TIM FERNANDO

A TYPE REDUCTION FROM PROOF-CONDITIONAL TO DYNAMIC SEMANTICS

(Received 1 December 1998; received in revised version 23 November 2000)

ABSTRACT. Dynamic and proof-conditional approaches to discourse (exemplified by Discourse Representation Theory and Type-Theoretical Grammar, respectively) are related through translations and transitions labeled by first-order formulas with anaphoric twists. Type-theoretic contexts are defined relative to a signature and instantiated model-theoretically, subject to change.

KEY WORDS: proof-conditional semantics, dynamic semantics, types, anaphora, discourse, natural language, truth, bisimulation

1. Introduction

Among the formal approaches to discourse semantics that have attracted some attention are "dynamic" formalisms such as *Discourse Representation Theory* (DRT, Kamp and Reyle [14]) and proof-conditional programs such as *Type-Theoretical Grammar* (TTG, Ranta [24]). Stretching traditions in formal logic, the former suggest

(**D**) a shift from truth to input/output interpretations

and the latter

(**P**) the insertion of proofs into well-formed formulas.

The present paper focuses on applications of (**D**) and (**P**) to anaphora, linking the approaches in a first-order setting. Previous comparisons (e.g. Ahn and Kolb [2]) are rigorously developed by taking up (on the one hand) model-theoretic interpretations and (on the other) type systems.

A simple (but telling) illustration of some of the differences at stake is provided by Geach's notorious donkey sentence, (s), read as (t).

(s) If a farmer owns a donkey, he beats it.
(t) $\forall x \forall y (\mathit{farmer}(x) \wedge \mathit{donkey}(y) \wedge \mathit{owns}(x, y) \supset \mathit{beats}(x, y))$.

A common point of departure for the approaches is the ill-formed formula (q) below, obtained from a piecemeal translation of (s), interpreting an indefinite description such as a farmer through existential quantification.

(q) $(\exists x \in farmer)(\exists y \in donkey)\ owns(x, y) \supset beats(\underline{\text{he}}, \underline{\text{it}})$.

As reformulated in *Dynamic Predicate Logic* (DPL, Groenendijk and Stokhof [11]), DRT reduces $\underline{\text{he}}$ to the variable x and $\underline{\text{it}}$ to the variable y, resulting in (d).

(d) $(\exists x \in farmer)(\exists y \in donkey) owns(x, y) \supset beats(x, y)$.

The trick is then to treat $\exists z$ as a random assignment to z (familiar from *Quantified Dynamic Logic*, Harel [12]), lifting the usual Tarskian satisfaction relation $\models$ between first-order formulas φ and models M to input/output relations $\|\varphi\|_M$ on functions f, f' from variables to objects in M such that roughly put,

$$f \|\varphi\|_M f' \quad \text{iff} \quad (M, f) \models \varphi \text{ with } \exists\text{-witnesses in } f'.$$

(Details in Section 3.1 below.) By contrast, TTG extracts $\underline{\text{he}}$ and $\underline{\text{it}}$ from a constructive proof of (q)'s premiss, $(\exists x \in farmer)(\exists y \in donkey)$ $owns(x, y)$. That is, interpreting existential quantification $(\exists x \in A)B$ as the dependent sum/product

$$\left(\sum x : A\right)B = \{\langle a, b\rangle | a \text{ in } A \text{ and } b \text{ in } B[x \mapsto a]\}$$

(generalizing Cartesian products) and implication $A \supset B$ as the dependent function space

$$\left(\prod z : A\right)B \ = \ \{\text{functions mapping } a \text{ in } A \text{ to some } b \text{ in } B[z \mapsto a]\},$$

TTG reduces (q) to (p), where lz and $l(rz)$ pick out the farmer and donkey witnesses encoded in a constructive proof z of (q)'s premiss (l and r being $\sum$'s *l*eft and *r*ight projections, respectively).

(p) $(\prod z : (\sum x : farmer)(\sum y : donkey)\ owns(x, y))\ beats(lz, l(rz))$.

Notice that just as the clause a farmer owns a donkey is hypothetical within (s), so too is the proof z in (p), raising the question of instantiating such variable proofs model-theoretically.

1.1. *Signatures, Rules and Interpretations*

For a systematic comparison of dynamic with proof-conditional semantics, it is convenient to fix a (many-sorted, relational) *signature* L consisting of *sorts* $(U, \ldots)$ and *relation symbols* $(R, \ldots)$ with associated *arities*. Let us agree to write $U \in L_0$ to mean U is a sort in L, and $R \in L(U_1 \cdots U_n)$ to mean that R is a relation symbol in L with n arguments of sorts U_1 to U_n. The *L-formulas* φ for dynamic semantics are generated from an infinite set Var of variables $x_1 \ldots$ according to

$$\begin{aligned}\varphi \ ::= \ & \bot \mid R(x_1 \ldots x_n) \mid \varphi \wedge \varphi \mid \varphi \supset \varphi \mid (\exists x \in U)\varphi \mid \\ & (\forall x \in U)\varphi,\end{aligned}$$

where $U \in L_0$ and $R \in L(U_1 \cdots U_n)$ for some $U_1, \ldots, U_n \in L_0$.[1] With the type-theoretic approach, the expressions are somewhat more complicated, and are formed using a set of *rules* specifying what *contexts* are and what judgments they license. Very roughly, $\wedge$ and $\exists$ turn into $\sum$, while $\supset$ and $\forall$ become $\prod$. And whereas a signature L fixes the possibilities for U and R, a rule set $\mathcal{D}$ determines what type-theoretic terms there are in addition to the variables x (in Var).

Just as the L-formulas φ of dynamic semantics are interpreted relative to the usual (many-sorted first-order) L-models, the type-theoretic L-expressions can be interpreted on the basis of the following notion. An *L-proof interpretation* (or *L-interpretation*, for short) is a function $[\![\cdot]\!]$ mapping

(i) every $U \in L_0$ to a set $[\![U]\!]$

and

(ii) every $R \in L(U_1 \cdots U_n)$ and $u_1 \in [\![U_1]\!], \ldots, u_n \in [\![U_n]\!]$ to a set $[\![R, u_1 \cdots u_n]\!]$.

An L-interpretation $[\![\cdot]\!]$ induces the many-sorted L-model $\mathcal{M}[\![\cdot]\!] = M$ that interprets $U \in L_0$ as $U_M = [\![U]\!]$ and $R \in L(U_1 \cdots U_n)$ as

$$R_M = \{u_1 \cdots u_n \in [\![U_1]\!] \times \cdots \times [\![U_n]\!] \mid [\![R, u_1 \cdots u_n]\!] \neq \emptyset\},$$

the intuition being that $[\![R, u_1 \cdots u_n]\!]$ consists of proofs of $R(u_1 \cdots u_n)$. Conversely, an arbitrary L-model M can be beefed up to an L-interpretation $\mathcal{P}(M) = [\![\cdot]\!]_M$ by taking $[\![U]\!]_M = U_M$ and

$$[\![R, u_1 \cdots u_n]\!]_M = \begin{cases} \{(R, u_1 \cdots u_n)\} & \text{if } R_M(u_1 \ldots u_n), \\ \emptyset & \text{otherwise.} \end{cases}$$

Alternatively, if we allow types to intersect, we could fix a single object 0 and set $[\![R, u_1 \cdots u_n]\!]_M = \{0\}$ whenever $R_M(u_1 \ldots u_n)$. In either case, $\mathcal{M}(\mathcal{P}(M)) = M$, although we cannot expect $\mathcal{P}(\mathcal{M}[\![\cdot]\!]) \cong [\![\cdot]\!]$ (as the sets $[\![R, u_1 \cdots u_n]\!]$ could be neither empty nor singletons). Whether or not we should be interested in $[\![\cdot]\!]$ beyond $\mathcal{M}[\![\cdot]\!]$ is a natural question investigated below. Along the way, $[\![\cdot]\!]$ is extended to interpret type expressions more complicated than those expressing $R_M(u_1 \ldots u_n)$.

1.2. *Outline and Note*

Section 2 relates dynamic with proof-conditional semantics syntactically, exploring translations between first-order systems that equate (d) with (p). Section 3 takes up the semantics of these fragments, extensions to which are considered in Section 4. Section 5 concludes by returning to the bold proposals (**D**) and (**P**) above, characteristic of dynamic and proof-conditional semantics.

Let us note at the outset that (**D**) and (**P**) are of interest beyond the particular applications to anaphora considered below – or, for that matter, variants involving, for example, "weak" readings of the donkey sentence (s) that only require every donkey-owning farmer to beat some donkey s/he owns. The hope is, however, that the present case study might throw some light on what (**D**) and (**P**) could more generally mean.

2. TRANSLATIONS AND THE RULE SET $\mathcal{D}_\circ$

Fix an infinite set Var of variables x and a signature L. Let the set Tm° of *preterms* t consist of Var and its closure under the projections l and r

$$t ::= x \mid lt \mid rt.$$

Corresponding to the L-formulas φ (for dynamic semantics) generated in Section 1.1 are the *L-pretypes* A generated from preterms $t_1 \dots t_n$ according to

$$A ::= \bot \mid R(t_1 \dots t_n) \mid \left(\sum x : A\right)A \mid \left(\prod x : A\right)A \mid \left(\sum x : U\right)A \mid \left(\prod x : U\right)A.$$

The present section refines the notions of L-formula and L-pretype, making the syntactic correspondence between these precise.

2.1. *The Novel Variable Condition*

A technical condition on L-formulas that will play a crucial role below is to ban a variable x from being bound (either by $\forall$ or $\exists$) after a reference to x has already been made in a formula. More precisely, the set of L-formulas φ that respect the *novel variable condition* (NVC) is defined inductively as follows:

(i) the atomic L-formulas $\bot$ and $R(x_1 \dots x_n)$ respect NVC,
(ii) if φ and ψ both respect NVC, and *no* variable bound in ψ occurs in φ (free or bound), then both $\varphi \wedge \psi$ and $\varphi \supset \psi$ respect NVC,

(iii) if φ respects NVC and the variable x does not occur bound in φ, then both $(\forall x \in U)\varphi$ and $(\exists x \in U)\varphi$ respect NVC (for $U \in L_0$).

Thus, $((\exists x \in U)R(x)) \wedge S(x)$ respects NVC, but $R(x) \wedge (\exists x \in U)S(x)$ does *not*. In ordinary predicate logic (classical or intuitionistic), NVC is innocuous insofar as every formula can be assumed to respect NVC, by renaming bound variables if necessary (e.g. from $R(x) \wedge (\exists x \in U)S(x)$ to $R(x) \wedge (\exists y \in U)S(y)$). Renaming may change the meaning of a formula in the present applications to anaphora, however (concerning which, NVC follows a condition in Heim [13] that variables introduced for indefinites such as a farmer be novel). We will repeatedly require L-formulas to respect NVC throughout this section, reconsidering NVC semantically in Sections 3.1 and 5.2.

2.2. *From L-formulas to L-pretypes and Back*

Next, we specify a translation of L-formulas to L-pretypes, systematizing the translation of (d) to (p).[2] An L-formula φ will be translated relative to

(i) a well-ordering $<$ of the infinite set Var of variables, from which we can define for every proper subset X of Var, a variable $\nu_< X \in \mathsf{Var} - X$ as the $<$-least variable *not* in X (making $\nu_< X$ a distinguished variable that is X-novel)

and

(ii) a function θ from a finite subset of Var to the set Tm° of preterms.

The function θ need not fully specify translations of variables in φ. Indeed, writing $\varphi^<_\theta$ for the type-theoretic translation of φ, we will, with θ set trivially to $\emptyset$, arrange

$$(\mathsf{d})^<_\emptyset = (\mathsf{p}), \quad \text{where } z = \nu_<\{x, y\}.$$

Let

$$\perp^<_\theta = \perp,$$
$$R(x_1 \ldots x_n)^<_\theta = \begin{cases} R(\theta(x_1) \ldots \theta(x_n)) & \text{if } \{x_1 \ldots x_n\} \subseteq dom(\theta), \\ \uparrow & \text{otherwise,} \end{cases}$$

where $dom(\theta)$ is the domain of θ and $\uparrow$ means undefined,[3]

$$((\exists x \in U)\varphi)^<_\theta = \left(\sum x : U\right)\varphi^<_{\theta \cup \{(x,x)\}},$$
$$((\forall x \in U)\varphi)^<_\theta = \left(\prod x : U\right)\varphi^<_{\theta \cup \{(x,x)\}}$$

and

$$\begin{aligned}
&(1) \qquad (\varphi \wedge \psi)^{<}_{\theta} &=& \Big(\sum z : \varphi^{<}_{\theta}\Big) \psi^{<}_{\theta \cup \{(x,\alpha(z)) \mid (x,\alpha) \in new(\varphi)\}}, \\
&(2) \qquad (\varphi \supset \psi)^{<}_{\theta} &=& \Big(\prod z : \varphi^{<}_{\theta}\Big) \psi^{<}_{\theta \cup \{(x,\alpha(z)) \mid (x,\alpha) \in new(\varphi)\}},
\end{aligned}$$

where in (1) and (2),

$$z = \nu_{<}\{y \in \mathsf{Var} \mid y \text{ in } \varphi, \psi \text{ or } \theta\}$$

while $new(\varphi)$ specifies the anaphoric possibilities offered by φ as follows. The L-formulas $\bot$, $R(x_1 \ldots x_n)$, $(\forall x \in U)\varphi$ and $\varphi \supset \psi$ are "static" in that

$$\begin{aligned}
new(\bot) &= \emptyset \\
&= new(R(x_1 \ldots x_n)) \\
&= new((\forall x \in U)\varphi) \\
&= new(\varphi \supset \psi)
\end{aligned}$$

while $(\exists x \in U)\varphi$ and $\varphi \wedge \psi$ are "dynamic" in that

$$\begin{aligned}
new((\exists x \in U)\varphi) &= \{(x, l)\} \cup \{(y, \alpha r) \mid (y, \alpha) \in new(\varphi)\}, \\
new(\varphi \wedge \psi) &= \{(y, \alpha l) \mid (y, \alpha) \in new(\varphi)\} \cup \\
&\qquad \{(y, \alpha r) \mid (y, \alpha) \in new(\psi)\}
\end{aligned}$$

(recalling that l and r are $\sum$'s left and right projections). Applying this recipe to $(\mathsf{d})^{<}_{\emptyset} = (\mathsf{p})$ shows that computing $\varphi^{<}_{\theta}$ decreases the complexity of φ at the expense of complicating θ

$$\begin{aligned}
&((\exists x \in farmer)(\exists y \in donkey)\ owns(x, y) \supset beats(x, y))^{<}_{\emptyset} \\
&\quad = \Big(\prod z : ((\exists x \in farmer)(\exists y \in donkey)\ owns(x, y))^{<}_{\emptyset}\Big)\ beats(x, y)^{<}_{\hat\theta} \\
&\quad = \Big(\prod z : \Big(\Big(\sum x : farmer\Big)\Big(\sum y : donkey\Big)\ owns(x, y)^{<}_{\{(x,x),(y,y)\}}\Big)\Big)\ beats(x, y)^{<}_{\hat\theta} \\
&\quad = \Big(\prod z : \Big(\sum x : farmer\Big)\Big(\sum y : donkey\Big)\ owns(x, y)\Big)\ beats(lz, l(rz)),
\end{aligned}$$

where $z = \nu_{<}\{x, y\}$ and $\hat\theta = \{(x, lz), (y, l(rz))\}$.

More generally, for φ's not derived from (d), there is the risk that θ may become non-functional. Fortunately, we can restrict computations of $\varphi^{<}_{\theta}$ to pairs φ, θ such that

(a) φ respects NVC

and

(b) θ is a partial function from Var to Tm° with domain

$$dom(\theta) = \{x \in \mathsf{Var} \mid x \text{ has a free but no bound occurrence in } \varphi\}.$$

Let us call φ, θ *convergent* if the pair satisfies (a) and (b). Note that if φ respects NVC, then $\varphi, \theta^{\varphi}$ is convergent, where

$$\theta^{\varphi} = \{(x, x) \mid x \in \mathsf{Var} \text{ has a free but no bound occurrence in } \varphi\}.$$

Moreover, an easy induction on φ establishes

PROPOSITION 1. *Let φ, θ be a convergent pair. Then $\varphi^{<}_{\theta}$ is well-defined, and can be computed using only values $\psi^{<}_{\theta'}$ for convergent pairs ψ, θ' with ψ a subformula of φ.*

To reverse translations $\cdot^{<}_{\theta}$, let us lift a partial function γ from Tm° to Var to a partial function $\gamma[\cdot]$ from L-pretypes A to L-formulas $\gamma[A]$ by

$$\begin{aligned}
\gamma[\bot] &= \bot, \\
\gamma[R(t_1 \dots t_n)] &= \begin{cases} R(\gamma(t_1) \dots \gamma(t_n)) & \text{if } \{t_1 \dots t_n\} \subseteq dom(\gamma), \\ \uparrow & \text{otherwise,} \end{cases} \\
\gamma\Big[\Big(\sum x : U\Big)A\Big] &= (\exists x \in U)\gamma^x_x[A], \\
\gamma\Big[\Big(\prod x : U\Big)A\Big] &= (\forall x \in U)\gamma^x_x[A], \\
\gamma\Big[\Big(\sum x : A\Big)B\Big] &= \gamma[A] \wedge (\gamma * x, A)[B], \\
\gamma\Big[\Big(\prod x : A\Big)B\Big] &= \gamma[A] \supset (\gamma * x, A)[B]
\end{aligned}$$

with $\gamma^x_x = \gamma \cup \{(x, x)\}$ and

$$(\gamma * x, A) = \gamma \cup \{(\alpha(x), y) \mid (\alpha, y) \in wen(A)\},$$

where, paralleling new,

$$\begin{aligned}
wen(A) &= \emptyset \quad \text{for } A = \bot,\ R(x_1 \dots x_n),\ \Big(\prod x : U\Big)B \\
&\qquad\qquad \text{or } \Big(\prod x : B\Big)C, \\
wen\Big(\Big(\sum x : U\Big)A\Big) &= \{(l, x)\} \cup \{(\alpha r, y) \mid (\alpha, y) \in wen(A)\}, \\
wen\Big(\Big(\sum x : A\Big)B\Big) &= \{(\alpha l, y) \mid (\alpha, y) \in wen(A)\} \cup \\
&\qquad \{(\alpha r, y) \mid (\alpha, y) \in wen(B)\}.
\end{aligned}$$

We have not only $\emptyset[(\mathsf{p})] = (\mathsf{d})$ but

PROPOSITION 2. *If a pair φ, θ is convergent and θ is* 1-1 (*with inverse* θ^{-1}), *then* $\theta^{-1}[\varphi^{<}_{\theta}] = \varphi$.

As with Proposition 1, Proposition 2 can be proved by induction on L-formulas φ. For some intuition about the seeds θ and γ to the translations $\cdot^{<}_{\theta}$ and $\gamma[\cdot]$, we turn next to type-theoretic notions of context and well-formedness.

2.3. *Projecting Left and Right: $\mathcal{D}_\circ$*

Following Martin-Löf [19], a context is a finite sequence of variable typings $x : T$, built from the empty sequence ϵ, the typings in which seep through an arrow $\Rightarrow$ connecting the context (to $\Rightarrow$'s left) with judgments (to $\Rightarrow$'s right) that the context supports.

$$\frac{}{\epsilon \text{ context}} \qquad \frac{\Gamma \text{ context}}{\Gamma \Rightarrow x : T}\ \text{‘}x : T\text{’ in } \Gamma \qquad \frac{\Gamma \Rightarrow T \text{ type}}{\Gamma, x : T \text{ context}}\ \text{NVC}'.$$

The side condition NVC′ on the last rule above is that x is a variable in Var different from any that occurs in Γ or T, bound or free. Going beyond usual formulations, the prohibition in NVC′ against reusing bound variables will make life easier. The stipulation is, however, harmless in a sense made precise in Section 3.2 below.

Departing somewhat more from Martin-Löf, let us introduce a symbol wff for well-formed L-pretypes, distinguishing these from sorts in L, which we will also take as types.[4]

$$\frac{\Gamma \text{ context}}{\Gamma \Rightarrow \bot \text{ wff}} \qquad \frac{\Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow A \text{ type}} \qquad \frac{\Gamma \text{ context}}{\Gamma \Rightarrow U \text{ type}}\ U \in L_0.$$

Notice that $\Rightarrow$ ought really to be decorated with L, but we will suppress the subscript L on $\Rightarrow_L$ for simplicity. L-formulas other than $\bot$ can be matched by rules for atomic L-formulas

$$\frac{\Gamma \Rightarrow t_1 : U_1 \cdots \Gamma \Rightarrow t_n : U_n}{\Gamma \Rightarrow R(t_1, \ldots, t_n) \text{ wff}}\ R \in L(U_1 \cdots U_n)$$

$\wedge$ and $\supset$

$$(\wedge)\ \frac{\Gamma, x : A \Rightarrow B \text{ wff} \quad \Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow (\sum x : A)B \text{ wff}},$$

$$(\supset)\ \frac{\Gamma, x : A \Rightarrow B \text{ wff} \quad \Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow (\prod x : A)B \text{ wff}}$$

and $\exists x \in U$ and $\forall x \in U$

$$\frac{\Gamma, x : U \Rightarrow A \text{ wff}}{\Gamma \Rightarrow (\sum x : U)A \text{ wff}} U \in L_0,$$
$$\frac{\Gamma, x : U \Rightarrow A \text{ wff}}{\Gamma \Rightarrow (\prod x : U)A \text{ wff}} U \in L_0.$$

Terms such as lz and $l(rz)$ in (p) can be formed from the familiar elimination rules for $\sum$.

$$\frac{\Gamma \Rightarrow t : (\sum x : T)A}{\Gamma \Rightarrow lt : T} \qquad \frac{\Gamma \Rightarrow t : (\sum x : T)A}{\Gamma \Rightarrow rt : A[x \mapsto lt]}.$$

Now, putting further rules aside for the moment, let $\mathcal{D}_\circ$ denote the set of rules above.

CONVENTIONS. L-pretypes obtained from the rules ($\wedge$) and ($\supset$) where the variable x bound does not occur in B are more perspicuously written $A \wedge B$ and $A \supset B$, respectively. Also, for $R \in L(U)$, let us write $(\sum x : R)A$ instead of $(\sum x : U)(R(x) \wedge A)$, and $(\prod x : R)A$ instead of $(\prod x : U)(R(x) \supset A)$. Thus, assuming that for some $U \in L_0$, *farmer*, *donkey* $\in L(U)$ and *owns*, *beats* $\in L(U, U)$, the expression (p) in Section 1 is $(\prod z : A)\mathit{beats}(lz, l(rz))$, where A unwinds to

$$\Big(\sum x : U\Big)\Big(\sum u : \mathit{farmer}(x)\Big)$$
$$\Big(\sum y : U\Big)\Big(\sum v : \mathit{donkey}(y)\Big)\ \mathit{owns}(x, y)$$

for certain novel (dummy) variables u and v.

As a set of rules, $\mathcal{D}_\circ$ induces the usual notion of a derivation. Let us collect the $\mathcal{D}_\circ$-derivable contexts in the set

$$\mathsf{context}^\circ = \{\Gamma \mid \text{there is a } \mathcal{D}_\circ\text{-derivation of `}\Gamma\ \mathsf{context}\text{'}\}$$

and write

$$\text{`}\Gamma \Rightarrow^\circ \Theta\text{'} \quad \text{to mean} \quad \text{there is a } \mathcal{D}_\circ\text{-derivation of `}\Gamma \Rightarrow \Theta\text{'}.$$

Notice that if '$\Gamma \Rightarrow^\circ \Theta$', then $\Gamma \in \mathsf{context}^\circ$. Let us gather the preterms t that a sequence Γ sorts in the set

$$\mathsf{Tm}^\circ[\Gamma] = \{t \mid \text{`}\Gamma \Rightarrow^\circ t : U\text{' for some } U \in L_0\}$$

with the understanding that for $\Gamma \notin \mathsf{context}^\circ$, $\mathsf{Tm}^\circ[\Gamma] = \emptyset$. Clearly, any element of $\mathsf{Tm}^\circ[\Gamma]$ is either a variable or has the form lt where '$\Gamma \Rightarrow^\circ$

$t : (\sum x : U)A$' for some $U \in L_0$ and A. Now, a suitable candidate for γ in Section 2.2 is the function $\cdot^{\circ}_{\Gamma}$ from $\mathsf{Tm}^{\circ}[\Gamma]$ to Var defined by

$$\begin{array}{rcl} x^{\circ}_{\Gamma} & = x & \text{for } x \in \mathsf{Var} \cap \mathsf{Tm}^{\circ}[\Gamma] \\ & & \text{(whence for some } U \in L_0, \text{ `}\Gamma \Rightarrow^{\circ} x : U\text{'}), \\ lt^{\circ}_{\Gamma} & = x & \text{where for some } U \in L_0 \\ & & \text{and } A, \text{ `}\Gamma \Rightarrow^{\circ} t : (\textstyle\sum x : U)A\text{'}. \end{array}$$

To ensure that an L-pretype A translates to an L-formula only if '$\Gamma \Rightarrow^{\circ} A$ wff', let us sharpen the translation $\cdot^{\circ}_{\Gamma}[A]$ of A to A°_{Γ},[5] defined by induction on A

$$\bot^{\circ}_{\Gamma} = \bot,$$

$$R(t_1 \ldots t_n)^{\circ}_{\Gamma} = \begin{cases} R(t_{1\Gamma}^{\circ} \ldots t_{n\Gamma}^{\circ}) & \text{if `}\Gamma \Rightarrow^{\circ} R(t_1 \ldots t_n) \text{ wff'}, \\ \uparrow & \text{otherwise} \end{cases}$$

(noting that if '$\Gamma \Rightarrow^{\circ} R(t_1 \ldots t_n)$ wff' then $\{t_1 \ldots t_n\} \subseteq \mathsf{Tm}^{\circ}[\Gamma]$)

$$\left(\left(\sum x : U\right)A\right)^{\circ}_{\Gamma} = \begin{cases} (\exists x \in U)A^{\circ}_{\Gamma,x:U} & \text{if } x \text{ not in } \Gamma, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\left(\left(\prod x : U\right)A\right)^{\circ}_{\Gamma} = \begin{cases} (\forall x \in U)A^{\circ}_{\Gamma,x:U} & \text{if } x \text{ not in } \Gamma, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\left(\left(\sum x : A\right)B\right)^{\circ}_{\Gamma} = \begin{cases} A^{\circ}_{\Gamma} \wedge B^{\circ}_{\Gamma,x:A} & \text{if } x \text{ not in } \Gamma \text{ nor } A, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\left(\left(\prod x : A\right)B\right)^{\circ}_{\Gamma} = \begin{cases} A^{\circ}_{\Gamma} \supset B^{\circ}_{\Gamma,x:A} & \text{if } x \text{ not in } \Gamma \text{ nor } A, \\ \uparrow & \text{otherwise.} \end{cases}$$

PROPOSITION 3. *Let* $\Gamma \in \mathsf{context}^{\circ}$ *and A be an L-pretype.*

(a) A°_{Γ} *is defined iff* '$\Gamma \Rightarrow^{\circ} A$ wff'.
(b) *If* '$\Gamma \Rightarrow^{\circ} A$ wff' *then* $A^{\circ}_{\Gamma} = \cdot^{\circ}_{\Gamma}[A]$.
(c) *If* '$\Gamma \Rightarrow^{\circ} A$ wff' *and* $theta(\Gamma)$ *is the inverse of* $\cdot^{\circ}_{\Gamma} : \mathsf{Tm}^{\circ}[\Gamma] \to \mathsf{Var}$ *restricted to the variables that occur in* A°_{Γ},[6] *then* $A^{\circ}_{\Gamma}, theta(\Gamma)$ *is convergent and*

$$(A^{\circ}_{\Gamma})^{<}_{theta(\Gamma)} \approx A$$

where $\approx$ is equality up to renaming of variables x bound by $(Qx : A)$ for some $Q \in \{\sum, \prod\}$ and $A \notin L_0$.

Proof. Part (a) follows by induction on L-pretypes A, parts (b) and (c) by induction on the length of $\mathcal{D}_{\circ}$-derivations. □

A simple example where $A^{\circ}_{\Gamma} \neq \cdot^{\circ}_{\Gamma}[A]$ is provided by $\Gamma = \epsilon$ and $A = (\sum x : U)R(x)$ where $R \in L(U')$ and $U \neq U'$. Sidestepping such sortal complications, we get

PROPOSITION 4. *Suppose L has exactly one sort. Then for every L-formula φ respecting NVC, there exists $\Gamma \in$* context° *such that '$\Gamma \Rightarrow^{\circ} \varphi^{<}_{theta(\Gamma)}$* wff*' and φ, theta(Γ) is convergent, whence $(\varphi^{<}_{theta(\Gamma)})^{\circ}_{\Gamma} = \varphi$.*

Proof. Let Γ be $x_1 : U, \ldots, x_n : U$ where U is the one sort of L, and $x_1 \ldots x_n$ is a non-repeating list of every variable with some free but no bound occurrence in φ. Observe that *theta*(Γ) is the function θ^{φ} mentioned before Proposition 1 in Section 2.2. □

An obvious extension of Proposition 4 is

If L has exactly one sort, then for every convergent pair φ, θ, there exists Γ such that theta$(\Gamma) = \theta$ and '$\Gamma \Rightarrow^{\circ} \varphi^{<}_{theta(\Gamma)}$ wff*'.*

That assertion is demonstrably false, however, as not every such θ is of the form *theta*(Γ) for some Γ. Take $\theta = \{(x, ly), (z, lly)\}$.

3. Interpretations: Transitions and Bisimulations

Next comes semantics. The notion of an L-interpretation described in Section 1.1 points in two opposite directions. We can reduce an L-interpretation $[\![\cdot]\!]$ to an L-model $\mathcal{M}[\![\cdot]\!]$, relative to which L-formulas φ can be interpreted dynamically. Or, going the other way, we can extend $[\![\cdot]\!]$ to interpret various constructs connected with $\mathcal{D}_{\circ}$. We will pursue both directions, sequentially and then in parallel. Among the results to be established is that whenever '$\Gamma \Rightarrow^{\circ} A$ wff', proof-conditional and dynamic semantics agree in that for every "$[\![\cdot]\!]$-instantiation" ρ of Γ,

$$A \text{ is "}[\![\cdot]\!]\text{-true for" } \Gamma, \rho \quad \text{iff} \quad A^{\circ}_{\Gamma} \text{ is "}\mathcal{M}[\![\cdot]\!]\text{-true for } (\Gamma, \rho)^{\circ}\text{".}$$

The precise statement of the equivalence is given in Lemma 6(c); but first we must understand what the phrases in quotes mean.

3.1. *Dynamic Semantics with Finite Functions*

As mentioned back in the introduction, dynamic semantics interprets an L-formula φ relative to an L-model M as an input/output relation $\|\varphi\|_M$ between functions f and f' from variables to objects in M

$$f \|\varphi\|_M f' \quad \text{iff} \quad \text{on input } f, \varphi \text{ can (in } M\text{) output } f'.$$

Intuitively, the program $\|\varphi\|_M$, on input f, checks if $M \models \varphi[f]$, halting exactly if this is so,

(3) $\quad f \in dom(\|\varphi\|_M) \quad$ iff $\quad \varphi$ is true at f (relative to M),

storing "suitable" witnesses in the output. Now, the semantic force of the novel variable condition, NVC, on φ is to avoid re-assigning values to variables, preserving (as it were) all witnesses ever found. Accordingly, it is useful to assume that f and f' are not defined on every variable in Var (contrary to DPL, but in line with DRT). In fact, we will make do with finite functions, drawing f and f' from the set Var_M of functions from finite subsets of Var to the universe $\bigcup_{U \in L_0} U_M$ of M

(4) $$\mathsf{Var}_M = \bigcup \left\{ X \to \bigcup_{U \in L_0} U_M \mid X \text{ is a finite subset of } \mathsf{Var} \right\}.$$

The binary relations $\|\varphi\|_M \subseteq \mathsf{Var}_M \times \mathsf{Var}_M$ encoding input/output pairs of L-formulas φ are then given as follows.[7] Keeping (3) in mind, let $\|\bot\|_M = \emptyset$ (as $\bot$ can never be true),

$$\begin{aligned}
f\|(R(x_1 \ldots x_n))\|_M f' \quad &\text{iff} \quad f = f', \{x_1 \ldots x_n\} \subseteq dom(f) \text{ and} \\
&\qquad R_M(f(x_1) \ldots f(x_n)), \\
f\|\varphi \wedge \psi\|_M f' \quad &\text{iff} \quad \text{for some } g,\ f\|\varphi\|_M g \text{ and } g\|\psi\|_M f' \\
f\|\varphi \supset \psi\|_M f' \quad &\text{iff} \quad f = f' \text{and} \\
&\qquad (\forall g \text{ such that } f\|\varphi\|_M g)\ g \in dom(\|\psi\|_M)
\end{aligned}$$

and, for $f^x_a = f \cup \{(x, a)\}$,

$$\begin{aligned}
f\|(\exists x \in U)\varphi\|_M f' \quad &\text{iff} \quad \text{for some } a \in U_M,\ f^x_a\|\varphi\|_M f', \\
f\|(\forall x \in U)\varphi\|_M f' \quad &\text{iff} \quad f = f' \text{and} \\
&\qquad (\forall a \in U_M)\ f^x_a \in dom(\|\varphi\|_M).
\end{aligned}$$

Note that if $x \in dom(f)$ and $f(x) \neq a$, then f^x_a falls outside Var_M and has no chance of being in the domain of any $\|\varphi\|_M$. The definition of f^x_a ensures that

(5) $\quad$ whenever $f\|\varphi\|_M f', \quad f \subseteq f'$

(with $f = f'$ in case $new(\varphi) = \emptyset$, as defined in Section 2.2). It will become clear shortly that the persistence described in (5) is instrumental in matching φ up with a type in $\mathcal{D}_\circ$.

That match-up involves the following standard notion from model theory (e.g. Keisler [15]). A *partial isomorphism between* L-models M and N (possibly the same) is a relation $I \subseteq \mathsf{Var}_M \times \mathsf{Var}_N$ such that $\emptyset I \emptyset$, and whenever fIg, (c1) for all $R \in L(U_1 \cdots U_n)$,

$$f \| R(x_1 \dots x_n) \|_M f \quad \text{iff} \quad g \| R(x_1 \dots x_n) \|_N g,$$

(c2) for all $a \in U_M$ and $x \notin dom(f)$, there exists $b \in U_N$ such that $f^x_a I g^x_b$, and

(c3) for all $b \in U_N$ and $x \notin dom(g)$, there exists $a \in U_M$ such that $f^x_a I g^x_b$.

Let us agree to write

$$(M, f) \cong_p (N, g) \quad \text{iff} \quad f I g \text{ for some partial isomorphism } I \text{ between } M \text{ and } N,$$

shortening $(M, \emptyset) \cong_p (N, \emptyset)$ to $M \cong_p N$. Among the many well-known facts about partial isomorphisms is that if the universes of M and N are countable and if for every $U \in L_0$, there is an $R \in L(U, U)$ such that R_M is = on U, then

$$M \cong_p N \quad \text{iff} \quad M \cong N.$$

Furthermore, partial isomorphisms are exactly *bisimulations* (Park [22]) over $(\| \cdot \|_M, \emptyset)$ and $(\| \cdot \|_N, \emptyset)$.[8] That is, a relation $I \subseteq \mathsf{Var}_M \times \mathsf{Var}_N$ such that $\emptyset I \emptyset$ is a partial isomorphism between M and N iff for all $f I g$,

(b1) whenever $f \|\varphi\|_M f'$, there is a g' such that $g \|\varphi\|_N g'$ and $f' I g'$, and
(b2) whenever $g \|\varphi\|_N g'$, there is an f' such that $f \|\varphi\|_M f'$ and $f' I g'$.

Notice that whereas φ ranges over all L-formulas in (b1) and (b2), the only L-formulas that figure in the previous clauses (c1)–(c3) defining partial isomorphisms are of the form

$$(*) \qquad R(x_1 \dots x_n) \quad \text{and} \quad (\exists x \in U)\top,$$

where $\top$ is an L-tautology such as $\bot \supset \bot$. The discrepancy here is accounted for by two facts (where $\neg\varphi$ is defined as $\varphi \supset \bot$):

(i) for every L-formula φ, an L-formula ψ can be assembled from L-formulas in $(*)$ using only $\wedge$ and $\neg$ such that

$$\|\varphi\|_M = \|\psi\|_M \quad \text{for every } L\text{-model } M$$

(as $\|\varphi \supset \psi\|_M = \|\neg(\varphi \wedge \neg\psi)\|_M$, $\|(\exists x \in U)\varphi\|_M = \|(\exists x \in U)\top \wedge \varphi\|_M$, and $\|(\forall x \in U)\varphi\|_M = \|\neg(\exists x \in U)\neg\varphi\|_M$),

(ii) the back-and-forth force common to (b1), (b2) and (c1)–(c3) builds in closure under the connectives $\wedge$ and $\neg$.

The remainder of this section is devoted to reducing a suitable notion of bisimulation equivalence for the rule set $\mathcal{D}_\circ$ to $\cong_p$. Towards that end, we will need to extend L-interpretations to various constructs in $\mathcal{D}_\circ$.

3.2. *Extending L-interpretations*

Fix an L-interpretation $[\![\cdot]\!]$ – i.e. a mapping from $U \in L_0$ to a set $[\![U]\!]$, and from $R \in L(U_1 \cdots U_n)$ and $u_1 \in [\![U_1]\!] \ldots u_n \in [\![U_n]\!]$ to a set $[\![R, u_1 \cdots u_n]\!]$. Applying $[\![\cdot]\!]$ to $\mathcal{D}_\circ$, let us define for every $\Gamma \in \mathsf{context}^\circ$,

(I) its set $[\![\Gamma]\!]^\circ$ of *instantiations* ρ interpreting Γ's variables x as $\rho(x)$

simultaneously (by induction on the length of a $\mathcal{D}_\circ$-derivation) with

(II) interpretations $[\![T]\!]^\circ_{\Gamma,\rho}$ of T such that '$\Gamma \Rightarrow^\circ T$ type' (with $\rho \in [\![\Gamma]\!]^\circ$)

and (again with $\rho \in [\![\Gamma]\!]^\circ$)

(III) interpretations $[\![t]\!]^\circ_{\Gamma,\rho}$ of t such that for some T, '$\Gamma \Rightarrow^\circ t : T$'.

The requirement that whenever $\rho \in [\![\Gamma]\!]^\circ$,

$$\text{if '}\Gamma \Rightarrow^\circ t : T\text{' then } [\![t]\!]^\circ_{\Gamma,\rho} \in [\![T]\!]^\circ_{\Gamma,\rho}$$

is easily met by agreeing that

(I)
$$[\![\epsilon]\!]^\circ = \{\emptyset\},$$
$$[\![\Gamma, x : T]\!]^\circ = \{\rho^x_t \mid \rho \in [\![\Gamma]\!]^\circ \text{and } t \in [\![T]\!]^\circ_{\Gamma,\rho}\},$$

where ρ^x_t is $\rho \cup \{(x, t)\}$,

(II)
$$[\![\bot]\!]^\circ_{\Gamma,\rho} = \emptyset,$$
$$[\![U]\!]^\circ_{\Gamma,\rho} = [\![U]\!],$$
$$[\![R(t_1 \ldots t_n)]\!]^\circ_{\Gamma,\rho} = [\![R, [\![t_1]\!]^\circ_{\Gamma,\rho} \cdots [\![t_n]\!]^\circ_{\Gamma,\rho}]\!]$$

and adopting $\sum$, $\prod$ notation at the meta-level, with : replaced by $\in$ (abusing the object level notation for L-formulas),

$$\left[\!\!\left[\left(\sum x : T\right)A\right]\!\!\right]^\circ_{\Gamma,\rho} = \left(\sum t \in [\![T]\!]^\circ_{\Gamma,\rho}\right)[\![A]\!]^\circ_{\Gamma,x:T,\rho^x_t},$$
$$\left[\!\!\left[\left(\prod x : T\right)A\right]\!\!\right]^\circ_{\Gamma,\rho} = \left(\prod t \in [\![T]\!]^\circ_{\Gamma,\rho}\right)[\![A]\!]^\circ_{\Gamma,x:T,\rho^x_t},$$

(III)
$$[\![x]\!]^\circ_{\Gamma,\rho} = \rho(x),$$
$$[\![lt]\!]^\circ_{\Gamma,\rho} = l[\![t]\!]^\circ_{\Gamma,\rho},$$
$$[\![rt]\!]^\circ_{\Gamma,\rho} = r[\![t]\!]^\circ_{\Gamma,\rho},$$

re-using l and r at the meta-language.

The definition above brings us very close to the truth equivalence mentioned at the beginning of this section. The impatient reader can jump to Lemma 6(c) in the next subsection, and verify it by induction (using the definition of $(\Gamma, \rho)^\circ$ given before the lemma). For an equivalence reaching beyond truth and applying to transitions, a bit more work is necessary, including some points about alphabetic invariance. Over sorts or L-pretypes T and T', let $\sim$ be the equivalence

$$T \sim T' \quad \text{iff} \quad T \text{ and } T' \text{ are identical up to renaming of bound variables}$$

(with the usual provisos for safe renaming). We extend $\sim$ to sequences of typings (that may or may not belong to $\mathsf{context}^\circ$) as follows

$$x_1 : T_1, \ldots, x_n : T_n \sim y_1 : T'_1, \ldots, y_m : T'_m$$
$$\text{iff} \quad n = m \text{ and for } 1 \leq i \leq n,\ x_i = y_i \text{ and } T_i \sim T'_i.$$

The following is trivial, but nevertheless worth recording.

PROPOSITION 5. (a) *If '*$\Gamma \Rightarrow^\circ A$ wff*' and* $\Gamma' \sim \Gamma$*, then for some* $A' \sim A$*, '*$\Gamma' \Rightarrow^\circ A'$ wff*'. Similarly for '*$\Gamma \Rightarrow^\circ t : A$*'.*

(b) *Suppose '*$\Gamma \Rightarrow^\circ A$ wff*', '*$\Gamma' \Rightarrow^\circ A'$ wff*',* $\Gamma \sim \Gamma'$ *and* $A \sim A'$*. Then*

$$[\![\Gamma]\!]^\circ = [\![\Gamma']\!]^\circ \text{ and for every } \rho \in [\![\Gamma]\!]^\circ,\ [\![A]\!]^\circ_{\Gamma,\rho} = [\![A']\!]^\circ_{\Gamma',\rho}.$$

One of the upshots of Proposition 5 is that the semantics above is unaltered by relaxing the side condition NVC′ in Section 2.3 to its usual formulation (i.e. x is not one of the variables Γ types). That aside, Proposition 5 will prove useful below in conjunction with $\mathcal{D}_\circ$ (defined using NVC′).

3.3. *Transitions between Context Instantiations*

The subscripts Γ, ρ on $[\![\cdot]\!]^\circ_{\Gamma,\rho}$ above are understood to belong to the set $\mathsf{CI}^\circ_{[\![\cdot]\!]}$ of context-instantiation pairs defined by

$$\mathsf{CI}^\circ_{[\![\cdot]\!]} = \Big(\sum \Gamma \in \mathsf{context}^\circ\Big)[\![\Gamma]\!]^\circ.$$

From $[\![A]\!]^\circ_{\Gamma,\rho}$ with $(\Gamma, \rho) \in \mathsf{CI}^\circ_{[\![\cdot]\!]}$, it is a small step to binary relations $[\![A]\!]_\circ$ on $\mathsf{CI}^\circ_{[\![\cdot]\!]}$ defined by

$$(\Gamma, \rho)[\![A]\!]_\circ(\Gamma', \rho') \quad \text{iff} \quad \rho \subset \rho',\ \text{`}\Gamma \Rightarrow^\circ A\ \mathsf{wff}\text{' and } \Gamma' = \Gamma, x : A \text{ for some } x \in \mathsf{Var}$$

for all $(\Gamma, \rho), (\Gamma', \rho') \in \mathsf{CI}^\circ_{[\![\cdot]\!]}$. That is, collecting all Γ's such that '$\Gamma \Rightarrow^\circ A$ wff' in

$$\mathsf{context}^\circ_A = \{\Gamma \in \mathsf{context}^\circ \mid \text{`}\Gamma \Rightarrow^\circ A\ \mathsf{wff}\text{'}\}$$

and pairing them with their instantiations in

$$\mathsf{Cl}^{\circ}_{[\![\cdot]\!]}(A) = \Big(\sum \Gamma \in \mathsf{context}^{\circ}_{A}\Big)[\![\Gamma]\!]^{\circ},$$

we can characterize $[\![A]\!]_{\circ}$ as

$$\Big(\sum(\Gamma, \rho) \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]}(A)\Big)\{(\Gamma, x : A, \rho^{x}_{a}) \mid a \in [\![A]\!]^{\circ}_{\Gamma,\rho}$$
$$\text{and } x \in \mathsf{Var} \text{ not in } \Gamma \text{ nor } A\}.$$

The idea now is to relate $[\![\cdot]\!]_{\circ}$ to the input/output schemes $\|\cdot\|_{\mathcal{M}[\![\cdot]\!]}$ from Section 3.1, where $\mathcal{M}[\![\cdot]\!]$ is the obvious L-model induced by $[\![\cdot]\!]$

$$\begin{aligned} U_{\mathcal{M}[\![\cdot]\!]} &= [\![U]\!], \\ R_{\mathcal{M}[\![\cdot]\!]} &= \{u_1 \cdots u_n \mid [\![R, u_1 \cdots u_n]\!] \neq \emptyset\}. \end{aligned}$$

For every $(\Gamma, \rho) \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]}$, let $(\Gamma, \rho)^{\circ}$ be the partial function from Var to $\bigcup_{U \in L_0}[\![U]\!]$ given by

$$(\Gamma, \rho)^{\circ} = \{(t^{\circ}_{\Gamma}, [\![t]\!]^{\circ}_{\Gamma,\rho}) \mid t \in \mathsf{Tm}^{\circ}[\Gamma]\}$$

with $\mathsf{Tm}^{\circ}[\Gamma]$ and t°_{Γ} as defined in Section 2.3. Recalling also the definition (4) of Var_{M} from Section 3.1, observe that

$$(\Gamma, \rho)^{\circ} \in \mathsf{Var}_{\mathcal{M}[\![\cdot]\!]}.$$

LEMMA 6. *Let A be an L-pretype, $[\![\cdot]\!]$ be an L-interpretation, and* $(\Gamma, \rho) \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]}$.

(a) *For every (Γ', ρ') such that $(\Gamma, \rho)[\![A]\!]_{\circ}(\Gamma', \rho')$,* $(\Gamma, \rho)^{\circ}\|A^{\circ}_{\Gamma}\|_{\mathcal{M}[\![\cdot]\!]}(\Gamma', \rho')^{\circ}$.

(b) *Conversely, for every $f \in \mathsf{Var}_{\mathcal{M}[\![\cdot]\!]}$ such that $(\Gamma, \rho)^{\circ}\|A^{\circ}_{\Gamma}\|_{\mathcal{M}[\![\cdot]\!]}f$,*

$$(\exists a \in [\![A]\!]^{\circ}_{\Gamma,\rho})\ (\forall x \in \mathsf{Var} \textit{ not in } \Gamma \textit{ nor } A)$$
$$f = (\Gamma, x : A, \rho^{x}_{a})^{\circ}\ (\textit{whence } (\Gamma, \rho)[\![A]\!]_{\circ}(\Gamma, x : A, \rho^{x}_{a})).$$

(c) *Whenever '$\Gamma \Rightarrow^{\circ} A$ wff',*

$$[\![A]\!]^{\circ}_{\Gamma,\rho} \neq \emptyset \quad \textit{iff} \quad (\Gamma, \rho)^{\circ} \in dom(\|A^{\circ}_{\Gamma}\|_{\mathcal{M}[\![\cdot]\!]}).$$

Proof. Part (c) is immediate from parts (a) and (b), which are proved by induction on A (conveniently enough matching the length of $\mathcal{D}_{\circ}$-derivations). □

Moving from truth (analyzed in Lemma 6(c)) to transitions, let us fix an L-interpretation $[\![\cdot]\!]'$, possibly (but not necessarily) identical to $[\![\cdot]\!]$. Given $(\Gamma, \rho) \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]}$ and $(\Gamma', \rho') \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]'}$, a $[\![\cdot]\!], [\![\cdot]\!]'$*-bisimulation* is a binary relation $\mathsf{B} \subseteq \mathsf{Cl}^{\circ}_{[\![\cdot]\!]} \times \mathsf{Cl}^{\circ}_{[\![\cdot]\!]'}$ such that for all $(\Gamma, \rho)\mathsf{B}(\Gamma', \rho')$,

(β0) $\mathsf{Tm}^\circ[\Gamma] = \mathsf{Tm}^\circ[\Gamma']$,
(β1) whenever $(\Gamma, \rho)[\![A]\!]_\circ \mathsf{p}$, there is a q such that $(\Gamma', \rho')[\![A]\!]'_\circ \mathsf{q}$ and pBq, and
(β2) whenever $(\Gamma', \rho')[\![A]\!]'_\circ \mathsf{q}$, there is a p such that $(\Gamma, \rho)[\![A]\!]_\circ \mathsf{p}$ and pBq.

Let us write

$$([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho')) \quad \text{iff} \quad (\Gamma, \rho)\mathsf{B}(\Gamma', \rho')$$
$$\text{for some } [\![\cdot]\!], [\![\cdot]\!]'\text{-bisimulation } \mathsf{B}.$$

To relate $\underline{\leftrightarrow}$ to partial isomorphism $\cong_p$ from Section 3.1, the following notion of Γ and Γ' being Tm°-*coincident* is useful. Define

$$\Gamma\langle\mathsf{Tm}^\circ\rangle\Gamma' \quad \text{iff} \quad \mathsf{Tm}^\circ[\Gamma] = \mathsf{Tm}^\circ[\Gamma']$$
$$\text{and } (\forall t \in \mathsf{Tm}^\circ[\Gamma])\ t^\circ_\Gamma = t^\circ_{\Gamma'},$$

the point of the second conjunct being that Γ and Γ' use the same variables $t^\circ_\Gamma = t^\circ_{\Gamma'}$ for $t \in \mathsf{Tm}^\circ[\Gamma]$. Now comes the type reduction from which the present paper gets its title. For all L-models M and N, and context-instantiation pairs $(\Gamma, \rho) \in \mathsf{CI}^\circ_{[\![\cdot]\!]}$ and $(\Gamma', \rho') \in \mathsf{CI}^\circ_{[\![\cdot]\!]'}$ where $\mathcal{M}[\![\cdot]\!] = M$ and $\mathcal{M}[\![\cdot]\!]' = N$, define

$$(M, (\Gamma, \rho)) \approx_\circ (N, (\Gamma', \rho')) \quad \text{iff}$$
$$(\exists\hat{\Gamma} \sim \Gamma)(\exists\hat{\Gamma}' \sim \Gamma')\hat{\Gamma}\langle\mathsf{Tm}^\circ\rangle\hat{\Gamma}' \text{ and}$$
$$(M, (\hat{\Gamma}, \rho)^\circ) \cong_p (N, (\hat{\Gamma}', \rho')^\circ).$$

The intuition behind $\approx_\circ$ is that before applying $\cdot^\circ$ and $\cong_p$, bound variables may need to be renamed to ensure that the contexts are Tm°-coincident.

THEOREM 7. *Let* $[\![\cdot]\!]$ *and* $[\![\cdot]\!]'$ *be L-interpretations, possibly but not necessarily identical. For all context-instantiation pairs* $(\Gamma, \rho) \in \mathsf{CI}^\circ_{[\![\cdot]\!]}$ *and* $(\Gamma', \rho') \in \mathsf{CI}^\circ_{[\![\cdot]\!]'}$,

$$([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho')) \quad \textit{iff}$$
$$(\mathcal{M}[\![\cdot]\!], (\Gamma, \rho)) \approx_\circ (\mathcal{M}[\![\cdot]\!]', (\Gamma', \rho')).$$

Proof. To simplify the notation, let us shorten $\mathcal{M}[\![\cdot]\!]$ to M, and $\mathcal{M}[\![\cdot]\!]'$ to N. Unlike previous proofs, the present argument proceeds *not* by induction on A, but by *co*-induction, the equivalence in question breaking into two halves.

(7a) The binary relation

$$\{((\Gamma, \rho)^\circ, (\Gamma', \rho')^\circ) \mid$$
$$([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho')) \text{ and } \Gamma\langle\mathsf{Tm}^\circ\rangle\Gamma'\}$$

is a partial isomorphism between M and N.

(7b) The binary relation

$$\{((\Gamma, \rho), (\Gamma', \rho')) \in \mathsf{Cl}^{\circ}_{[\![\cdot]\!]} \times \mathsf{Cl}^{\circ}_{[\![\cdot]\!]'} \mid (M, (\Gamma, \rho)) \approx_{\circ} (N, (\Gamma', \rho'))\}$$

is a $[\![\cdot]\!]$, $[\![\cdot]\!]'$-bisimulation.

It is immediate from the definition of $\underline{\leftrightarrow}$ that (7b) gives the right-to-left direction of the equivalence above. To see that (7a) yields the converse, suppose $([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho'))$. Since $\mathsf{Tm}^{\circ}[\Gamma] = \mathsf{Tm}^{\circ}[\Gamma']$, we can find $\hat{\Gamma} \sim \Gamma$ and $\hat{\Gamma}' \sim \Gamma'$ such that $\hat{\Gamma}\langle\mathsf{Tm}^{\circ}\rangle\hat{\Gamma}'$ (by renaming bound variables, if necessary). But by Proposition 5, $[\![\cdot]\!]_{\circ}$ and $[\![\cdot]\!]'_{\circ}$ are invariant under such renaming, whence $([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho'))$ implies $([\![\cdot]\!], (\hat{\Gamma}, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\hat{\Gamma}', \rho'))$. Thus, (7a) gives $(M, (\hat{\Gamma}, \rho)^{\circ}) \cong_p (N, (\hat{\Gamma}', \rho')^{\circ})$, as required.

To prove (7a), suppose

$$(6) \qquad ([\![\cdot]\!], (\Gamma, \rho)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\Gamma', \rho')) \quad \text{and} \quad \Gamma\langle\mathsf{Tm}^{\circ}\rangle\Gamma'$$

and let π be the inverse $\{(t^{\circ}_{\Gamma}, t) \mid t \in \mathsf{Tm}^{\circ}[\Gamma]\}$ of the function $\cdot^{\circ}_{\Gamma}$ with domain $\mathsf{Tm}^{\circ}[\Gamma]$. We are to verify clauses (c1)–(c3) in Section 3.1 defining partial isomorphisms, with $f = (\Gamma, \rho)^{\circ}$ and $g = (\Gamma', \rho')^{\circ}$. For (c1), if $(\Gamma, \rho)^{\circ} \| R(x_1 \ldots x_n) \|_M (\Gamma, \rho)^{\circ}$ then by Lemma 6(b),

$$(\Gamma, \rho)\ [\![R(\pi(x_1) \ldots \pi(x_n))]\!]_{\circ}\ (\Gamma, z : R(\pi(x_1) \ldots \pi(x_n)), \rho^{z}_{a})$$

for some z and a. By (6),

$$(\Gamma', \rho')\ [\![R(\pi(x_1) \ldots \pi(x_n))]\!]'_{\circ}\ (\Gamma', z : R(\pi(x_1) \ldots \pi(x_n)), \rho'^{z}_{b})$$

for some b, whence by Lemma 6(a),

$$(\Gamma', \rho')^{\circ}\ \| R(x_1 \ldots x_n) \|_N\ (\Gamma', \rho')^{\circ},$$

as required. Similarly for the converse. And indeed for (c2): if

$$(\Gamma, \rho)^{\circ}\ \| (\exists x \in U\Big)\top \|_M \Big(\Gamma, z : \Big(\sum x : U\Big)\top, \rho^{z}_{\langle a,p\rangle}\Big)^{\circ}$$

then by Lemma 6(b),

$$(\Gamma, \rho)\ \Big[\!\!\Big[\Big(\sum x : U\Big)\top\Big]\!\!\Big]_{\circ} \Big(\Gamma, z : \Big(\sum x : U\Big)\top, \rho^{z}_{\langle a,p\rangle}\Big)$$

whence by (6),

$$(\Gamma', \rho')\ \Big[\!\!\Big[\Big(\sum x : U\Big)\top\Big]\!\!\Big]'_{\circ} \Big(\Gamma', z : \Big(\sum x : U\Big)\top, \rho'^{z}_{\langle b,q\rangle}\Big)$$

for some b, q, where $(\Gamma, z : (\sum x : U)\top, \rho^z_{\langle a,p\rangle}) \underline{\leftrightarrow} (\Gamma', z : (\sum x : U)\top, \rho'^z_{\langle b,q\rangle})$. Lemma 6(a) then gives

$$(\Gamma', \rho')^\circ \ \|(\exists x \in U)\top\|_N \left(\Gamma', z : \left(\sum x : U\right)\top, \rho'^z_{\langle b,q\rangle}\right)^\circ ,$$

as required. Ditto for (c3).

Proving (7b) is no harder. Suppose $(M, (\Gamma, \rho)) \approx_\circ (N, (\Gamma', \rho'))$, say

(7) $\quad \hat{\Gamma} \sim \Gamma,\ \hat{\Gamma}' \sim \Gamma',\ \hat{\Gamma}\langle \mathsf{Tm}^\circ\rangle\hat{\Gamma}'$ and
$(M, (\hat{\Gamma}, \rho)^\circ) \cong_p (N, (\hat{\Gamma}', \rho')^\circ)$.

Recalling clauses (β0)–(β2) above (for $[\![\cdot]\!]$, $[\![\cdot]\!]'$-bisimulations), (β0) is immediate: $\mathsf{Tm}^\circ[\Gamma] = \mathsf{Tm}^\circ[\Gamma']$. As for ($\beta$1), if $(\Gamma, \rho)[\![A]\!]_\circ(\Gamma, z : A, \rho^z_a)$, then as $\hat{\Gamma} \sim \Gamma$, Proposition 5 gives

$$(\hat{\Gamma}, \rho)\ [\![A]\!]_\circ\ (\hat{\Gamma}, y : A, \rho^y_a)$$

for some y, whence by Lemma 6(a),

$$(\hat{\Gamma}, \rho)^\circ\ \|A^\circ_\Gamma\|_M\ (\hat{\Gamma}, y : A, \rho^y_a)^\circ.$$

By (7) and the characterization of $\cong_p$ in terms of the bisimulation conditions (b1) and (b2) in Section 3.1,

$$(\hat{\Gamma}', \rho')^\circ\ \|A^\circ_\Gamma\|_N\ g$$

for some g such that $(M, (\hat{\Gamma}, y : A, \rho^y_a)^\circ) \cong_p (N, g)$. By Lemma 6(b) and $\hat{\Gamma}\langle \mathsf{Tm}^\circ\rangle\hat{\Gamma}'$,

$$(\hat{\Gamma}', \rho')\ [\![A]\!]'^\circ\ (\hat{\Gamma}', y : A, \rho'^y_b)$$

for some b, which by $\hat{\Gamma}' \sim \Gamma'$ and Proposition 5, turns to

$$(\Gamma', \rho')\ [\![A]\!]'^\circ\ (\Gamma', v : A, \rho'^v_b)$$

for some v. Clearly, $(M, (\Gamma, z : A, \rho^z_a)) \approx_\circ (N, (\Gamma', v : A, \rho'^v_b))$, taking care of ($\beta$1). A similar chain of implications gives (β2). □

Theorem 7 says that the extra bit an L-interpretation $[\![\cdot]\!]$ has over the L-model $\mathcal{M}[\![\cdot]\!]$ is inconsequential if, as is commonly agreed, the "essential" structure of transitions such as those in $[\![\cdot]\!]_\circ$ is invariant under bisimulations. Writing $[\![\cdot]\!]_\circ \underline{\leftrightarrow} [\![\cdot]\!]'_\circ$ for $([\![\cdot]\!], (\epsilon, \emptyset)) \underline{\leftrightarrow} ([\![\cdot]\!]', (\epsilon, \emptyset))$, Theorem 7 specializes to

COROLLARY 8. $[\![\cdot]\!]_\circ \underline{\leftrightarrow} [\![\cdot]\!]'_\circ$ *iff* $\mathcal{M}[\![\cdot]\!] \cong_p \mathcal{M}[\![\cdot]\!]'$.

We cannot expect to strengthen $[\![\cdot]\!]_\circ \underline{\leftrightarrow} [\![\cdot]\!]'_\circ$ to an isomorphism, as $\mathcal{M}[\![\cdot]\!]$ forgets the proofs in $[\![\cdot]\!]$. On the other hand, as noted in Section 3.1, we can, for countable purposes, read $\cong_p$ as $\cong$. And since in all cases, isomorphisms are partial isomorphisms, Corollary 8 yields

(8) if $\mathcal{M}[\![\cdot]\!] \cong \mathcal{M}[\![\cdot]\!]'$ then $[\![\cdot]\!]_\circ \underline{\leftrightarrow} [\![\cdot]\!]'_\circ$.

4. Extensions

The two previous sections have concentrated on simple fragments of dynamic and proof-conditional semantics. Do the results above carry over to extensions of these fragments? We turn first to proof-conditional semantics.

4.1. *$\mathcal{D}$ and $\mathcal{D}$*[**Ax**]

For the match-up above with L-formulas, the rule set $\mathcal{D}_\circ$ was confined to term constructors l and r. Obvious omissions include function application *ap*, pairing $\langle\cdot,\cdot\rangle$ and λ-abstraction, which figure respectively in the elimination rule for $\prod$

$$\frac{\Gamma \Rightarrow t : \left(\prod x : T\right)A \quad \Gamma \Rightarrow t' : T}{\Gamma \Rightarrow ap(t,t') : A[x \mapsto t']}$$

and the introduction rules for $\sum$

$$\frac{\Gamma \Rightarrow t : T \quad \Gamma \Rightarrow t' : A[x \mapsto t] \quad \Gamma, x : T \Rightarrow A \text{ type}}{\Gamma \Rightarrow \langle t,t'\rangle : \left(\sum x : T\right)A}$$

and for $\prod$

$$\frac{\Gamma, x : T \Rightarrow t : A}{\Gamma \Rightarrow \lambda x.t : \left(\prod x : T\right)A}.$$

In addition, from $\bot$, we get a term 0 saying that $\bot$ entails every well-formed formula

$$\frac{\Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow 0 : \bot \supset A}$$

and, should we wish, terms δ_A for double negation

$$\frac{\Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow \delta_A : \neg\neg A \supset A}$$

(where, as usual, $\neg A$ is $A \supset \bot$). For reasons to become clear shortly, we take care to attach type subscripts A onto δ_A.

Concentrating on the subscript-less constructs $l, r, ap, \langle \cdot, \cdot \rangle, \lambda$ and 0 (and dropping the superscript in Tm°), let Tm be the set of (extended) preterms t given by

$$t ::= x \mid lt \mid rt \mid ap(t, t) \mid \langle t, t \rangle \mid \lambda x.t \mid 0,$$

from which (extended) L-pretypes are then generated as before. (We leave "extended" out from here on, unless the contrast with the non-extended notions is relevant.) Let $\mathcal{D}$ be the rule set $\mathcal{D}_{\circ}$ plus the rules above excluding those for double negation δ_A, and let $\mathsf{context}^{\mathcal{D}}$ and $\Rightarrow^{\mathcal{D}}$ be defined as in $\mathsf{context}^{\circ}$ and $\Rightarrow^{\circ}$, with $\mathcal{D}_{\circ}$ extended to $\mathcal{D}$. Given an L-interpretation $[\![\cdot]\!]$, we can interpret a preterm $t \in \mathsf{Tm}$ in a model of the type *free* calculus embedding $[\![\cdot]\!]$, imposing a system of types on top of it so that for all $\Gamma \in \mathsf{context}^{\mathcal{D}}$ and $\rho \in [\![\Gamma]\!]^{\mathcal{D}}$,

$$\text{if `}\Gamma \Rightarrow^{\mathcal{D}} t : T\text{' then } [\![t]\!]^{\mathcal{D}}_{\Gamma,\rho} \in [\![T]\!]^{\mathcal{D}}_{\Gamma,\rho}.$$

Indeed, Feferman [5] and Aczel [1] provide type-free interpretations for dependent types, verifying rules such as β-reduction

$$\frac{\Gamma, x : T \Rightarrow t : T' \quad \Gamma \Rightarrow t' : T}{\Gamma \Rightarrow ap(\lambda x.t, t') = t[x \mapsto t'] : T'[x \mapsto t']}.$$

Whereas $l, r, ap, \langle \cdot, \cdot \rangle, \lambda$ and 0 can be interpreted uniformly over all types T (by typing type-free combinators), a similar interpretation for double negation δ_A is problematic. The obvious solution is to let the interpretation of δ_A vary with A. If $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset$, we can set $[\![\delta_A]\!]^{\mathcal{D}}_{\Gamma,\rho} = \lambda x.a$, for some $a \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$. Otherwise, $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} = \emptyset$ and $[\![\delta_A]\!]^{\mathcal{D}}_{\Gamma,\rho} = [\![0]\!]^{\mathcal{D}}_{\Gamma,\rho}$ will do. (Notice that this argument depends on the law of excluded middle for $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} = \emptyset$, as well as some choice principle.[9])

More generally, a set $\mathbf{Ax}$ of L-pretypes that we wish to treat as axioms might be introduced via

$$(9) \qquad \frac{\Gamma \Rightarrow A \text{ wff}}{\Gamma \Rightarrow p_A : A} A \in \mathbf{Ax}$$

(with δ_A amounting to the instance $p_{\neg\neg A \supset A}$). In fact, to allow $\mathbf{Ax}$ to include pretypes A built from terms p_B, we ought to generate the set $\mathsf{Tm}[L]$ of L-preterms t *simultaneously* with (further extended) L-pretypes A

$$t ::= x \mid lt \mid rt \mid ap(t, t) \mid \langle t, t \rangle \mid \lambda x.t \mid 0 \mid p_A,$$

$$A ::= \bot \mid R(t_1 \ldots t_n) \mid \left(\sum x : A\right) A \mid \left(\prod x : A\right) A \mid \\ \left(\sum x : U\right) A \mid \left(\prod x : U\right) A.$$

Given a collection **Ax** of these L-pretypes, we can then define $\mathcal{D}[\mathbf{Ax}]$ to be $\mathcal{D} + (9)$. **Ax** may, of course, fail to, in any natural sense, hold in an arbitrary L-interpretation $[\![\cdot]\!]$ – especially as **Ax** may contain a pair A and $A \supset \bot$. Blatant contradictions aside, we might for every $\Gamma \in \mathsf{context}^{\mathcal{D}}$ and $\rho \in [\![\Gamma]\!]^{\mathcal{D}}$, form the set $\mathcal{D}[\![\Gamma, \rho]\!]$ of L-pretypes supported by Γ, ρ,

$$\mathcal{D}[\![\Gamma, \rho]\!] = \{A \mid [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset\},$$

choosing for $A \in \mathcal{D}[\![\Gamma, \rho]\!] \cap \mathbf{Ax}$, an element of $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$ to interpret p_A (relative to Γ, ρ).

Turning next to the preterms t that a $\mathcal{D}$-context Γ assigns sorts, let

$$\mathsf{Tm}^{\mathcal{D}}[\Gamma] = \{t \mid \text{‘}\Gamma \Rightarrow^{\mathcal{D}} t : U\text{’for some } U \in L_0\},$$

and note that as with $\mathsf{Tm}^{\circ}[\Gamma]$, all terms in $\mathsf{Tm}^{\mathcal{D}}[\Gamma]$ are either variables $x \in \mathsf{Var}$ or of the form lt. But unlike $\mathsf{Tm}^{\circ}[\Gamma]$, if $\mathsf{Tm}^{\mathcal{D}}[\Gamma]$ is non-empty, then $\mathsf{Tm}^{\mathcal{D}}[\Gamma]$ must be infinite – if only for the rather uninteresting reason that if ‘$\Gamma \Rightarrow^{\mathcal{D}} t : U$’, then $l\langle t, \lambda x.x\rangle \in \mathsf{Tm}^{\mathcal{D}}[\Gamma]$ (since ‘$\Gamma \Rightarrow^{\mathcal{D}} \langle t, \lambda x.x\rangle : (\sum y : U)(\bot \supset \bot)$’). This is uninteresting because $[\![l\langle t, \lambda x.x\rangle]\!]^{\mathcal{D}}_{\Gamma,\rho} = [\![t]\!]^{\mathcal{D}}_{\Gamma,\rho}$ for any L-interpretation $[\![\cdot]\!]$. More interesting is the $\mathcal{D}$-context

$$x_1 : \left(\prod y : U\right)\left(\sum z : U\right) Succ(y, z), \\ x_2 : \left(\sum x : U\right)\top$$

(where $U \in L_0$ and $Succ \in L(U, U)$), relative to which the preterms

$$lx_2, l(ap(x_1, lx_2)), l(ap(x_1, l(ap(x_1, lx_2)))), \ldots, \\ t, l(ap(x_1, t)), \ldots$$

all have type U, denoting, under the obvious L-interpretation and instantiation, the natural numbers $0, 1, 2, \ldots, n, n + 1, \ldots$. By contrast, for every $\Gamma \in \mathsf{context}^{\circ}$, $\mathsf{Tm}^{\circ}[\Gamma]$ is finite.

Now, the difficulty in extending our reduction of $\mathcal{D}_{\circ}$ to $\mathcal{D}$ is evident. Writing $\top$ for $\bot \supset \bot$ and assuming $R \in L(U)$, let Γ be $x_1 : \top$ and Γ' be $x_1 : (\prod y : U)(\sum z : U)R(z)$, both of which belong to $\mathsf{context}^{\circ}$ and hence $\mathsf{context}^{\mathcal{D}}$. While

$$\mathsf{Tm}^{\circ}[\Gamma] = \mathsf{Tm}^{\circ}[\Gamma'] = \mathsf{Tm}^{\mathcal{D}}[\Gamma] = \mathsf{Tm}^{\mathcal{D}}[\Gamma'] = \emptyset,$$

Γ and Γ' have very different wff's:

$$\text{‘}\Gamma' \Rightarrow^{\mathcal{D}} \left(\sum x_2 : U\right) R(l(ap(x_1, t))) \text{ wff’}$$

but

$$\text{‘}\Gamma \not\Rightarrow^{\mathcal{D}} \left(\sum x_2 : U\right) R(l(ap(x_1, t))) \text{ wff’}$$

for t equal to any of $x_2, l(ap(x_1, x_2)), l(ap(x_1, l(ap(x_1, x_2)))), \ldots$.

The previous example suggests a more sophisticated reduction of $\Gamma \in \mathsf{context}^{\mathcal{D}}$ than $\mathsf{Tm}^{\mathcal{D}}[\Gamma]$. Writing Sk for Skolem functions,[10] let

$$\begin{aligned}\mathsf{Sk}[\Gamma] = \Big\{ t \mid \text{‘}\Gamma \Rightarrow^{\mathcal{D}} t : \left(\prod x_1 : U_1\right) \cdots \\ \left(\prod x_n : U_n\right)\left(\sum x : U\right) A\text{’ for some} \\ U_1 \ldots U_n, U \in L_0 \text{ and } x_1, \ldots, x_n, x, A \Big\},\end{aligned}$$

the idea being that from $t : (\prod x_1 : U_1) \cdots (\prod x_n : U_n)(\sum x : U)A$, we can extract the A-Skolem function

$$\begin{aligned}&\lambda x_1 \cdots \lambda x_n . l(ap(\cdots ap(t, x_1) \cdots, x_n)) : \\ &\quad (U_1 \to \cdots (U_n \to U) \cdots) \cong (U_1 \times \cdots \times U_n) \to U.\end{aligned}$$

$\mathsf{Sk}[\Gamma]$ incorporates $\mathsf{Tm}^{\mathcal{D}}[\Gamma]$ in that whenever ‘$\Gamma \Rightarrow^{\mathcal{D}} t : U$’,

$$\text{‘}\Gamma \Rightarrow^{\mathcal{D}} \langle t, \lambda x.x \rangle : \left(\sum z : U\right) \top\text{’}.$$

The problem, however, is that Sk misses out on preterms that take proofs as arguments,[11] such as a variable of type $(\prod x : U)(R(x) \supset (\sum y : V)\top)$ where $R \in L(U)$ and $U, V \in L_0$. To press the point, let $[\![\cdot]\!]$ be an L-interpretation such that $[\![U]\!] = \{u\}$, $[\![V]\!] = \{1, 2\}$ and $[\![R, u]\!] = \{3, 4\}$. The prenex form $(\prod x : U)(\sum y : V)(R(x) \supset \top)$ has (up to extensionality) two Skolem functions in $[\![\cdot]\!]$, $u \mapsto 1$ and $u \mapsto 2$, neither of which can (alone) match the map $\alpha : \langle u, 3\rangle \mapsto 1, \langle u, 4\rangle \mapsto 2$ from $[\![(\prod x : U)(R(x) \supset (\sum y : V)\top)]\!]^{\mathcal{D}}_{\epsilon}$ in that if $S \in L(V)$ differentiates 1 from 2 in $[\![\cdot]\!]$ – say, $[\![S, 1]\!] = \emptyset \neq [\![S, 2]\!]$ – then α contains witnesses to both $(\exists z \in V)S(z)$ and $(\exists z \in V)\neg S(z)$, whereas neither $u \mapsto 1$ nor $u \mapsto 2$ does (on its own). Although such examples may impress only readers concerned with the structure of transitions up to bisimulation, there is no denying that reification of proofs gives $\mathcal{D}$ considerable scope. That said, Skolem functions have attracted some attention in the discourse literature (e.g. Schubert [26]), and carve out a fragment of $\mathcal{D}$ that we explore next.

4.2. *$\mathcal{D}_\forall$ and a Dynamic Universal Quantifier*

A modest extension of $\mathcal{D}_\circ$ capturing Skolem functions is the rule set $\mathcal{D}_\forall$ consisting of $\mathcal{D}_\circ$ and

$$\frac{\Gamma \Rightarrow t : (\prod x : U)A \qquad\qquad \Gamma \Rightarrow t' : U}{\Gamma \Rightarrow ap(t, t') : A[x \mapsto t']} U \in L_0.$$

Replacing $\mathcal{D}_\circ$ by $\mathcal{D}_\forall$, the notions $\mathsf{context}^\circ$, $\Rightarrow^\circ$, $[\![\cdot]\!]^\circ$ and Tm° convert smoothly into $\mathsf{context}^\forall$, $\Rightarrow^\forall$, $[\![\cdot]\!]^\forall$ and $\mathsf{Tm}^\forall$, the set of preterms at stake being Var closed under l, r and ap. The obvious question is how to modify the translations A°_Γ, reductions $(\Gamma, \rho)^\circ$ and transitions $\|\cdot\|$ (in dynamic semantics) to $A^\forall_\Gamma$, $(\Gamma, \rho)^\forall$ and $\|\cdot\|^\forall$ so that, for instance, the truth equivalence given by Lemma 6(c) becomes

PROPOSITION 9. *Whenever '$\Gamma \Rightarrow^\forall A$ wff',*

$$[\![A]\!]^\forall_{\Gamma,\rho} \neq \emptyset \quad \textit{iff} \quad (\Gamma, \rho)^\forall \in dom\big(\|A^\forall_\Gamma\|^\forall_{\mathcal{M}[\![\cdot]\!]}\big)$$

for all L-interpretations $[\![\cdot]\!]$ and $\rho \in [\![\Gamma]\!]^\forall$.

Let us start with the translations $A^\forall_\Gamma$, and see what extension to the notion of L-formula (in dynamic semantics) that leads to.[12] As with $\cdot^\circ_\Gamma$, the definition proceeds by induction on A (with $A^\forall_\Gamma = A^\circ_\Gamma$ whenever '$\Gamma \Rightarrow^\circ A$ wff').

$$\bot^\forall_\Gamma = \bot,$$

$$R(t_1 \dots t_n)^\forall_\Gamma = \begin{cases} R(t_1{}^\forall_\Gamma \dots t_n{}^\forall_\Gamma) & \text{if '}\Gamma \Rightarrow^\forall R(t_1 \dots t_n) \ \mathsf{wff}\text{,'} \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\Big(\Big(\sum x : U\Big)A\Big)^\forall_\Gamma = \begin{cases} (\exists x \in U)A^\forall_{\Gamma,x:U} & \text{if } x \text{ not in } \Gamma, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\Big(\Big(\prod x : U\Big)A\Big)^\forall_\Gamma = \begin{cases} (\forall x \in U)A^\forall_{\Gamma,x:U} & \text{if } x \text{ not in } \Gamma, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\Big(\Big(\sum x : A\Big)B\Big)^\forall_\Gamma = \begin{cases} A^\forall_\Gamma \wedge B^\forall_{\Gamma,x:A} & \text{if } x \text{ not in } \Gamma \text{ nor } A, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\Big(\Big(\prod x : A\Big)B\Big)^\forall_\Gamma = \begin{cases} A^\forall_\Gamma \supset B^\forall_{\Gamma,x:A} & \text{if } x \text{ not in } \Gamma \text{ nor } A, \\ \uparrow & \text{otherwise,} \end{cases}$$

where the translation $t^\forall_\Gamma$, for every t such that $\Gamma \Rightarrow^\forall t : U$ (for some $U \in L_0$), is given by

$$x^\forall_\Gamma = x \quad \text{for } x \in \mathsf{Var},$$

$$(lt)^\forall_\Gamma = (x, sb(t)_\Gamma) \quad \text{where '}\Gamma \Rightarrow^\forall t : \Big(\sum x : U\Big)A\text{'}$$

with $sb(t)$ recording the substitutions in ap-subterms of t as follows

$$\begin{aligned}
&sb(x)_\Gamma = \emptyset \quad \text{for } x \in \mathsf{Var}\\
&sb(lt)_\Gamma = sb(t)_\Gamma\\
&sb((rt)_\Gamma = sb(t)_\Gamma\\
&sb(ap(t,t'))_\Gamma = sb(t)_\Gamma \cup \{(x, {t'}^{\forall}_{\Gamma})\} \quad \text{where `}\Gamma \Rightarrow^{\forall} t' : U\text{'}\\
&\qquad\qquad\qquad \text{and `}\Gamma \Rightarrow^{\forall} t : \left(\prod x : U\right) A\text{'.}
\end{aligned}$$

The difference with $\mathcal{D}_\circ$ (Section 2.3) is the simultaneous definition of

$$t^{\forall}_{\Gamma} \text{ for } t \text{ such that } \Gamma \Rightarrow^{\forall} t : U \text{ (for some } U \in L_0)$$

and

$$sb(t')_\Gamma \text{ for } t' \text{ such that } \Gamma \Rightarrow^{\forall} t' : A \text{ (for some } A).$$

An example should clarify matters.

EXAMPLE. Assuming $R \in L(U,U,U,U,U)$ and $S \in L(U,U)$, let B be

$$\left(\prod x_1 : U\right)\left(\prod x_2 : U\right)\left(\sum y_1 : U\right)\left(\prod x_3 : U\right)$$
$$\left(\sum y_2 : U\right) R(x_1, x_2, y_1, x_3, y_2)$$

and A be

$$\left(\sum z : B\right)\left(\sum z_1 : U\right)\left(\prod z_2 : U\right) S(t, t'),$$

where t is $l(ap(ap(z,z_1),z_2))$ and t' is $l(ap(r(ap(ap(z,z_1),z_2)),t)))$. Then $A^{\forall}_{\epsilon}$ is the $L_{\forall}$-formula

$$B^{\forall}_{\epsilon} \wedge (\exists z_1 \in U)(\forall z_2 \in U)\ S(t^{\forall}_{\Gamma}, {t'}^{\forall}_{\Gamma}),$$

where $B^{\forall}_{\epsilon} = B^{\circ}_{\epsilon}$ is

$$(\forall x_1 \in U)(\forall x_2 \in U)(\exists y_1 \in U)(\forall x_3 \in U)(\exists y_2 \in U)$$
$$R(x_1, x_2, y_1, x_3, y_2),$$

Γ is $z : B, z_1 : U, z_2 : U$, and

$$\begin{aligned}
t^{\forall}_{\Gamma} &= (y_1, sb(ap(ap(z,z_1),z_2))_\Gamma)\\
&= (y_1, sb(ap(z,z_1))_\Gamma \cup \{(x_2, z_2)\})\\
&= (y_1,\ \{(x_1,z_1),(x_2,z_2)\}),
\end{aligned}$$

$$\begin{aligned} t'^{\forall}_{\Gamma} &= (y_2, sb(ap(r(ap(ap(z, z_1), z_2)), t))_{\Gamma}) \\ &= (y_2, sb(r(ap(ap(z, z_1), z_2)))_{\Gamma} \cup \{(x_3, t^{\forall}_{\Gamma})\}) \\ &= (y_2, sb(ap(ap(z, z_1), z_2))_{\Gamma} \cup \{(x_3, t^{\forall}_{\Gamma})\}) \\ &= (y_2,\ \{(x_1, z_1), (x_2, z_2), (x_3, t^{\forall}_{\Gamma})\}). \end{aligned}$$

The translations $A^{\forall}_{\Gamma}$ require that the $\forall$-extended L-formulas – let us call them $L_{\forall}$*-formulas* – have terms/arguments drawn not only from Var but from the least set $\mathsf{Var}^{\forall}$ satisfying the equation

$$\mathsf{Var}^{\forall} = \mathsf{Var} \cup (\mathsf{Var} \times Sb(\mathsf{Var})),$$

where, writing $Pow_{\text{fin}}(X)$ for the set of finite subsets of X,

$$Sb(\mathsf{Var}) = Pow_{\text{fin}}(\mathsf{Var} \times \mathsf{Var}^{\forall}).$$

To interpret elements of $\mathsf{Var}^{\forall}$, let us fix an L-model M, and lift a variable assignment $f \in \mathsf{Var}_M$ to a partial function $f^{\forall}$ from $\mathsf{Var}^{\forall}$ to $\bigcup_{U \in L_0} U_M$ defined inductively as follows

(a) $f \subseteq f^{\forall}$ (i.e., for every $x \in dom(f)$, $x \in dom(f^{\forall})$ and $f^{\forall}(x) = f(x)$), and
(b) for all $x \in dom(f)$ and $\alpha \in Sb(\mathsf{Var})$, if for every $(y, s) \in \alpha$,

$$y \in dom(f),\ s \in dom(f^{\forall}) \text{ and } f(y) = f^{\forall}(s),$$

then $(x, \alpha) \in dom(f^{\forall})$ and $f^{\forall}((x, \alpha)) = f(x)$.

For anaphoric uses of universally quantified variables (illustrated by the variables x_1, x_2 and x_3 in the example above), let us interpret $s \in \mathsf{Var}^{\forall}$ relative not only to a variable assignment f in Var_M but also a non-empty subset F of Var_M. That interpretation $s^{F,f}$ is defined as follows.

If s is a variable, say x, then $x^{F,f}$ is defined iff $x \in dom(f)$, in which case $x^{F,f}$ is $f(x)$.
Otherwise (s is not a variable), $s^{F,f}$ is defined iff there is a $g \in F$ such that $s \in dom(g^{\forall})$ and for all $h \in F$ such that $s \in dom(h^{\forall})$, $g^{\forall}(s) = h^{\forall}(s)$. In that case, $s^{F,f}$ is $g^{\forall}(s)$.

Notice that $x^{F,f}$ is independent of F, whereas if $s \notin \mathsf{Var}$, then $s^{F,f}$ is independent of f.

Now, writing $\mathsf{Var}^{\forall}_M = Pow(\mathsf{Var}_M) - \{\emptyset\}$ for the family of non-empty subsets of Var_M, let us define the input/output interpretation $\|\varphi\|^{\forall}_M$ of an $L_{\forall}$-formula φ to be a binary relation on $\mathsf{Var}^{\forall}_M$ as follows, where $F, F' \in \mathsf{Var}^{\forall}_M$.

$$F \|R(s_1 \ldots s_n)\|^{\forall}_M F' \quad \text{iff} \quad \begin{array}{l} F = F' \text{ and for every } f \in F, \\ {s_i}^{F,f} \text{ is defined for } 1 \leq i \leq n \\ \text{and } R_M({s_1}^{F,f} \ldots {s_n}^{F,f}) \end{array}$$

and so if the arguments s_i are variables x_i, $F\|R(x_1 \ldots x_n)\|^{\forall}_M F'$ iff $F = F'$ and for every $f \in F$, $f\|R(x_1 \ldots x_n)\|_M f$. The clauses

$$\begin{aligned}
&\|\bot\|^{\forall}_M = \emptyset, \\
&F\|\varphi \wedge \psi\|^{\forall}_M F' \quad \text{iff} \quad F\|\varphi\|^{\forall}_M G \text{ and } G\|\psi\|^{\forall}_M F' \text{for some } G, \\
&F\|\varphi \supset \psi\|^{\forall}_M F' \quad \text{iff} \quad F = F' \text{ and whenever } F\|\varphi\|^{\forall}_M G, \\
&\qquad\qquad\qquad\qquad G \in dom(\|\psi\|^{\forall}_M), \\
&F\|(\exists x \in U)\varphi\|^{\forall}_M F' \quad \text{iff} \quad \text{for some } u \in U_M, \\
&\qquad\qquad\qquad\qquad \{f^x_u | f \in F\}\|\varphi\|^{\forall}_M F'
\end{aligned}$$

are pretty much as in Section 3.1 (with $\|(\exists x \in U)\varphi\|^{\forall}_M$ assigning x the same value u, regardless of the $f \in F$). The clause for $\forall$ diverges from Section 3.1 if $U_M \neq \emptyset$,

$$\begin{aligned}
F\|(\forall x \in U)\varphi\|^{\forall}_M F' \quad \text{iff} \quad &(U_M \neq \emptyset \text{ and for some} \\
&p : U_M \to \mathsf{Var}^{\forall}_M, \\
&(\forall u \in U_M)\{f^x_u \mid f \in F\}\|\varphi\|^{\forall}_M p(u) \\
&\text{and } F' = \bigcup\{p(u) \mid u \in U_M\}) \\
&\text{or } (U_M = \emptyset \text{ and } F = F'),
\end{aligned}$$

the intuition behind p being that it is the "image collapse" of a proof in the dependent function space interpretation $(\prod x : U)$ of $(\forall x \in U)$.

Next, given $\Gamma \in \mathsf{context}^{\forall}$ and $\rho \in [\![\Gamma]\!]^{\forall}$, the reduction $(\Gamma, \rho)^{\forall} \in \mathsf{Var}^{\forall}_{\mathcal{M}[\![\cdot]\!]}$ in Proposition 9 collects together variables assignments according to

$$\begin{aligned}
&(\epsilon, \emptyset)^{\forall} = \{\emptyset\}, \\
&(\Gamma, x : U, \rho^x_u)^{\forall} = \{f^x_u \mid f \in (\Gamma, \rho)^{\forall}\}, \\
&(\Gamma, x : A, \rho^x_a)^{\forall} = \{f \cup g \mid f \in (\Gamma, \rho)^{\forall} \text{and } g \in (A, a)_{\Gamma,\rho}\},
\end{aligned}$$

where $(A, a)_{\Gamma,\rho}$ is, for all A and a such that '$\Gamma \Rightarrow^{\forall} A$ wff' and $a \in [\![A]\!]^{\forall}_{\Gamma,\rho}$, a semantically interpreted $\forall$-analog of the functions *new* and *wen* in Section 2.2. To be precise, we define $(A, a)_{\Gamma,\rho} \in \mathsf{Var}^{\forall}_{\mathcal{M}[\![\cdot]\!]}$ by induction on A, leaving $\bot$ out (as there is no $a \in [\![\bot]\!]^{\forall}_{\Gamma,\rho}$), treating $L_{\forall}$-atomic formulas and implications "statically"

$$\begin{aligned}
&(R(t_1 \ldots t_n), a)_{\Gamma,\rho} = \{\emptyset\}, \\
&\left(\left(\prod x : A\right)B, c\right)_{\Gamma,\rho} = \{\emptyset\}
\end{aligned}$$

and allowing the rest to be "dynamic"

$$\begin{aligned}
&\left(\left(\sum x : A\right)B, \langle a, b\rangle\right)_{\Gamma,\rho} \\
&\quad = \{f \cup g \mid f \in (A, a)_{\Gamma,\rho} \text{ and } g \in (B, b)_{\Gamma,x\,:\,A,\rho^x_a}\}, \\
&\left(\left(\sum x : U\right)A, \langle u, a\rangle\right)_{\Gamma,\rho} \\
&\quad = \{f^x_u \mid f \in (A, a)_{\Gamma,x\,:\,U,\rho^x_u}\}, \\
&\left(\left(\prod x : U\right)A, c\right)_{\Gamma,\rho} \\
&\quad = \begin{cases} \bigcup_{u\in[\![U]\!]} \{f^x_u \mid f \in (A, c(u))_{\Gamma,x\,:\,U,\rho^x_u}\} & \text{if } [\![U]\!] \neq \emptyset, \\ \{\emptyset\} & \text{otherwise.} \end{cases}
\end{aligned}$$

NVC′ guarantees that in all occurrence of $f \cup g$ above, $dom(f) \cap dom(g) = \emptyset$ (whence $f \cup g \in \mathsf{Var}_{\mathcal{M}[\![\cdot]\!]}$). This completes the ingredients for Proposition 9, which now follows by a routine induction on $\mathcal{D}_\forall$-derivations of $\Gamma \Rightarrow A$ wff.

It is perhaps worth noting that there is more to $L_\forall$-formulas than the fragment Proposition 9 covers. This includes simple translations of discourses such as (c) from Groenendijk and Stokhof [11].

(c) Every player chooses a pawn. He puts it on square one.
$(\forall x \in \mathit{player})(\exists y \in \mathit{pawn})\ \mathit{chooses}(x, y) \wedge \mathit{puts\text{-}on\text{-}sqr\text{-}one}(x, y)$

The interested reader is referred to Fernando [7], where the step from $f \in \mathsf{Var}_M$ to non-empty sets $F \subseteq \mathsf{Var}_M$ of such is linked to *conjunctive* branching, introduced into dynamic logic in Peleg [23] and widely found in so-called and/or graphs for 2-person games.

5. Discussion

Having associated dynamic and proof-conditional semantics (in the introduction) with the intriguing proposals (**D**) and (**P**), suggesting that truth be updated and proofs used in constructing well-formed formulas, let us conclude by reviewing how these ideas have fared in the synthesis attempted above. With an eye on not only truth but context change, we pushed our semantic analysis (in Section 3) of the translations A°_Γ (from Section 2) beyond truth equivalence, Lemma 6(c), to bisimulation equivalence, Theorem 7. However edifying the extra work in forming bisimulations may have been, we carried our analysis of $A^\forall_\Gamma$ (in Section 4) only up to Proposition 9, the ∀-analog of Lemma 6(c). In falling short of the ∀-analog of Theorem 7, we invited speculation as to whether the author had finally run out of steam or had decided not to try the patience of his indulgent reader further. Be that as it may, an obvious defect in the link forged between dynamic and proof-conditional semantics is the omission of many of the rules in the set $\mathcal{D}$ (defined in Section 4.1).

5.1. *Derivation versus Interpretation*

While we may have expelled various rules from $\mathcal{D}_\circ$ and $\mathcal{D}_\forall$, we also introduced oracles $[\![\cdot]\!]$ telling us if, for example, $R(u)$ is true (i.e. $[\![R, u]\!] \neq \emptyset$), whether or not there is a $\mathcal{D}$-derivation settling the matter. The concept of an interpretation independent of *any* derivation is manifestly as suspect as truth independent of proof. But for any rule set $\mathcal{D}'$ that is intuitionistically justifiable (e.g. any subset of $\mathcal{D}$), the incompleteness of $\mathcal{D}'$-derivations leaves room for a notion of L-interpretation transcending those limitations. Classical and constructive logicians may, of course, disagree on what L-interpretations $[\![\cdot]\!]$ there are, and on how $[\![\cdot]\!]$ extends to $\mathcal{D}'$-contexts, types and terms. But surely we can be forgiven for trying to steer clear of these controversies, concentrating instead on matters that bear directly on applications to discourse.

In the type reduction above, $\mathcal{D}_\circ$-derivations and L-interpretations serve very different purposes. $\mathcal{D}_\circ$-derivations feed into the assembly of well-formed formulas, while L-interpretations $[\![\cdot]\!]$ provide the L-models $\mathcal{M}[\![\cdot]\!]$, over which dynamic semantics defines input/output relations. These i/o relations specify programs which, under the reduction above, amount to searches for proofs in $[\![\cdot]\!]$ – proofs which may or may not belong to $\mathcal{D}'$. The link established in Theorem 7 respects the "propositions-as-types"[13] intuition lost in the embedding described in Muskens [21] of DRT in classical higher-order logic. But by deleting (from $\mathcal{D}_\circ$) all of $\mathcal{D}$'s introduction and elimination rules save those for the projections l and r, have we not trivialized propositions-as-types? Only if we steadfastly refuse to step beyond the fragment $\mathcal{D}_\circ$ of $\mathcal{D}$. A well-known example lying outside $\mathcal{D}_\circ$ but, as it happens, within $\mathcal{D}_\forall$ (and hence $L_\forall$) is (g).

(g) If each child is given a gift for Christmas, some child will open $^?$it/[her gift] today.

$\mathcal{D}_\forall$- $(\prod u : (\prod x : \mathit{child})(\sum y : \mathit{Xgift})\mathit{given}(x, y))\ (\sum z : \mathit{child})$ $\mathit{op\text{-}td}(z, l(ap(u, z)))$,

$L_\forall$- $(\forall x \in \mathit{child})(\exists y \in \mathit{Xgift})\mathit{given}(x, y) \supset (\exists z \in \mathit{child})$ $\mathit{op\text{-}td}(z, (y, \{(x, z)\}))$.

Differences in empirical coverage between formalisms can be a touchy matter, especially when trade-offs are involved (e.g. for DRT vs TTG, see Watson [28]). Obviously, extra-logical "linguistic" considerations must enter into translations of English to well-formed formulas. But it is far from clear whether to tolerate, in their absence, cases of overgeneration or of undergeneration. In other words, should linguistics contribute *constraints* or *generative* mechanisms? The former would suit TTG, as the inferential possibilities TTG generates ought to be constrained by linguistic factors. Among these factors, many have observed, is the difference in (g) above between definite descriptions such as her gift and pronouns such as it.

5.2. *NVC Revisited: A Clash of Paradigms and Beyond*

By admitting formalizations of proofs relative to rule sets into proof-conditional semantics, have we reduced dynamic semantics into a proof-conditional fragment given by some rule set such as $\mathcal{D}_\circ$ (or perhaps $\mathcal{D}_\forall$)? This would be surprising, given that proof-conditional semantics can (under propositions-as-types) be seen as a declarative (functional) programming language (e.g. Girard, Lafont and Taylor [10]) whereas dynamic semantics is commonly associated with assignment-based (imperative) programs. Indeed, the thrust of (**D**) has more than once been identified with a procedural turn in semantics. Declarative though it may be, however, TTG is arguably more dynamic than dynamic semantics. Type-theoretic context change yields (in $\mathcal{D}$, if not $\mathcal{D}_\circ$) many more terms for anaphoric reference. But then the potential of procedural programming is barely scratched by the applications above. One of the culprits is the novel variable condition (NVC), which bans re-assignments to variables.

Why rob procedural programming of some of its power by imposing NVC? The reason can be traced to the reduction of programs in dynamic logic to their input/output relations. A crucial clause is relational composition

(10) $f \|\varphi \wedge \psi\|_M f'$ iff $f\|\varphi\|_M g$ and $g\|\psi\|_M f'$ for some g,

discarding intermediate outputs g. This is evidently at gross variance with the retention of witnesses in $\sum$ (witnesses to existential claims being one of the characteristic pre-occupations of intuitionistic logic). But now, if each output f' to $\|\psi\|_M$ encodes all the information in the input g, then nothing is lost in throwing out intermediate outputs g in (10). Enter persistence (5) from Section 3.1, stating that $f \subseteq f'$ whenever $f\|\varphi\|_M f'$.

Should we then never destroy information, and enshrine persistence as a principle of interpretation? Arguably not. While anaphoric possibilities may multiply during a discourse, some possibilities may also drop out (or at the very least degrade). Beyond that, non-monotonicity has become a significant feature of various accounts of discourse interpretation (e.g. Asher and Lascarides [3]). Nevertheless, there is a widespread feeling that persistence has its place, immune from counter-examples of the kind just mentioned. Some such space is afforded by the distinction drawn in Kamp and Reyle [14] between

(i) an algorithm translating English sentences (or some syntactic analyses thereof) into well-formed formulas

and

(ii) a scheme (e.g. $\| \cdot \|_M$) interpreting such well-formed formulas model-theoretically.

Persistence may find a home in (ii), if not in (i). Now, the trouble with characterizing DRT as a fragment of proof-conditional semantics is that, just as intuitionistic type theory reaches far beyond $\mathcal{D}_\circ$, the bulk of work in DRT revolves not around the input/output relations $\| \cdot \|_M$, but on some algorithm constructing logical forms for English discourse (e.g. van der Sandt [25]).[14] What impact could replacing models by proofs have on the design of such an algorithm? Some answers are under way (e.g. Krause [18], Krahmer and Piwek [17], Fernando [8]).

Acknowledgments

I am very grateful to an anonymous referee for taking this paper on at a stage when it was hardly readable, and for much help in trying to get it past that stage. I thank also Krister Segerberg for not giving up on it/me.

NOTES

[1] Refinements to L-formulas will be made below. In the meantime, note that for $R \in L(U)$, it is natural to abbreviate $(\exists x \in U)(R(x) \wedge A)$ to $(\exists x \in R)A$, and $(\forall x \in U)(R(x) \supset A)$ to $(\forall x \in R)A$. In particular, assuming that *farmer*, *donkey* $\in L(U)$ for some $U \in L_0$, the antecedent in (d) above is recorded in abbreviated form.

[2] Although (d) and (p) are written in abbreviated form, the translations operate on their official forms in exactly the same manner.

[3] The reader troubled by the use of equality = between possibly non-denoting terms should feel free to write Kleene equality $\simeq$ instead.

[4] The separation of wff from sorts (and relations) accords with the distinction between propositions and properties advocated in Fox [9].

[5] The notation $\cdot^\circ_\Gamma$ is being overloaded here, serving both as a function from $\mathsf{Tm}^\circ[\Gamma]$ to Var and as a partial function from L-pretypes to L-formulas.

[6] The side condition NVC$'$ above ensures that as a function from $\mathsf{Tm}^\circ[\Gamma]$ to Var, $\cdot^\circ_\Gamma$ is 1-1, and therefore has an inverse.

[7] Diffferences between these and definitions in, for example, Zeevat [29], Dekker [4] and Vermeulen [27] are largely inessential – at least for the translations A°_Γ considered below.

[8] This observation is a starting point for the study of bisimulations and predicate logic reported in Fernando [6].

[9] Intuitionistic versus classical logic aside, differences in the models embedding $[\![\cdot]\!]$ may, of course, lead to conflicting predictions as to whether or not $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$ is inhabited. For instance, whether or not an induction principle holds over an interpretation of arithmetic may depend on just what functions and classes we have at our disposal. We will proceed naively below, following the usual practice in model theory of making as little fuss as possible about the set-theoretic universe from which models are drawn.

[10] Analogously, an alternative notation for Tm is He, for Henkin witnesses.

[11] A basic difference between classical and intuitionistic logic is that assuming x does not occur free in φ, the equivalence

$$\varphi \supset (\exists x)\psi \equiv (\exists x)(\varphi \supset \psi)$$

holds in classical logic, but breaks down intuitionistically, as the x-witness in $\varphi \supset (\exists x)\psi$ may well depend on the proof of φ (which is unavailable at the start of $(\exists x)(\varphi \supset \psi)$). Since $\mathcal{D}$-derivations are intuitionistically sound, we cannot resort to prenex normal forms.

[12] That extension can be motivated independently of proof-conditional semantics (Fernando [7]).

[13] A linguist familiar with this slogan is likely to have heard it via categorial grammar – i.e., in connection not with TTG but with *Type Logical Grammar* (TLG, Morrill [20]). TLG should *not* be confused with proof-conditional semantics, as proofs in TLG pertain exclusively to parsing, lacking the truth-conditional force that $\mathcal{D}$-derivations have.

[14] It is perhaps in such construction algorithms where the force of a “procedural turn in semantics” can be felt. Allied to (and at the same time constituting a twist to) dynamic semantics is *Dynamic Syntax* (Kempson, Meyer-Viol and Gabbay [16]), featuring dynamic logic in a propositional (as opposed to quantified) form, employed for representationalist ends.

REFERENCES

1. Aczel, P.: Frege structures and the notions of proposition, truth and set, in J. Barwise, H. J. Keisler, and K. Kunen (eds), *The Kleene Symposium*, North-Holland, Amsterdam, 1980.
2. Ahn, R. and Kolb, H.-P.: Discourse representation meets constructive mathematics, in L. Kálmán and L. Pólos (eds), *Papers from the Second Symposium on Logic and Language*, Akademiai Kiado, Budapest, 1990.
3. Asher, N. and Lascarides, A.: Bridging, *J. Semantics* **15**(1) (1998).
4. Dekker, P.: *Transsentential Meditations: Ups and Downs in Dynamic Semantics*, ILLC Dissertation Series, Number 1, University of Amsterdam, 1993.
5. Feferman, S.: A language and axioms for explicit mathematics, in J. N. Crossley (ed.), *Algebra and Logic*, Lecture Notes in Math. 450, Springer-Verlag, Berlin, 1975.
6. Fernando, T.: Bisimulations and predicate logic, *J. Symbolic Logic* **59**(3) (1994).
7. Fernando, T.: Generalized quantifiers as second-order programs – ‘dynamically’ speaking, naturally, in P. Dekker and M. Stokhof (eds), *Proc. Ninth Amsterdam Colloquium*, ILLC, University of Amsterdam, 1994.
8. Fernando, T.: Three processes in natural language interpretation, in *Festschrift for Solomon Feferman*, ASL Lecture Notes, to appear.
9. Fox, Ch.: Discourse representation, type theory and property theory, in H. Bunt, R. Muskens, and G. Rentier (eds), *Proc. International Workshop on Computational Semantics*, ITK, Tilburg, 1994.

10. Girard, J.-Y., Lafont, Y., and Taylor, P.: *Proofs and Types*, Cambridge Tracts in Theoret. Comput. Sci. 7, Cambridge University Press, 1989.
11. Groenendijk, J. and Stokhof, M.: Dynamic predicate logic, *Linguistics and Philosophy* **14** (1991).
12. Harel, D.: Dynamic logic, in D. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic*, Vol. 2, Reidel, Dordrecht, 1984.
13. Heim, I.: *The Semantics of Definite and Indefinite Noun Phrases*, Dissertation, University of Massachusetts, Amherst, 1982. Published by Garland Press, New York, 1988.
14. Kamp, H. and Reyle, U.: *From Discourse to Logic*, Kluwer Academic Publishers, Dordrecht, 1993.
15. Keisler, H. J.: Fundamentals of model theory, in J. Barwise (ed.), *Handbook of Mathematical Logic*, North-Holland, Amsterdam, 1977.
16. Kempson, R., Meyer-Viol, W., and Gabbay, D.: *Dynamic Syntax: The Flow of Language Understanding*, Blackwell, Oxford, 2000.
17. Krahmer, E. and Piwek, P.: Presupposition projection as proof construction, in *Computing Meaning*, Kluwer Academic Publishers, Dordrecht, 1999.
18. Krause, P.: Presupposition and abduction in type theory, in E. Klein et al. (eds), *Computational Logic and Natural Language Processing*, South Queensferry, Scotland, 1995.
19. Martin-Löf, P.: *Intuitionistic Type Theory*, Bibliopolis, Napoli, 1984. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.
20. Morrill, G. V.: *Type Logical Grammar*, Kluwer Academic Publishers, Dordrecht, 1994.
21. Muskens, R.: Combining Montague semantics and discourse representation, *Linguistics and Philosophy* **19**(2) (1996).
22. Park, D.: Concurrency and automata on infinite sequences, in P. Deussen (ed.), *Proc. 5th GI Conference*, Lecture Notes in Comput. Sci. 104, Springer-Verlag, Berlin, 1981.
23. Peleg, D.: Concurrent dynamic logic, *J. Assoc. Comput. Mach.* **34**(2) (1987).
24. Ranta, A.: *Type-Theoretical Grammar*, Oxford University Press, Oxford, 1994.
25. Van der Sandt, R. A.: Presupposition projection as anaphora resolution, *J. Semantics* **9**(4) (1992).
26. Schubert, L.: Dynamic Skolemization, in *Computing Meaning*, Kluwer Academic Publishers, Dordrecht, 1999.
27. Vermeulen, C. F. M.: *Explorations of the Dynamic Environment*, Dissertation, Utrecht University, 1994.
28. Watson, M.: A critique of a proof-theoretic account of anaphora, in P. Dekker and M. Stokhof (eds), *Proc. Tenth Amsterdam Colloquium*, ILLC, University of Amsterdam, 1996.
29. Zeevat, H.: A compositional approach to discourse representation theory, *Linguistics and Philosophy* **12** (1989).

Computer Science,
Trinity College,
Dublin 2, Ireland
E-mail: tim.fernando@tcd.ie