...

(38)

m:	1	2	3	4	...
	kim	$\#_e$	*kirsten*	$\#_e$	...

Update functions, then, will be ordered list of inhabitants to add to the input state. The sentence *A delegate arrived* would update the input state

[18] The terminology actually varies here too, but I will use *stack* to refer to any data structure which is a list where new referents get added to the end.

by adding some individual who arrived to the next uninhabited register. If *kim* is in the denotation of *delegate* and *arrived*, a possible update would take $s_{\emptyset}$ to j. That means the names we use for discourse referents are completely inessential, and we can sequence states by simply adding the inhabited registers of the second state to those of the first. For example, sequencing j and k yields l. The fact that *kirsten* moves from the first register in k to the second register in l lets us avoid overwrites.

The ordered and partial nature of states in partial CDRT makes them very similar in concept to *stacks*, which were also devised to avoid the overwrite problem in dynamic semantics. In the original implementation (Vermeulen 1993), there is stack associated with each discourse referent. When a discourse referent is reused, the new value is pushed to its stack, which also tracks the values of previous uses of that discourse referent. van Eijck (2001) then showed that it is possible to do with one stack, thus avoiding the use of discourse referents altogether. For concreteness, we base our presentation here on the stack-based theory developed in van Eijck & Unger (2010, chapter 12), which also works with a single stack, though everything we will have to say about stacks carry over to the other approaches, including those that use variables and multiple stacks.

In the approach of van Eijck & Unger (2010), stacks are lists of individuals, or in other words, (partial) functions from indices to inviduals. $\hat{}$ is the append operator: if c is a stack, then $c\hat{}x$ is the result of appending x to c. Dynamic existential quantification is defined as in (39).

(39) $\exists := \lambda c.\lambda c'.\exists x.c\hat{}x = c'$

So $\exists$ has the effect of extending an input stack c with an element x from the domain, and creating an output stack $c\hat{}x$. If the sentence we analyse is *A delegate arrived*, then x must be in the denotation of *delegate* and *arrived*. This is exactly the same thing as what happens in partial CDRT. The same could be said about other stack-based approaches, but the single-stack approach in van Eijck & Unger (2010) makes the similarity especially striking. One is tempted to ask what the advantage of our new formal machinery really is.

The crucial difference is that in partial CDRT states track *occurrences* of discourse referents, whereas stacks track *values* of discourse referents. This difference shows up if we extend the discourse with *She registered*. In partial CDRT, all referring expressions, also anaphoric ones, will extend the state with a new inhabited register. If the referring expression is explicitly anaphoric, as is the case with *she*, then the semantics of anaphoricity will have to make sure that that register must corefer with some previously introduced register. Notice that we are returning to the classical DRT approach, which is otherwise abandoned in all compositional versions of dynamic semantics. As we saw in section 3.1, Kamp *et al.* (2011) assume that anaphoric expressions do introduce new discourse referents, whose reference is then pinned down by conditions like $y = x$ in (13).

In stack-based approaches, things work differently. Anaphoric expressions do not push new values onto the stack; instead they pick an element from the stack as their referent. How this is done varies from approach to approach. One way is to use indexation so that she_n picks up the n-th element from the stack: this is what we find in the formalism developed in the first half of chapter 12 in van Eijck & Unger (2010), and in the main body of the article van Eijck (2001) as well as in Dekker's Predicate Logic with Anaphora (Dekker 1994), Bittner's Logic of Change with Centering (Bittner 2001, 2007) and Schlenker's semantic approach to binding theory (Schlenker 2005). The idea is expressed in the following way by Nouwen (2007: 132): '[p]ronouns pick out antecedents by choosing stack positions, not by choosing a variable name.' Since at each point in the discourse, the stack contains all and only the accessible antecedents, this provides a semantic account of accessibility, unlike coindexation approaches, which as we saw in section 2.2, need to stipulate syntactic constraints on accessibility.

However while such indices are less problematic than coindexation, where they come from remains just as mysterious as in coindexation. One could perhaps try to see she_n as a function from integers to individuals, where the integer is to be provided by a pragmatic component that deals with anaphoric resolution. So the denotation of *She registered* would be a function from integers n to truth values such that we get truth if the n'th element on the stack is in the denotation of *registered*. However, it would still be necessary for the pragmatic component to disambiguate n before we process the next sentence; for once we do that, the stack changes, and there is no way we can extract past states of the stack from the semantic representation of the discourse. This means we cannot keep monotonic and non-monotonic content properly separated across stretches of discourse, and we get the problems observed in section 2.4.

Another way is to assume lexical ambiguity. After all, while it seems preposterous to claim that languages have a special word he_{16} that is used to refer back to a_{16} and $john_{16}$ and so on (as a lexical view of the coindexation approach would require), it seems less counterintuitive that there should be a special anaphor that refers back to the last mentioned or most topical referent, and other anaphors to refer to less central participants in the discourse. Pronouns distinguishing the fourth most topical participant from the fifth most are less appealing, though. There is also a problem pointed out by Nouwen (2003: 138f.): the lexical entry for an anaphoric pronoun will really have to be a list of entries he_1, he_2. . ., he_n. Using he_{16} in a context where the stack is only five elements long will yield an undefined meaning, so we do have a semantic characterization of its malformedness. However, 'by making indexation context dependent, we expect that, in principle, there should be a way of deriving defined interpretations only' (Nouwen 2003: 139).

Aware that indexation is a simplification, van Eijck (2001: 349) suggests that 'pronouns can be translated as invitations to pick a reference from the current context. The author is aware that this sketch is far too concise, but a more detailed account of all this will have to wait for another occasion'. A similar idea is found in de Groote (2006), who assumes that anaphors are resolved by choice operators that act as 'oracles', taking a left context as their argument and yielding back an individual. The second part of chapter 12 in van Eijck & Unger (2010) is a bit more precise and implements pronoun resolution as a function picking out the elements from the stack that are compatible with the gender information in the pronoun in order of salience, yielding a list of possible antecedents in order of preference (for our purposes, we can ignore how salience is spelled out).

But appealing though it might be to see anaphoric resolution as the picking of a reference from the current context, there are problems with this view. As we saw in section 2.4, anaphoric resolution can often be guided by material in the context *following* the anaphor. It is unclear how to deal with this in systems where anaphoric resolution is a (deterministic or non-deterministic) 'function' from left contexts to resolutions. Again, there is no way to retrieve past contexts from the semantic representations, and there is no way to 'downdate' the interpretation by going back to the unresolved anaphor but keeping the new, following context.

There is also a more technical reason why this view of anaphoric resolution goes wrong. As first pointed out by Nouwen (2003: 140), we get the wrong result in quantified structures. Consider (40), with two alternative continuations in (40-a) and (40-b).

(40) France_1 is a monarchy. Every_2 nation cherishes its_3 king.
  a. So the French love King François.
  b. That is because his appearances in the UN have been so convincing.

(40-a) suggests an interpretation where 3 = 2, that is, for each nation, it is the case that that nation cherishes its king. (40-b), on the other hand, suggests that 3 = 1, that is, every nation cherishes the King of France. So the resolution of 3 is ambiguous. But ever since Kamp (1981) and Heim (1982), dynamic semantics analyses *every* as quantifying over contexts: *every P Q* is a test on contexts that succeeds just in case every extension of the global context that satisfies *P* can be extended to a context that satisfies *Q*. This means that if pronouns are resolved in context (in the technical sense of dynamic semantics), then the reference of 3 must be resolved for *each context P* that provides a value for the quantifier *every nation*; possibly, it would be resolved to 2 in some of these contexts and 1 in others. In other words, we predict that there is a reading of (40) where every nation cherishes either the French king or its own king. This is clearly wrong.[19]

Nouwen (2003: 140–44) offers a representational solution to this kind of problem, which is sufficient for his purposes, the analysis of problematic kinds of plural anaphora. But such a solution is directly at odds with the task we have set ourselves here, which is to develop a non-representational account of anaphora.[20] To my knowledge, there is no such account in the literature.

How can a system that tracks occurences of discourse referents rather than their values fare better than the approaches we have seen? There are two crucial points: first, in this way, when anaphoric expressions too leave their mark on the context, we have a way of reconstructing the context of that anaphor from the semantic representation: it is simply the context up to the anaphor. Contrast this with stack-based approaches where we cannot reconstruct previous states of the stack from the semantic representation.

Second, we get a way of referring to an anaphoric expression *irrespective of its reference*. This yields a clear distinction between monotonic

[19] While this problem is quite general to all approaches that model anaphora as 'functions' from contexts to value(s), it is perhaps most easily seen in the framework of Groote (2006). *Every farmer who owns a donkey beats it* is represented as in (i).

(i) $\lambda e.\lambda\phi.(\forall x.farmer(x) \rightarrow (\forall y.(donkey(y) \wedge own(x,y)) \rightarrow beat(x,\ sel_{it}(x :: y :: e)))) \wedge \phi(e)$

e is a left context, which is modelled as a set of discourse referents. ϕ is a function from left contexts to right contexts, so $\phi(e)$ just says that the right context should take the sentence's left context as input, that is the sentence itself does not change the global context (because quantification only introduces embedded contexts). The interesting part is $sel_{it}(x :: y :: e)$. sel_{it} is an 'oracle' picking an antecedent for *it* from a context, in this case the context consisting of y and x appended to the original left context e. We see that $sel_{it}(x :: y :: e)$ occurs in the scope of the quantifiers $\forall x$ and $\forall y$, so we cannot be sure that it makes the same choice for all values of these variables.

[20] On the other hand, we will have nothing to say about plural anaphora, which will have to be left to future work.

and non-monotonic content. Consider (13) again. The condition $x = y$ represents non-monotonic content; the rest is monotonic. What is lacking in DRT is a way of representing accessibility in stretches of discourse: if we sequenced (13) with a representation of the following discourse, the result would be a new DRS with possibly more discourse referents in its universe, and these would be accessible inside the DRS. This is why it is essential to resolve preliminary DRSs before DRS sequencing. The variable y gives us what we need to talk about the anaphoric expression without fixing its reference (as seen in the traditional notation $y = ?$), but the unordered nature of DRS universes makes it impossible to leave the value of y open across stretches of discourse.[21]

5.2 *Introducing boxes*

The logic developed in section 4 extends CDRT with the ingredients that will allow us to develop a clear model-theoretic interpretation of anaphora and accessibility. But like classical CDRT, the representations are somewhat cumbersome and the real attraction lies in the possibility of defining abbreviations which allow us to work with representations quite close to those of 'normal DRT'. We will therefore adopt abbreviations similar to those of Muskens (1996) given in (20), but we need some extra machinery to deal with the expressions for registers. This is to make sure that our abbreviations keep track of the ordering of registers in the underlying language.

We want whatever discourse referents occur in the universe of a box to be abbreviations for registers picked out by the functions s and $\mathcal{L}$ (remember that x_{i_1} abbreviates $\mathcal{L}(i)$, which gives us the first uninhabited register in state i). So we will let x_α abbreviate x_{i_α},[22] where i is in input state of the DRS in whose universe x_α occurs. This means we no longer use constants to refer to registers. Instead, the registers that are introduced in the universe of a box are always the next registers 'in line' and the overwrite problem is avoided. Moreover, since the registers are ordered, we can generalize DRT's notion of accessibility to registers which are inhabited in the same state, that is inside DRSs.

[21] In fact, the ambiguity of anaphoric expressions can even be more radical than what we have discussed in this section, and involve more than just uncertainty of reference. The majority of the world's languages lack a definite article altogether, and in such languages NPs are systematically ambiguous as to whether they are anaphoric or not. Again, this ambiguity might only be resolvable in the following context. Given the very different way stack-based approaches treat anaphoric and non-anaphoric expressions, it is not obvious how they could deal with this situation without assuming a lexical ambiguity.

[22] It is sometimes necessary to renumber the discourse referent when we expand abbreviations, see below.

The abbreviations in (41) will achieve what we want. K^{α}_{β} is a DRS abbreviation in whose universe the discourse referents $x_{\alpha} \ldots x_{\beta}$ are declared, where $x_{\alpha} \ldots x_{\beta}$ is a set of variables with a continuous range of subscripts $\alpha \ldots \beta$ with $\alpha > 0$. (For the purposes of index calculations, an (unembedded) DRS with an empty universe is K^{1}_{0}).

(41)
$$
\begin{array}{ll}
[x_{\alpha} \ldots x_{\beta} | \Gamma_1, \ldots, \Gamma_{\gamma}] & \lambda i.\lambda o.\partial(i[x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}}]o) \wedge \Gamma^{*}_{1}(o) \wedge \cdots \wedge \Gamma^{*}_{\gamma}(o) \\
R(\delta_{\alpha}, \ldots, \delta_{\beta}) & \lambda i.R(\nu(i)(\delta_{\alpha}), \ldots, \nu(i)(\delta_{\beta})) \\
\partial(R(\delta_{\alpha}, \ldots, \delta_{\beta})) & \lambda i.\partial(R(\nu(i)(\delta_{\alpha}), \ldots, \nu(i)(\delta_{\beta}))) \\
\delta_{\alpha} \text{ is } \delta_{\beta} & \lambda i.\nu(i)(\delta_{\alpha}) = \nu(i)(\delta_{\beta}) \\
\neg K & \lambda i.\neg\exists j.K(i)(j) \\
K \vee L & \lambda i.\exists j.K(i)(j) \vee L(i)(j) \\
K^{\alpha}_{\beta} \Rightarrow L^{\delta}_{\varepsilon} & \lambda i.\forall j.K(i)(j) \rightarrow \exists k.L^{+}(j)(k) \\
K^{\alpha}_{\beta} \; ; L^{\delta}_{\varepsilon} & \lambda i.\lambda o.\exists k.K(i)(k) \wedge L^{+}(k)(o)
\end{array}
$$

The abbreviated notation is somewhat less expressive than the full language: the full language indexes discourse registers relative to a particular state, whereas the abbreviations do not express relative to which state we index. Therefore we need to make sure that in all abbreviations x_{α}, α is always interpreted relative to the same state, the (input state of the) main DRS in the expression. This is why we allow DRS universes to start with indices higher than 1. When such universes occur alone, the translation alters the index the amount of the 'offset' $1-\alpha$, for in such cases we index relative to the input state of that DRS. But when such a DRS is embedded, the index is altered relative to the offset in the main DRS, which is typically zero, since we now index relative to the input state of the main DRS.

The *- and +-rules help us translate the abbreviated discourse referents to the full representations. To define them, we need to introduce a distinction between *bound* and *free* discourse referents in the abbreviated language.[23] We first introduce a relation 'is directly embedded under', $L < K$, which holds between two DRSs iff 1. K contains a condition of the form $\neg L$, or 2. K contains a condition of the form $L \vee M$ or $M \vee L$, or 3. K contains a condition of the form $L \Rightarrow M$, or 4. K and L occur in a condition of the form $K \Rightarrow L$. or 5. K and L occur in an expression of the form K ; L. $<$ defines an accessibility path from a DRS K, and the transitive closure of $<$ gives us the set of DRSs accessible from K. An occurrence of a discourse referent δ in a condition Γ in

[23] The distinction we define here pertains to binding from DRS universes, for which we need special renaming rules to keep track of accessibility. Discourse referents can also be *lambda-bound*, in which case the usual renaming rules (alpha equivalence) apply. In the full language, freedom and bondage of variables are of course as defined in type theory.

a DRS K is said to be bound from the universe of L iff δ occurs in the universe of L and either $K = L$ or L is the first DRS along the accessibility path from K where δ occurs in the universe.

We can now define * and +. Γ^* is the formula that results from changing Γ by changing every bound occurrence of x_η (whether bound from the universe of the main or an embedded DRS) to $x_{i_{\eta+(1-\alpha)}}$. L^+ is the box that results from changing every free occurrence of x_η that is free in L but bound in $K \Rightarrow L$ or $K\,;\,L$ to $x_{i_{\eta+(1-\alpha)}}$.[24] We see that only discourse referents that are bound from the universe of their own or an embedding DRS (except those bound from the antecedent of a conditional or in sequencing) are changed by the *-rule. The +-rule deals with cases where a discourse referent is bound from the antecedent of a condition or in sequencing.[25] Consider the expressions in (42).

(42) a. $[x_1|P(x_1)] \Rightarrow [x_2|Q(x_1,\ x_2)]$
b. $[x_1|P(x_1)];\ [\,|Q(x_1)]$

In these cases, x_1 is free in the rightmost DRS but bound in the entire expression. Hence, the +-rule applies and ensures that it is translated as a bound variable. Let us look at (42-b). By the definition of the sequencing abbreviation, (42-b) becomes (43).

(43) $\lambda i.\lambda o.\exists k.[x_1|P(x_1)](i)(k) \wedge [\,|Q(x_{i_1})](k)(o)$

We see that x_1 in the second box has been replaced by x_{i_1}, where i is the input state of the first box. By the definition of the box abbreviations (and subsequent reduction), this becomes (44).

(44) $\lambda i.\lambda o.\exists k.\partial(i[x_{i_1}]k) \wedge P(x_{i_1})(k) \wedge \partial(k[\ ]o) \wedge Q(x_{i_1})(o)$

Here we see that the occurrences of x_1 in the first box also end up as x_{i_1}, so we get the desired effect of binding in sequencing.

[24] In the usual case, universes start with x_1, in which case $(1-\alpha)$ is 0 and both the + -rule and the *-rule simply replace x_η with x_{i_η}. Observe that the +-rule will not apply to conditionals when they are embedded as conditions in another box, since x_η is bound and will be changed by the *-rule. But + is needed to give a meaning to conditionals out of context. The difference is that we index relative to different input states, that of the antecedent DRS in the contextless case, that of the main DRS in other cases.

[25] Notice that free discourse referents remain unaltered, as in $[|Q(x_1)]$, which rewrites as $\lambda i.\lambda o.\partial(i[\,]o) \wedge Q(x_1)$. x_1 is a free variable in the full representation too. This means that a DRS abbreviation with free discourse referents (an 'improper DRS' in the terminology of DRT) gets translated as an open formula. In other words, we have a semantic characterization of the (im)properness of DRSs.

In general, sequencing in the abbreviated language behaves as in (45), provided there are no free variables in $[x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma]$; $[x_\delta \ldots x_\epsilon | \Gamma'_1, \ldots, \Gamma'_\zeta]$.[26]

(45) $[x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma]$; $[x_\delta \ldots x_\epsilon | \Gamma'_1, \ldots, \Gamma'_\zeta] =$
$$[x_\alpha \ldots x_\beta x_{\beta+1} \ldots x_{\epsilon+\beta+(1-\delta)} | \Gamma_1, \ldots, \Gamma_\gamma, \Gamma'^{\dagger}_1, \ldots, \Gamma'^{\dagger}_\zeta]$$

$\Gamma'^{\dagger}$ is the formula which results from Γ' by adding $\beta + (1-\delta)$ to the index of any discourse referent in Γ' that is bound from the universe of the second DRS.[27] This replacement lets us preserve the well-order on discourse referents in the underlying logic in our abbreviations as well. The equivalence in (45) is the abbreviated form of the merging lemma, which we prove in Appendix A.

Finally, in order to have the subscripts on discourse referents in the abbreviations accurately reflect the order of registers in the full language, we need a convention for renumbering bound discourse referents during beta reduction: whenever we substitute a term containing a DRS K in a position where it is embedded by another DRS L^α_β, we must add β to the index of all bound discourse referents in K. Whenever we subsitute in a position where K^α_β embeds another DRS L, we must add β to the index of all bound discourse referents in L. These renumberings take place before further beta reduction. In other words, we will renumber as in (46).

(46) $\lambda Q.[x_1 | \neg(Q(x_1))](\lambda x.[x_1 | see(x, x_1)]) \rightsquigarrow$
$[x_1 | \neg((\lambda x.[x_2 | see(x, x_2)])(x_1))] \equiv$
$[x_1 | \neg[x_2 | see(x_1, x_2)]]$

The effect is that embedded DRSs always start indexing referents with the lowest index not used in their accessibility path when they are embedded via functional application.

Observe that the renumbering is not an arbitrary convention to avoid variable clashes, but a reflection in the abbreviations of what happens in the unabbreviated formalism: Whenever we embed a DRS, we can start counting its discourse referents relative to the embedding DRS; and if we wish to be able to sequence the embedded DRS with other DRSs, we *must* do this. The worked out example in section 5.3 gives several examples of the application of this rule, and the unabbreviated version

[26] Among other things, this proviso ensures that we do not get 'reverse binding' from L to K in K ; L. A similar proviso is found in classical CDRT.

[27] In the usual case, universes will start with x_1, in which case $(1-\delta)$ is 0 and we simply add β to the index of bound discourse referents, that is, we start counting discourse referents in the second DRS at the point where the universe of the first DRS ends.

(of parts of the example) in Appendix B shows what happens in the unabbreviated representation.

5.3 *A discourse example*

We can now see how to deal with a simple discourse. Consider (47)

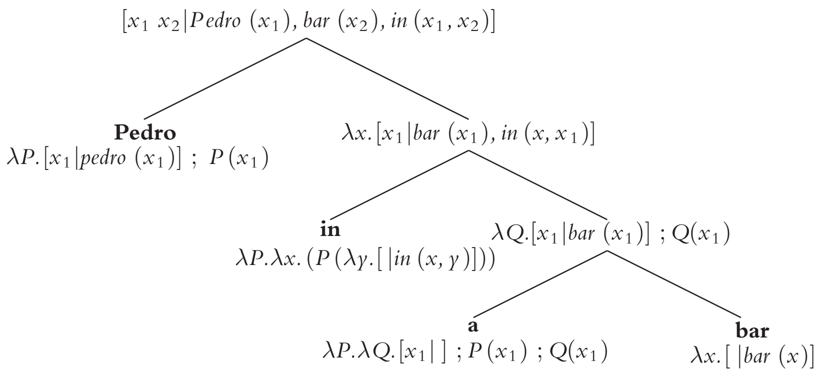
(47) Pedro is in a bar. Every woman who ever dated a man despises him. He is sad.

We build the meaning of the first sentence compositionally using the word meanings in (48) (assuming that the copula makes no semantic contribution apart from tense, which we ignore).

(48)
in	$\lambda P.\lambda x.(P(\lambda y.[\,|in(x, y)]))$
bar	$\lambda x.[\,|bar(x)]$
Pedro	$\lambda P.[x_1|pedro(x_1)]\ ;\ P(x_1)$
a	$\lambda P.\lambda Q.[x_1|]\ ;\ P(x_1)\ ;\ Q(x_1)$

Since we want to remain completely agnostic about the syntax, we make no attempt to model the syntax–semantics interface, but simply provide the semantic tree in (49), which shows how the meaning is built. Also, we do not try to show how the order of discourse referents within a sentence can be represented, as this again necessarily hinges on details of the syntactic framework.

(49)



It is worth saying something about the uppermost functional application, where **Pedro** applies to **in a bar**. The second occurrence of x_1 in **Pedro** is bound in sequencing, but would be accidentally bound by x_1 from **in a bar** after β reduction. However, when we subsitute for P in the embedded position in **Pedro**, we must renumber indices in the DRS we insert. This gives us (50).

(50) $[x_1|pedro(x_1)]\ ;\ (\lambda x.[x_2|bar(x_2),\ in\ (x,\ x_2)])(x_1)$

This works because crucially, and unlike in classical CDRT, discourse referents are bound variables. (50) reduces to (51).

(51) $[x_1|pedro(x_1)]\ ;\ [x_2|bar(x_2),\ in(x_1,\ x_2)]$

The result of this sequencing operation is as in the top node of (49), or the more pictorial version in (52).

(52)
$$\begin{array}{|l|}\hline x_1\,x_2 \\ \hline \\ Pedro\,(x_1) \\ bar\,(x_2) \\ in\,(x_1, x_2) \\ \\ \hline \end{array}$$

We now go on to the second sentence of (47). The new lexical entries we need are as in (53) (we ignore the contribution of *ever*).

(53)

every	$\lambda P.\lambda Q.[\,|([x_1|]\ ;\ P(x_1)) \Rightarrow Q(x_1)]$
woman	$\lambda x.[\,|(x)]$
who	$\lambda P.\lambda Q.\lambda x.Q(x)\ ;\ P(x)$
dated	$\lambda P.\lambda x.(P(\lambda y.[\,|date(x, y)]))$
man	$\lambda x.[\,|man(x)]$
despises	$\lambda P.\lambda x.(P(\lambda y.[\,|despise(x, y)]))$
him	$\lambda P.[x_1|male(x_1), ant(x_1)]\ ;\ P(x_1)$

For now, $ant(x_1)$ is just an unspecified predicate that marks x_1 as an anaphoric discourse referent. The challenge will be to spell out this predicate in *a* convincing way, and this will be the task of section 5.4. For now we look at how the meaning of the sentence is built compositionally. In this section we do that using the abbreviated notation; the reader who is interested in how things work on the unabbreviated level can consult Appendix B.

The VP *despises him* composes straightforwardly as in (54).

(54)

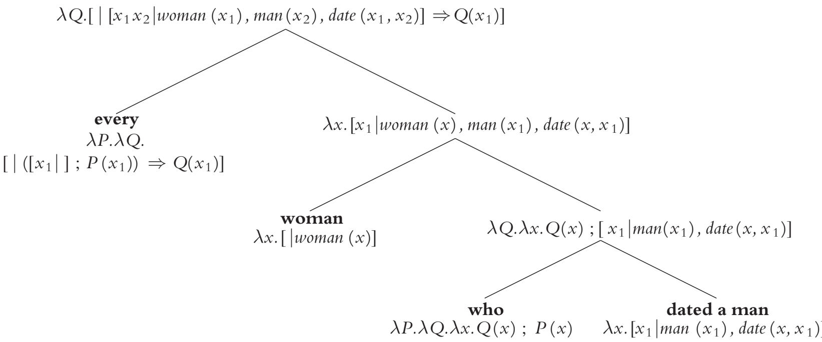
$\lambda x.[x_1|despise\ (x, x_1), male\ (x_1), ant\ (x_1)]$

despises $\lambda P.\lambda x.(P(\lambda y.[\,|despise\,(x, y)]))$

him $\lambda P.[x_1|\,male\,(x_1), ant(x_1)]\ ;\ P(x_1)$

(55) gives the composition of the subject NP *every women who dated a man*. (Since the relative clause VP composes analogously to the matrix VP, the composition has already been performed.)

(55)



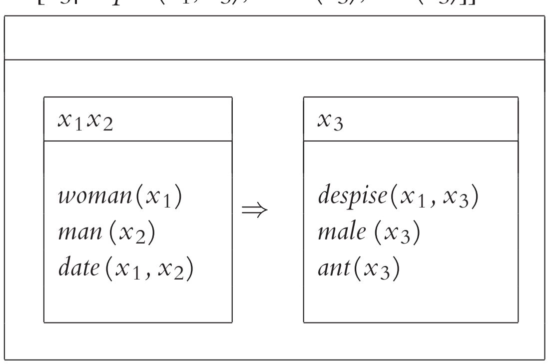
Observe that in the uppermost functional application, where **every** combines with its restrictor, we need to renumber x_1 in the second DRS to x_2, because we now index discourse referents relative to the first DRS in the sequencing.[28] Applying the subject meaning in (55) to the VP meaning in (54) gives us the sentence meaning in (56):

(56) $\lambda Q.[\,|[x_1x_2|woman(x_1),\ man(x_2),\ date(x_1,x_2)] \Rightarrow Q(x_1)]$
$(\lambda x.[x_1|despise(x,\ x_1),\ male(x_1),\ ant(x_1)])$

We need to renumber x_1 in the argument to x_3, the lowest index that is not already used in the accessibility path. [See the (89)–(90) in Appendix B for how this works in the unabbreviated language.] The expression then reduces to (57), shown in box notation in (58).

(57) $[\,|[x_1x_2|woman(x_1),\ man(x_2),\ date(x_1,x_2)] \Rightarrow$
$[x_3|despise(x_1,x_3),\ male(x_3),\ ant(x_3)]]$

(58)



Now if we sequence (52) and (58) we get (59).

[28] The interested reader can consult (87) in Appendix B to see what happens in the unabbreviated notation.

(59)

$$\begin{array}{|l|}\hline x_1 x_2 \\ \hline \mathit{Pedro}\,(x_1) \\ \mathit{bar}\,(x_2) \\ \mathit{in}\,(x_1, x_2) \\ \begin{array}{|l|}\hline x_3\ x_4 \\ \hline \mathit{woman}\,(x_3) \\ \mathit{man}\,(x_4) \\ \mathit{date}(x_3, x_4) \\ \hline \end{array} \Rightarrow \begin{array}{|l|}\hline x_5 \\ \hline \mathit{despise}(x_3, x_5) \\ \mathit{male}\,(x_5) \\ \mathit{ant}\,(x_5) \\ \hline \end{array} \\ \hline \end{array}$$

The final sentence of (47) straightforwardly gets the representation in (60).

(60)

$$\begin{array}{|l|}\hline x_1 \\ \hline \mathit{sad}\,(x_1) \\ \mathit{ant}\,(x_1) \\ \hline \end{array}$$

We then sequence this with the representation of the third sentence (60). Observe that in this sequencing operation x_1 gets rewritten as x_3, even though x_3 already occurs in an embedded DRS. The sequencing rule only calls for renumbering discourse referents starting with the first one not in the main universe of the first DRS. This will in fact help us avoid unwanted accessibility. We therefore get (61), where we have labelled the DRSs for convenience in the following discussion.

(61)

$$o: \begin{array}{|l|}\hline x_1\ x_2\ x_3 \\ \hline \mathit{Pedro}\,(x_1) \\ \mathit{bar}\,(x_2) \\ \mathit{in}\,(x_1, x_2) \\ k: \begin{array}{|l|}\hline x_3\ x_4 \\ \hline \mathit{woman}(x_3) \\ \mathit{man}\,(x_4) \\ \mathit{date}\,(x_3, x_4) \\ \hline \end{array} \Rightarrow l: \begin{array}{|l|}\hline x_5 \\ \hline \mathit{despise}\,(x_3, x_5) \\ \mathit{male}\,(x_5) \\ \mathit{ant}(x_5) \\ \hline \end{array} \\ \mathit{ant}(x_3) \\ \mathit{male}\,(x_3) \\ \mathit{sad}(x_3) \\ \hline \end{array}$$

(61) is certainly *a* plausible representation of the discourse, but we obviously need to make sense of the *ant*-predicate. To do that properly, we need an account of accessibility. Observe that (61) gives an intuitive representation of accessibility, not only across DRSs (as in DRT) but also inside them: we follow the accessibility path as defined in section 5.2, $o < \mathrm{k} < \mathrm{l}$, and always choose the first instance of a register name. Thus x_5 in l has access to x_4 and x_3 in k, and to x_2 and x_1 in o, but not to x_3 in o, since there is already an x_3 in the accessibility path. Inside boxes, accessibility is determined by the order of discourse referents: x_1 and x_2 are accessible to x_3. If the discourse is extended, there will be subsequent referents x_4 and so on in o, but these will be inaccessible to x_3 (both in l and o). This means that unlike in DRT, it is possible to sequence two boxes where one of them contains an anaphoric discourse referent without changing the accessible referents of the unresolved anaphors.

How does this notion of accessibility in the abbreviations play out in the full representations? Consider the full representation of (61) in (62). All the conditions except *ant* have been removed, since we are only interested in these. Observe that in the full representations, *ant* is evaluated in the output state of the DRS in which it occurs, just like other conditions. This is why *ant* takes both a state argument and a register.

(62) $\lambda i.\lambda o.\partial(i[x_{i_1} x_{i_2} x_{i_3}]o) \wedge ant(o)(x_{i_3}) \wedge \forall k.o[x_{i_3}\ x_{i_4}]k \rightarrow \exists l.(k[x_{i_5}]l \wedge ant(l)(x_{i_5}))$

This is a promising representation. We see that the model theoretic characterization of the accessibility relation between referents in DRT carries over to partial CDRT, unlike what happens in classical CDRT. For example, the discourse referents x_{i_3}, x_{i_4} and x_{i_5} that are introduced in the conditional are not accessible in the global DRS and this is reflected in the fact that k and l are allowed to differ from o (the main DRS) wrt. the inhabitants of these registers. On the other hand, x_{i_1} and x_{i_2} introduced in o are accessible inside the conditional and this is reflected in that the fact that k, l and o have the same inhabitants for these registers, as the conditions $o[x_{i_3}\ x_{i_4}]k$ and $k[x_{i_5}]l$ make sure. Put another way, the order on discourse referents gives us the accessibility relation, as long as we make sure that we interpret both registers *in the same state*.

5.4 *Interpreting anaphora*

How can we best represent anaphoric relations? We saw in section 5.1 that because our object language quantifies over contexts, it is problematic to think of anaphora as discourse referents picking antecedents in context. Instead, we will return to the simple idea that underlies the coindexation approach, namely that anaphoric relations are just relations between *words*. On the other hand, we want to abandon the idea that these relations are pre-semantic or part of the monotonic content, so we should be able to give semantic representations of discourses with unresolved anaphoric expressions—waiting, as it were, for an anaphoric resolution (which will always remain 'hypothetical', liable to change as other non-monotonic content) to provide the reference of the anaphoric expressions. However, since we want a semantic characterization of accessibility, we do want the monotonic content to be able to put constraints on the resolution. A full interpretation of a discourse, then, will be a tuple $\langle K, P \rangle$ where K is a DRS, including among other things a proper semantic characterization of accessibility, and P is a set of pragmatic enrichments of the interpretation, containing an anaphoric resolution $\mathcal{R}$, among other things such as Gricean inferences, etc.

If anaphoric relations hold between words, we can represent $\mathcal{R}$ as a function mapping integers to integers. These integers will arise as indices on linguistic tokens in the syntax. The indexation we require is entirely uncontroversial: it only serves to 'name' words and so keep linguistic tokens apart. For example, a stretch of discourse may contain several instances of *she*, which will be she_x and she_y where $x \neq y$, irrespective of whether they are interpreted as coreferent or not: we only require the syntax to be able to tell them apart. This is certainly not an unreasonable assumption, and it is one which is explicitly embodied in some syntactic frameworks, for example through the uniqueness of PRED features in Lexical-Functional Grammar.

Even if we model anaphora as a relation between words, we want our semantic language to able to talk about the referents of these words, and to do that, we need access to the registers they introduce. As we see from (61), there is no one-to-one relationship between words and registers. For example, x_3 is declared in the universe of k, where it corresponds to *a woman*, but also in the universe of o, where it corresponds to *he*. However, as long as we confine our attention to a single DRS (or a single state in the unabbreviated language), there *is* a one-to-one relationship between words and registers.

In the formal language, we can capture this by letting our models include a set of partial injections $\mathcal{I}_s$, one for each state s, mapping registers inhabited in s to the index of the word that introduced that register and uninhabited registers to the undefined integer. Since these are injections,[29] the set of $\mathcal{I}_s^{-1}$ are also (partial) functions mapping each

[29] Notice that things may be different when we consider examples such as Muskens' *Bill and Sue own a donkey* (= (6)). The distributive NP coordination is translated as in (i) and the VP as in (ii).

word that introduces a register in state s to that register. We can use these injections to transform our anaphoric relations between *words* into relations between *discourse referents*.

Consider the indexed version of the discourse in (47), given in (63).

(63) Pedro$_1$ is$_2$ in$_3$ a$_4$ bar$_5$. Every$_6$ woman$_7$ who$_8$ ever$_9$ dated$_{10}$ a$_{11}$ man$_{12}$ despises$_{13}$ him$_{14}$. He$_{15}$ is$_{16}$ sad$_{17}$.

Consulting the DRS in (61), we see that $\mathcal{I}_o(1) = x_1$, $\mathcal{I}_o(4) = x_2$. x_3 is introduced by different words in different states, so both $\mathcal{I}_o(15)$ and $\mathcal{I}_k(6)$ gives us x_3 (and inversely, $\mathcal{I}_o^{-1}(x_3) = 15$, but $\mathcal{I}_k^{-1}(x_3) = 6$).

Let us get back to anaphoric resolution. Consider the resolution of (63) that we see in (64), representing an interpretation where $\mathcal{R}(14) = 11$ and $\mathcal{R}(15) = 1$.

(64) $\boxed{\text{Pedro}_1}$ is$_2$ in$_3$ $\boxed{\text{a}_4}$ bar$_5$. $\boxed{\text{Every}_6}$ woman$_7$ who$_8$ ever$_9$ dated$_{10}$ $\boxed{\text{a}_{11}}$ man$_{12}$ despises$_{13}$ $\boxed{\text{him}_{14}}$ $\boxed{\text{He}_{15}}$ is$_{16}$ sad$_{17}$.

We can now 'translate' these dependencies between words into dependencies between registers. The basic idea is to use $\mathcal{I}_s$ to get from a register in a particular state to the index of the word which introduced it, then $\mathcal{R}$ to go from that index to the antecedent index, and then $\mathcal{I}_s^{-1}$ to get to the register corresponding to the antecedent index. The composed function $\mathcal{I}_s \circ \mathcal{R} \circ \mathcal{I}_s^{-1}$ (which we will abbreviate $\mathcal{A}(s)(x)$) will take us from the anaphoric register x in state s to its antecedent in that same state.

Figure 1 gives a pictorial representation of how this works. o is the output state of the main DRS in (61), k and l are the output states of the antecedent and the consequent of the conditional respectively. The solid lines represent $\mathcal{R}$, the function from anaphor indices to antecedent indices. The figure also shows the functions $\mathcal{I}_l$ (dotted) and $\mathcal{I}_o$ (dash-dotted), mapping registers in states l and o to words. Consider now

(i) $\lambda P.[x_1 x_2 | bill(x_1),\ sue(x_2)]\ ;\ P(x_1)\ ;\ P(x_2)$
(ii) $\lambda x.[x_3 | donkey(x_3),\ owns(x, x_3)]$.

Combining these yields (iii)

(iii) $[x_1 x_2 | bill(x_1),\ sue(x_2)];\ [x_3 | donkey(x_3),\ owns(x_1, x_3)];\ [x_3 | donkey(x_3),\ owns(x_2, x_3)]$

The second instance of the multiply declared referent x_3 will have to be renamed to x_4 and we get the correct result. But in such cases, $\mathcal{I}_s^{-1}$ mapping words (indices) to the referents they introduce in state s will no longer be a function, since *a donkey* introduces both x_3 and x_4. It is possible that this could be useful in dealing with distributive anaphora such as in *They keep it in the barn*, where *it* refers to both donkeys [see Krifka (1996), Nouwen (2003), Nouwen (2007) for discussion and analysis of these and similar cases]. We leave this for further research.

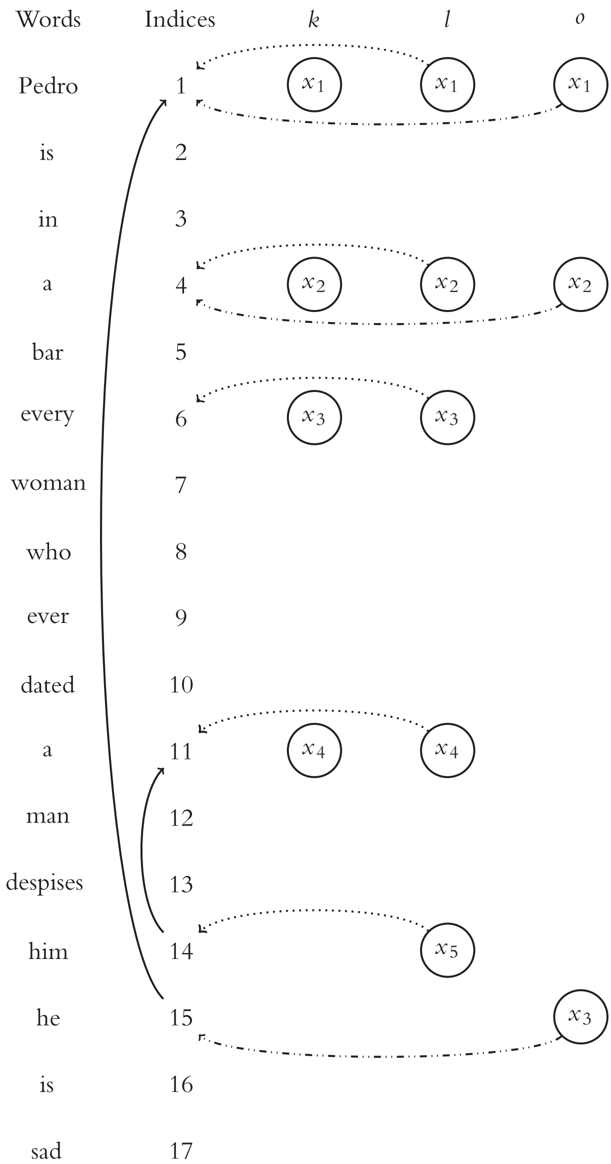


Figure 1 Relationship between words, states and registers.

the anaphoric relation between him_{14} and a_{11} *man*. $\mathcal{I}_l$ takes us from x_5 to 14 (*him*). Then $\mathcal{R}$ takes us from 14 to 11 (*a man*). Finally, $\mathcal{I}_l^{-1}$ takes us from 11 to x_4. In other words, $\mathcal{R}(14) = 11$ leads to an anaphoric relationship between x_5 and x_4 in state l; in terms of our formalism, $\mathcal{A}(l)(x_5) = x_4$.

With this much in place, we can return to the so far undefined predicate *ant*(s)(x). What we want *ant* to do is to restrict $\mathcal{R}$ to accessible antecedents, that is we want to make sure that, given some resolution $\mathcal{R}$, $\mathcal{A}(s)(x)$ is accessible to x in s and corefers with x in s. Formally, we define *ant* as an abbreviation of (65).

(65) $\nu(s)(x) = \nu(s)(\mathcal{A}(s)(x)) \wedge \mathcal{A}(s)(x) < x$

In other words, *ant*(s)(x) is true iff in state s register x (the anaphor) has an inhabitant that is identical with the inhabitant of the antecedent register,[30] which we can find by following $\mathcal{I}$ to get the index of the anaphoric word, then $\mathcal{R}$ to get the index of the antecedent word, then $\mathcal{I}^{-1}$ to find the register corresponding to the antecedent word in state s. Moreover, *ant*(s)(x) requires the antecedent register to be lower than the anaphoric register. This gives us a dynamic effect: the antecedent must be present in the input state already. It also ensures that the anaphoric possibilities do not increase as the discourse is extended.

On the salient reading of this discourse, it is not possible for x_3 in state o to be bound to an antecedent inside the conditional, since there is no register corresponding to *every woman* or *a man* which is inhabited in o. So setting for example. So setting for example $\mathcal{R}(15) = 14$ does not make sense. This follows directly from our account. $\mathcal{I}_o$ will take us from x_3 to 15, and $\mathcal{R}$ will take us further to 14, but $\mathcal{I}_o^{-1}(14)$ is undefined.

However, there is another reading of our discourse which we cannot yet capture, namely the one where *him* in the consequent of the conditional is anaphorically related to *Pedro* in the first sentence, that is every woman who ever dated a man hates Pedro.[31] On that reading, it is also possible for *he* in the last sentence to be bound to *him* in the consequent of the conditional (66).

(66) $\boxed{\text{Pedro}_1}$ is$_2$ in$_3$ $\boxed{\text{a}_4}$ bar$_5$. $\boxed{\text{Every}_6}$ woman$_7$ who$_8$ ever$_9$ dated$_{10}$ $\boxed{\text{a}_{11}}$ man$_{12}$ despises$_{13}$ $\boxed{\text{him}_{14}}$ $\boxed{\text{He}_{15}}$ is$_{16}$ sad$_{17}$.

From a purely semantic point of view, we could represent (66) as in (67), where 15 is not dependent on 14, but directly on 1.

[30] To capture more complex 'bridging' anaphora, we could let the relation between the reference of the anaphor and the antecedent be supplied by the non-monotonic anaphoric resolution, with identity being just a default. We leave this for future research.

[31] This is the reading that is prompted by the alternative third sentence 'he is a well-known date crasher' in (7).

(67) $\boxed{\text{Pedro}_1}$ is$_2$ in$_3$ $\boxed{\text{a}_4}$ bar$_5$. $\boxed{\text{Every}_6}$ woman$_7$ who$_8$ ever$_9$ dated$_{10}$ $\boxed{\text{a}_{11}}$ man$_{12}$ despises$_{13}$ $\boxed{\text{him}_{14}}$ $\boxed{\text{He}_{15}}$ is$_{16}$ sad$_{17}$.

This is in a sense what stack-based approaches are forced to if they rely on anaphors picking antecedents in context, because only the value of 1 will be present in 15's context. Since 14 is also dependent on 1, 15 and 14 will be coreferent by transitivity of the identity relation. So the solution is semantically fine. But it cannot serve as a sound basis for reasoning about anaphoric relationships. Surely, part of what favours *him*$_{14}$ as the antecedent of *he*$_{15}$ is the short distance between the two words.

This means we need some way of making sure that $\mathcal{R}(15) = 14$ makes sense in case $\mathcal{R}(14) = 1$. We achieve this by recursively defining a function $\mathcal{R}^*$ from states and indices to indices as in (68).

(68)
$$\mathcal{R}^*(s)(i) \begin{cases} \mathcal{R}(i) & \text{if } \exists e.\nu(s)(\mathcal{I}_s^{-1}(\mathcal{R}(i))) = e \text{ is true} \\ \mathcal{R}^*(s)(\mathcal{R}(i)) & \text{if } \exists n.\mathcal{R}(i) = n \text{ is true} \\ \text{undefined} & \text{otherwise} \end{cases}$$

The idea is that $\mathcal{R}^*$ follows the $\mathcal{R}$-path until it meets an index that maps to a register which is inhabited in the given state. If it does not find one, it returns the undefined integer. $\mathcal{R}^*$ is always well-defined if we assume that the left context is finite (or alternatively, if we assume that there is an upper bound on how far back anaphoric resolution can reach). We see that if $\mathcal{R}(15) = 14$ and $\mathcal{R}(14) = 1$, then $\mathcal{R}^*(o)(15) = \mathcal{R}^*(o)(\mathcal{R}(15)) = \mathcal{R}^*(o)(14) = \mathcal{R}(14) = 1$. On the other hand, if $\mathcal{R}(14) = 11$, then setting $\mathcal{R}(15) = 14$ will leave $\mathcal{R}^*(o)(15)$ undefined.

We still model the output of anaphoric resolution as the function $\mathcal{R}$ from anaphors to antecedents, and we still define *ant* as in (65), but we use the derived function $\mathcal{R}^*$ to define $\mathcal{A}$, which takes us from an anaphoric register in a particular state s to its antecedent register, as in (69).

(69) $\mathcal{A}(s)(x) \equiv \mathcal{I}_s^{-1}(\mathcal{R}^*(s)(\mathcal{I}_s(x)))$

In this way, we can have the pragmatics establish anaphoric relationships between words, even in cases where the antecedent is embedded and not strictly speaking accessible to the anaphor, but where it 'scopes out' of its embedding by its own anaphoric relation.

5.5 *Truth and partiality*

In classical CDRT, mimicking DRT's notion of truth, a DRS K is true in a state i iff there is an output state o such that $K(i)(o)$ is true; and a

DRS is true simpliciter iff it is true for all input states in our model, i.e. if $\forall i.\exists o.K(i)(o)$ is true. But when dealing with anaphora, it is useful to consider another definition of truth, namely, that a DRS K is true simpliciter iff it is true in the empty context, i.e. $\exists o.K(s_\emptyset)(o)$.[32] Let us now consider what this means for a reduced variant of (61), containing only the first and last sentences *Pedro*$_1$ *is*$_2$ *in*$_3$ *a*$_4$ *bar*$_5$. *He*$_{15}$ *is*$_{16}$ *sad*$_{17}$, as in (70).

(70)

$$\begin{array}{|l|}\hline x_1\ x_2\ x_3 \\ \hline \\ pedro\,(x_1) \\ bar\,(x_2) \\ sad(x_3) \\ male\,(x_3) \\ ant(x_3) \\ \\ \hline \end{array}$$

(70) will be true iff (71) is true.

(71) $\exists o.\partial(s_\emptyset[x_{s_{\emptyset_1}}\ x_{s_{\emptyset_2}}\ x_{s_{\emptyset_3}}]o) \wedge pedro(\nu(o)(x_{s_{\emptyset_1}})) \wedge bar(\nu(o)(x_{s_{\emptyset_2}}))\wedge$
$sad(\nu(o)(x_{s_{\emptyset_3}})) \wedge male(\nu(o)(x_{s_{\emptyset_3}})) \wedge ant(o)(x_{s_{\emptyset_3}})$

What about the partiality of our language? So far, we have used partiality in the definition of states and in the definition of transitions, such as $\partial(i[x_{i_1}\ x_{i_2}\ x_{i_3}]o)$, which is built into the abbreviation rules in (41). This is necessary because we do not want states which manipulate registers in violation of the constraints on transitions to falsify the transition constraint, but to be nonsensical transition outputs.

Should we extend the use of partiality in our language? The question is most easily approached using familiar first-order formulae rather than the full representations. By the unselective binding lemma, which is proven in Appendix C, we can, given an anaphoric resolution,[33] reduce (71) to a standard first-order formula. Let us assume $\mathcal{R}(15) = 1$, so *he* and *Pedro* corefer. Then unselective binding tells us that (71) is equivalent to (72).

[32] Alternatively, if one also considers deictic expressions, it would be better to consider truth in a contextually defined state which includes objects that can be referred to without a previous linguistic introduction. We do not pursue this here.

[33] In order to keep the underdetermined resolution function in the translation to first-order logic, we need to consider a weakest precondition calculus. The weakest precondition calculus is also needed to translate DRSs with complex conditions (embedded DRSs) to first-order logic. See Appendix D for details.

(72) $\exists x_1.\exists x_2.\exists x_3.pedro(x_1) \wedge bar(x_2) \wedge sad(x_3) \wedge male(x_3) \wedge x_3 = x_1 \wedge 1 < 3$

The problem with this account is that the descriptive content of a referring expression, which is translated as a predicate applied to that referent, does not play a different role from other predicates in the sentence. According to (72), the discourse *Pedro is in a bar. He is sad.* is just as false when Pedro is not sad, as when Pedro is sad, but happens to be a woman; or when the anaphor does not corefer with its antecedent (i.e. $x_3 \neq x_1$); or when $\mathcal{R}$ violates accessibility constraints (as Appendix C shows, $1 < 3$ renders the accessibility constraint).

There is of course a long tradition going back to Frege (1892) and Strawson (1950) for assuming that whenever definite descriptions fail to refer (or even whenever there is presupposition failure), the sentence as a whole has no truth value. I take no stance on whether using truth value gaps is a correct approach to presupposition in general, but I think it is fruitful for failing anaphoric expressions. Intuitively, *he is sad* is false if the referent of *he* is not sad, whereas if there there is no appropriate referent for *he*, the sentence is somehow inappropriate.

If we want a semantics which treats presupposition failure as a truth value gap, then obviously conditions like $ant(i)(\delta)$ as well as the descriptive material associated with an anaphoric expressions must be interpreted within the scope of ∂. $ant(i)(\delta)$ requires that the antecedent is located in the preceding context; resolving the anaphor to an 'antecedent' that only becomes available in the following discourse, is just as much an anaphoric failure as resolving to an antecedent that makes the descriptive content of the anaphor false. Adopting this semantics of anaphora, we will have to revise (71) to (73).

(73) $\exists o.\partial(s_{\emptyset}[x_{s_{\emptyset_1}}\ x_{s_{\emptyset_2}}\ x_{s_{\emptyset_3}}]o) \wedge pedro(\nu(o)(x_{s_{\emptyset_1}})) \wedge bar(\nu(o)(x_{s_{\emptyset_2}})) \wedge$
$sad(\nu(o)(x_{s_{\emptyset_3}})) \wedge \partial(male(\nu(o)(x_{s_{\emptyset_3}})) \wedge ant(o)(x_{s_{\emptyset_3}}))$

6 CONCLUSION

Unresolved anaphors are dealt with in our system by the predicate *ant*, which is true of a register in a state iff the anaphoric resolution links it to a lower register in that state which has the same inhabitant, or in more DRT-like terms, iff there is a coreferent discourse referent in the available preceding context. This faithfully formalizes the intuition that it is part of the meaning of an anaphoric expression that it should have an antecedent. This is an improvement on Muskens (2011), who treats anaphoric discourse referents as variables over registers and is unable to state semantic constraints on the value of that variable: nothing in the semantics will prevent that variable to be spelled out as a brand new register. Our theory is also an obvious improvement on the other dynamic theories, which offer no model-theoretic characterization of unresolved anaphors at all.

Regarding **constraints on anaphora**, we have seen that DRT's semantic characterization of accessibility carries over to partial CDRT. None of the other dynamic theories achieve this, although FCS could do it if it was spelled out with a partial theory of types. As we just saw, the approach in Muskens (2011) cannot impose coreference at all in a semantic way; *a fortiori* it cannot impose constraints on coreference.

We also avoid the **overwrite problem** because we do not use free variables or constants to refer to registers, but instead rely on a function to pick out the next free register. This crucially relies on partiality as well as on the ordered nature of discourse referents.

Finally, partial CDRT gives us a strict **separation of monotonic and non-monotonic content**. At each stage, the discourse interpretation is a pair $\langle K, P\rangle$, where P is a set of pragmatic enrichments of K, including the resolution function $\mathcal{R}$. The outcome of extending a discourse with a new DRS K' is a new interpretation $\langle K\,;\,K', P'\rangle$, where; is of course a monotonic function, whereas the update from P to P' (and in particular, from $\mathcal{R}$ to $\mathcal{R}'$) can be non-monotonic and follow from reasoning over $K\,;\,K'$, because our setup lets us merge discourse interpretations without resolving anaphors.

In sum, then, partial CDRT avoids the problems of other compositional dynamic theories, while keeping the appearance of being just DRT with lambdas. The *ant* predicate can be thought of as a way of spelling out an interpretation of what it means for a discourse referent in DRT to be declared in the universe of a presuppositional DRS. And the solution to the overwrite problem treats discourse referents as bound variables, which corresponds to the intuition that *John loves* x_1 and *John loves* x_2 should not be different semantic objects.

Finally, we have at a number of occasions pointed to the possibility of extending partial CDRT with a theory of actual anaphoric resolution as non-monotonic reasoning about $\mathcal{R}$ with the monotonic contents of the discourse as (parts of) the premises. Since registers are object-language entities in CDRT, it would also be possible to extend our DRSs with predicates of registers and not only their inhabitants. In that way, we could for example differentiate between centered registers and background registers, to use an idea from centering theory revived by Bittner (2001, 2007) [see also Beaver (2004)]. Even though I believe that the stack-based foundation of Bittner's Logic of Change with Centering fails, as we saw in section 5.1, it should be possible to reconstruct many of the insights in partial CDRT, since both approaches build on Muskens' Logic of change. In this way, my approach can be extended with actual principles of anaphoric resolution. Partial CDRT already offers the machinery to state (defeasible) constraints such as 'prefer resolution to the most recent centered register'. Much work remains to be done in detecting and formalizing preferences for anaphoric resolution, but I believe that partial CDRT offers an attractive framework for this kind of research.

APPENDIX A: THE MERGING LEMMA

We need to prove that the equivalence in (45) holds. Let us unpack the abbreviations on the left side. The +-rule is defined for boxes, but for notational convenience we will use Γ'^{+} to refer to the conditions of a box after application of the +-rule.

(74) $[x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma]\ ;\ [x_\delta \ldots x_\varepsilon | \Gamma'_1, \ldots, \Gamma'_\zeta] \rightsquigarrow$
$\lambda i.\lambda o.\exists k.[x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma](i)(k) \wedge [x_\delta \ldots x_\varepsilon | {\Gamma'_1}^{+}, \ldots, {\Gamma'_\zeta}^{+}](k)(o) \rightsquigarrow$
$\lambda i.\lambda o.\exists k.\partial(i[x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}}]k \wedge \Gamma^*_1(k) \wedge \ldots \wedge \Gamma^*_\gamma(k) \wedge \partial(k[x_{k_1} \ldots x_{k_{\varepsilon+(1-\alpha)}}]o) \wedge$
${\Gamma'_1}^{+}{*}(o), \ldots, {\Gamma'_\zeta}^{+}{*}(o)$

On the right side of (45), the abbreviations unpack as follows:

(75) $[x_\alpha \ldots x_\beta x_{\beta+1} \ldots x_{\varepsilon+\beta+(1-\delta)} | \Gamma_1, \ldots, \Gamma_\gamma, {\Gamma'_1}^{\dagger}, \ldots, {\Gamma'_\zeta}^{\dagger}] \rightsquigarrow$
$\lambda i.\lambda o.\partial(i[x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}} x_{i_{(\beta+1)+(1-\alpha)}} \ldots x_{i_{\varepsilon+\beta+(1-\delta)+(1-\alpha)}}]o) \wedge \Gamma^*_1(o) \wedge \ldots \wedge$
$\Gamma^*_\gamma(o) \wedge {\Gamma'_1}^{\dagger *}(o) \wedge \ldots \wedge {\Gamma'_\zeta}^{\dagger *}(o)$

We need to prove that the unpacked representations in (74) (the left side) and (75) (the right side) are equivalent, provided there are no free variables in $[x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma]\ ;\ [x_\delta \ldots x_\epsilon | \Gamma'_1, \ldots, \Gamma'_\zeta]$.

Proof.
States i and k on the left side differ only in that k extends i with $x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}}$, so the registers $x_{k_1} \ldots x_{k_{\epsilon+(1-\delta)}}$ must be $s(x_{i_{\beta+(1-\alpha)}}) \ldots s^{\epsilon+(1-\delta)}(x_{i_{\beta+(1-\alpha)}})$, which we abbreviate $x_{i_{(\beta+1)+(1-\alpha)}} \ldots x_{i_{\epsilon+\beta+(1-\delta)+(1-\alpha)}}$. Hence the expression $\partial(i[x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}} x_{i_{(\beta+1)+(1-\alpha)}} \ldots x_{i_{\epsilon+\beta+(1-\delta)+(1-\alpha)}}]o)$ on the right side is equivalent to the conjunction of $\partial(i[x_{i_1} \ldots x_{i_{\beta+(1-\alpha)}}]k)$ and $\partial(k[x_{k_1} \ldots x_{k_{\epsilon+(1-\delta)}}]o)$ on the left side.

For conditions from the left box in the sequencing operations, we need to prove that $\Gamma^{*}(k)$ and $\Gamma^{*}(o)$ are always equivalent. There are two cases: 1. Γ^{*} is a simplex condition. Then $\Gamma^{*}(k)$ and $\Gamma^{*}(o)$ are equivalent iff k and o assign the same inhabitants to the registers in Γ^{*}. The *-rule replaces any bound discourse referent x_η in Γ by $x_{i_{\eta+(1-\alpha)}}$. k and o differ only with respect to inhabitants in register $x_{\beta+1}$ and higher, and we know that $\eta \leq \beta$ since x_η is bound. Hence $\Gamma^{*}(k)$ and $\Gamma^{*}(o)$ are equivalent. 2. Γ^{*} is a complex condition. Then k in $\Gamma^{*}(k)$ and o in $\Gamma^{*}(o)$ only occur in expressions of the form $k[x_{i_\eta}\ldots x_{i_\zeta}]\sigma$ and $o[x_{i_\eta}\ldots x_{i_\zeta}]\sigma$. The relation $[x_{i_\eta}\ldots x_{i_\zeta}]$ holds between two states iff they have the same inhabitants in all registers except $x_{i_\eta}\ldots x_{i_\zeta}$. We know that $x_{i_\eta} \leq x_{i_{\beta+(1-\alpha)+1}}$ since we have followed the convention of renumbering with the lowest possible index both in variable renumbering and in sequencing. Hence, k and o have the same inhabitants in all registers lower than x_{i_η}. In general, we do not know whether k and o have the same inhabitants in registers following x_{i_ζ}. But for all states σ, our axioms ensure there are states σ' that differ from σ only in having the same inhabitants as k or o in registers following x_{i_ζ}. Since registers following x_{i_ζ} do not occur in the complex condition (because they would be free), moving from σ to σ' does not change the interpretation of the box. Hence $\Gamma^{*}(k)$ and $\Gamma^{*}(o)$ are equivalent.

$\Gamma'^{+*}(o)$ and $\Gamma'^{\dagger *}(o)$ are equivalent iff applying the +-rule and then the *-rule always leads to the same result as applying the †-rule and then the *-rule. For a discourse referent x_η, there are two cases: 1. x_η is free in the second DRS but bound in sequencing from the universe of the first DRS. On the left side, the +-rule turns x_η into $x_{i_{\eta+(1-\alpha)}}$ and the *-rule does not apply. On the right side, the †-rule does not apply, but the *-rule turns x_η into $x_{i_{\eta+(1-\alpha)}}$. 2. x_η is bound in the second DRS. On the left side, the +-rule does not apply and the *-rule turns x_η into $x_{k_{\eta+(1-\delta)}}$. On the right side, the †-rule turns x_η into $x_{\eta+\beta+(1-\delta)}$, which the *-rule turns into $x_{i_{\eta+\beta+(1-\delta)+(1-\alpha)}}$. By the same reasoning as above, $x_{k_{\eta+(1-\delta)}}$ and $x_{i_{\eta+\beta+(1-\delta)+(1-\alpha)}}$ are equivalent. □

APPENDIX B: UNABBREVIATED DERIVATION OF *EVERY WOMAN WHO DATED A MAN DESPISES HIM*

Let us first compose the VP **despises(him)**. This is as in (76)

(76) $\lambda P.\lambda x.(P(\lambda y.\lambda i.\lambda o.\partial(i[\,]o) \wedge \mathit{despise}(\nu(o)(x), \nu(o)(y))))$
$(\lambda P'.\lambda i.\lambda o.\exists k.\partial(i[x_{i_1}]k) \wedge \mathit{male}(\nu(k)(x_{i_1})) \wedge \mathit{ant}(k)(x_{i_1}) \wedge P'(x_{i_1})(k)(o))$

(76) reduces to (77):

(77) $\lambda x.\lambda i.\lambda o.\exists k.\partial(i[x_{i_1}]k) \wedge \partial(male(\nu(k)(x_{i_1})) \wedge ant(k)(x_{i_1})) \wedge \partial(k[\ ]o) \wedge despise(\nu(o)(x), \nu(o)(x_{i_1}))$

By the merging lemma we get (78).

(78) $\lambda x.\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge \partial(male(\nu(o)(x_{i_1})) \wedge ant(o)(x_{i_1})) \wedge despise(\nu(o)(x), \nu(o)(x_{i_1}))$

We now compose the subject NP *every woman who dated a man*. **dated a man** is analogous to **despises him**, as in (79).

(79) $\lambda x.\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge man(\nu(o)(x_{i_1})) \wedge date(\nu(o)(x), \nu(o)(x_{i_1}))$

who(**dated a man**), then, will be as in (80).

(80) $\lambda P.\lambda Q.\lambda x.\lambda i.\lambda o.\exists k.Q(x)(i)(k) \wedge P(x)(k)(o)$
$(\lambda x.(\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge\ man(\nu(o)(x_{i_1})) \wedge date(\nu(o)(x), \nu(o)(x_{i_1}))))$

(80) reduces to (81).

(81) $\lambda Q.\lambda x.\lambda i.\lambda o.\exists k.Q(x)(i)(k) \wedge \partial(k[x_{k_1}]o) \wedge man(\nu(o)(x_{k_1})) \wedge date(\nu(o)(x), \nu(o)(x_{k_1}))$

which applies to **woman** in (82).

(82) $\lambda Q.\lambda x.\lambda i.\lambda o.\exists k.Q(x)(i)(k) \wedge \partial(k[x_{k_1}]o) \wedge man(\nu(o)(x_{k_1})) \wedge date(\nu(o)(x),\nu(o)(x_{k_1}))\ (\lambda x.\lambda i.\lambda o.\partial(i[\,]o) \wedge woman(\nu(o)(x)))$

This reduces as in (83).

(83) $\lambda x.\lambda i.\lambda o.\exists k.\partial(i[\ ]k) \wedge woman(\nu(k)(x)) \wedge \partial(k[x_{k_1}]o)\wedge man(\nu(o)(x_{k_1}))\ \wedge date(\nu(o)(x), \nu(o)(x_{k_1}))$

By the merging lemma we get (84).

(84) $\lambda x.\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge woman(\nu(o)(x)) \wedge man(\nu(o)(x_{i_1})) \wedge date(\nu(o)(x), \nu(o)(x_{i_1}))$

We now apply **every** to this in (85).

(85) $\lambda P.\lambda Q.\lambda i.\lambda o.\partial(i[\ ]o) \wedge \forall j.(\exists j'.\partial(o[x_{o_1}]j') \wedge P(x_{o_1})(j')(j))) \rightarrow \exists k.Q(x_{o_1})(j)(k)(\lambda x.\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge woman(\nu(o)(x))\wedge man(\nu(o)(x_{i_1})) \wedge\ date(\nu(o)(x), \nu(o)(x_{i_1}))) \wedge date(\nu(o)(x), \nu(o)(x_{i_1})))$

This reduces as in (86).

(86) $\lambda Q.\lambda i.\lambda o.\partial(i[\ ]o) \wedge \forall j.(\exists j'.\partial(o[x_{o_1}]j') \wedge \partial(j'[x_{j'_1}]j) \wedge woman(\nu(j)(x_{o_1})) \wedge man(\nu(j)(x_{j'_1})) \wedge date(\nu(j)(x_{o_1}), \nu(j)(x_{j'_1}))) \rightarrow \exists k.Q(x_{o_1})(j)(k)$

By the merging lemma, we get (87), which is the final meaning of the subject NP. Observe that after this sequencing, $x_{j'_1}$ comes out as x_{o_2}—this is what corresponds to the renaming of the second occurrence of x_1 in the uppermost functional application in the abbreviated (55).

(87) $\lambda Q.\lambda i.\lambda o.\partial(i[\,]o) \wedge \forall j.(\partial(o[x_{o_1}x_{o_2}]j) \wedge woman(\nu(j)(x_{o_1})) \wedge man(\nu(j)(x_{o_2})) \wedge date(\nu(j)(x_{o_1}), \nu(j)(x_{o_2}))) \rightarrow \exists k.Q(x_{o_1})(j)(k)$

If we combine the subject meaning in (87) and the VP meaning in (78), we get (88).

(88) $\lambda Q.\lambda i.\lambda o.\partial(i[\,]o) \wedge \forall j.(\partial(o[x_{o_1}x_{o_2}]j) \wedge woman(\nu(j)(x_{o_1})) \wedge man(\nu(j)(x_{o_2})) \wedge date(\nu(j)(x_{o_1}), \nu(j)(x_{o_2}))) \rightarrow \exists k.Q(x_{o_1})(j)(k)$
$(\lambda x.\lambda i.\lambda o.\partial(i[x_{i_1}]o) \wedge \partial(male(\nu(o)(x_{i_1})) \wedge ant(o)(x_{i_1})) \wedge despise(\nu(o)(x), \nu(o)(x_{i_1})))$

This reduces to (89).

(89) $\lambda i.\lambda o.\partial(i[\,]o) \wedge \forall j.(\partial(o[x_{o_1}x_{o_2}]j) \wedge woman(\nu(j)(x_{o_1})) \wedge man(\nu(j)(x_{o_2})) \wedge date(\nu(j)(x_{o_1}), \nu(j)(x_{o_2}))) \rightarrow (\exists k.\partial(j[x_{j_1}]k) \wedge despise(\nu(k)(x_{o_1}), \nu(k)(x_{j_1})) \wedge \partial(male(\nu(k)(x_{j_1})) \wedge ant(k)(x_{j_1})))$

Observe that since j only extends o with x_{o_1} and x_{o_2}, and k extends j with x_{j_1}, it follows that $x_{j_1} = x_{o_3}$. This is what corresponds to the renaming between (56) and (57). In other words, we can rewrite (89) as (90) to make the order of discourse referents more transparent.

(90) $\lambda i.\lambda o.\partial(i[\,]o) \wedge \forall j.(\partial(o[x_{o_1}x_{o_2}]j) \wedge woman(\nu(j)(x_{o_1})) \wedge man(\nu(j)(x_{o_2})) \wedge date(\nu(j)(x_{o_1}), \nu(j)(x_{o_2}))) \rightarrow (\exists k.\partial(j[x_{o_3}]k) \wedge despise(\nu(k)(x_{o_1}), \nu(k)(x_{o_3})) \wedge male(\nu(k)(x_{o_3})) \wedge ant(k)(x_{o_3}))$

This is indeed equivalent to the abbreviated version in (58).

APPENDIX C: UNSELECTIVE BINDING

The intuition behind unselective binding is that the whole machinery with states and registers does not matter as far as truth is concerned: if our model makes it possible to satisfy the DRS conditions on the domain D_e, then our axioms guarantee that there is a state which assigns exactly the right inhabitants to registers $u_1, \ldots, u_n$. In other words, unselective binding lets us move from states and registers to entities.

Let $u_1, \ldots, u_n$ be terms of type π_e, $\mathcal{N}$ be a strictly monotonic function from $u_1, \ldots, u_n$ to the integers $1, \ldots, n$ (i.e. $\mathcal{N}(u_x) < \mathcal{N}(u_y) \leftrightarrow$

$u_x < u_y$), $x_1, \ldots, x_n$ be distinct variables of type e, ϕ be a formula which does not contain $x_1, \ldots, x_n$ or $\mathcal{R}$ (i.e. unselective binding is relative to an anaphoric resolution, so that we can substitute for all occurrences of $\mathcal{R}$ in the original formula) and write ϕ^* for the formula which results from ϕ by simultaneously substituting x_1 for $\nu(i)(u_1)$ and ... and x_n for $\nu(i)(u_n)$ and $\mathcal{N}(u_x) < \mathcal{N}(u_y)$ for $u_x < u_y$, then

(91) $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi \ = \ \exists x_1 \ldots \ \exists x_n.\phi^*$

Proof.
We first prove prove that if $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ has a classical truth value then $\exists x_1 \ldots \ \exists x_n.\phi^*$ has the same truth value.

Suppose $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ is true. From the monotonicity of $\mathcal{N}$ it follows that the substitution of $\mathcal{N}(u_x) < \mathcal{N}(u_y)$ for $u_x < u_y$ is truth preserving. Since ϕ is true, there is a state which makes ϕ true and is inhabited in exactly $u_1, \ldots, u_n$. Call the inhabitants of these registers $e_1, \ldots, e_n$. Clearly $\nu(o)(u_1) = e_1$ and so on and we can replace $\nu(o)(u_1)$ by e_1. It follows by existential generalization that $\exists x_1 \ldots \exists x_n\phi^*$ is true.

Suppose now $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ is false. Then at least one state inhabited in exactly $u_1, \ldots, u_n$ makes ϕ false, and no state inhabited in exactly $u_1, \ldots, u_n$ makes ϕ true.

But suppose for contradiction that $\exists x_1 \ldots \ \exists x_n.\phi^*$ is true. Call the witnesses for $x_1, \ldots, x_n$ $e_1, \ldots, e_n$. Now by an induction over states using AX1 and AX2 there is a state which is inhabited by e_1 in u_1 and ... and e_n in u_n. This state is inhabited in exactly $u_1, \ldots, u_n$ and it makes ϕ true as we see by a reasoning similar to that above. Contradiction. Suppose again for contradiction that $\exists x_1 \ldots \ \exists x_n.\phi^*$ is #. Then ϕ is undefined for all witnesses. It follows that we cannot construct a state inhabited in exactly $u_1, \ldots, u_n$ which makes ϕ false. Contradiction. We now prove that if $\exists x_1 \ \ldots \ \exists x_n\phi^*$ has a classical truth value then $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ has the same truth value. Consider first the case where $\exists x_1 \ldots \exists x_n\phi^*$ is true. The reasoning is as above. Call the witnesses for $x_1, \ldots, x_n$ $e_1, \ldots, e_n$. There is a state which is inhabited by e_1 in u_1 and ... and e_n in u_n and this state makes both ϕ and $\partial(s_{\emptyset}[u_1, \ldots, u_n]o)$ true. Hence $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ is true.

Finally, consider the case where $\exists x_1 \ldots \exists x_n\phi^*$ is false. Then, by a reasoning as above, at least one state inhabited in exactly $u_1, \ldots, u_n$ makes ϕ false, and no state inhabited in exactly $u_1, \ldots, u_n$ makes ϕ true. So $\exists o.\partial(s_{\emptyset}[u_1, \ldots, u_n]o) \wedge \phi$ is false. □

Observe that in the case where $\exists x_1 \ldots \exists x_n \ \phi^*$ is false, there will be states that are inhabited $u_1, \ldots, u_n$ and u_{n+1}. All of these will make

$\partial(s_\emptyset[u_1, \ldots, u_n]o)$ undefined and hence $\partial(s_\emptyset[u_1, \ldots, u_n]o) \wedge \phi$ undefined. But by the semantics of quantification defined in section 4.3, $\exists o.P(o)$ is undefined just in case P is undefined for *all* values of o and this is not the case here, as shown in the proof.

APPENDIX D: THE WEAKEST PRECONDITION CALCULUS

Muskens (1996) uses a variant of Hoare logic (first adapted for linguistic purposes in Van Eijck and de Vries 1992) to produce truth conditions for complex DRSs that are expressible as standard first-order formulae, or in other words to provide an embedding of CDRT into first-order logic.

There are two parts to this embedding, a one-place function **tr** which sends conditions to first-order formulae, and a two-place function (weakest precondition), which maps pairs of a box and a first-order formula to a first-order formula. The idea is that $\mathbf{wp}(K,\chi)$ gives us the first-order formula that must hold in the input state i for us to be able to process K and end up in a state where χ holds. So $\mathbf{wp}(K,\top)$ gives the truth conditions of K, and as we shall see, they are identical with the truth definition from section 5.5.

For our purposes, a one-place function **tr** will not quite do, since we need to keep apart multiple occurrences of the same register in several DRSs when we assign antecedents. To achieve this, we assign a unique label n to each DRS K to be translated, and let **tr** be a two-place function from terms/conditions and DRS labels to first-order terms/formulae. To deal with anaphoric resolution, we introduce a set of resolutions functions $\mathcal{R}_n$ in our first-order representations, one for each DRS K_n. These functions will map integers (variable indices) to integers.

Let $x_\alpha \ldots x_\beta$ be variables for registers and $y_\alpha \ldots y_\beta$ variables for individuals. Then we can define **tr** and **wp** as in (92). Notice that p, the DRS label argument to **tr**, is only relevant for translating ant conditions.

(92)
$$
\begin{array}{lcl}
\mathbf{tr}(R(x_\alpha \ldots x_\beta), p) & = & R(y_\alpha \ldots y_\beta) \\
\mathbf{tr}(ant(x_\alpha), p) & = & y_\alpha = (y_{\mathcal{R}_p(\alpha)}) \wedge \mathcal{R}_p(\alpha) < \alpha \\
\mathbf{tr}(\partial(\Gamma), p) & = & \partial(\mathbf{tr}(\Gamma, p)) \\
\mathbf{tr}(\neg K, p) & = & \neg\mathbf{wp}(K, \top) \\
\mathbf{tr}(K \vee L, p) & = & \mathbf{wp}(K, \top) \vee \mathbf{wp}(L, \top) \\
\mathbf{tr}(K \Rightarrow L, p) & = & \neg\mathbf{wp}(K, \neg\mathbf{wp}(L, \top)) \\
\mathbf{wp}([x_\alpha \ldots x_\beta | \Gamma_1, \ldots, \Gamma_\gamma]_p, \chi) & = & \exists y_\alpha \ldots \exists y_\beta.(\mathbf{tr}(\Gamma_1, p) \wedge \ldots \mathbf{tr}(\Gamma_\gamma, p) \wedge \chi) \\
\mathbf{wp}(K; L, \chi) & = & \mathbf{wp}(K, \mathbf{wp}(L, \chi))
\end{array}
$$

Let us apply these to (93), which is the labelled box representation from (61) with ∂ added to the anaphoricity conditions as per the discussion in section 5.5.

(93)
$$o:\ \begin{array}{|l|}\hline x_1\ x_2\ x_3\\ \hline \mathit{Pedro}\,(x_1)\\ \mathit{bar}\,(x_2)\\ \mathit{in}\,(x_1,x_2)\\ k:\begin{array}{|l|}\hline x_3\ x_4\\ \hline \mathit{woman}\,(x_3)\\ \mathit{man}\,(x_4)\\ \mathit{date}(x_3,x_4)\\ \hline\end{array}\ \Rightarrow\ l:\begin{array}{|l|}\hline x_5\\ \hline \mathit{despise}\,(x_3,x_5)\\ \partial(\mathit{male}\,(x_5))\\ \partial(\mathit{ant}(x_5))\\ \hline\end{array}\\ \partial(\mathit{ant}(x_3))\\ \partial(\mathit{male}\,(x_3))\\ \mathit{sad}\,(x_3)\\ \hline\end{array}$$

By applying the DRS rule to the main DRS and **tr** to all of its conditions, we get (94):

(94) $\exists y_1.\exists y_2.\exists y_3.\mathit{Pedro}(y_1) \wedge \mathit{bar}(y_2) \wedge \mathit{in}(y_1, y_2) \wedge \neg\mathbf{wp}([x_3x_4|\mathit{woman}(x_3),$ $\mathit{man}(x_4), \mathit{date}\ (x_3, x_4)]_k, \neg\mathbf{wp}([x_5|\mathit{despise}(x_3, x_5), \partial(\mathit{male}(x_5)), \partial(\mathit{ant}(x_5))]_l),$ $\top) \wedge \partial(y_3 = y_{\mathcal{R}_o(3)} \wedge \mathcal{R}_1(3) < 3) \wedge \partial(\mathit{male}(y_3)) \wedge \mathit{sad}(y_3)$

Resolving $\mathbf{wp}([x_5|\mathit{despise}(x_3,x_5), \partial(\mathit{male}(x_5)), \partial(\mathit{ant}(x_5))]_l), \top)$ gives us (95).

(95) $\exists y_5.\mathit{despise}(y_3, y_5) \wedge \partial(\mathit{male}(y_5)) \wedge \partial(y_5 = y_{\mathcal{R}_l(5)} \wedge \mathcal{R}_l(5)5)$

This gives us (96) for the entire embedded conditional.

(96) $\neg(\exists y_3.\exists y_4.(\mathit{woman}(y_3) \wedge \mathit{man}(y_4) \wedge \mathit{date}(y_3, y_4) \wedge$ $\neg(\exists y_5.(\mathit{despise}(y_3, y_5) \wedge \partial(\mathit{male}(y_5)) \wedge \partial(y_5 = y_{\mathcal{R}_l(5)} \wedge \mathcal{R}_l(5)5)))))$

(96) is identical to (97).

(97) $\forall y_3.\forall y_4.(\mathit{woman}(y_3) \wedge \mathit{man}(y_4) \wedge \mathit{date}(y_3, y_4) \rightarrow \exists y_5.(\mathit{despise}(y_3, y_5) \wedge$ $\partial(\mathit{male}(y_5)) \wedge \partial(y_5 = y_{\mathcal{R}_l(5)} \wedge \mathcal{R}_l(5)5)))$

So for the entire DRS we get (98):

(98) $\exists y_1.\exists y_2.\exists y_3.Pedro(y_1) \wedge bar(y_2) \wedge in(y_1, y_2) \wedge \forall y_3.\forall y_4.(woman(y_3) \wedge man(y_4) \wedge date(y_3, y_4) \rightarrow \exists y_5.(despise(y_3, y_5) \wedge \partial(male(y_5)) \wedge \partial(y_5 = y_{\mathcal{R}_l(5)} \wedge \mathcal{R}_l(5)5))) \wedge \partial(y_3 = y_{\mathcal{R}_o(3)} \wedge \mathcal{R}_o(3)3) \wedge \partial(male(y_3)) \wedge sad(y_3)$

To calculate the truth value, we need to resolve the anaphors $\mathcal{R}_l(5)$ and $\mathcal{R}_o(3)$. Notice that there is no way the final anaphor can be bound to one of the variables inside the translation of the conditional, since the anaphor is outside the scope of the quantifiers that bind these variables. So if we tried to set $\mathcal{R}_o(3) = 4$ (which of course does not satisfy $\mathcal{R}_o(3) < 3$ but that is immaterial here), the resulting y_4 will be a free variable. This reflects our purely model-theoretic characterization of accessibility: accessibility in our lambda terms turns up as quantifier scopings in the translations to first-order logic, and coindexations that violate accessibility leads to free variables in the translations.

If $\mathbf{wp}(K,\top)$ is a correct truth calculation for K, then it should have the same truth value as $\exists s.K(s_\emptyset)(s)$, provided that the set of anaphoric resolution functions that occurs in the translation is congruent with the resolution $\mathcal{R}$ which occurs in the original. We can define congruence by exploiting the one-to-one mapping from DRS labels in the translation to DRS output states in the original. Writing i indiscriminately for DRS labels and for states, we say that the set $\{\mathcal{R}_\alpha \ldots \mathcal{R}_\beta\}$ is congruent with $\mathcal{R}$ iff for all labels $i \in \{\alpha \ldots \beta\}$ and for all n,[34]

(99) $x_{\mathcal{R}_i(n)} = \mathcal{A}(i)(x_n)$

In words, if $\mathcal{A}$ resolves the antecedent of the register x_n in state i to register x_m, then a congruent $\mathcal{R}_i$ should map n to m.

We now show $\mathbf{wp}(K, \top) \leftrightarrow \exists\, s.K(s_\emptyset)(s)$ for congruent resolutions.

Proof.

Notice first that, apart from the obvious rule for ∂, the only substantial difference between (92) and the setup in Muskens (1996) is the rule for $ant(x_n)$. For the other cases, the DRS label p is eliminable so that Muskens' proof carries over.

[34] To avoid clutter, we will here and in the remainder of this appendix use x_n for the n'th free register in $s_\emptyset$ (i.e. what we have previously written as $x_{s_{\emptyset_n}}$).

For any formula ϕ and state variable i let $(\phi)^i$ denote the result of replacing each free individual variable y_k in ϕ with $\nu(i)(x_k)$. We prove that $\mathbf{tr}(\mathrm{ant}(x_n), i)^i$ is equivalent to $ant(x_n)(i)$:

(100)
$$\begin{array}{lll}
\mathbf{tr}(ant(x_n), i)^i & \leftrightarrow & \text{by the definition of } \mathbf{tr} \\
(y_n = (y_{\mathcal{R}_i(n)}) \wedge \mathcal{R}_i(n) < n)^i & \leftrightarrow & \text{by the definition of } (\phi)^i \\
(\nu(i)(x_n) = \nu(i)(x_{\mathcal{R}_i(n)}) \wedge \mathcal{R}_i(n) < n & \leftrightarrow & \text{by congruence} \\
(\nu(i)(x_n) = \nu(i)(\mathcal{A}(i)(x_n)) \wedge \mathcal{R}_i(n) < n & \leftrightarrow & \text{by the successor function} \\
(\nu(i)(x_n) = \nu(i)(\mathcal{A}(i)(x_n)) \wedge x_{\mathcal{R}_i(n)} < x_n & \leftrightarrow & \text{by congruence} \\
(\nu(i)(x_n) = \nu(i)(\mathcal{A}(i)(x_n)) \wedge \mathcal{A}(i)(x_n) < x_n & \leftrightarrow & \text{by the definition of } ant \\
ant(i)(x_n). & &
\end{array}$$

From this and the proof in Muskens (1996), it follows that $\mathbf{wp}(K, \top) \leftrightarrow \exists\, s.K(s_\emptyset)(s)$ for congruent resolutions. □

Acknowledgements

I would like to thank Gianluca Giorgolo, Corien Bary and Kjell Johan Sæbø for having read and commented upon earlier versions of this article, and Jan Tore Lønning for help with references on partial type theory. I also thank the reviewers and the editor Rick Nouwen for useful comments. This research was supported by the Norwegian Research Council [YFF grant 180632].

DAG TRYGVE TRUSLEW HAUG
Department of Philosophy, Classics, History of Arts and Ideas,
PO Box 1020 Blindern N-0315 Oslo Norway,
e-mail: daghaug@ifikk.uio.no

REFERENCES

Asher, N. & A. Lascarides. (2003), *Logics of Conversation*. Cambridge University Press. Cambridge.

Beaver, D. (1992), 'The kinematics of presupposition'. In P. Dekker & M. Stokhof (eds.), In *Proceedings of the Eight Amsterdam Colloquium*. ILLC. Amsterdam. 17–36.

Beaver, D. (2002), 'Pragmatics and that's an order'. In D. Barker-Plummer, D. Beaver, J. van Benthem & P. S. di Luzio (eds.), *Logic, Language and Visual Information*. CSLI Publications. Stanford. 191–215.

Beaver, D. (2004), 'The optimization of discourse anaphora'. *Linguistics and Philosophy* **27:**3–56.

Bittner, M. (2001), 'Surface composition as bridging'. *Journal of Semantics* **18:** 127–177.

Bittner, M. (2007), 'Online update: Temporal, modal and de se anaphora in polysynthetic discourse'. In C. Barker & P. Jacobson (eds.), *Direct*

Compositionality. Oxford University Press. Oxford. 363–404.

Blamey, S. (1995), 'Partial logic'. In D. Gabbay & F. Guenther (eds.), *Partial Logic*. D. Reidel. Dordrecht. 1–70.

Cann, R., R. Kempson & L. Marten. (2005), *The Dynamics of Language: an Introduction,* Vol. 35 of *Syntax and Semantics*. Elsevier. Amsterdam.

Dalrymple, M. (ed.) (1999), *Semantics and Syntax in Lexical Functional Grammar*. MIT Press. Cambridge, MA.

de Groote, P. (2006), 'Towards a Montagovian account of dynamics'. In M. Gibson & J. Howell (eds.), In *Proceedings of SALT XVI*. Cornell University. Ithaca, NY. 1–16.

Dekker, P. (1994), 'Predicate logic with anaphora'. In *Proceedings of SALT IV Lynn Santelmann and Mandy*. DMLL Publications, Cornell University. 79–95.

Fox, D. (2008), 'Two short notes on Schlenker's theory of presupposition projection'. *Theoretical Linguistics* **34:** 237–52.

Frege, G. (1892), 'Über Sinn und Bedeutung'. *Zeitschrift für Philosophie und philosophische Kritik* **100:**25–50.

Geurts, B. & D. Beaver. (2011), 'Discourse representation theory'. In E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (http://plato.stanford.edu/archives/fall2011/entries/discourse-representation-theory) last accessed 21 July 2013.

Groenendijk, J. & M. Stokhof. (1991), 'Dynamic predicate logic'. *Linguistics and Philosophy* **14:**39–100.

Heim, I. (1982), 'On the Semantics of Definite and Indefinite Noun Phrases'. *Ph. D. thesis*. University of Massachusetts, Amherst.

Jacobson, P. (1999), 'Towards a variable-free semantics'. *Linguistics and Philosophy* **22:**117–84.

Kamp, H. (1981), 'A theory of truth and semantic representation'. In J. Groenendijk, T. Janssen & M. Stokhof (eds.), *Formal Methods in the Study of Language*. Mathematisch Centrum. Amsterdam. 277–322.

Kamp, H. & C. Rohrer. (1983), 'Tense in texts'. In R. Bäuerle, C. Schwarze & A. von Stechow (eds.), *Meaning, Use, and Interpretation of Language*. de Gruyter. Berlin. 250–69.

Kamp, H., J. van Genabith & U. Reyle. (2011), 'Discourse representation theory'. In D. M. Gabbay & F. Günthner (eds.), *Handbook of Philosophical Logic*, Vol. 15. Springer. Dordrecht. 125–394.

Karttunen, L. (1976), 'Discourse referents'. In J. McCawley (ed.), *Syntax and Semantics 2: Notes From the Linguistic Underground*. Academic Press. New York. 363–85.

Krahmer, E. (1998), *Presupposition and Anaphora*. CSLI Publications. Stanford.

Krifka, M. (1996), 'Parametrized sum individuals for plural anaphora'. *Linguistics and Philosophy* **19:**555–98.

Lapierre, S. (1992), 'A functional partial semantics for intensional logic'. *Notre Dame Journal of Formal Logic* **33:** 517–41.

Lepage, F. (1992), 'Partial functions in type theory'. *Notre Dame Journal of Formal Logic* **33:**493–516.

Muskens, R. (1995), *Meaning and Partiality*. CSLI Publications. Stanford.

Muskens, R. (1996), 'Combining Montague semantics and discourse representations'. *Linguistics and Philosophy* **19:** 143–86.

Muskens, R. (2011), 'A squib on anaphora and coindexing'. *Linguistics and Philosophy* **34:**85–89.

Nouwen, R. (2003), *Plural Pronominal Anaphora in Context*. Ph. D. thesis, Utrecht Institute of Linguistics OTS.

Nouwen, R. (2007), 'On dependent pronouns and dynamic semantics'. *Journal of Philosophical Logic* **36:**123–54.

Rothschild, D. (2011), 'Explaining presupposition projection with dynamic semantics'. *Semantics and Pragmatics* 4 (http://dx.doi.org/10.3765/sp).

Schlenker, P. (2005), 'Non-redundancy: Towards a semantic reinterpretation of binding theory'. *Natural Language Semantics* **13:**1–92.

Strawson, P. (1950), 'On referring'. *Mind* **59:**320–44.

Tichý, P. (1982), 'Foundations of partial type theory'. *Reports on Mathematical Logic* **14:**59–72.

van Benthem, J. (1986), *Essays in Logical Semantics*. Reidel. Dordrecht.

van Eijck, J. (2001), 'Incremental dynamics'. *Journal of Logic, Language and Information* **10:**319–51.

van Eijck, J. & F.-J. de Vries. (1992), 'Dynamic interpretation and Hoare deduction'. *Journal of Language, Logic and Information* **1:**1–44.

van Eijck, J. & C. Unger. (2010), *Computational Semantics with Functional Programming*. Cambridge University Press. Cambridge.

van Eijck, J. & A. Visser. (2012), 'Dynamic semantics'. In E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2012 edition) (http://plato.stanford.edu/archives/win2012/entries/dynamic-semantics/) last accessed 21 July 2013.

Vermeulen, K. (1993), 'Sequence semantics for dynamic predicate logic'. *Journal of Logic, Language and Information* **2:** 217–54.

First version received: 27.09.2012
Second version received: 05.06.2013
Accepted: 10.06.2013