A Simultaneous Abstraction Calculus and Theories of Semantics

Peter Ruhrberg*
peru@cogsci.ed.ac.uk

May 19, 1998

Abstract

I define a simple calculus of simultaneous abstraction, and show how it can provide the basis for property-theoretic reconstructions of Dynamic Montague Grammar and Discourse Representation Theory. The system can also be used for some forms of Situation Theory, instead of parametric objects and Aczel–Lunnon abstraction.

1 Introduction

There are close similarities between some of the most influential semantic theories of our time, such as Discourse Representation Theory (DRT), see [Kamp 1981] and [Zeevat 1991], Dynamic Montague Grammar (DMG), see [Groenendijk & Stokhof 1990], especially when based on the Dynamic Property Theory (DPT) of [Chierchia 1992], and Situation Theory (ST), as described in [Barwise & Cooper 1991], with its underlying theory of abstraction (AL) developed in [Aczel & Lunnon 1991]. They all can be construed as involving forms of *simultaneous abstraction*. In DRT we have a set of discourse markers as one ingredient of a DRS, though one usually doesn't think of them as being abstracted over. We will show a pretty straightforward way of rendering them as abstractions. In DPT the $^\cap$-operator can be understood as abstracting over the infinite set of discourse markers, which are a subsort of the variables. In AL we take a one-one mapping of roles to parameters as input to abstract over a set of parameters in one go.

Abstraction over an unordered set of variables, or parameters, creates the problem of defining a sensible operation of application which unambiguously determines which of the objects applied to fill which of the positions abstracted over. The solution to this difficulty comes in the form of *application to assignments*. These abstracts denote, or correspond to, functions from assignments to objects of some kind. They

*I thank Lawrence Cavedon, Robin Cooper, Rachel Lunnon and Paul Schweizer for criticism and support. This research was funded in part by ESPRIT BR6852 (DYANA-2) and LRE 62-051 (FraCaS).

are thus rather like open formulas in standard logics which have their semantics given in terms of functions from assignments to denotations. In the same vein DRSs can be taken to denote, under an assignment, a set of verifying "embeddings," that is, a function from variable assignments to truth values. In DPT functions from discourse marker assignments to propositions are first class citizens of the semantic domain itself, and in AL the application operation is defined on assignments of entities to role indices.

It seems worthwhile to look for a common core which brings out the similarities as well as the differences between these theories more clearly and with formal precision. This core is our *Simultaneous Abstraction Calculus*. Semantic theories can be obtained from it by giving axioms of properties, propositions and truth, which again may be shared to some extent across competing theories. One may group the axioms into convenient packages under the banner of "abstract specification," thus creating a framework that should facilitate the comparison of existing theories as well as the development of alternatives. DMG and DRT for example, which we concentrate on in this paper, share a common theory of identity and propositional logic, but diverge in the kinds of quantification they employ. ST might be obtained by adding a logic of situation types to such unsituated theories of propositions. I can only give a sketchy account of these matters here, and also must leave the topic of relating these semantic objects to natural language utterances largely in the background.

1.1 Variables: "Bound yet Free"

As a consequence of the idea of simultaneous abstraction we will face a certain *loss of α-equivalence*. This is an inevitable result of the way in which application has to work for such terms. A renaming of variables will normally change the object as it will yield different results when applied to the same assignment. This should come as no surprise: it is of course well known that free variables cannot be renamed, but neither can the "abstracted" discourse markers of a DRS be changed without consequences. The same holds true for DPT discourse markers under $^{\sqcap}$-abstraction. Parameters in AL can be renamed, if abstracted over, but role indices cannot without changing the object.

I will not embark into an argument about whether it is wrong to speak of "abstraction" when there is no α-equivalence. It is more important to see that it is not a virtue in itself to be able to rename variables, as this means that computational work has to be done to establish a trivial identity of objects. Its virtue, as we shall see, lies in obtaining a total substitution operation, which is essential for the full exploitation of β-equivalences.

The idea that variables may be essential to the identity of a semantic object is certainly now fairly accepted, due to DRT and its "dynamic" children. From a more philosophical perspective I would argue that using variables in this way is no more a contamination of semantic objects by non semantic concepts than the common view that the order of arguments to a many place function should matter.

2 The Simultaneous Abstraction Calculus

In this section I introduce the basic formalism for simultaneous abstraction. I then add an operation of partial application to it, and compare the system to a similar one which is closer to Aczel and Lunnon's approach. Finally I take a look at possibly non well-founded structured objects.

2.1 The Basic Calculus

Instead of abstracting over single variables, as in standard λ-calculus, we allow λ-abstraction over any set of them. Terms are applied to records assigning terms to variables. For the rest of the paper, differently indexed variables are assumed to be non-identical.

Definition 1 *The language* Λ *consists of* TERMS $t, t' \ldots$ *built up from basic type free* CONSTANTS $c, c' \ldots$, *and* $\#$ *for "undefined," and* VARIABLES $x, y \ldots \in Var$ *by means of* ABSTRACTION $\lambda M.t$, *for* $M \subseteq Var$, *and* APPLICATION $t(x_1.t_1, \ldots, x_n.t_n)$.

We in fact allow infinite sets of variables to be abstracted over and also infinite records $(x_i.t_i)_{i \in I}$ to be applied to. We often write $\{x_1..x_n\}t$ instead of $\lambda\{x_1, .., x_n\}.t$, without meaning to imply that n has to be a finite number. A similar remark holds for application terms. Ways of introducing standard abstraction and application into such a system will be discussed later on.

Semantically, we will be working in the category CPO of complete partial orders and continuous functions. So a DOMAIN will always be understood to be a cpo with a least element $\bot$, a mapping between cpo's to be continuous, and the operation $\rightarrow$ to form the space of continuous functions, ordered pointwise.[1] g_f^M is defined by $g_f^M(x) = f(x)$ if $x \in M$, otherwise $g_f^M(x) = g(x)$, for all $x \in Var$. Letting $dom(f) = \{a | f(a) \neq \bot\}$ I will use the notation $g \wr f$ for $g_f^{dom(f)}$. We write $\langle x_i \mapsto \xi_i \rangle_{i \in I}$ for the function f such that $f(x_i) = \xi_i$ for $i \in I$ and $f(x) = \bot$ otherwise. A RETRACTION between two domains, written $D \rightleftharpoons_{\Phi}^{\Psi} C$, is a pair of mappings $\Psi : D \rightarrow C$ and $\Phi : C \rightarrow D$ such that $\Psi \circ \Phi = id_C$.

Definition 2 *A* Λ-MODEL *consists of a retraction* $D \rightleftharpoons_{\Phi}^{\Psi} (Var \rightarrow D) \rightarrow D$, *and an interpretation* $\Im$ *which maps constants into* D, *with* $\Im(\#) = \bot_D$. *The denotation of terms under a variable assignment* g *is given by:*[2]

- $\|c\|^g = \Im(c)$;
- $\|x\|^g = g(x)$;
- $\|\lambda M.t\|^g = \Phi\ \lambda f \|t\|^{g_f^M}$;

[1] In some cases they will be required to be *strict*, preserving the bottom element. For background information notions see [Barendregt 1984]. Notice that every partial assignment function is continuous (if strict), given that Var is a flat domain with an added $\bot$.

[2] The third clause has its predecessor in [Zeevat 1990], where abstraction over the sets of variables of a common type was defined in a reconstruction of DMG. The 'λ' on the right is used as an expression of the meta language in this clause, with the expected meaning.

- $\|t(x_i.t_i)_{i\in I}\|^g = \Psi(\|t\|^g)\,(\langle x_i \mapsto \|t_i\|^g\rangle_{i\in I})$.

The denotations are only well-defined because the following holds:[3]

Theorem 1 *$\lambda f \|t\|^{g_f^M}$ is continuous for all terms t.*

Definition 3 *The* FREE VARIABLES *of a term are defined by*

- $FV(x) = \{x\}$;
- $FV(c) = \emptyset$;
- $FV(\lambda M.t) = FV(t) \setminus M$;
- $FV(t(x_i.t_i)_{i\in I}) = FV(t) \cup \bigcup_{i\in I} FV(t_i)$.

Lemma 2 *If $x \notin FV(t)$ then $\|t\|^g = \|t\|^{g_d^x}$ for all g and d.*

In such a case the denotation of t is independent of what we assign to x, but this does not mean one could rename x by some y of which t is independent as well. For example $\{x\}x$ is independent of x and y, but $\|\{x\}x\|^g \neq \|\{y\}y\|^g$. The latter becomes clear when we apply these two functions to suitable assignments e.g., $\|\{x\}x\,(x.a, y.b)\|^g = \|a\|^g \neq \|b\|^g = \|\{y\}y\,(x.a, y.b)\|^g$, presuming our model is non trivial.

Definition 4 SIMULTANEOUS SUBSTITUTION $[s_j/y_j]_{j\in J}$ *(dropping the index set when no confusion can arise) is a partially defined operation given by:*

- $[s_j/y_j]_{j\in J}x = s_j$, *if* $x = y_j$ *for some* $j \in J$, *otherwise* x;
- $[s_j/y_j]c = c$;
- $[s_j/y_j]_{j\in J}\lambda M.t = \lambda M.[s_i/y_i]_{i\in I}t$, *where* $I = J \setminus \{j |\ y_j \in M\}$, *if* $M \cap \bigcup_{i\in I} FV(s_i) = \emptyset$, *otherwise undefined;*
- $[s_j/y_j]\ t(x_i.t_i)_{i\in I} = [s_j/y_j]t\ (x_i.[s_j/y_j]t_i)_{i\in I}$.

An example of an undefined substitution would be $[y/x]\{y\}t_{xy}$, where replacing x in t_{xy} by y would give undesired results and renaming y is not a possibility. To avoid such trouble we require a FRESH VARIABLE for a term to occur neither free nor bound in it.

Let us say that $\Lambda \models t = t'$ iff for every Λ-model and assignment $\|t\|^g = \|t'\|^g$.

Theorem 3 *1.* $\Lambda \not\models \{x_1..x_n\}t = \{y_1..y_n\}[y_i/x_i]t$,

2. $\Lambda \not\models \{x_1..x_n\}\{x_{n+1}..x_{n+m}\}t = \{x_1..x_{n+m}\}t$.

Lemma 4 $\|t\|^\gamma = \|[t_i/x_i]_{i\in I}\,t\|^g$ *if $[t_i/x_i]t$ is defined, where $\gamma = g \wr \langle x_i \mapsto \|t_i\|^g\rangle_{i\in I}$.*

[3] Proofs are generally omitted from this paper. None of them is particularly interesting or involved.

The following equations can now be shown to hold, where the list is not meant to be complete. The fact that substitution is not always defined is responsible for the difficulties in giving such a list. Fact 5.3, which allows for some amount of renaming of variables, is just one example of an equation that cannot be derived from 5.2 for this reason. It seems a good idea to look for a complete calculus for Λ_{AL}, defined below, before doing the same for Λ since substitution can be totally defined in that system.

Theorem 5 *1.* $\Lambda \models \lambda M.t = \lambda N.t$ *for* $N = M \cap FV(t)$*;*

2. $\Lambda \models \{x_i\}_{i \in I} t \ (x_j.t_j)_{j \in J} = [t_j/x_j, \#/x_i]_{j \in J', i \in I'} \ t$,
with $J' = J \cap I$ *and* $I' = I \setminus J$*, if the substitution is defined.*

3. $\Lambda \models \{x_i\}_{i \in I} t \ (x_i.t_i)_{i \in I} = \{z_i\}_{i \in I} [z_i/x_i]_{i \in I} t \ (z_i.t_i)_{i \in I}$,
where all z_i *are fresh.*

2.2 Partial Application

It is very natural to ask for an operation for partially filling the argument roles of a relation, leaving some of them simply open for more to come. We use the notation $t[x_1.t_1, .., x_n.t_n]$ for partial application. The semantics is as follows:

$$\|t[x_i.t_i]_{i \in I}\|^g =_{df} \Phi \lambda f \ \Psi \|t\|^g \ (f \wr \langle x_i \mapsto \|t_i\|^g \rangle_{i \in I}) \ .$$

The following properties of partial application are of particular interest, letting application associate to the left.

Theorem 6 *1.* $\Lambda \models \{x_i\}_{i \in I} t \ [x_j.t_j]_{j \in J} = [t_j/x_j]_{j \in J'} \{x_i\}_{i \in I'} t$*, if defined,*
where $J' = J \cap I$ *and* $I' = I \setminus J$*;*

2. $\Lambda \models t[x_i.t_i]_{i \in I} [y_j.s_j]_{j \in J} = t[x_i.t_i, y_j.s_j]_{i \in I, j \in J'}$
where $J' = J \setminus \{j \mid \exists i \ y_j = x_i\}$*;*

3. $\Lambda \models t[x_i.t_i]_{i \in I} (y_j.s_j)_{j \in J} = t(x_i.t_i, y_j.s_j)_{i \in I, j \in J'}$
where $J' = J \setminus \{j \mid \exists i \ y_j = x_i\}$*.*

2.3 Aczel–Lunnon Abstraction

The SAC is closely related to Barwise and Cooper's Extended Kamp Notation (EKN), see [Barwise & Cooper 1991], which is based on Aczel and Lunnon's theory of abstraction. In EKN λ-abstraction operates on injective functions from a set of *role indices* to *parameters*. In our case those two notions are merged into one, the variables (thought of as argument roles), which allows us to replace the injections by simple sets. This makes sense because in contrast to AL our semantics does not treat the application to variables in a term like $t(x.y)$ as a filling of the role x by some "indeterminate object" y, but rather as a linking of two roles. There is a price we pay for our simple mindedness: we cannot link a role y with any role x inside the scope of an abstraction over y by a simple application. The same problem was encountered in undefined substitutions such as $[y/x]\{y\}t_{xy}$, which indicated a certain deficit of elegance for our system. The following system revises the basic SAC to overcome this problem, using AL-style syntax for abstraction. We will show then that no additional expressive power is gained from such a move.

Definition 5 *The* TERMS *of* Λ_{AL} *are built up from* CONSTANTS $c, \#, ..$ *and* PARAMETERS $X, Y, ..$ *by means of* ABSTRACTION $\lambda\chi.t$ *over injective functions* χ *from parameters to variables (=roles),*[4] *and* TOTAL/PARTIAL APPLICATION $t(x_i.t_i)_{i\in I}$ *and* $t[x_i.t_i]_{i\in I}$.

Semantically we stay in the same space of domains $D \rightleftharpoons^{\Psi}_{\Phi} (Var \to D) \to D$ as for Λ, with the only difference being that we now interpret under assignments of objects to parameters, called ANCHORS, and that these have to be obtained from assignments to roles when we define the meaning of abstraction.

Definition 6 *A* Λ_{AL}-MODEL *is a* Λ*-model, where the denotation of terms under an anchor* g *is given by:*

- $\|c\|^g = \Im(c)$;
- $\|X\|^g = g(X)$;
- $\|\lambda\chi.t\|^g = \Phi\ \lambda f \|t\|^{g\wr(f\circ\chi)}$;
- $\|t(x_i.t_i)_{i\in I}\|^g = \Psi(\|t\|^g)(\langle x_i \mapsto \|t_i\|^g\rangle_{i\in I})$;
- $\|t[x_i.t_i]_{i\in I}\|^g = \Phi\lambda f\ \Psi\|t\|^g\,(f \wr \langle x_i \mapsto \|t_i\|^g\rangle_{i\in I})$.

This solves the problems described above as substitution is total by means of renaming parameters e.g., $[X/Y]\lambda\langle X\mapsto z\rangle.t_{XY} = \lambda\langle Z\mapsto z\rangle.t_{ZX}$. The question that needs to be addressed is whether there is more to be gained, namely terms which have no equivalent in the original language Λ. To show that there are no such terms we only face one difficulty, namely that of dealing with multiple uses of roles in different levels of abstraction. Such "role recycling" is made easy by using different parameters in a term like $\lambda\langle Y \mapsto x\rangle\lambda\langle Z \mapsto x\rangle.t_{YZ}$. In creating a Λ-term with the same role x used twice we face the potential problem of the outer abstraction to become vacuous. The solution is to inject an intermediate role y to arrive at $\{x\}(\{y\}\{x\}t_{yx}\ (y.x))$. To have such roles available we assume some form of restriction on Λ_{AL} abstraction to that effect. Alternatively one might employ a way of expanding Var suitably for the purposes of translation. I will be sloppy about the details of this in the following elaboration.

Definition 7 *The* TRANSLATION $[x_i/X_i]^{\dagger}_{i\in I} : \Lambda_{AL} \to \Lambda$ *under a substitution of all free parameters of a* Λ_{AL}*-term by variables is defined by:*

- $[x_i/X_i]^{\dagger}_{i\in I} X_i = x_i$;

[4]In AL, the functions take roles to parameters injectively. In turning them around we could be more liberal in allowing several parameters to be mapped to the same role, which is equivalent to renaming those parameters by one of them which then gets mapped to the role in question. Notice that associating one parameter with many roles in one abstraction clearly makes no sense. Our semantics for Λ_{AL} will not presuppose injectivity, but we will assume it here nevertheless for convenience.

- $[x_i/X_i]^\dagger c = c$;
- $[x_i/X_i]^\dagger_{i\in I}\,\lambda\langle Y_j \mapsto y_j\rangle_{j\in J}.t$
  $= (\lambda\{z_i\}_{i\in I''}\lambda\{y_j\}_{j\in J}.[y_j/Y_j]_{j\in J}[x_i/X_i]_{i\in I'}[z_i/X_i]_{i\in I''}{}^\dagger t)\,(z_i.x_i)_{i\in I''}$,
  where $K = I\setminus\{i|\exists j_{\in J}\ X_i = Y_j\}$, $I' = K\setminus\{i|\exists j_{\in J}\ x_i = y_j\}$,
  $I'' = K\cap\{i|\exists j_{\in J}\ x_i = y_j\}$ *and the* z_i *are fresh;*
- $[x_i/X_i]^\dagger t(y_j.t_j)_{j\in J} = [x_i/X_i]^\dagger t\ (y_j.[x_i/X_i]^\dagger t_j)_{j\in J}$;
- $[x_i/X_i]^\dagger t[y_j.t_j]_{j\in J} = [x_i/X_i]^\dagger t\ [y_j.[x_i/X_i]^\dagger t_j]_{j\in J}$.

Lemma 7 $\|t\|^g = \|[x_i/X_i]^\dagger_{i\in I}t\|^{g\circ\pi}$, *where* $\pi = \langle x_i \mapsto X_i\rangle_{i\in I}$.

Translation from Λ to Λ_{AL} is a trivial matter of renaming variables by parameters, and abstracted sets by mappings, under some fixed bijection between Par and Var. Hence we conclude the following.

Theorem 8 *There are faithful translations from* Λ *to* Λ_{AL} *and vice versa.*

It is clear that we can define unary abstraction and application in Λ_{AL} in terms of a designated role used solely for that purpose. By the above theorem we see that the same can be done in Λ as well. The SAC thus contains the unary λ-calculus. In particular, α-equivalence for λ-terms becomes an instance of Theorem 5.3 above.

Corollary 9 λ *can be embedded into* Λ.

2.4 Structured Objects and Systems of Equations

Using some encoding of standard abstraction, which exists for Λ by the previous corollary, we define the following PREDICATION OPERATION:

$$\langle\!\langle t; (x_i.t_i)_{i\in I}\rangle\!\rangle =_{df} \lambda x.\,(x(t)\,(x_i.t_i)_{i\in I}).$$

Theorem 10 $\langle\!\langle t; (x_i.t_i)_{i\in I}\rangle\!\rangle = \langle\!\langle t'; (x_i.t'_i)_{i\in I}\rangle\!\rangle \to t = t' \wedge \bigwedge_{i\in I} t_i = t'_i$.

We thus have structured objects and operations of abstraction and application in place that appear to satisfy much of the needs of current Situation Theory. A further demand is to have non well-founded objects, obtained from cyclic sets of equations. To solve systems of equations[5] $(x_i = t_i)_{i\in I}$ we need to encode assignments by some kind of terms. We could use predication for this, but it is still simpler to define $\ll x_i.t_i \gg_{i\in I} =_{df} \lambda x.x(x_i.t_i)_{i\in I}$ and $\pi_z =_{df} \lambda x.x(\{z\}z)$. Notice that

$$\pi_{x_i}(\ll x_j.t_j \gg_{j\in J}) = t_i \quad for\ \ i\in J.$$

We say that an assignment $\bar{a}$ SOLVES A SYSTEM OF EQUATIONS $(x_i = t_i)_{i\in I}$ iff $\pi_{x_i}(\bar{a}) = [\pi_{x_j}(\bar{a})/x_j]_{j\in I}t_i$ for $i\in I$.

[5]We need to assume that at least three variables do not occur in such a system, to allow us to form the terms below without accidentally binding some of the x_i inside them.

Theorem 11 *Every system of equations has a solution.*

Proof: Recall that $\Upsilon =_{df} \lambda f.\,(\lambda x.f(xx))(\lambda x.f(xx))$ is a fixed point combinator with $t(\Upsilon t) = (\Upsilon t)$ for any term t. Define from a system of equations $(x_i = t_i)_{i\in I}$ the functor $F =_{df} \lambda z. \ll x_i.[\pi_{x_j}(z)/x_j]_{j\in I} t_i \gg_{i\in I}$. Then (ΥF) solves the system, as $\pi_{x_i}(\Upsilon F) = \pi_{x_i}(F(\Upsilon F)) = \pi_{x_i} \ll x_i.[\pi_{x_j}(\Upsilon F)/x_j]_{j\in I} t_i \gg_{i\in I} = [\pi_{x_j}(\Upsilon F)/x_j]_{j\in I} t_i$. □

As things stand we do not necessarily have unique solutions to such equations. It may be consistent to assume uniqueness at least for some kinds of systems, in the spirit of [Aczel & Lunnon 1991].

2.5 Equations for Λ_{AL}

The complications in clause three of the translation from Λ_{AL} to Λ indicate why a complete and reasonably simple set of equations for Λ will be hard to find. The slightly more complex language Λ_{AL} seems more suitable for this task. So I give a set of equations for it that I hope to be complete if used in conjunction with rules for congruence relations. The notions of free parameters and substitution are assumed to be defined in the standard way.

Theorem 12 *1.* $\models t(x_i.t_i, x_j.\#)_{i\in I, j\in J} = t(x_i.t_i)_{i\in I}$;

2. $\models \lambda\chi.t = \lambda\xi.t$ *for* $\xi = \chi|_{FP(t)}$;

3. $\models \lambda\langle X_i \mapsto x_i\rangle_{i\in I} t = \lambda\langle Y_i \mapsto x_i\rangle_{i\in I}[Y_i/X_i]_{i\in I} t$ *for fresh* Y_i*'s;*

4. $\models \lambda\langle X_i \mapsto x_i\rangle_{i\in I} t\ (x_j.t_j)_{j\in J} = [t_j/X_j, \#/X_i]_{j\in J', i\in I'} t$
with $J' = J \cap I$ *and* $I' = I \setminus J$;

5. $\models \lambda\langle X_i \mapsto x_i\rangle_{i\in I} t\ [x_j.t_j]_{j\in J} = [t_j/X_j]_{j\in J'} \lambda\langle X_i \mapsto x_i\rangle_{i\in I'} t$
with $J' = J \cap I$ *and* $I' = I \setminus J$;

6. $\models t[x_i.t_i]_{i\in I}[y_j.s_j]_{j\in J} = t[x_i.t_i, y_j.s_j]_{i\in I, j\in J'}$
where $J' = J \setminus \{j \mid \exists i\ y_j = x_i\}$;

7. $\models t[x_i.t_i]_{i\in I}(y_j.s_j)_{j\in J} = t(x_i.t_i, y_j.s_j)_{i\in I, j\in J'}$
where $J' = J \setminus \{j \mid \exists i\ y_j = x_i\}$.

3 Dynamic Montague Grammar

I now show how the SAC can provide a framework for specifying semantic theories, especially recent ones, which do not easily fit into the traditional λ-calculus. The basic idea for semantics as conceived of in this paper is to think of properties as functions from assignments to propositions.[6] I will use the term *ω-properties* to distinguish them from ordinary functions which take entities into propositions. Equipped with this notion we can reconstruct DMG along the lines of [Chierchia 1992].

[6]Propositions can be very fine grained—there is no need to think of propositions as sets of possible worlds or situations. Even so, Bealer's fondalee/rajneesh puzzle, see [Bealer 1989], suggests that propositional functions cannot cope with all problems of intensional grain, or at least that function application should be distinguished from a structure preserving predication operation, as was done in 2.4.

3.1 Using the SAC for DMG

For DMG, variables need to come in two kinds: the DISCOURSE MARKERS DM, and the remaining ones, which I call META VARIABLES MV. I use greek letters $\alpha, \beta \ldots$ for meta variables, dotted letters $\dot{x}, \dot{y} \ldots$ for discourse markers, and $x, y \ldots$ for variables that can be of either kind. The sorting of our variables is not a matter of the kinds of denotation that they take. It is rather a matter of protecting some of them, namely the meta variables, from the influence of abstraction so that we can bind variables inside the scope of $\sqcap$.

The crucial point of DMG can now be captured easily by abstracting and applying to the infinite set of discourse markers via the following definitions:

- $^{\sqcap}t =_{df} \lambda DM.t$, and
- $^{\sqcup}t =_{df} t(\dot{x}.\dot{x})_{\dot{x} \in DM}$.

All other abstractions and applications will be the traditional unary ones. Notice that every discourse marker is bound in a term of the form $^{\sqcap}t$, and occurs free in $^{\sqcup}t$. We thus can β-convert a term $\lambda\alpha.t\,(^{\sqcap}t')$ even if some discourse markers of t' end up in the scope of an abstraction in t. Discourse markers are not free in $^{\sqcap}t'$ and hence do not become bound by a new operator inside t. Another reduction we use in DMG beside the more standard β-conversions is $^{\sqcup\sqcap}t = t$.

The sentence 'A man walks in' can now be rendered in our notation as

$$\lambda\alpha.\Sigma\dot{x}(man(\dot{x}) \cap walkin(\dot{x}) \cap {}^{\sqcup}\alpha)$$

which can applied to $^{\sqcap}whistle(\dot{x})$ ('he whistles') to yield

$$\Sigma\dot{x}(man(\dot{x}) \cap walkin(\dot{x}) \cap whistle(\dot{x}))$$

The important action happens at the level of conjunction of ω-properties, which is achieved by abstracting over "possible continuations" of the discourse.[7]

We use unary abstraction and application in the system in order to get a fairly standard logic that does not require us to rethink the setup of DMG. We saw that these notions can be defined within the confines of Λ.[8] Logical operations are treated as unary or binary constants whose semantic behavior is captured by our theories of truth, which we choose to give separate from the general model theory in an axiomatic way. The following logical constants are singled out: $\cap, \cup, -, \supset, \Sigma, \Pi, \doteq$, intended to be the operations of conjunction, disjunction, negation, implication, existential and universal quantification, and equality. I will take $\cap, -, \Sigma, \doteq$ as primitive, and the others to be defined in the standard way.[9] Logical combinators are written in the conventional form, $t \cap t'$ for $\cap(t)(t')$, $\Sigma x\, t$ for $\Sigma(\lambda x.t)$ etc., to enhance readability.

[7] Lambda conversion on $^{\sqcap}whistle(\dot{x})$ is only one way to make the conjunction happen; Unification of $whistle(\dot{x})$ with the free variable α, as in UCG, is just as intuitive for this. Hence the choice of the term 'meta variables', reminiscent of its use in [Zeevat 1991].

[8] As an alternative to taking these unary operations as defined in terms of Λ-expressions, one could introduce them as basic operations, and give them a semantics in an expanded domain $D \rightleftharpoons^{\Psi}_{\Phi} ((Var \to D) \to D) + (D \to D)$, in the way [Chierchia 1992] does.

[9] There may be reasons not to do this, such as to avoid unintended identities in attitude contexts, or to be able to use the less symmetric notion of implication of [Aczel 1980].

3.2 Frege Structure Axioms

For semantics, we need a theory of truth for the propositions that we hope to denote by the Λ-terms. To that end, we introduce a formal language in which to state such a theory. One might do the job in an informal model theoretic way along the lines of [Aczel 1980] but I will make things look more like the formal treatment of [Turner 1990].

Definition 8 *We define WFFs of* FOL_Λ *in the standard way:*

- *if* $t, t' \in TERM$ *then* $T(t), t = t' \in WFF$,
- *if* $\psi, \phi \in WFF$ *then* $\neg\phi, \phi \wedge \psi, \exists\alpha\phi \in WFF$.

The interpretation of WFFs proceeds in the standard way. T is a truth predicate. We define $F(t) =_{df} T(-t)$ (falsity), $P(t) =_{df} T(t) \vee F(t)$ (being a proposition), and $PTY^n(t) =_{df} \forall x_1 \ldots x_n\, (\bigwedge_{i=1..n} x_i \neq \#) \rightarrow P(t(x_1)\ldots(x_n))$ (being an n-place property). Here are the basic axioms:

$$\begin{array}{rrl}
 & P(t \doteq t') \wedge & T(t \doteq t') \leftrightarrow t = t' \\
P(t) \rightarrow & P(-t) \wedge & T(-t) \leftrightarrow \neg T(t) \\
P(t) \wedge P(t') \rightarrow & P(t \cap t') \wedge & T(t \cap t') \leftrightarrow T(t) \wedge T(t') \\
PTY^1(t) \rightarrow & P(\Sigma\, t) \wedge & T(\Sigma t) \leftrightarrow \exists z\, T(t(z))
\end{array}$$

3.3 Some DMG Translations

We can only give a short sketch here and refer the reader to [Chierchia 1992] and [Chierchia 1992b] for a more complete treatment. Sentence denotations in DMG are not propositions, but *context change potentials (ccp's)*, where a ccp is a function from ω-properties to propositions. Two operations are handy to convert props into ccp's and back: $\uparrow t =_{df} \lambda\alpha(t \cap {}^{\sqcup}\alpha)$ and $\downarrow t =_{df} t({}^{\sqcap}true)$. Ccp's allow for clever ways of leaving holes within quantified structures to be filled by later material in the discourse so that one can use functional composition $t; t' =_{df} \lambda\alpha.t({}^{\sqcap}(t'(\alpha)))$ for discourse sequencing:

'$^{\dot{x}}$[a]' $\Longrightarrow$
$\lambda\alpha\lambda\beta\ \lambda\gamma\Sigma\dot{x}({}^{\sqcup}\alpha\,\dot{x})\,{}^{\sqcap}({}^{\sqcup}\beta\,(\dot{x})(\gamma))$

'$^{\dot{x}}$[a man] walks in. $^{\dot{x}}$[he] whistles.'$\Longrightarrow$
$[\lambda\gamma\Sigma\dot{x}(man(\dot{x}) \cap walkin(\dot{x}) \cap {}^{\sqcup}\gamma)]\,;\,[\lambda\gamma.\,whistle(\dot{x}) \cap {}^{\sqcup}\gamma]$
$= \lambda\gamma(\Sigma\dot{x}\ man(\dot{x}) \cap walkin(\dot{x}) \cap whistle(\dot{x}) \cap {}^{\sqcup}\gamma)$

The flexibility of ccp's is shown by various possibilities to define dynamic kinds of universal quantification for donkey sentences:

'every' $\Longrightarrow$
$\lambda\alpha\lambda\beta\uparrow\Pi\dot{x} - [({}^{\sqcup}\alpha\,\dot{x})({}^{\sqcap}- \downarrow({}^{\sqcup}\beta\,\dot{x}))]$

'man with ${}^{\dot y}$[a donkey]' $\Longrightarrow$
$\lambda\dot x\ \lambda\gamma(man(\dot x) \cap \Sigma\dot y(donkey(\dot y) \cap with(\dot x, \dot y) \cap {}^{\sqcup}\gamma))$

'beats ${}^{\dot y}$[it]' $\Longrightarrow$
$\lambda\dot x \uparrow beat(\dot x, \dot y)$

By intensional function application $t({}^{\sqcap}t')$ on the translations one gets:

$$\uparrow\Pi\dot x - [man(\dot x) \cap \Sigma\dot y(donkey(\dot y) \cap with(\dot x, \dot y) \cap -(beat(\dot x, \dot y) \cap true))]$$

An alternative to the above "strong" reading of the universal, would assign the translation:

'every' $\Longrightarrow$
$\lambda\alpha\lambda\beta \uparrow\Pi\dot x[- \downarrow({}^{\sqcup}\alpha\, \dot x)\ \cup\ ({}^{\sqcup}\alpha\, \dot x)({}^{\sqcap} \downarrow({}^{\sqcup}\beta\, \dot x))]$

resulting in the "weak" reading:

$\uparrow\Pi\dot x[-(man(\dot x) \cap \Sigma\dot y(donkey(\dot y) \cap with(\dot x, \dot y) \cap true))$
$\cup(man(\dot x) \cap \Sigma\dot y(donkey(\dot y) \cap with(\dot x, \dot y) \cap beat(\dot x, \dot y) \cap true))]$

4 Discourse Representation Theory

The SAC mirrors very closely the use of discourse referents in DRSs. The similarity with Zeevat's semantics for those, in terms of pairs consisting of a set of variables and a function from assignments to truth, is undeniable, see [Zeevat 1991]. We have put propositions in the place of truth values and defined an abstraction operation that takes such pairs into appropriate new functions. In addition, we can iterate such abstractions indefinitely and we have a simple notion of application at our disposal.

Our approach is most closely related to Cooper's Situation Theoretic DRT, see [Cooper 1993], who uses EKN with Aczel–Lunnon abstraction to similar effect. As in Cooper's treatment our version of DRT imposes a need for some additional abstractions at the right level. I won't go into the resulting complications for the syntax–semantics interface here.

4.1 ω-Properties

To be a propositional function in the argument roles $x_1 \ldots x_n$ is defined by:[10]

$$PF^{x_1..x_n}(t) =_{df} \forall z_1..z_n \bigwedge_{i=1..n} z_i \neq \# \ \rightarrow\ P(t(x_1.z_1, \ldots, x_n.z_n)).$$

[10]We will use an infinitary version of FOL_Λ to express the theory, though one might get away with axiom shemata in the finitary logic if we stick to finite n here. In particular the definition of $PTY^M(t)$ below would have to be replaced by an infinite number of implicational axioms.

This is not a sufficiently strict notion of ω-properties to support the forthcoming definitions and theorems. How strict a notion of ω-property one ultimately wants remains to be seen, but a function f that for example maps an assignment $\langle x \mapsto d\rangle$ to the proposition p while mapping $\langle x' \mapsto d'\rangle$ to $-p$ seems useless for semantics. We exclude such pathological cases by requiring something to the effect that

$$PTY^M(d) \Rightarrow \forall N\; PF^N(d) \leftrightarrow M \subseteq N.$$

A too strict definition would have $=$ instead of $\subseteq$. This would exclude any abstracts from denoting ω-properties, because they are insensitive to overdefined assignments:

$$\Lambda \models \{x_1..x_n\}t\,(x_1.t_1,..,x_n.t_n) = \{x_1..x_n\}t\,(x_1.t_1,..,x_{n+m}.t_{n+m})$$

Still one may ask whether the application of a ω-property to an assignment should ever result in something other than a proposition or $\bot$. The only sensible possibility seems to be to have application to "underdefined" assignments result in a new ω-property whose argument roles are those that haven't been filled on the first attempt. We have introduced the operation of *partial application* for these purposes, which will also enable us to define the duplex conditions of modern DRT. So for now I stick to the view that (full) application of ω-properties to assignments, if yielding a definite result at all, always yields a proposition.

Definition 9 $PTY^{\{x_i|i\in I\}}(t) =_{df} \bigwedge_{I\subseteq J} PF^{\{x_i|i\in J\}}(t) \wedge \bigwedge_{I\not\subseteq J} \overset{i\in J}{\forall} z_i\; t(x_i.z_i)_{i\in J} = \#.$

4.2 Kamp Structures

Let us now try to give a theory of ω-properties and propositions and truth that is capable of dealing with basic ideas of DRT. We add to the logical operations $\cap, -, \doteq$ two new quantificational operators, namely a non selective existential Ξ and a binary conditional $\gg$. All these are required to be (bi)strict: yielding $\bot$ whenever one of their arguments is $\bot$. Here are the axioms for Kamp structures:

$$\begin{array}{rrl}
 & P(t \doteq t') \wedge & T(t \doteq t') \leftrightarrow t = t' \\
P(t) \;\rightarrow & P(-t) \wedge & T(-t) \leftrightarrow \neg T(t) \\
P(t) \wedge P(t') \;\rightarrow & P(t \cap t') \wedge & T(t \cap t') \leftrightarrow T(t) \wedge T(t') \\
PTY^{x_1..x_n}(t) \;\rightarrow & P(\Xi\, t) \wedge & T(\Xi t) \leftrightarrow \exists z_1..z_n\, T(t(x_1.z_1,..,x_n.z_n))
\end{array}$$

$$PTY^{\{x_i|i\in I\}}(t) \wedge PTY^{\{x_j|j\in J\}}(s) \;\rightarrow\; P(t \gg s) \wedge$$

$$T(t \gg s) \;\leftrightarrow\; \overset{i\in I}{\forall} z_i\; T(\,t\,(x_i.z_i)_{i\in I}\,) \rightarrow \overset{j\in J\setminus I}{\exists} z_j\;\; T(\,s\,(x_i.z_i, x_j.z_j)_{i\in J\cap I, j\in J\setminus I}\,)$$

Notice that only bound variables in s are quantified in the conditional. To get some kind of "dynamic conjunction" of DRSs we can introduce the operation $\oplus$ into the system.[11] For example by means of this definition:

[11]This operation is used in [Cooper 1993] for the interpretation of discourse. It becomes definable in a slightly richer language that employs variables over assignments.

$$\|t \oplus t'\|^g =_{df} \Phi\ \lambda f\ \Psi\|t\|^g\,(f) \cap \Psi\|t'\|^g\,(f)$$

The following consequences can now be derived:

Theorem 13 *1.* $PTY^X(t) \wedge PTY^Y(t') \rightarrow PTY^{X\cup Y}(t \oplus t')$;

2. $\models t \oplus t'\,(x_1.t_1,..,x_n.t_n) = t(x_1.t_1,..,x_n.t_n) \cap t'(x_1.t_1,..,x_n.t_n)$

3. $\models \lambda M.t \oplus \lambda N.s = \lambda M \cup N\,.\,t \cap s$,
if $M \cap FV(\lambda N.s) = N \cap FV(\lambda M.t) = \emptyset$.

Using partial application we can define DRT-type quantifiers Q_x for the so called "duplex conditions," by means of standard generalized quantifiers Q as relations between ω-properties. The weak and strong readings are obtained as follows:

$$t\langle Q_x\rangle^w t' =_{df} Q(\lambda z.\Xi\, t[x.z]\,,\ \lambda z.\Xi\, t \oplus t'[x.z]),$$

$$t\langle Q_x\rangle^s t' =_{df} Q(\lambda z.\Xi\, t[x.z]\,,\ \lambda z.\, t[x.z] \gg t'[x.z]).$$

4.3 True Dynamics

It is now time to ask what the relation is between the SAC and systems like the Dynamic Predicate Logic (DPL) of [Groenendijk & Stokhof 1991], that are based on relational interpretations. DPL replaces the non-standard DRT syntax by FOL formulas, but they have shown how to give the relational interpretation for DRSs too. Let us view relations between assignments as functions from assignments to sets of assignments. Call sets of assignments ω-sets, and take conditions to denote just ω-sets. We then get the following denotation for a DRS:

$$\|\{x_1...x_n\}\phi\|^g_{DPL} = \{h \mid g \overset{\{x_1..x_n\}}{\sim} h\ \wedge\ h \in \|\phi\|\}.$$

This should be compared with extensionalized SAC denotations, which we obtain when we replace propositions by truth values. We then also get ω-sets as denotations under an assignment, but they are different from the DPL ones:

$$\|\{x_1...x_n\}\phi\|^g_{SAC} = \{h \mid g^{\{x_1..x_n\}}_{h} \in \|\phi\|\}.$$

The point is that the h's in our set can assign anything they like to the variables outside $\{x_1...x_n\}$, while for the DPL style of denotation they have to agree with the incoming g.

A natural development, carried further in [Dekker 1993], is to view "information states" as such ω-sets, and then lift denotations of DRSs to functions from ω-sets to ω-sets, such as:

$$\|\phi\|_{DPL\uparrow}(G) = \bigcup_{g \in G} \|\phi\|^g_{DPL}$$

which is quite different from the trivial lifting from ϕ to $\lambda p.(p \oplus \phi)$ that might be used in our system.

The lack of a requirement for assignments to agree on certain variables has consequences for semantics. Thus, in contrast to DPL and most forms of DRT, we must abstract over variables in order to link them anaphorically to preceding discourse by $\oplus$. If we would use DPL style composition for discourse conjunction instead the link could not be sustained over more than two sentences. A similar divergence can be found in the treatment of the conditional. Again we have to put anaphoric variables into to abstracted set of the consequence, but only those which are anaphoric to the antecedent, not those which refer further back. Abstraction in Λ really is just a way of partitioning roles into different levels, in view of how we want to fill them or quantify over them.

There is something of a philosophical gap between the approach of this paper and the "truly dynamic" ones, which is brought out by these considerations. On the SAC view the functions from assignments to ω-properties are not regarded as ways of "updating" information states by incoming material. It makes no sense on this view to talk of "input" and "output" in reference to the assignments. The guiding intuition is rather that semantic objects should be plausible as things to have attitudes towards. These objects, I take it, are rather static creatures whose role in the changes of mental states are another matter.

5 Conclusion

We have defined a calculus of simultaneous abstraction and shown it to be equivalent to a less simple minded one that is more reminiscent of Aczel and Lunnon's system. We established the existence of structured, and non well-founded objects in the system, by an embedding of the lambda calculus in it. We then obtained versions of Dynamic Montague Grammar and Discourse Representation Theory, using axiomatic theories of truth. It seems plausible that the same can be done for Situation Theory as well, supporting the claim that we have found a framework which is general enough for a wide range of semantic theories which take notions of propositions and truth as fundamental for the enterprise. It is at odds though with forms of "dynamic" semantics that take the idea of updating (sets of) assignments as their starting point. Many important logical questions remained unanswered in this paper, but I hope to have shown that they may be worth asking.

References

[Aczel 1980] P. Aczel: Frege Structures and the Notions of Proposition, Truth and Set. In Barwise et al. (ed.) *The Kleene Symposion*, North Holland Studies in Logic 1980.

[Aczel & Lunnon 1991] P. Aczel and R. Lunnon: Universes and Parameters. In Barwise et al. (ed.) *Situation Theory and its Applications, Vol. 2.* CSLI, Stanford 1991.

[Barendregt 1984] H.P. Barendregt: *The Lambda Calculus. Its Syntax and Semantics.* Revised edition, North-Holland, Amsterdam 1984.

[Barwise & Cooper 1991] J. Barwise and R. Cooper: Simple Situation Theory and its Graphical Representation. In Jerry Seligman (ed.) *Partial and Dynamic Semantics 3*, DYANA deliverable R2.1.C, Centre for Cognitive Science, Edinburgh, 1991.

[Bealer 1989] G. Bealer: On the Identification of Properties and Propositional Functions. In *Linguistics & Philosophy 12*, 1-14, Kluwer, 1989.

[Chierchia 1992] G. Chierchia: Intensionality and Context Change. Towards a Dynamic Theory of Propositions and Properties. Manuscript 1992. To appear in the *Journal of Logic, Language and Information.*

[Chierchia 1992b] G. Chierchia: Anaphora and Dynamic Binding. In *Linguistics & Philosophy 15/2* 1992.

[Cooper 1993] R. Cooper: *Towards a General Semantic Framework.* In R. Cooper (ed.): Integrating Semantic Theories. DYANA-II deliverable R2.1.A, Centre for Cognitive Science, Edinburgh 1993.

[Dekker 1993] P. Dekker: *Transsentential Meditations. Ups and downs in dynamic semantics.* ILLC dissertation, Amsterdam 1993.

[Groenendijk & Stokhof 1990] J. Groenendijk and M. Stokhof: Dynamic Montague Grammar. Technical Report, ITLI Amsterdam.

[Groenendijk & Stokhof 1991] J. Groenendijk and M. Stokhof: Dynamic Predicate Logic. In *Linguistics & Philosophy 14*, 1991.

[Kamp 1981] H. Kamp: A Theory of Truth and Semantic Representation. In J. Groenendijk et al. (eds.) *Formal Methods in the Study of Language.* Mathematical Centre, Amsterdam 1981.

[Turner 1990] R. Turner: *Truth and Modality for Knowledge Representation.* Pitman, London 1990.

[Zeevat 1990] H. Zeevat: *Static Semantics.* In J. van Benthem (ed.): *Partial and Dynamic Semantics I*, DYANA deliverable, Centre for Cognitive Science, Edinburgh 1991.

[Zeevat 1991] H. Zeevat: *Aspects of Discourse Semantics and Unification Grammar.* Ph.D. Thesis, Amsterdam 1991.