Dynamic Semantics Lite

Martin Jansche

Linguistics 780
October 27, 1998

As an appetizer, we will review the semantics of Predicate Logic in Section 1. In Section 2 we digest the first order system of Dynamic Predicate Logic. In Section 3 we order more.

1 Predicate Logic

A model of a particular language of First-Order Predicate Logic (PL) is the usual pair $\langle D, F\rangle$ consisting of a domain and an interpretation function that respects the choice of non-logical constants of the language. The semantics of a particular language of PL is stated with respect to such a model M and the set V of variables used in the language.

Terms are interpreted with respect to a model $M = \langle D, F\rangle$ and an assignment function $g \in D^V$:

Definition 1 (interpretation of terms)

1. $[\![c]\!]^{M,g} := F(c)$ iff c is a constant
2. $[\![v]\!]^{M,g} := g(v)$ iff v is a variable

Formulas are assigned truth-conditions rather than truth values:

Definition 2 (PL interpretation of atomic formulas)

1. ${}_{\text{PL}}[\![R^n t_1 \dots t_n]\!]^M := \{g \in D^V \mid \langle [\![t_1]\!]^{M,g}, \dots, [\![t_n]\!]^{M,g}\rangle \in F(R)\}$
2. ${}_{\text{PL}}[\![t_1 = t_2]\!]^M := \{g \in D^V \mid [\![t_1]\!]^{M,g} = [\![t_2]\!]^{M,g}\}$
3. ${}_{\text{PL}}[\![\top]\!]^M := D^V$
4. ${}_{\text{PL}}[\![\bot]\!]^M := \emptyset$

Definition 3 (PL interpretation of propositional connectives)

1. ${}_{\text{PL}}[\![\neg\phi]\!]^M := D^V \setminus {}_{\text{PL}}[\![\phi]\!]^M$
2. ${}_{\text{PL}}[\![(\phi \wedge \psi)]\!]^M := {}_{\text{PL}}[\![\phi]\!]^M \cap {}_{\text{PL}}[\![\psi]\!]^M$
3. ${}_{\text{PL}}[\![(\phi \vee \psi)]\!]^M := {}_{\text{PL}}[\![\phi]\!]^M \cup {}_{\text{PL}}[\![\psi]\!]^M$
4. ${}_{\text{PL}}[\![(\phi \rightarrow \psi)]\!]^M := (D^V \setminus {}_{\text{PL}}[\![\phi]\!]^M) \cup {}_{\text{PL}}[\![\psi]\!]^M$

To interpret the first-order quantifier symbols we need to define a relation that holds between sufficiently similar assignment functions:

Definition 4 (similarity of assignment functions)
For any $x \in V$ define the relation $[x]$ on D^V to be

$$\{\langle g, h\rangle \in (D^V \times D^V) \mid \forall v \in (V \setminus \{x\})\ g(v) = h(v)\}.$$

Of two assignment functions $g, h \in D^V$ such that $g[x]h$ we say that g is identical to h except for possibly assigning a different value to x.

Now we can interpret the quantified formulas of PL:

Definition 5 (PL interpretation of quantified formulas)

1. ${}_{\text{PL}}[\![\exists x\phi]\!]^M := \{g \in D^V \mid \exists h(g[x]h \wedge h \in {}_{\text{PL}}[\![\phi]\!]^M)\}$
2. ${}_{\text{PL}}[\![\forall x\phi]\!]^M := \{g \in D^V \mid \forall h(g[x]h \rightarrow h \in {}_{\text{PL}}[\![\phi]\!]^M)\}$

At this point we introduce the derived notion of truth:

Definition 6 (PL truth and validity)

1. ϕ is true (false) in a model M with respect to g, written $M, g \models \phi$ ($M, g \not\models \phi$), iff it is (not) the case that $g \in {}_{\text{PL}}[\![\phi]\!]^M$.
2. ϕ is true in a model M, written $M \models \phi$, iff ${}_{\text{PL}}[\![\phi]\!]^M = D^V$.
3. ϕ is false in a model M, written $M \not\models \phi$, iff ${}_{\text{PL}}[\![\phi]\!]^M = \emptyset$.
4. ϕ is valid (a contradiction), written $\models \phi$ ($\not\models \phi$), iff it is true (false) in all models.

Finally, note an important fact that will figure prominently in a following proof and in the comparison of PL and DPL.

Fact 1 $\models \forall x(\phi \rightarrow \psi)$ iff $\models \exists x\phi \rightarrow \psi$ provided x is not free in ψ.
(See e.g. [H82, p. 50] or [GS91, §3.4].)

Whereas this equivalence holds only in PL if the variable bound in the antecedent does not occur freely in the consequent, that last proviso can be dropped in the system of Dynamic Predicate Logic, which was explicitly designed to account for so-called sage-plant anaphora.

2 First order: Dynamic Predicate Logic

2.1 Semantics of DPL

Dynamic Predicate Logic (DPL) differs from classical (static) Predicate Logic in that it uses a different active ingredient. While ${}_{\mathrm{PL}}[\![\ ]\!]^M$ is a function $\mathcal{L} \rightarrow \wp(D^V)$ that maps formulas of Predicate Logic to sets of assignment functions, the semantics of DPL is given in terms of sets of pairs of assignment functions by a function ${}_{\mathrm{DPL}}[\![\ ]\!]^M : \mathcal{L} \rightarrow \wp(D^V \times D^V)$.

Note that Groenendijk and Stokhof [GS91] insist on the two functions having the same domain, i.e., DPL and PL assign potentially different denotations to the formulas of a language of Predicate Logic. However, in order to keep things compatible with Dynamic Montague Grammar [GS90] in Section 3, I'll use different symbols for the DPL connectives, giving it implicitly a slightly different syntax from that of PL.

For each well-formed DPL formula ϕ its denotation ${}_{\mathrm{DPL}}[\![\phi]\!]^M$ can be seen as a relation on the set D^V. Intuitively, the relation holds between two assignment functions (members of D^V) if the second is the result of updating the information encoded in the first with the information contained in ϕ. For ease of exposition, I'll drop all sub- and superscripts and simply write $[\![\phi]\!]$ for the DPL denotation of ϕ relative to some relevant model. We can then talk about two assignment functions being related by writing e.g. $g[\![\phi]\!]h$, meaning that g can be continued successfully with ϕ resulting in h.

The central notion of all dynamic theories of meaning is that conjunction is interpreted as some sort of sequencing operation. In the relational semantics of DPL, dynamic conjunction ; (written $\wedge$ in [GS91]) is interpreted as relational composition:

Definition 7 (DPL interpretation of conjunction)
$[\![(\phi\ ;\ \psi)]\!] := [\![\psi]\!] \circ [\![\phi]\!]$, or in other terms, $g[\![(\phi\ ;\ \psi)]\!]h$ iff $\exists j(g[\![\phi]\!]j \wedge j[\![\psi]\!]h)$.

Since relational composition is associative but not commutative, so (and neither, respectively) is dynamic conjunction.

The transportation of information via assignment functions is already very reminiscent of Heim's [H82] File Change Semantics (FCS). As in FCS, an atomic formula will serve to eliminate incompatible assignment functions,

retaining only those that already support the (static) meaning of the formula. For example, $[\![Mx]\!]$ lets all assignment functions that map x to some entity in the extionsion of M pass through unaltered, filtering out all incompatible assignments. In general, an atomic formula α denotes some partial identity relation on D^V:

Definition 8 (DPL interpretation of atomic formulas)
${}_{\text{DPL}}[\![\alpha]\!]^M := \Delta\,{}_{\text{PL}}[\![\alpha]\!]^M$, where $\Delta X := \{\langle x, y\rangle \in (X \times X) \mid x = y\}$, or in other tems, $g[\![\alpha]\!]h$ iff $g = h$ and $g \in {}_{\text{PL}}[\![\alpha]\!]$.

Before we introduce more connectives, let's think about how to define the usual notions of truth and validity:

Definition 9 (DPL truth and validity)

1. ϕ is true (false) in a model M with respect to g, written $M, g \models \phi$ ($M, g \not\models \phi$), iff it is (not) the case that $\exists h\ g[\![\phi]\!]h$.
2. ϕ is true in a model M, written $M \models \phi$, iff the relation $[\![\phi]\!]$ is serial.
3. ϕ is false in a model M, written $M \not\models \phi$, iff the relation $[\![\phi]\!]$ is empty.
4. ϕ is valid (a contradiction), written $\models \phi$ ($\not\models \phi$), iff it is true (false) in all models.

The notion of truth with respect to an assignment can be grammaticized and integrated into the syntax as a new connective, namely $\Diamond$ (read "might"):

Definition 10 (DPL interpretation of continuation modality)
${}_{\text{DPL}}[\![\Diamond\phi]\!]^M := \Delta\{g \in D^V \mid M, g \models \phi\}$, or in other terms, $g[\![\Diamond\phi]\!]h$ iff $g = h$ and $\exists j\ g[\![\phi]\!]j$.

Because of the fact that $\models \Diamond\Diamond\phi$ iff $\models \Diamond\phi$, the connective $\Diamond$ defines a closure operation. Note that we could do the converse and define $\models$ in terms of $\Diamond$: $M \models \phi$ iff ${}_{\text{DPL}}[\![\Diamond\phi]\!]^M = \Delta(D^V)$.

Analogous to the notion of truth corresponding to a modality expressing the possibility of a successful continuation, we can also introduce a modality corresponding to falsehood into the syntax. We write $\sim\phi$ (where Groenendijk and Stokhof [GS91] have $\neg\phi$) to mean that any attempt to continue with ϕ would fail. This new connective is often called static negation, as it performs a static closure over its argument in addition to its truth-conditional impact.

Definition 11 (DPL interpretation of static negation)
$_{\mathrm{DPL}}[\![\sim\phi]\!]^M := \Delta\{g \in D^V \mid M, g \not\models \phi\}$, or in other terms, $g[\![\sim\phi]\!]h$ iff $g = h$ and $\neg\exists j\ g[\![\phi]\!]j$.

In fact, $\Diamond$ is now dispensable. We could define $\Diamond\phi :\equiv \sim\sim\phi$, because of the following observation:

Fact 2 $\models \Diamond\phi$ iff $\models \sim\sim\phi$.

Proof

$$\begin{aligned}
g[\![\sim\sim\phi]\!]h &\Leftrightarrow\ g = h \wedge \neg\exists j\ g[\![\sim\phi]\!]j \\
&\Leftrightarrow\ g = h \wedge \neg\exists j(g = j \wedge \neg\exists k\ g[\![\phi]\!]k) \\
&\Leftrightarrow\ g = h \wedge \forall j\neg(g = j \wedge \neg\exists k\ g[\![\phi]\!]k) \\
&\Leftrightarrow\ g = h \wedge \forall j(g = j \rightarrow \exists k\ g[\![\phi]\!]k) \\
&\Leftrightarrow\ g = h \wedge ((\exists j\ g = j) \rightarrow \exists k\ g[\![\phi]\!]k) \\
&\Leftrightarrow\ g = h \wedge \exists k\ g[\![\phi]\!]k
\end{aligned}$$

If a formula ϕ is atomic or of the form $\sim\psi$ or $\Diamond\psi$ we find that $[\![\phi]\!] \subseteq \Delta(D^V)$. Whenever a formula has this property, we call it a test.

Just as $\wedge$ and $\neg$ form the basis for defining all other PL connectives, so do ; and $\sim$ for DPL.

Definition 12 (DPL definition of propositional connectives)

1. $(\phi \Rightarrow \psi) :\equiv \sim(\phi\ ;\ \sim\psi)$ (Groenendijk and Stokhof [GS91] have $(\phi \rightarrow \psi)$)
2. $(\phi$ **or** $\psi) :\equiv (\sim\phi \Rightarrow \psi)$ (Groenendijk and Stokhof [GS91] have $(\phi \vee \psi)$)

Let's see how the semantics of internally dynamic implication $\Rightarrow$ works:

$$\begin{aligned}
g[\![\phi \Rightarrow \psi]\!]h &\Leftrightarrow\ g[\![\sim(\phi\ ;\ \sim\psi)]\!]h \\
&\Leftrightarrow\ g = h \wedge \neg\exists j\ g[\![(\phi\ ;\ \sim\psi)]\!]j \\
&\Leftrightarrow\ g = h \wedge \neg\exists j\exists k(g[\![\phi]\!]k \wedge k[\![\sim\psi]\!]j) \\
&\Leftrightarrow\ g = h \wedge \neg\exists j\exists k(g[\![\phi]\!]k \wedge k = j \wedge \neg\exists l\ k[\![\psi]\!]l) \\
&\Leftrightarrow\ g = h \wedge \forall j\forall k((g[\![\phi]\!]k \wedge k = j) \rightarrow \exists l\ k[\![\psi]\!]l) \\
&\Leftrightarrow\ g = h \wedge \forall k(\exists j(g[\![\phi]\!]k \wedge k = j) \rightarrow \exists l\ k[\![\psi]\!]l) \\
&\Leftrightarrow\ g = h \wedge \forall k((g[\![\phi]\!]k \wedge \exists j\ k = j) \rightarrow \exists l\ k[\![\psi]\!]l) \\
&\Leftrightarrow\ g = h \wedge \forall k(g[\![\phi]\!]k \rightarrow \exists l\ k[\![\psi]\!]l)
\end{aligned}$$

Note that we could in principle define externally dynamic implication as $g[\![\phi \Rrightarrow \psi]\!]h \Leftrightarrow \forall k(g[\![\phi]\!]k \rightarrow k[\![\psi]\!]h)$. Then it would be possible to define $(\phi \Rightarrow \psi) :\equiv \Diamond(\phi \Rrightarrow \psi)$.

Disjunction is completely static:

$$\begin{aligned} g[\![(\phi \textbf{ or } \psi)]\!]h &\Leftrightarrow g[\![(\sim\phi \Rightarrow \psi)]\!]h \\ &\Leftrightarrow g = h \wedge \forall k(g[\![\sim\phi]\!]k \rightarrow \exists l\ k[\![\psi]\!]l) \\ &\Leftrightarrow g = h \wedge \forall k((g = k \wedge \neg\exists j\ g[\![\phi]\!]j) \rightarrow \exists l\ k[\![\psi]\!]l) \\ &\Leftrightarrow g = h \wedge (\exists k(g = k \wedge \neg\exists j\ g[\![\phi]\!]j) \rightarrow \exists l\ g[\![\psi]\!]l) \\ &\Leftrightarrow g = h \wedge (\neg\neg\exists j\ g[\![\phi]\!]j \vee \exists l\ g[\![\psi]\!]l) \\ &\Leftrightarrow g = h \wedge \exists j(g[\![\phi]\!]j \vee g[\![\psi]\!]j) \end{aligned}$$

The following important fact illustrates the close connection between dynamic conjunction and implication:

Fact 3 $\models (\phi \Rightarrow (\psi \Rightarrow \chi))$ iff $\models ((\phi\ ;\ \psi) \Rightarrow \chi)$.

Proof

$$\begin{aligned} & g[\![(\phi \Rightarrow (\psi \Rightarrow \chi))]\!]h \\ \Leftrightarrow\ & g = h \wedge \forall j(g[\![\phi]\!]j \rightarrow \exists k\ j[\![(\psi \Rightarrow \chi)]\!]k) \\ \Leftrightarrow\ & g = h \wedge \forall j(g[\![\phi]\!]j \rightarrow \exists k(j = k \wedge \forall l(j[\![\psi]\!]l \rightarrow \exists m\ l[\![\chi]\!]m))) \\ \Leftrightarrow\ & g = h \wedge \forall j(g[\![\phi]\!]j \rightarrow \forall l(j[\![\psi]\!]l \rightarrow \exists m\ l[\![\chi]\!]m)) \\ \Leftrightarrow\ & g = h \wedge \forall l\,\forall j(g[\![\phi]\!]j \rightarrow (j[\![\psi]\!]l \rightarrow \exists m\ l[\![\chi]\!]m)) \\ \Leftrightarrow\ & g = h \wedge \forall l\,\forall j((g[\![\phi]\!]j \wedge j[\![\psi]\!]l) \rightarrow \exists m\ l[\![\chi]\!]m) \\ \Leftrightarrow\ & g = h \wedge \forall l(\exists j(g[\![\phi]\!]j \wedge j[\![\psi]\!]l) \rightarrow \exists m\ l[\![\chi]\!]m) \\ \Leftrightarrow\ & g = h \wedge \forall l(g[\![(\phi\ ;\ \psi)]\!]l \rightarrow \exists m\ l[\![\chi]\!]m) \\ \Leftrightarrow\ & g[\![((\phi\ ;\ \psi) \Rightarrow \chi)]\!]h \end{aligned}$$

Finally let's look at quantified formulas. These require the auxiliary notion of a variable update. We write $g[\![\oplus x]\!]h$ to indicate that the assignment function h is just like g except that any information about the value of the variable x is lost (because of this, $\oplus x$ is occasionally called to bring about an information downdate) and a new information regarding x can be accumulated afterwards.

Definition 13 (DPL interpretation of variable update)
$[\![\oplus x]\!] := [x]$.

Quantification is defined in terms of variable update and the internally dynamic connectives:

Definition 14 (DPL definition of quantifiers)

1. $\mathcal{E}x\phi :\equiv (\oplus x\ ;\ \phi)$ (Groenendijk and Stokhof [GS91] have $\exists x\phi$)
2. $\mathcal{A}x\phi :\equiv (\oplus x \Rightarrow \phi)$ (Groenendijk and Stokhof [GS91] have $\forall x\phi$)

Now we are in a position to digest the key equivalence of DPL.

Fact 4 $\models \mathcal{A}x(\phi \Rightarrow \psi)$ iff $\models (\mathcal{E}x\phi \Rightarrow \psi)$.

Proof

$$\begin{array}{rcl} \mathcal{A}x(\phi \Rightarrow \psi) & \equiv & (\oplus x \Rightarrow (\phi \Rightarrow \psi)) \\ & \equiv & ((\oplus x \,;\, \phi) \Rightarrow \psi) \\ & \equiv & (\mathcal{E}x\phi \Rightarrow \psi) \end{array}$$

The above equivalence provides a formal counterpart to the alleged equivalence of sage-plant sentences such as the following:

- If Rose-Marie^1 buys a^2 sage-plant, she$_1$ eats it$_2$.
- Every sage-plant that Rose-Marie^1 buys she$_1$ eats.

Compare what $\mathcal{A}x(\phi \Rightarrow \psi)$ means with the meaning of a universally quantified formula in FCS [H82, p. 363f.]:

$$\begin{array}{rcl} a[\![\mathcal{A}x(\phi \Rightarrow \psi)]\!]z & \Leftrightarrow & a[\![((\oplus x \,;\, \phi) \Rightarrow \psi)]\!]z \\ & \Leftrightarrow & a = z \wedge \forall b'(a[\![(\oplus x \,;\, \phi)]\!]b' \rightarrow \exists c\; b'[\![\psi]\!]c) \\ & \Leftrightarrow & a = z \wedge \forall b'(\exists b(a[\![\oplus x]\!]b \wedge b[\![\phi]\!]b') \rightarrow \exists c\; b'[\![\psi]\!]c) \\ & \Leftrightarrow & a = z \wedge \forall b \forall b'((a[x]b \wedge b[\![\phi]\!]b') \rightarrow \exists c\; b'[\![\psi]\!]c) \end{array}$$

2.2 Problems with DPL

Many familiar properties of classical Predicate Logic are lost. For example:

1. DPL entailment is not reflexive, e.g., $(Px \,;\, \mathcal{E}xQx) \Rightarrow (Px \,;\, \mathcal{E}xQx)$ is not a valid DPL formula, which becomes more transparent if we write it equivalently as $(Px \,;\, \mathcal{E}yQy) \Rightarrow (Py \,;\, \mathcal{E}zQz)$.
2. DPL entailment isn't transitive either.
3. Renaming of variables isn't easy, since they can be bound an unbounded distance to the right of an existential quantifier. Moreover, $[\![\mathcal{E}xPx]\!] \neq [\![\mathcal{E}yPy]\!]$.

A proof theory for DPL is developed in [vE98].

Apart from the above technical problems, DPL has the following more fundamental deficiencies:

1. Like many other dynamic semantic theories it assumes that indexation of pronouns is done by some component prior to semantic interpretation. This is not easily remedied.

2. DPL semantics is based on the naïve assumption that pairs of sage-plant sentences such as 'If a farmer owns a sage-plant, he waters it' and 'Every farmer who owns a sage-plant waters it' are *identical* in meaning. Kamp makes the same assumption, only Heim doesn't.

3. Constraints on (un)familiarity of (in)definites à la Heim are missing.

4. Although DPL has a compositional semantics, its only important advantage over FCS is its clarity, perspicuity, and lucidity. A higher-order logic is needed to fully implement the compositional program. A system that tries to do that is Dynamic Montague Grammar.

3 Second order: Dynamic Montague Grammar

3.1 Introductory polemic

In their influential paper [GS82] Groenendijk and Stokhof gave a presentation of a semantics for questions whose clarity and lucidity stemmed to a large extent from their use of Ty2 rather than IL as the underlying logic. Ty2 makes quantification over indices (expressions of type s) explicit, and although it seems to be more verbose than Montague's IL, having the additional baggage around is actually a virtue, for IL carries it too but tries to hide it in the semantics.

In their equally influential paper [GS90] however, neither Groenendijk nor Stokhof seemed particularly keen on keeping up with the clarity and lucidity of their earlier work. Rather than making the complex interactions of states and discourse markers explicit in the underlying logic, they chose to hide it in the semantics of DIL, the Dynamic Intensional (a misnomer, as they readily admit) Logic behind Dynamic Montague Grammar (DMG). The only excuse for this obfuscation might be that they wanted to present DMG along the same lines as Montague Grammar was first presented.

For those of you already familiar with DMG based on DIL, the two key differences to the version of DMG I'm about to present (based on the higher-order logic TyME) are the absence of states and the strict extensionality. TyME replaces DIL's states (expressions of type s) with explicit assignment functions (of type $m \to e$) that assign individuals to discourse markers. Whereas DIL formulas are interpreted with respect to a designated

state, TyME formulas aren't, and as a consequence propositions have type $(m \to e) \to t$ in TyME, corresponding to t in DIL, where we make explicit that the truth of the proposition depends on a state (assignment) of type $m \to e$.

3.2 TyME

The extensional typed language TyME is in all but minor respects identical to Ty2. In addition to a type t for truth values, TyME has two other basic types, e for entities and m for discourse markers. Moreover, we introduce the following type abbreviations:

- s abbreviates $m \to e$, the type of states or assignment functions
- p abbreviates $s \to t$ or $(m \to e) \to t$, the type of static propositions
- d abbreviates $p \to p$ or $((m \to e) \to t) \to (m \to e) \to t$, the type of dynamic propositions

Semantically TyME is, for my taste, superior to DIL, as it can be defined entirely in terms of equality, abstraction, and application in the usual manner and doesn't need the postulates that DIL uses in order to incorporate the notion of a state.

3.3 Some definitions and some facts

Definition 15 (uparrow)
If neither p nor g occurs freely in ϕ:

$\uparrow : p \to d$
$\uparrow : p \to (p \to p)$
$\uparrow : (s \to t) \to ((s \to t) \to s \to t)$

$\uparrow\phi := \lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))$

Definition 16 (downarrow)
$\downarrow : d \to p$
$\downarrow : (p \to p) \to p$
$\downarrow : ((s \to t) \to s \to t) \to s \to t$

$\downarrow\Phi := \Phi(\lambda g.\, \top)$

Fact 5 ($\downarrow\uparrow$-elimination) $\downarrow\uparrow\phi = \phi$

Proof

$$
\begin{array}{ll}
 & \downarrow\uparrow\phi \\
= & \downarrow(\lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))) \\
= & [\lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))](\lambda h.\, \top) \\
\rightarrow_\beta & \lambda g.\, (\phi(g) \wedge [\lambda h.\, \top](g)) \\
\rightarrow_\beta & \lambda g.\, (\phi(g) \wedge \top) \\
= & \lambda g.\, \phi(g) \\
\rightarrow_\eta & \phi
\end{array}
$$

Fact 6 (failure of $\uparrow\downarrow$-elimination) $\uparrow\downarrow\Phi \neq \Phi$

Proof

$$
\begin{array}{ll}
 & \uparrow\downarrow\Phi \\
= & \uparrow(\Phi(\lambda h.\, \top)) \\
= & \lambda p.\, \lambda g.\, (\Phi(\lambda h.\, \top)(g) \wedge p(g))
\end{array}
$$

Set $\Phi := \lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(k))$, then continue:

$$
\begin{array}{ll}
= & \lambda p.\, \lambda g.\, ([\lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(k))](\lambda h.\, \top)(g) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ([\lambda g'.\, (\phi(g') \wedge [\lambda h.\, \top](k))](g) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ((\phi(g) \wedge [\lambda h.\, \top](k)) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ((\phi(g) \wedge \top) \wedge p(g)) \\
= & \lambda p.\, \lambda g.\, (\phi(g) \wedge p(g)) \\
=_\alpha & \lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(g'))
\end{array}
$$

Definition 17 (static negation)
If g does not occur freely in Φ:

$$\sim : d \rightarrow d$$
$$\sim : (p \rightarrow p) \rightarrow (p \rightarrow p)$$
$$\sim : ((s \rightarrow t) \rightarrow s \rightarrow t) \rightarrow (s \rightarrow t) \rightarrow s \rightarrow t$$

$$\sim\Phi := \uparrow(\lambda g.\, \neg[\downarrow\Phi](g))$$

N.B.: If $\neg$ would denote generalized negation, we could say $\sim\Phi := \uparrow\neg\downarrow\Phi$.

Definition 18 (dynamic conjunction)
If p does not occur freely in Φ or Ψ:

$$; : d \rightarrow d \rightarrow d$$
$$; : (p \rightarrow p) \rightarrow (p \rightarrow p) \rightarrow p \rightarrow p$$

$$\Phi\ ;\ \Psi := \lambda p.\, \Phi(\Psi(p))$$

Definition 19 (update)
What it means to update an assignment function:

$\mathsf{update} : m \to e \to s \to s$
$\mathsf{update} : m \to e \to (m \to e) \to m \to e$

$\mathsf{update}(d)(x)(g)(d') := g(d')$ provided $d \neq d'$
$\mathsf{update}(d)(x)(g)(d) := x$

Generously add syntactic sugar: $\{x/d\}g := \mathsf{update}(d)(x)(g)$

Definition 20 (dynamic existential quantifier)
If none of p, g, x occurs freely in Φ:

$\mathcal{E} : m \to d \to d$
$\mathcal{E} : m \to (p \to s \to t) \to p \to s \to t$

$\mathcal{E}d\Phi := \lambda p.\, \lambda g.\, \exists x \Phi(p)(\{x/d\}g)$

Definition 21 (remaining connectives)

1. internally dynamic implication: $(\Phi \Rightarrow \Psi) := \sim(\Phi\ ;\ \sim\Psi)$
2. static disjunction: $(\Phi \textbf{ or } \Psi) := (\sim\Phi \Rightarrow \Psi)$
3. static universal quanitifier: $\mathcal{A}d\Phi := \sim\mathcal{E}d\sim\Phi$

Fact 7 $(\mathcal{E}d\Phi \Rightarrow \Psi) = \mathcal{A}d(\Phi \Rightarrow \Psi)$

3.4 Dynamic Montague Grammar

Definition 22 (translation of basic expressions)
$f(a^i) := \lambda P.\, \lambda Q.\, \mathcal{E}d_i(P(d_i)\ ;\ Q(d_i))$
$f(man) := \lambda d.\, \uparrow\lambda g.\, \mathsf{man}(g(d)) = \lambda d.\, \lambda p.\, \lambda g.\, (\mathsf{man}(g(d)) \wedge p(g))$
$f(walks) := \lambda d.\, \uparrow\lambda g.\, \mathsf{walk}(g(d))$
$f(he_i) := \lambda Q.\, Q(d_i)$
$f(talks) := \lambda d.\, \uparrow\lambda g.\, \mathsf{talk}(g(d))$

$f(a^1\ man) = \lambda Q.\, \mathcal{E}d_1([\lambda d.\, \lambda p.\, \lambda g.\, (\mathsf{man}(g(d)) \wedge p(g))](d_1)\ ;\ Q(d_1))$
$= \lambda Q.\, \mathcal{E}d_1(\lambda p.\, \lambda g.\, (\mathsf{man}(g(d_1)) \wedge p(g))\ ;\ Q(d_1))$

$f(a^1\ man\ walks)$
$= \mathcal{E}d_1(\lambda p.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge p(g'))\ ;\ [\lambda d''.\,\lambda p''.\,\lambda g''.\,(\mathsf{walk}(g''(d'')) \wedge p''(g''))](d_1))$
$= \mathcal{E}d_1(\lambda p.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge p(g'))\ ;\ \lambda p''.\,\lambda g''.\,(\mathsf{walk}(g''(d_1)) \wedge p''(g'')))$
$= \mathcal{E}d_1\lambda p'.\,[\lambda p.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge p(g'))]([\lambda p''.\,\lambda g''.\,(\mathsf{walk}(g''(d_1)) \wedge p''(g''))](p'))$
$= \mathcal{E}d_1\lambda p'.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge [\lambda g''.\,(\mathsf{walk}(g''(d_1)) \wedge p'(g''))](g'))$
$= \mathcal{E}d_1\lambda p'.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge \mathsf{walk}(g'(d_1)) \wedge p'(g'))$
$= \lambda p.\,\lambda g.\,\exists x[\lambda p'.\,\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge \mathsf{walk}(g'(d_1)) \wedge p'(g'))](p)(\{x/d_1\}g)$
$= \lambda p.\,\lambda g.\,\exists x[\lambda g'.\,(\mathsf{man}(g'(d_1)) \wedge \mathsf{walk}(g'(d_1)) \wedge p(g'))](\{x/d_1\}g)$
$= \lambda p.\,\lambda g.\,\exists x(\mathsf{man}(\{x/d_1\}g(d_1)) \wedge \mathsf{walk}(\{x/d_1\}g(d_1)) \wedge p(\{x/d_1\}g))$
$= \lambda p.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge p(\{x/d_1\}g))$

$f(he_1\ talks) = [\lambda Q.\,Q(d_1)](\lambda d.\,\lambda p.\,\lambda g.\,(\mathsf{talk}(g(d)) \wedge p(g)))$
$= [\lambda d.\,\lambda p.\,\lambda g.\,(\mathsf{talk}(g(d)) \wedge p(g))](d_1)$
$= \lambda p.\,\lambda g.\,(\mathsf{talk}(g(d_1)) \wedge p(g))$

$f(a^1\ man\ walks.\ he_1\ talks) = f(a^1\ man\ walks)\ ;\ f(he_1\ talks)$
$= \lambda p.\,[\lambda p'.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge p'(\{x/d_1\}g))]([\lambda p''.\,\lambda g''.\,(\mathsf{talk}(g''(d_1)) \wedge p''(g''))](p))$
$= \lambda p.\,[\lambda p'.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge p'(\{x/d_1\}g))](\lambda g''.\,(\mathsf{talk}(g''(d_1)) \wedge p(g'')))$
$= \lambda p.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge [\lambda g''.\,(\mathsf{talk}(g''(d_1)) \wedge p(g''))](\{x/d_1\}g))$
$= \lambda p.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(\{x/d_1\}g(d_1)) \wedge p(\{x/d_1\}g))$
$= \lambda p.\,\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge p(\{x/d_1\}g))$

$\models f(a^1\ man\ walks.\ he_1\ talks)$
$\Leftrightarrow \downarrow f(a^1\ man\ walks.\ he_1\ talks) = [\lambda g.\,\top]$
$\Leftrightarrow [\lambda g.\,\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x))] = [\lambda g.\,\top]$
$\Leftrightarrow \forall g(\exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x)) \leftrightarrow \top)$
$\Leftrightarrow \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x))$

References

[vE98] Jan van Eijck. 1998. Axiomatizing dynamic logics for anaphora. Universiteit van Amsterdam, ILLC preprint series #LP-1998-07.

[GS82] Jeroen Groenendijk and Martin Stokhof. 1982. Semantic analysis of *wh*-complements. *Linguistics & Philosophy* 5.

[GS90] Jeroen Groenendijk and Martin Stokhof. 1990. Dynamic Montague Grammar.

[GS91] Jeroen Groenendijk and Martin Stokhof. 1991. Dynamic Predicate Logic. *Linguistics & Philosophy* 14.

[H82] Irene R. Heim. 1982. The semantics of definite and indefinite noun phrases. UMass Amherst dissertation.