Partial dynamic semantics

Dag Trygve Truslew Haug

March 12, 2014

Abstract
The development of dynamic semantics was to a large extent motivated by the desire to account for anaphoric expressions (such as pronominals, but also anaphoric tense), which do not in themselves have reference, but must be resolved in the discourse context, possibly outside the sentence in which they occur. Yet it is striking that no dynamic semantic framework can give an interpretable denotation for unresolved anaphora. In this talk I present a new dynamic framework, partial CDRT (Haug 2013) based on Reinhard Muskens’ CDRT (Muskens 1996) grafted on a partial logic. In this framework, we can give an interpretable semantics for unresolved anaphora, thereby separating the monotonic semantics from non-monotonic processes such as a anaphora resolution.

1 Introduction

It is usual in dynamic semantics to discuss sentences and discourses like (1)-(2).

(1) Every\(^2\) farmer who owns a\(^7\) donkey beats it\(^7\).

(2) a. A\(^3\) cat appeared. It\(_3\) meowed.
   b. No\(^5\) cat appeared. *It\(_5\) meowed.

• What do those little numbers mean, and who put them there anyway? (Beaver 1999)
• Is it possible to give a semantics for words such as every, a, and even it – without the little number attached?

One way around this is as in (3)-(4).

(3) Every\(^2\) farmer who owns a\(^7\) donkey beats it\(_?\).

(4) a. A\(^3\) cat appeared. It\(_?\) meowed.
   b. No\(^5\) cat appeared. *It\(_?\) meowed.

This is fine as far as it goes. But ideally we want to make sure that

• the resolution of ? is properly constrained by the semantics.
• we can change the resolution non-monotonically.
• we can interpret the sentences independently of resolution.
1.1 The motivation for dynamic semantics

Let us first consider a static semantics for indefinites:

(5) A cat appeared. \(\leadsto \exists x. \text{cat}(x) \land \text{appear}(x)\)

The good news:

- The indefinite is not referential so (5) can be true no matter which cat appeared
- Because the variable is bound, its denotation does not depend on an assignment
- Because the variable is bound, it doesn’t matter which variable we choose

(6) It meowed. \(\leadsto ?\)

The bad news:

- Because the variable is bound, we cannot access it outside the scope of the quantifier

We need to combine features of bound and free variables to get the right semantics in such cases.

Dynamic predicate logic: tinker with scope and variable binding (Groenendijk and Stokhof 1991)

DRT: use free variables, and tinker with the truth definition (DRT, (Kamp 1981)).

I will follow DRT here.

1.2 DRT’s solution

1.2.1 DRSs and conditions

A DRS for (5):

\[
\begin{array}{c|cc}
   x & \text{cat}(x) & \text{appear}(x) \\
\end{array}
\]

A DRS is interpreted as a set of verifying assignments. Some notation:

- \(i, o\) are partial assignments of individuals to variables,
- \(i \subseteq_{\{x\}} o\) (\(o\) extends \(i\) with \(\{x\}\)) means that \(o\) assigns the same individual as \(i\) to all drefs in the domain of \(i\), but also assigns an individual to \(x\),
- \(\mathcal{I}\) interprets the non-logical constants of the language, mapping \(n\)-ary predicates to sets of \(n\)-tuples in \(D^n\).

Verification of DRSs and simple conditions:

- \(\langle i, o \rangle\) verifies the DRS \(\langle U, \text{Con} \rangle\) iff \(i \subset U\) \(o\) and for all \(\gamma \in \text{Con}\), \(o\) verifies \(\gamma\).
- \(o\) verifies a simple condition \(P(x_1, \ldots, x_n)\) iff \(\langle o(x_1), \ldots, o(x_n) \rangle \in \mathcal{I}(P)\)
This gives us (8) as an interpretation of (7)

\[ \{ \langle i, o \rangle \mid i \subset \{ x \} \land o(x) \in \mathcal{I}(cat) \land o(x) \in \mathcal{I}(appear) \} \]

Truth relative to an input context \( i \) is defined as the existence of an output assignment \( o \). Truth simpliciter is truth relative to the empty input context.

### 1.2.2 Connectives: negation

Negation is represented as a complex condition (embedded DRS):

(9) No cat appeared.

\[
\begin{array}{|c|}
\hline
x \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\neg \quad \text{cat}(x) \\
\hline
\begin{array}{|c|}
\hline
\text{appear}(x) \\
\hline
\end{array}
\end{array}
\]

(10) \( o \) verifies \( \neg K \) iff there is no assignment \( j \) such that \( \langle o, j \rangle \) verifies \( K \)

(11) \( \{ \langle i, o \rangle \mid i = o \) and there is no \( j \) such that \( o \subset \{ x \} \land j(x) \in \mathcal{I}(cat) \land j(x) \in \mathcal{I}(appear) \} \)

The output assignment \( o \) is unchanged!

### 1.2.3 Connectives: conditionals

Conditionals introduce two layers of DRS embedding.

(13) Every cat eats a mouse.

\[
\begin{array}{|c|}
\hline
x \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\text{cat}(x) \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
y \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\text{mouse}(y) \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\text{eat}(x, y) \\
\hline
\end{array}
\]

(14) \( o \) verifies \( K \Rightarrow K' \) iff for all assignments \( j \) such that \( \langle o, j \rangle \) verifies \( K \), there is an assignment \( k \) such that \( \langle j, k \rangle \) verifies \( K' \).

(15) \( \{ \langle i, o \rangle \mid i = o \) and for all \( j \) such that \( o \subset \{ x \} \land j(x) \in \mathcal{I}(cat) \), there is a \( k \) such that \( j \subset \{ y \} \land k(y) \in \mathcal{I}(mouse) \) and \( \langle k(x), k(y) \rangle \in \mathcal{I}(eat) \}\)

The output assignment \( o \) is again unchanged. Note that \( k(x) \) is defined and equal to \( j(x) \).
1.2.4 Sequencing
Sequencing two DRSs – written \( K ; K' \) – is defined as in (17).

\[
[K ; K'] := \{ \langle i, o \rangle | \text{there is an } m \text{ such that } \langle i, m \rangle \text{ verifies } K \text{ and } \langle m, o \rangle \text{ verifies } K' \}
\]

Intuitively, the output of the first DRS serves as the input to the second.

1.2.5 DRS and anaphoric accessibility
There is a truth-preserving embedding of DRT into first order logic. For example, (7) and (10) translate into (18) and (19) respectively.

\[
\begin{align*}
(18) & \quad \exists x.\text{cat}(x) \land \text{appear}(x) \\
(19) & \quad \neg \exists x.\text{cat}(x) \land \text{appear}(x)
\end{align*}
\]

However, while these are truth-conditionally equivalent to (8) and (11), they do not capture the difference in anaphoric potential which arise from the interpretation of DRSs.

In (8), the output assignment \( o \) is extended with a dref \( x \) which is therefore available for anaphoric uptake, but in (11) (and (16)), there is no such extension. We therefore correctly predict that (8) and (11) differ in their anaphoric potential:

\[
\begin{align*}
(20) & \quad \text{A cat appeared. It meowed.} \\
(21) & \quad \text{No cat appeared. *It meowed.}
\end{align*}
\]

In other words, constraints on anaphoric accessibility fall out from the semantics of the connectives. In other words, we have a model-theoretic account of semantic accessibility. This is one of the main achievements of DRT.

1.3 Remaining problems
1.3.1 Unresolved anaphora
Let us get back to the discourse in (22).

\[
(22) \quad \text{A cat appeared. It meowed.}
\]

The second sentence gets the representation in (23).\(^1\)

\[
\begin{array}{c|c|c}
\hline
x & y & y \\
\hline
\text{cat}(x) & \text{non-human}(y) & \text{meowed}(y) \\
\text{appear}(x) & \text{representation} & \\
\text{context} & \{ \} & \\
\hline
\end{array}
\]

\(^1\)I follow exposition of Two Stage Bottom-Up DRS Construction in (Kamp, van Genabith, and Reyle 2011, p. 139–144). The old notation in terms of conditions like \( y = ? \) is equivalent for our purposes.
Such representations are *not interpreted*, but must be resolved. Resolution involves identifying any drefs in the universe of a presuppositional DRS with a dref in the context and augmenting the assertion with that identification. Furthermore, given this anaphoric resolution, the context must entail any conditions in the presuppositional DRS (under this identification), or at least allow accommodation of these conditions. In (23), we can augment the assertion with \(y = x\). The context entails \(\text{non-human}(x)\)

\[
\begin{array}{|c|}
\hline
x \\
\hline
\text{cat}(x) \\
\text{appear}(x) \\
\text{non-human}(x) \\
\hline
\end{array}
\]

context

\[
\begin{array}{|c|}
\hline
y \\
\hline
\text{meowed}(y) \\
\hline
\end{array}
\]

representation

and we can merge the context and the representation to yield a new DRS (25).

\[
\begin{array}{|c|}
\hline
x y \\
\hline
\text{cat}(x) \\
\text{appear}(x) \\
y = x \\
\text{non-human}(y) \\
\text{meowed}(y) \\
\hline
\end{array}
\]

DRT’s resolution strategy is explicitly designed to ensure that

1. resolution always takes place in a context which does not include information provided by the part of the sentence which contains the presupposition trigger
2. the ordering information is preserved up to sequencing and then discarded

1 may be motivated for other kinds of presuppositions, but not for anaphora. Surely the fact that *it* is the subject of *meowed* is relevant to the resolution of the anaphor (in a context with competing antecedents). Even the following discourse is relevant, which means that 2 is problematic too.

(26) 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.

We also need “backtracking” in our anaphor resolution, and 2 makes that impossible.

(27) 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.

Finally, in addition to the coreference problem there is a dual ‘overwrite problem’: how do we make sure that we pick an unused discourse referent? This is connected to the use of free variables.
2 Data structures for tracking discourse referents

There are a variety of different data structures used to track discourse referents:

- **DRT** partial assignments
- **DPL** total assignments
- **others** lists (Vermeulen 1993; Dekker 1994; van Eijck 2001; Nouwen 2003; Nouwen 2007; Bittner 2001; Bittner 2007; Schlenker 2005; van Eijck and Unger 2010)

A list is a data structure with two operations

\[\wedge \text{ takes a list and an individual to a new list with the individual added at the end}\]

\[\lbrack n\rbracket \text{ returns the } n\text{th last individual in the list}\]

Dynamic existential quantification is non-deterministic appending to the list:

\[(28) \ \exists := \lambda c.\lambda c'.\exists x.\wedge^x c = c'\]

Anaphoric expressions do not append an individual to the list but rather extract one from it:

\[(29) \ \text{Every}\brace{k}[f] \text{ farmer who owns a}\brace{k}[f]\cdot d \text{ donkey beats it}_1.\brace{[1]}\]

\[(30) \ a. \ \text{A}\brace{k}[c] \text{ cat appeared.}^{[c]} \text{ It}_1.\brace{[1]} \text{ meowed.}\]

\[b. \ \text{No}\brace{k}[c] \text{ cat appeared.}^{[c]} \text{ It}_1.\brace{[1]} \text{ meowed.}\]

This again requires indices! To avoid that, van Eijck (2001, p. 349) suggests that, “pronouns can be translated as invitations to pick a reference from the current context.”

\[(31) \ \lambda c.\forall c'.\exists x.y.\wedge^x c = c' \land farmer(x) \land donkey(y) \land own(x, y) \rightarrow beats(x, sel(c'))\]

\(sel(c')\) is in the scope of \(\forall c'.\) This falsely predicts a reading of (32) where each nation cherishes either their own king or the king of France.

\[(32) \ \text{France}_1 \text{ is a monarchy. Every}_2 \text{ nation cherishes its}_3 \text{ king.}\]

We also run into problems with backtracking. We observed that the resolution of its is likely to depend on whether we get the continuation in (33) a or b.

\[(33) \ a. \ \text{So the French love King François.}\]

\[b. \ \text{That is because his appearances in the UN have been so convincing.}\]

We cannot directly access past states of the list. More individuals will have been added, and the array may have been permuted by resolution of other anaphora.

The list approach requires us to resolve anaphora in contexts, whereas we want it to be done “globally”, taking the whole discourse into account. I will argue that the best way to do this is to return to the original DRT intuition that even anaphors introduce discourse referents, which are then subsequently resolved to some antecedent. This makes it possible to talk about anaphora resolution.

\[\Rightarrow \text{ lists of dref occurrences, with anaphora as discourse-level identification of occurrences.}\]
3 Partial CDRT

A common complaint about DRT is that it is not compositional, because it uses unification rather than functional application to build DRSs. Functional application has at least two advantages:

1. Using functional application instead would let us use the Curry-Howard isomorphism to provide a syntax-semantics interface, as in Categorial Grammar (van Benthem 1986), Dynamic Syntax (Cann, Kempson, and Marten 2005) or Glue Semantics (Dalrymple 1999).

2. Intermediate representations can be eliminated

Compositional DRT (Muskens 1996) is one attempt to equip DRT with lambdas. The basic idea is to inject assignments, which are part of the metalanguage in DRT, into the object language. We will do this, but within a partial rather than a classical logic.

3.1 Partial type theory

We need to be careful in setting up our partial type theory, as there are potential lurking problems (Lapierre 1992, p. 521f.). Our logic is based on the intuition that undefinedness is absurdity, not lack of information (as in e.g. strong Kleene logic). Here are some key features, formal details are in Haug (2013).

Undefined objects: We model partiality through undefined objects. For each type $\alpha$ (simple or complex) there is an undefined object $\#_\alpha$, which is the bottom of a semi-lattice $\langle D'_\alpha, \sqsubseteq_\alpha \rangle$, where $\sqsubseteq_\alpha$ is a partial order (definedness). $*_\alpha$ is a constant denoting the undefined object of type $\alpha$.

Non-monotonicity: $*_\alpha = *_\alpha$ for all $\alpha$, so the identity predicate is classical and non-monotonic: $*_t = *_t$ is true, but $\top = *_t$ is false.

Weak Kleene semantics for the connectives: Undefinedness is absurdity and propagates through formulae.

Beaver’s $\partial$ (Beaver 1992) for presuppositions: $\partial(\phi)$ is true iff $\phi$ is true, otherwise undefined.

Quantification over defined objects only: $\exists x. f(x)$ is true iff there is a defined object such that $f$ is true of it; false iff $f$ is false of all defined objects; and undefined iff $f$ is undefined for all defined objects. Dually, $\forall x. f(x)$ is true if $f$ is true of all defined objects; false if there is a defined object such that $f$ is false of it; and undefined iff $f$ is undefined for all defined objects.

Undefined objects have no properties except self-identity:

$P(a_1, \ldots, a_n) \rightarrow \exists x_1. \ldots \exists x_n. P(x_1, \ldots, x_n)$ where $P$ is a predicate distinct from identity.

3.2 Partial CDRT

3.2.1 States and registers

registers (type $\pi$) are like addresses of memory locations; at any given state of the system (here: the discourse), that memory location may hold a different value.

states (type $s$) are the possible states of the system (discourse)

the non-logical constant $\nu$ of type $(s(\pi, e))$, which in a given state $s$ maps all registers to an individual (possibly the undefined one); pairs of registers and states reconstruct DRT drefs

well-ordering of registers: we can always pick out the lowest unused register in a given state $s$, written $x_{s_1}$
We use an abbreviation to say that two states $i$ and $j$ differ at most in the inhabitants they assign to registers $\delta_1 \ldots \delta_n$.

$$i[\delta_1 \ldots \delta_n]j = \text{abr} \forall \delta ((\delta_1 \neq \delta \land \cdots \land \delta_n \neq \delta) \rightarrow \nu(i)(\delta) = \nu(j)(\delta))$$

Axioms for states:

1. $\exists s. \forall \delta. \neg \exists e. \nu(s)(\delta) = e$  \hspace{1cm} (empty state)
2. $\forall s. \forall e. \exists s'[x_{s1}]s' \land \nu(s')(x_{s1}) = e$  \hspace{1cm} (expansion)
3. $\forall s. \forall \delta. \forall \delta'. (\exists e. \nu(s)(\delta) = e \land \delta' < \delta) \rightarrow \exists e'. \nu(s)(\delta') = e'$  \hspace{1cm} (no gaps)

So in each state, $\nu$ maps a (possibly empty) initial subset of the set of registers defined inhabitants and any remaining registers to the undefined individual.

- Axiom 2 guarantees that we never “run out of states”.
- If there are $n$ individuals that satisfy an $n$-place predicate in our model, then there is also a state that has these individuals inhabiting $n$ registers.
- Quantification over states works as unselective quantification over individuals.

The intuition is that the state grows as the discourse continues. When an expression introduces a new discourse referent, that will always be $x_{s1}$, of type $(s\pi)$ – the first uninhabited register in its input context. So we avoid the problems with using free variables (as in DRT) or constants (as in classical CDRT). Whenever negation, quantification, conditionals or intensionality create temporary contexts, these will “branch off” from the main state and be closed. Notice that even anaphoric expressions will introduce discourse referents, which are subject to special constraints.

### 3.2.2 DRSs and conditions

We can view DRSs as abbreviations for terms in ordinary type theory, as in (35).

$$[o_\alpha \ldots o_{\alpha+n} | \Gamma_1, \ldots, \Gamma_\gamma] := \lambda i. \lambda o. \partial(i[x_{i1} \ldots x_{in}]o) \land \Gamma_1^*(o) \land \cdots \land \Gamma_\gamma^*(o)$$

(This is a deviation from (Haug 2013). We still use the subscript index to track the relative ordering of registers in the full language, but we also use the letter to track the state in which the register is interpreted.)

$\Gamma^*$ is the formula that results from making the same substitutions in $\Gamma$ as we did in the universe of the DRS containing $\Gamma$ and expanding the result according to the following rules:

$$R(\delta_1, \ldots, \delta_n) \quad \lambda i. R(\nu(i)(\delta_1), \ldots, \nu(i)(\delta_n))$$

$\delta_1$ is $\delta_2$ \hspace{1cm} $\lambda i. \nu(i)(\delta_1) = \nu(i)(\delta_2)$

not $K$ \hspace{1cm} $\lambda i. \neg \exists j. K(i)(j)$

$K$ or $L$ \hspace{1cm} $\lambda i. \exists j. K(i)(j) \lor L(i)(j)$

$K \Rightarrow L$ \hspace{1cm} $\lambda i. \forall j. L(i)(j) \rightarrow \exists k. L(j)(k)$

It can be seen that what we doing when we expand abbreviation is ‘injecting’ the DRT definition of verification (see e.g. (11) and (15)) into the object language.

$$\text{sequencing:} \; K ; K' \leadsto \lambda i. \lambda o. \exists k. K(i)(k) \land K'(k)(o)$$
So again, the intuition is that sequencing requires there to be an intermediate state that is the output of $K$ and the input to $K'$. The merging lemma guarantees that we can always get rid of this intermediate state and write the result as a single DRS. In the abbreviated language, we then need to renumber discourse referents, but only if they are declared in the second DRS:

(38) a. $[o_1|P(o_1)];[j_1|Q(j_1)] = [o_1 o_2|P(o_1), Q(o_2)]$

b. $[o_1|P(o_1)];[Q(o_1)] = [o_1|P(o_1), Q(o_1)]$

A similar renumbering principle holds when we embed DRS into others via negation, disjunction or the conditional. Unlike, (Muskens 1996), our abbreviated language is closed under sequencing, i.e. the result of sequencing two DRSs can always be written as a DRS.

Finally, truth can be defined analogously to the DRT definition, as the existence of an output state given the empty input context $(s_0)$, as in (39).

(39) $\exists o.\partial(s_0[x_{s_01}, \ldots x_{s_0n}, o]) \land \Gamma^*_1(o) \land \cdots \land \Gamma^*_\gamma(o)$

### 3.3 A discourse example

We are now ready to see how we can perform compositional analysis over the abbreviated language, without having to worry about the expansions.

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

Word meanings in (41). *in* has been type-raised following Muskens’ treatment of transitive verbs. (This shouldn’t be necessary and depends on your overall framework.)

(41) $\lambda P. \lambda x. (P(\lambda y. [\text{in}(x, y)]))$

$\lambda x. [\text{bar}(x)]$

Pedro $\lambda P. [o_1|\text{pedro}(o_1)]; P(o_1)$

a $\lambda P. \lambda Q. [o_1]; P(o_1); Q(o_1)$

(42) $[o_1 o_2|\text{pedro}(o_1), \text{bar}(o_2), \text{in}(o_1, o_2)]$

Observe the renumbering in the uppermost functional application! The state at this stage can be represented as in (43), where the double circle represents the “grafting point” for the further discourse.
Lexical entries for the second sentence are as in (44). Overlining marks a register as anaphoric. (This corresponds to the predicate ant in (Haug 2013). We will return to its exact meaning). Again, there is type-raising (in transitive verbs).

(44) every $\lambda P.\lambda Q.[ | ([o_1] ; P(o_1)) \Rightarrow Q(o_1) ]$
woman $\lambda x.[ | woman(x) ]$
who $\lambda P.\lambda Q.\lambda x.P(x) ; P(x)$
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 $\lambda P.[o_1| \partial(male(o_1))] ; P(o_1)$

Straightforward composition gives us the sentence meaning in (45):

(45) $[ | [k_1, k_2| woman(k_1), man(k_2), date(k_1, k_2)] \Rightarrow [l_3| despise(l_1, l_3), \partial(male(l_3))] ]$

Observe that $l_1 = k_1$, so the reference of the relative pronoun is induced by the grammar and is not anaphoric.

Sequencing with the previous discourse in (42) involves some renumbering and yields (46).

(46) $[o_1, o_2| Pedro(o_1), bar(o_2), in(o_1, o_2),$
$k_3, k_4| woman(k_3), man(k_4), date(k_3, k_4)] \Rightarrow [l_5| despise(l_3, l_5), male(l_5)] ]$

This can be visualized as in (47), where we have grafted a branch onto (43). The octagon indicates an anaphoric discourse referent.

(47)

The final sentence of (40) straightforwardly gets the representation in (48).

(48) $[o_1| sad(o_1), \partial(male(o_1))]$

We merge everything and get the DRS in (49).
The drefs can be visualized as in (50).

(50)

- We preserve the notion of accessibility between DRSs that we have in DRT. This is a partial order which can be read off geometrically (leftwards and upwards) in (49).
- Register ordering (which is total) also gives us accessibility inside universes.
- Together, these gives us accessibility as a partial order on discourse referents (construed as pairs of states and registers), as illustrated in (50).
- Unlike DRT, we can now merge DRSs without resolving anaphora, because grafting onto (50) does not increase the set of accessible antecedents for l_5 and o_3.

4 Anaphora

- A holistic, discourse perspective on anaphora (rather than context-based).
- Anaphoric relations as relations between words (more precisely, syntactic tokens).
- Modelled as a function $\mathcal{R}$ taking anaphoric expressions to their antecedents.
- A discourse interpretation is a tuple $\langle K, P \rangle$ where $K$ is a DRS and $P$ is a set of pragmatic enrichments including $\mathcal{R}$.

Here is one example:

(51)
\( \mathcal{R} \) is isomorphic to a function \( \mathcal{A} \) between drefs, i.e. pairs of states and registers. Given the indexes in (49) we get (52).

\[
(52) \quad \mathcal{A} : \{l_5 \mapsto k_4, o_3 \mapsto o_1\}
\]

We can now capture dref overlining as two constraints on anaphoric resolution:

- the antecedent should be coreferent with the anaphor
- the antecedent should be accessible to the anaphor

Formally:

\[
(53) \quad \nu(s)(x) = \nu(s)(\mathcal{A}(s)(x)) \land \mathcal{A}(s)(x) < x
\]

In Haug (2013), these conditions were added to the DRS in whose universe the anaphoric register is introduced. This gives the wrong result in intensional contexts, because the anaphoric relationship ends up as part of modal context. As an alternative, we can consider overlining to abbreviate a predicate \textit{ant} singling out anaphoric discourse referents (pairs of states and registers) and amend the truth definition as in (54) to lift the condition on anaphoricity to the main DRS.

\[
(54) \quad \forall \delta, \forall s. \text{ant}(s)(\delta) \rightarrow \partial(\nu(s)(x) = \nu(s)(\mathcal{A}(s)(x)) \land \mathcal{A}(s)(x) < x) \land \ldots
\]

This gives us a model-theoretic characterization of the monotonic part of the meaning of an anaphoric expression, viz. that it should corefer with an accessible antecedent. At the same time, we keep the pragmatic resolution itself, \( \mathcal{R} \), out of the monotonic content.

### 4.1 Reference to embedded discourse referents

There is another situation we need to capture:

\[
(55) \quad \text{Pedro}_1 \text{ is}_2 \text{ in}_3 \text{ a}_4 \text{ bar}_5. \text{ Every}_6 \text{ woman}_7 \text{ who}_8 \text{ ever}_9 \text{ dated}_10 \text{ a}_11 \text{ man}_12 \text{ despises}_13 \text{ him}_14. \text{ He}_15 \text{ is}_16 \text{ sad}_17.
\]

\( \text{He}_{15}/o_3 \) can corefer with \( \text{he}_{14}/k_5 \) because \( \text{he}_{14}/k_5 \) corefers with \( \text{Pedro}_1/o_1 \), which belongs in the main DRS. In other words, because \( k_5 \) corefers with a dref in the main DRS, it gets ‘lifted’ to the main DRS and becomes accessible there.

By the transitivity of identity, this situation is identical to one in which \( o_3 \) takes \( o_1 \) as its antecedent directly:

\[
(56) \quad \text{Pedro}_1 \text{ is}_2 \text{ in}_3 \text{ a}_4 \text{ bar}_5. \text{ Every}_6 \text{ woman}_7 \text{ who}_8 \text{ ever}_9 \text{ dated}_10 \text{ a}_11 \text{ man}_12 \text{ despises}_13 \text{ him}_14. \text{ He}_15 \text{ is}_16 \text{ sad}_17.
\]

DRT (and list-based approaches) are forced to adopt such an analysis, but it is hard to square with a realistic account of anaphoric resolution.

To solve this, we can have pragmatics supply (55), but map that onto the truth-conditionally equivalent (56). Formally, we map map \( \mathcal{R} \) onto \( \mathcal{R}^* \) as follows:
\[
R^*(s)(i) = \begin{cases} 
R(i) & \text{if } i \text{ is inhabited in } s \\
R^*(s)(R(i)) & \text{if } R(i) \text{ is defined} \\
\text{undefined} & \text{otherwise}
\end{cases}
\]

Now we can set \(R(15)=14\), resulting in \(R^*(15)=R(14)\) iff \(R(14)\) refers to some syntactic token that introduces a discourse referent in the main DRS.

### 4.2 Bridging and split antecedents

The semantics of anaphoricity in (53)-(54) enforces identity between an anaphor and its antecedent. This is too strong if we want to deal with bridging (58) and split antecedence (59).

(58)  The\(_1\) bus arrived. The\(_2\) driver was drunk.

(59)  John\(_1\) met Mary\(_2\) and they\(_3\) went for a drink.

We need a richer notion of antecedence: to capture bridging, the value of \(R\) should be a pair consisting of an antecedent and a coreference relation (defaulting to identity) between the anaphor and the antecedent. E.g. for (58).

(60)  \(R(2) = \langle 1, \lambda x.\lambda y.\text{driver}(x, y) \rangle\)

If we write \(\text{antcd}\) for the antecedent – i.e. \(\text{fst}(A(s)(x))\) – and \(B\) – i.e. \(\text{snd}(A(s)(x))\) – for the coreference relation, we can define anaphoricity as in (61).

(61)  \(B(\nu(s)(\text{antcd}), \nu(s)(x)) \land \text{antcd} < x\)

We also need to alter our definition of \(R^*\), to compose relations along the path. Again using \(\text{antcd}\) (for \(\text{fst}(R(s)(i))\)) and \(B\) (for \(\text{snd}(R(s)(i))\)), we get (62).

(62)  \(R^*(s)(i) = \begin{cases} 
R(i) & \text{if } i \text{ is inhabited in } s \\
\langle \text{fst}(R^*(s)(\text{antcd})), \text{snd}(R^*(s)(\text{antcd}) \circ B) \rangle & \text{if } R(i) \text{ is defined} \\
\text{undefined} & \text{otherwise}
\end{cases}\)

Here is an example:

(63)  Ivan’s car\(_1\) does not start. If it’s not due to the alarm system, the\(_2\) engine must have broken down. It\(_3\) has already malfunctioned several times before, and Ivan has been thinking of replacing it.

Assume \(R = \{2 \mapsto \langle 1, \text{engine} \rangle, 3 \mapsto \langle 2, = \rangle\}\). We compute \(R^*(o)(3)\):

(64)  \(R^*(o)(3) \equiv \langle \text{fst}(R^*(o)(2)), \text{snd}(R^*(o)(2) \circ =) \rangle \equiv \langle 1, \text{engine} \circ = \rangle \equiv \langle 1, \text{engine} \rangle\)

This is not a theory of bridging, but provides a formal language that could be the basis for such a theory. Similarly PCDRT, like DRT, is no theory of anaphoric resolution, just a formal underpinning. For split antecedence we need the first element of the pair to be a set of antecedents, plus a theory of plurals.
5 Conclusions

Unresolved anaphora receive a model-theoretic interpretation (unlike other dynamic theories)
Constraints on anaphora are also captured model-theoretically (as in DRT)
The overwrite problem disappears (as in array-based approaches)
Separation of monotonic and non-monotonic content: updating $\langle K, P \rangle$ with $K'$ yields $\langle K ; K', P' \rangle$ where $;$ is monotonic but $P \rightarrow P'$ isn’t necessarily

References


