Centrum voor Wiskunde en Informatica
REPORT *RAPPORT*

Incremental Dynamics

D.J.N. van Eijck

Information Systems (INS)

INS-R9811 November 1998

Report INS-R9811
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Incremental Dynamics

Jan van Eijck

CWI, Amsterdam, ILLC, Amsterdam, Uil-OTS, Utrecht

EMAIL: jve@cwi.nl

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

An incremental semantics for a logic with dynamic binding is developed on the basis of a variable free notation for dynamic logic. The variable free indexing mechanism guarantees that active registers are never overwritten by new quantifier actions. The resulting system has the same expressive power as Dynamic Predicate Logic or Discourse Representation Theory, but comes with a more well behaved consequence relation. A calculus for dynamic reasoning with anaphora is presented and its soundness and completeness are established. Incremental dynamic logic provides an explicit account of anaphoric context and yields new insight into the dynamics of anaphoric linking in reasoning.

1991 Mathematics Subject Classification: 03B65, 68Q55
1991 Computing Reviews Classification System: F.3.1, F.3.2, I.2.4, I.2.7
Keywords and Phrases: dynamic semantics of natural language, complete calculus for dynamic reasoning with anaphora, incremental interpretation, monotonic semantics, anaphora and context
Note: Work carried out under project P4303. Submitted for publication in the Journal of Logic, Language and Information.

1. Updating Anaphoric Contexts

In recent developments of natural language semantics, problems of pronominal reference and anaphoric linking have inspired logicians to a dynamic turn in natural language semantics. This started with Discourse Representation Theory (Kamp [15]) and File Change Semantics (Heim [13]), and various attempts at rational reconstruction of these proposals, with Barwise [3] and Groenendijk and Stokhof [11] as the most prominent ones. The gist of these proposals is that the static variable binding regime from standard predicate logic gets replaced by a dynamic regime, where meanings are viewed as relations between variable states in a model.

In the original version of the 'dynamic shift', the basic ingredients are contexts and constraints on contexts. A Kamp-style representation for a piece of text (or: discourse) looks basically like this:

context
constraints on context

The informal picture of how the information conveyed by a piece of text grows is that of ‘updating’ of representation structures:

context		new context
constraints on context	$\longrightarrow$	new constraints on context

This picture can only be made to work if we make sure that the contexts are represented smartly. Contexts are essentially sets of variables: a context just is a list of dynamically bound variables. These variables represent the antecedents which are available in any extension of that context. Embedded contexts (contexts occurring inside the constraints on a given context) and extensions of contexts (representing extensions of anaphoric possibilities) should always employ fresh variables, for if they do not, existing anaphoric possibilities get blocked off by destructive value assignment.

The rational reconstructions of dynamic discourse representation given by Barwise [3] and Groenendijk and Stokhof [11] essentially represent introduction of new antecedents by means of random assignment to a variable. The meaning of $\exists x$ becomes the relation between variable states f, g with the property that f and g differ at most in their x value:

$${}_f[\![\exists x]\!]_g \text{ iff } f[x]g.$$

This does indeed solve the problem of how to use dynamic scoping of variables to account for unbounded anaphoric linkings, but it does not give a rational reconstruction of the fact that discourse representation is supposed to work incrementally.

What one would like is illustrated by the following example, where we assume an initial representation for the sentence ‘A man entered’, which gets updated by subsequent processing of ‘A woman entered’, and next of ‘He smiled at her.’

x		x y		x y
Mx Ex	→ ‘A woman entered’ →	Mx Ex Wy Ey	→ ‘He smiled at her’ →	Mx Ex Wy Ey Sxy

In a rational reconstruction of this, one would assume that the sentences to be added to the existing representation have a representation of their own, so one would get something like:

x		y		x y
Mx Ex	+	My Ey	=	Mx Ex Wy Ey

Problem: what happens if we get a variable clash:

$$\begin{array}{|l|}\hline \mathrm{x} \\ \hline \mathrm{Mx} \\ \mathrm{Ex} \\ \\ \hline \end{array} \quad + \quad \begin{array}{|l|}\hline \mathrm{x} \\ \hline \mathrm{Mx} \\ \mathrm{Ex} \\ \\ \hline \end{array} \quad = \quad ?$$

In Kamp's original version of discourse representation theory, and also in the extended version presented in Kamp and Reyle [16], this problem does not occur, for the algorithm presented there always parses new sentences in the context of an existing representation structure, and for any indefinite noun phrase it encounters, it simply gives the instruction: 'take a fresh variable.' In other words, Kamp never *merges* representation structures, for an extension of a discourse with a new sentence always takes place in the context of a DRS.

If one is interested in constructing discourse representations in a bottom-up fashion, the picture changes. In a bottom up approach, one can still represent a context as a DRS. A new sentence S is processed by first translating S into a DRS K, and next merging K with the context DRS. But *how* should the DRSs be merged? There are various approaches to the merge problem for DRSs; see Van Eijck and Kamp [10] for an overview. These strategies amount to various ways of avoiding destructive assignments to variables, i.e., to various ways of arriving at structures which can be interpreted monotonically in terms of an information ordering on the meanings of the representation structures.

In this paper we will argue that a variable free representation of dynamic logic leads to a very natural monotonic interpretation, and thus to a natural approach to the merge problem. We get 'fresh variables' for free if we replace the dynamic variable binding mechanism of dynamic predicate logic with an indexing mechanism.

2. Dynamic Predicate Logic Without Variables

Predicate logics without variables have a long history. A key paper is Quine [20]. Based on this, Kuhn [18] and Purdy [19] have proposed variable free representations for natural language understanding. Based on an even older approach (Peirce's existential graphs), Sanchez [21] has developed a variable free natural logic. There is also a long tradition of variable free notation in lambda calculus: combinatory logic (see [2]) and De Bruijn indices [6] come to mind here. We will take our cue from this tradition.

The De Bruijn notation for lambda calculus consists of replacing variables by indices that indicate the distance to their binding lambda operator. The lambda term $\lambda x \lambda y.(\lambda z.(y(zx))(yx)$ is written in De Bruijn notation as $\lambda\lambda.(\lambda.(2\ (1\ 3))(1\ 2)$. This approach carries over to predicate logic in a straightforward fashion. Rather than carry out the program in any detail we refer to Ben-Shalom [4] (but note that the connection to De Bruijn is not mentioned there).

As an alternative to the De Bruijn style binding regime, where the binding quantifier is found by counting from the inside out, it is also possible to count from the outside in. This is similar to the way lambdas are counted in Cartesian closed category models of the lambda calculus (see e.g. Gunter [12, Ch. 3]; also Aczel [1]). Call this 'reverse De Bruijn style.'

If $X \subseteq \mathbb{N}$ and X finite, then $\sup(X)$ is defined by:

$$\begin{array}{lcl} \sup(\emptyset) & := & 0, \\ \sup(\{n\} \cup Y) & := & \max(n, \sup(Y)). \end{array}$$

Thus, $\sup(X)$ gives the maximum of X in case X is non-empty, 0 otherwise.

The language L of variable free dynamic predicate logic consists of the union $\bigcup_{n\in\mathbb{N}} L_n$, where each L_n gives the formulas that assume a *context* of size n. Each formula is a pair (n, ϕ), where n gives the size of the context. The languages L_n are defined by simultaneous recursion, as follows.

Definition 1 (Terms and formulas of L)

$$
\begin{array}{lcll}
v & ::= & 1 \mid 2 \mid \cdots & \\
L & ::= & (n, \bot) & \\
 & \mid & (n, \exists) & \\
 & \mid & (n, Pv_1 \cdots v_m) & \textit{provided } \sup\{v_1, \ldots, v_m\} \leq n \\
 & \mid & (n, v_1 \doteq v_2) & \textit{provided } \max(v_1, v_2) \leq n \\
 & \mid & (n, \neg\phi) & \textit{provided } (n, \phi) \in L \\
 & \mid & (n, \bot; \phi) & \textit{provided } (n, \phi) \in L \\
 & \mid & (n, Pv_1 \cdots v_m; \phi) & \textit{provided } (n, Pv_1 \cdots v_m) \in L \textit{ and } (n, \phi) \in L \\
 & \mid & (n, v_1 \doteq v_2; \phi) & \textit{provided } (n, v_1 \doteq v_2) \in L \textit{ and } (n, \phi) \in L \\
 & \mid & (n, \exists; \phi) & \textit{provided } (n+1, \phi) \in L \\
 & \mid & (n, \neg(\phi_1); \phi_2) & \textit{provided } (n, \phi_1) \in L \textit{ and } (n, \phi_2) \in L
\end{array}
$$

Note that we have built into the language that ; creates a flat list structure.

The language L has no individual constants. Such constants can easily be simulated by fixing part of the context, so this omission is without loss of generality.

We will omit unnecessary parentheses, writing $(3, \neg(R2\ 3))$ as $3, \neg R2\ 3$, etcetera. Occasionally, we will write $\exists; \phi$ as $\exists\phi$. Also, we abbreviate $\neg\bot$ as $\top$, and $\neg(\phi_1; \cdots; \phi_n; \neg\phi_{n+1})$ as $(\phi_1; \cdots; \phi_n) \rightarrow \phi_{n+1}$.

The static interpretation is replaced by a dynamic one. Let $\mathcal{M} = (M, I)$ be a first order model, and σ an element of M^*. We use $l(\sigma)$ for the length of $\sigma \in M^*$, and $\sigma[n]$ for the n-th element of σ. Then the term interpretation with respect to $\mathcal{M}$ and σ is given by (we use $\uparrow$ for 'undefined'):

$$
[\![n]\!]^{\mathcal{M}}_{\sigma} \quad := \quad \begin{cases} \sigma[n] & \text{if } n \leq l(\sigma), \\ \uparrow & \text{otherwise.} \end{cases}
$$

In what follows, we will often extend a stack $\sigma \in M^*$ with a single value $a \in M$. Notation for this: $\sigma\hat{\ }a$. Concatenation of two stacks $\sigma, \tau \in M^*$, in that order, is written as $\sigma\hat{\ }\tau$. If $\sigma, \tau \in M^*$ we use $\sigma \sqsubseteq \tau$ for: there is a $\theta \in M^*$ with $\sigma\hat{\ }\theta = \tau$. It is easy to see that $\sqsubseteq$ is a partial order on M^* (reflexive, transitive and anti-symmetric).

The reason why it is more convenient to use reverse De Bruyn indexing rather than regular De Bruyn style is this. The key feature of dynamic anaphora logics is the ability of the existential quantifier to bind variables outside its proper scope. Consider the DPL text $\exists x; Px; \exists y; Qy; Rxy$. Here the x and y of Rxy are bound outside of the proper scope by $\exists x$ and $\exists y$ respectively, so variables can be viewed as anaphoric elements linked to a preceding existential quantifier that introduces a referent. Similarly, in DRT, the introduction of a reference marker

$$
\begin{array}{|c|}
\hline
x \\
\hline
\\
\\
\\
\hline
\end{array}
$$

acts as an existential quantifier with dynamic scope.

The regular De Bruijn analogue of the above DPL formula would be the following (we assume that the anaphoric context is empty):

$$(0, \exists; P1; \exists; Q1; R2\ 1)$$

The index 1 in $P1$ and the index 2 in $R2\ 1$ are bound by the same quantifier (the leftmost occurrence of $\exists$). This illustrates that anaphoric coreference (or: dynamic binding) is no longer encoded by use of the same index, but the antecedent of an index has to be worked out by taking the 'existential depth' of the intervening formula into account.

The awkwardness in antecedent recovery can be avoided by using reverse De Bruijn indexing. The reverse De Bruyn analogue of the example $\exists x; Px; \exists y; Qy; Rxy$ looks like this:

$$(0, \exists; P1; \exists; Q2; R1\ 2)$$

All occurrences of 1 are bound by the same quantifier (the leftmost occurrence of $\exists$), and similarly, all occurrences of 2 are bound by the rightmost occurrence of $\exists$.

The semantic definition of satisfaction, for dynamic logic without quantifiers under the reverse De Bruyn indexing scheme, runs as follows (A ranges over $\perp$, $Pv_1 \cdots v_n$, $v_1 \doteq v_2$, $\neg(\phi)$):

Definition 2 (Satisfaction for L)

$$\begin{array}{rll}
{}_\sigma[\![n, \perp]\!]_\tau^{\mathcal{M}} & & \textit{never}, \\
{}_\sigma[\![n, \exists]\!]_\tau^{\mathcal{M}} & \textit{iff} & l(\sigma) = n \textit{ and } \tau = \sigma\hat{\ }a \textit{ for some } a \in M, \\
{}_\sigma[\![n, Pv_1 \cdots v_m]\!]_\tau^{\mathcal{M}} & \textit{iff} & l(\sigma) = n, \sigma = \tau \textit{ and } \langle [\![v_1]\!]_\sigma^{\mathcal{M}} \ldots, [\![v_m]\!]_\sigma^{\mathcal{M}} \rangle \in I(P), \\
{}_\sigma[\![n, v_1 \doteq v_2]\!]_\tau^{\mathcal{M}} & \textit{iff} & l(\sigma) = n, \sigma = \tau \textit{ and } [\![v_1]\!]_\sigma^{\mathcal{M}} = [\![v_2]\!]_\sigma^{\mathcal{M}}, \\
{}_\sigma[\![n, \neg\phi]\!]_\tau^{\mathcal{M}} & \textit{iff} & l(\sigma) = n, \sigma = \tau \textit{ and } \textit{ there is no } \theta \in M^* \textit{ with } {}_\sigma[\![n, \phi]\!]_\theta^{\mathcal{M}}, \\
{}_\sigma[\![n, \exists; \phi]\!]_\tau^{\mathcal{M}} & \textit{iff} & {}_\sigma[\![n, \exists]\!]_\theta^{\mathcal{M}} \textit{ and } {}_\theta[\![\phi]\!]_\tau^{\mathcal{M}} \textit{ for some } \theta \in M^*, \\
{}_\sigma[\![n, A; \phi]\!]_\tau^{\mathcal{M}} & \textit{iff} & {}_\sigma[\![n, A]\!]_\sigma^{\mathcal{M}} \textit{ and } {}_\sigma[\![n, \phi]\!]_\tau^{\mathcal{M}}.
\end{array}$$

Note that in the semantic clauses for $(n, Pv_1 \cdots v_m)$ and $(n, v_1 \doteq v_2)$ the proviso $l(\sigma) = n$ guarantees that the term functions $[\![v_i]\!]_\sigma^{\mathcal{M}}$ are well defined.

The definition of the semantics for L is in fact a straightforward adaptation of the dynamic semantics for predicate logic defined in Groenendijk and Stokhof [11], which is in turn closely related to a proposal made by Barwise in [3]. However, this semantics is *not* equivalent to the semantics given by Groenendijk and Stokhof, but has an important advantage over it. In Groenendijk and Stokhof's semantics for DPL, a repeated assignment to a single variable by means of a repeated use of the same existential quantifier-variable combination blocks off the individual introduced by the first use of the quantifier from further anaphoric reference. After $\exists x Px; \exists x Qx$, the variable x will refer to the individual introduced by $\exists x Qx$, and the individual introduced by $\exists x Px$ has become inaccessible.

In the sequence semantics proposed by Vermeulen [23] this problem is solved by making every variable refer to a stack, and interpreting an existential quantification for variable x as a push operation on the x stack. The quantification $\exists x$ now gets a counterpart xE, interpreted as a pop of the x stack. In our reverse De Bruyn semantics for L we use a single finite stack, and we do not allow pops. This ensures that existential quantification is non-destructive, in other words that our semantics is *incremental*. The push stack operations are replaced by a single push operation (the interpretation of the existential quantifier). Note that quantifications never can destroy previous dynamic assignments in the same formula, nor can they overwrite initially given values. Indeed, the definition of the semantics for L ensures that positions $1, \ldots, k$ of an L-formula (k, ϕ) are not affected by the stack dynamics of the existential quantifier. The values of these positions are read from the input state; these are the anaphoric references picked up from the surrounding context. Positions higher up on the 'stack' get their value from an existential quantifier action inside ϕ. This is made formal in the following proposition.

Lemma 3 (Incrementality) *If* ${}_\sigma[\![n,\phi]\!]^{\mathcal{M}}_\tau$ *then* $\sigma \sqsubseteq \tau$.

Proof. Induction on the structure of ϕ, with induction hypothesis of the form *for all* $n \in \mathbb{N}$, *…* . □

The language L is designed for the translation of open texts: texts that may contain occurrences of pronouns which take their reference from the surrounding context. After all, the process of picking up antecedents from context is the essence of anaphoric linking.

For a translation from L to standard DPL we need a term translation $^\circ$ and a formula translation $^\bullet$. The term translation is given by $n^\circ := x_n$. The formula translation (again, we let A range over $\perp$, $Pv_1 \cdots v_n$, $v_1 \doteq v_2$, $\neg(\phi)$):

$$
\begin{array}{rcl}
(n,\perp)^\bullet & := & \perp \\
(n,\exists)^\bullet & := & \exists x_{n+1} \\
(n, Pv_1 \cdots v_m)^\bullet & := & Pv_1^\circ \cdots v_m^\circ \\
(n, v_1 \doteq v_2)^\bullet & := & v_1^\circ \doteq v_2^\circ \\
(n, \neg(\phi))^\bullet & := & \neg(n,\phi)^\bullet \\
(n, \exists; \phi)^\bullet & := & \exists x_{n+1}; (n+1,\phi)^\bullet \\
(n, A; \phi)^\bullet & := & (n,A)^\bullet; (n,\phi)^\bullet .
\end{array}
$$

E.g., L-formula $(2, \exists; R(1,3); S(2,3))$ gets translated by $^\bullet$ into the DPL formula $\exists x_3; Rx_1x_3; Sx_2x_3$.

Note that the DPL translations of L formulas are rather special, for they will contain no destructive assignments (all quantifications are over 'fresh' variables).

To show that the translation function is correct in the sense that it preserves satisfaction of formulas, assume a DPL language over a set of variables $V = \{x_i \mid i \in \mathbb{N}^+\}$. Then a DPL state over a model $\mathcal{M} = (M, I)$ is a member of M^V. We use $\mathcal{M}, s, s' \models_{dpl} \phi$ for: the state pair s, s' satisfies the DPL formula ϕ in model $\mathcal{M}$.

If s is a DPL state over $\mathcal{M}$, and $\sigma \in M^*$ we define the state s_σ as follows.

$$
s_\sigma(x_i) \quad := \quad \begin{cases} \sigma[i] & \text{if } i \leq l(\sigma), \\ s(x_i) & \text{otherwise.} \end{cases}
$$

Proposition 4 *For all* $(n,\phi) \in L$*, all models* $\mathcal{M}$*, all* $\sigma, \tau \in M^*$*, all* $s \in M^V$*:*

$${}_\sigma[\![(n,\phi)]\!]^{\mathcal{M}}_\tau \textit{ iff } \mathcal{M}, s_\sigma, s_\tau \models_{dpl} (n,\phi)^\bullet .$$

Proof. Induction on the structure of ϕ. Here is the existential quantifier case.

${}_\sigma[\![(n,\exists)]\!]^{\mathcal{M}}_\tau$ iff $l(\sigma) = n$ and for some $a \in M$, $\sigma\hat{\ }a = \tau$ iff for all $s \in M^V$, s_σ and s_τ differ at most in the value for x_{n+1} iff $\mathcal{M}, s_\sigma, s_\tau \models_{dpl} \exists x_{n+1}$ iff $\mathcal{M}, s_\sigma, s_\tau \models_{dpl} (n,\exists)^\bullet$. □

To illustrate the considerable expressive power of dynamic logic without variables, here is a translation function from FOL to L. We assume a set of first order variables $V = \{x_i \mid i \in \mathbb{N}^+\}$. If f is an assignment to V in some domain M and $\sigma \in M^k$, f_σ is defined in the obvious way, by putting $f_\sigma(x_i) := \sigma[i]$ for $i \leq k$, $f_\sigma(x_i) := f(x_i)$ for $i > k$. For term translations, let $^\ominus$ be the inverse of $^\circ$.

To translate a first order formula ϕ, let $k := \sup\{i \mid x_i \in FV(\phi)\}$. Then the L translation of ϕ is

(k, ϕ^k), where the translation functions ^{k}, for $k \in \mathbb{N}$, are defined as follows:

$$\begin{array}{rcl}
(\bot)^k & := & \bot \\
(Pv_1 \cdots v_m)^k & := & Pv_1^{\ominus} \cdots v_m^{\ominus} \\
(\neg\phi)^k & := & \neg(\phi)^k \\
(\phi_1 \wedge \phi_2)^k & := & \neg\neg(\phi_1)^k; (\phi_2)^k \\
(\exists x_i \phi)^k & := & \exists; ([x_{k+1}/x_i]\phi)^{k+1}.
\end{array}$$

The substitution $[x_{k+1}/x_i]\phi$ is subject to the usual condition that x_{k+1} should be free for x_i in ϕ. If necessary, replace ϕ by an alphabetic variant that meets the condition.

Proposition 5 *Let ϕ be a FOL formula, and let $k := \sup\{i \mid x_i \in FV(\phi)\}$.*

For all models $\mathcal{M} = (M, I)$, all stacks $\sigma \in M^k$, all variable assignments f:

$\mathcal{M}, f_\sigma \models \phi$ iff there is a τ with ${}_\sigma[\![k, \phi^k]\!]_\tau^{\mathcal{M}}$.

Proof. Induction on the structure of ϕ. The induction hypothesis of the form: suppose the property holds for all subformulas ψ of ϕ, for all m with $k \leq m \leq k + d(\phi)$, where $d(\phi)$ measures the quantifier depth of ϕ. □

Combining the well-known translation from DPL to FOL with the above translation, we get in two steps a translation from DPL to L. Still, it is instructive to define a direct translation. The fixed occurrences of variables in a DPL formula are the variable occurrences that are neither classically bound nor in the dynamic scope of an existential quantifier. Let ϕ be a DPL formula, and let $k := \sup\{i \mid x_i$ has a fixed occurrence in $\phi\}$. Then ϕ translates into $(k, \phi^{(k)})$, with ${}^{(k)}$ given by:

$$\begin{array}{rcl}
(\bot)^{(k)} & := & \bot \\
(\exists x_i)^{(k)} & := & \exists \\
(Pv_1 \cdots v_m)^{(k)} & := & Pv_1^{\ominus} \cdots v_m^{\ominus} \\
(v_1 \doteq v_2)^{(k)} & := & v_1^{\ominus} \doteq v_2^{\ominus} \\
(\neg\phi)^{(k)} & := & \neg(\phi)^{(k)} \\
((\phi_1; \phi_2); \phi_3)^{(k)} & := & (\phi_1; (\phi_2; \phi_3))^{(k)} \\
(A; \phi)^{(k)} & := & A^{(k)}; \phi^{(k)} \\
(\exists x_i; \phi)^{(k)} & := & \exists; ([x_{k+1}/x_i]\phi)^{(k+1)}.
\end{array}$$

Here $[x_{k+1}/x_i]\phi$ denotes dynamic substitution, i.e., substitution of x_{k+1} for all occurrences of x_i that are neither classically or dynamically bound, while taking care, through appropriate switches to alphabetic variants, that the replacing occurrences of x_{k+1} are dynamically free in the result (i.e., are not in the dynamic scope of an existential quantifier).

The following proposition can be proved by induction on the structure of ϕ:

Proposition 6 *For all models $\mathcal{M} = (M, I)$, all $\sigma \in M^k$, all $s \in V \to M$, where $V = \{x_i \mid i \in \mathbb{N}^+\}$, the following holds:*

$$\exists t \in V \to M : \ \mathcal{M}, s_\sigma, t \models_{dpl} \phi \iff \exists \tau \in M^* : \ {}_\sigma[\![k, \phi^{(k)}]\!]_\tau^{\mathcal{M}}.$$

3. Variable Free Dynamic Logic and Discourse Representation

Next, we want to show that L formulas correspond exactly to (canonical forms of) Discourse Representation Structures in the sense of Kamp [15] (so-called *pure* DRSs). DRSs are defined by the following mutual recursion. Again, we assume for simplicity that there are no individual constants (and again, nothing hinges on this).

$$
\begin{array}{rcl}
v & ::= & x_1 \mid x_2 \mid \cdots \\
C & ::= & Pv_1 \cdots v_n \mid v_1 \doteq v_2 \mid \neg D \\
D & ::= & (\{v_1, \ldots, v_n\}, \{C_1, \ldots, C_m\})
\end{array}
$$

If $D = (\{v_1, \ldots, v_n\}, \{C_1, \ldots, C_m\})$ and D' are DRSs, then $D \Rightarrow D'$ abbreviates the condition

$$\neg(\{v_1, \ldots, v_n\}, \{C_1, \ldots, C_m, \neg D'\}).$$

We give a translation function $^\oplus$ from L formulas to DRSs, as follows (using $(n, \phi)^\oplus_0$ and $(n, \phi)^\oplus_1$ for the first and second components of $(n, \phi)^\oplus$):

$$
\begin{array}{rcl}
(n, \bot)^\oplus & := & (\emptyset, \{\bot\}) \\
(n, \exists)^\oplus & := & (\{x_{n+1}\}, \emptyset) \\
(n, Pv_1 \cdots v_m)^\oplus & := & (\emptyset, \{Pv_1^\circ \cdots v_m^\circ\}) \\
(n, v_1 \doteq v_2)^\oplus & := & (\emptyset, \{v_1^\circ \doteq v_2^\circ\}) \\
(n, \neg(\phi))^\oplus & := & (\emptyset, \{\neg(n, \phi)^\oplus\}) \\
(n, \bot; \phi)^\oplus & := & (\emptyset, \{\bot\}) \\
(n, \exists; \phi)^\oplus & := & (\{x_{n+1}\} \cup (n+1, \phi)^\oplus_0, (n+1, \phi)^\oplus_1) \\
(n, Pv_1 \cdots v_m; \phi)^\oplus & := & ((n, \phi)^\oplus_0, \{Pv_1^\circ \cdots v_m^\circ\} \cup (n, \phi)^\oplus_1) \\
(n, v_1 \doteq v_2; \phi)^\oplus & := & ((n, \phi)^\oplus_0, \{v_1^\circ \doteq v_2^\circ\} \cup (n, \phi)^\oplus_1) \\
(n, \neg(\phi); \psi)^\oplus & := & ((n, \psi)^\oplus_0, \{\neg(n, \phi)^\oplus\} \cup (n, \psi)^\oplus_1).
\end{array}
$$

Some examples:

$$
(2, R1\ 2; \exists; R1\ 3)^\oplus \quad = \quad
\begin{array}{|c|}
\hline
x_3 \\
\hline
Rx_1x_2 \\
Rx_1x_3 \\
\hline
\end{array}
$$

$$
(2, R1\ 2; \neg(\exists; R1\ 3))^\oplus \quad = \quad
\begin{array}{|l|}
\hline
\\
\hline
Rx_1x_2 \\
\neg \begin{array}{|c|}
\hline
x_3 \\
\hline
Rx_1x_3 \\
\hline
\end{array} \\
\hline
\end{array}
$$

We will now show that the translation is adequate. Assume a model $\mathcal{M} = (M, I)$. An embedding function in the sense of DRT is a function from a finite subset of the set of variables $\{x_i \mid i \in \mathbb{N}^+\}$ to M. We use $\sigma^\oplus$ for the function in $\{x_1, \ldots x_{l(\sigma)}\} \to M$ that corresponds to stack $\sigma \in M^*$, in the obvious sense (namely, by setting $\sigma^\oplus(x_i) := \sigma[i]$).

Proposition 7 *If $(n, \phi) \in L$ then:*

1. $(n, \phi)^{\oplus}$ *is a DRS.*
2. ${}_{\sigma}[\![n, \phi]\!]_{\tau}^{\mathcal{M}}$ *iff* $\tau^{\oplus}$ *verifies* $(n, \phi)^{\oplus}$ *in* $\mathcal{M}$ *with respect to* $\sigma^{\oplus}$ *(in the sense of DRT).*

Proof. Both claims are proved by induction on the structure of ϕ. □

The DRS translations have the additional property that they yield *pure* DRSs: If K' is a sub-DRS of K then their sets of introduced markers will be disjoint.

A special case is the case of L formulas of the form $(0, \phi)$. These correspond precisely to so-called *proper* pure DRSs, i.e., pure DRSs without 'fixed' variable occurrences.

a

To capture the precise meaning of 'fixed' variables in a DRS, we need to distinguish three kinds of variable occurrences in a DRS: (1) fixed by the larger context, (2) fixed in the current context, and (3) fixed in a subordinate context. Here are the definitions of these sets.

Definition 8 (fix, intro, cbnd)

- $\mathit{fix}(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\}) := \bigcup_i \mathit{fix}(C_i) - \{v_1, \dots, v_n\}$.
- $\mathit{intro}(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\}) := \{v_1, \dots, v_n\}$.
- $\mathit{cbnd}(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\}) := \bigcup_i \mathit{cbnd}(C_i)$.
- $\mathit{fix}(Pv_1 \cdots v_n) := \{v_1, \dots v_n\}$, $\mathit{intro}(Pv_1 \cdots v_n) := \emptyset$, $\mathit{cbnd}(Pv_1 \cdots v_n) := \emptyset$.
- $\mathit{fix}(v_1 \doteq v_2) := \{v_1, v_2\}$, $\mathit{intro}(v_1 \doteq v_2) := \emptyset$, $\mathit{cbnd}(v_1 \doteq v_2) := \emptyset$.
- $\mathit{fix}(\neg D) := \mathit{fix}(D)$, $\mathit{intro}(\neg D) := \emptyset$, $\mathit{cbnd}(\neg D) := \mathit{intro}(D) \cup \mathit{cbnd}(D)$.

We will now define a translation from DRSs to L formulas, using a technique similar to the mapping of FOL to L. Let D be a DRS, and let k be $\sup\{i | x_i \in \mathit{fix}(D)\}$. Then the L-translation of D is the formula $(k, D^{[k]})$, where the functions ${}^{[k]}$ are given by:

$$
\begin{array}{rcl}
(\emptyset, \{C_1, \dots, C_m\})^{[k]} & := & C_1^{[k]}; \dots ; C_m^{[k]} \\
(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\})^{[k]} & := & \exists; (\{v_2, \dots, v_n\}, \{[x_{k+1}/v_1]C_1, \dots, [x_{k+1}/v_1]C_m\})^{[k+1]} \\
(\bot)^{[k]} & := & \bot \\
(Pv_1 \cdots v_m)^{[k]} & := & Pv_1^{\ominus} \cdots v_m^{\ominus} \\
(v_1 \doteq v_2)^{[k]} & := & v_1^{\ominus} \doteq v_2^{\ominus} \\
(\neg\phi)^{[k]} & := & \neg(\phi)^{[k]}.
\end{array}
$$

Note that there is an element of indeterminism in this translation instruction, for $\{v_1, \dots, v_n\}$ is a set, and the translation recipe instructs us to take its elements one by one. If the reader does not like this, she can use the order on $\{v_1, \dots, v_n\}$ imposed by the indices of the variables to always pick the smallest element from this set.

Again, we have to ensure that x_{k+1} is free for v_i in $[x_{k+1}/v_i]C$, i.e., that x_{k+1} does not have contextually bound occurences in C. If this condition is not met, we have to replace C by an alphabetic variant first. For all purposes, contextually bound variables in DRT behave exactly like bound variables in FOL. Note that x_{k+1} cannot have fixed occurrences in C, by an inductive argument based on the fact that the initial choice of index is the highest index of the initial set of fixed occurrences, and that fixed occurrences of variables that are not in the initial set would have caused a variable clash at the level where they got introduced.

For the next proposition, we have to relate embedding functions to stacks of elements of a domain. If f is a function in $\{x_i \mid 1 \leq i \leq k\} \to M$ for some $k \in \mathbb{N}$, then $f^* \in M^k$ is given by $f^*[i] := f(x_i)$.

Proposition 9 *Let D be a DRS, and let k be $\sup\{i \mid x_i \in \mathit{fix}(D)\}$. Then the following hold:*

1. $(k, D^k) \in L$,
2. *For all models $\mathcal{M} = (M, I)$, all functions $f : \mathit{fix}(D) \to M$: there is a $g : \mathit{fix}(D) \cup \mathit{intro}(D) \to M$ such that g verifies D with respect to f in $\mathcal{M}$ (in the sense of DRT) iff there is a $\tau \in M^*$ with ${}_{f^*}[\![k, D^k]\!]^{\mathcal{M}}_{\tau}$.*

Proof. Both claims are proved by induction on the structure of D. □

In this section we have shown that L and DRT have exactly the same expressive power. Moreover, L formulas are isomorphic to DRSs in canonical form, in the following sense. L formulas correspond to pure DRSs, and L formulas of the form $(0, \phi)$ correspond to proper pure DRSs. The advantage of L over DRT will reveal itself when we are going to define logical consequence for L, in Section 5.

4. Merging Formulas and Merging Representation Structures

For the following we need the notion of the 'existential depth' of a formula. Intuitively, the existential depth of (n, ϕ) calculates the number of positions by which the stack grows during the semantic processing of ϕ. E.g., the existential depth of $(n, \exists)$ is 1, for any n.

If $(n, \phi) \in L$, the existential depth of ϕ is given by (A ranges over $\perp$, $Pv_1 \cdots v_n$, $v_1 \doteq v_2$, $\neg(\phi)$):

$$\begin{array}{lcl} e(\exists) & := & 1 \\ e(A) & := & 0 \\ e(\exists; \phi) & := & 1 + e(\phi) \\ e(A; \phi) & := & e(\phi). \end{array}$$

Suppose we want to 'merge' two formulas (n, ϕ) and (m, ψ) in left-to-right order, in such a way that the output of (n, ϕ) serves as input to (m, ψ). One could introduce a merge operation $\bullet$ as a partial operation on L formulas, as follows:

$$(n, \phi) \bullet (m, \psi) := \begin{cases} (n, \phi; \psi) & \text{if } m = n + e(\phi), \\ \uparrow & \text{otherwise.} \end{cases}$$

In case the result of merging (n, ϕ) and (m, ψ) is undefined all is not lost, however. The undefinedness may be due to the fact that the context is too large ($n + e(\phi) > m$) or to the fact that the context is too small ($n + e(\phi) < m$). In the former case, the problem can be remedied by performing a 'write memory shift operation' on (m, ψ), as follows:

$$\frac{(m, \psi)}{(m + k, [{}^{m}_{+k}]\psi)}$$

Here, $[{}^{m}_{+k}]\psi$ is the index substitution which replaces every $i > m$ by $i + k$.

Proposition 10 ${}_{\sigma}[\![m,\psi]\!]^{\mathcal{M}}_{\sigma\hat{}\tau}$ *iff for all* $\theta \in M^k$*:* ${}_{\sigma\hat{}\theta}[\![m+k,[^{\,m}_{+k}]\psi]\!]^{\mathcal{M}}_{\sigma\hat{}\theta\hat{}\tau}$.

The other case where the result of merging (n,ϕ) and (m,ψ), in that order, is undefined, is the case where the context is too small $(n+e(\phi)<m)$. In this case we can use 'existential padding'. A useful abbreviation for this is $\exists^k$, defined recursively by:

$$\begin{array}{rcl} \exists^0 & := & \top \\ \exists^{k+1} & := & \exists;\exists^k \end{array}$$

Existential padding is applied as follows:

$$\frac{(m+k,\psi)}{(m,\exists^k;\psi)}$$

Proposition 11 ${}_{\sigma}[\![m,\exists^k;\psi]\!]^{\mathcal{M}}_{\tau}$ *iff* ${}_{\sigma\hat{}\tau[m+1..m+k]}[\![m+k,\psi]\!]^{\mathcal{M}}_{\tau}$.

The rules for memory shift and existential padding are built into the calculus of Section 6.

As Propositions 7 and 9 have shown us, the variable free dynamic logic L can be viewed as a rational reconstruction of DRT (in a way that DPL *cannot* be viewed as such). In fact, the reconstruction has made us sensitive to a distinction which often remains implicit in DRT: the distinction between representation structures which contain reference markers not introduced in the structure itself but imported from a pre-existing representation on one hand and representation structures which do not on the other (no reference markers are imported from outside; every marker gets introduced in the structure itself).

The variable constraint imposed in DRT (always take fresh variables when extending a DRT structure) avoids the destructive assignment problem from DPL, but the penalty imposed for this is a top-down DRS construction algorithm. A bottom up perspective on the semantics reveals itself via the link with the variable free notation.

Several possible solutions to the merge problem for DRT are discussed in Van Eijck and Kamp [10]. If one wants merge to be a total operation on DRSs, the merge of DRSs D and D', in that order, may involve substitution of the introduced variables of D'. The present variable free perspective on dynamic logic suggests a particular choice for the merge operation. The DRS translations of L-formulas have the following general form (this is the general form of a pure DRS):

$$\begin{array}{|c|} \hline x_{m+1},\dots,x_n \\ \hline \\ C_1 \\ \vdots \\ C_k \\ \\ \\ \hline \end{array}$$

Here it is assumed that all the markers occurring in $C_1,\dots,C_k$ are among $x_1,\dots,x_n$. The markers $x_1,\dots,x_m$ are the fixed markers of the DRS, the markers $x_{m+1}\cdots x_n$ the introduced reference markers.

Assuming that DRSs are all in this canonical form, we can merge them as follows, using substitution to avoid variable clashes:

$$
\begin{array}{|c|}\hline x_{p+1},\ldots,x_{p+m} \\ \hline C_1 \\ \vdots \\ C_k \\ \\ \hline \end{array}
\quad \bullet_{[x_{p+m+1}/x_{q+1},\ldots,x_{p+m+n}/x_{q+n}]} \quad
\begin{array}{|c|}\hline x_{q+1},\ldots,x_{q+n} \\ \hline C_1' \\ \vdots \\ C_r' \\ \\ \hline \end{array}
$$

$$
\Longrightarrow \quad
\begin{array}{|c|}\hline x_{p+1},\ldots,x_{p+m+n} \\ \hline C_1 \\ \vdots \\ C_k \\ C_1'[x_{p+m+1}/x_{q+1},\ldots,x_{p+m+n}/x_{q+n}] \\ \vdots \\ C_r'[x_{p+m+1}/x_{q+1},\ldots,x_{p+m+n}/x_{q+n}] \\ \\ \hline \end{array}
$$

Example:

$$
\begin{array}{|c|}\hline x_2,x_3 \\ \hline Rx_1x_2 \\ Sx_2x_3 \\ \\ \hline \end{array}
\quad \bullet_{[x_4/x_3,x_5/x_4]} \quad
\begin{array}{|c|}\hline x_3,x_4 \\ \hline Tx_1x_4 \\ Vx_3x_4 \\ \\ \hline \end{array}
\quad \Longrightarrow \quad
\begin{array}{|c|}\hline x_2,x_3,x_4,x_5 \\ \hline Rx_1x_2 \\ Sx_2x_3 \\ Tx_1x_5 \\ Vx_4x_5 \\ \hline \end{array}
$$

This example corresponds to the merge of $(1,\exists;\exists;R(1,2);S(2,3))$ and $(2,\exists;\exists;T(1,4),V(3,4))$, in that order, after memory shift right of the second formula over one position, to get $(3,\exists;\exists;T(1,5),V(4,5))$, with end result:

$$(1,\exists;\exists;R(1,2);S(2,3);\exists;\exists;T(1,5),V(4,5).$$

The switch rules of the calculus of Section 6 permit to transform this in turn into:

$$(1,\exists;\exists;\exists;\exists;R(1,2);S(2,3);T(1,5);V(4,5),$$

which again corresponds to (a canonical, i.e., pure, representation of) the result DRS.

5. A Transitive Notion of Dynamic Consequence

A piece of text containing anaphoric references can either be self-contained, in case all anaphors find their antecedent in the text itself, or it can be linked to a context, in case some pronouns refer to an antecedent outside the text itself, e.g. an antecedent mentioned in previous discourse, or introduced by another speaker, or introduced by an act of pointing, and so on. We can say that texts of the latter kind have an *anaphoric presupposition*. In order to establish the truth conditions of such a text one needs access to the context that provides antecedents for the outward pointing anaphors, and in that sense the anaphoric context is presupposed.

Still, it is clear that we can make (minimal) sense of a piece of text containing unresolved anaphors, even without access to the context. We can *abstract* from the context, in the usual way, by viewing the meaning of a piece of text with anaphoric presupposition as a *function* from contexts to denotations. The full information content of the text reveals itself once the anaphoric context is plugged in. As long as the context is unknown, the anaphors with an outside link have the weakest possible information content: they carry the same information as a wide scope existential quantifier.

The modelling of anaphoric presupposition as existentially-quantified-over context, with this context in turn treated as a piece of 'read-only memory', suggests a very natural consequence notion for 'reasoning under anaphoric presupposition'.

The anaphoric presupposition of a formula (n, ϕ) is given by its 'offset' n, for the number of anaphoric elements that need (possibly) different outside referents. It should be noted, though, that not every index i in $\{1, \ldots, n\}$ need occur in ϕ. We can now say that (n, ϕ) entails (m, ψ) iff for all models, the interpretation of (n, ϕ) is 'more informative' than that of (m, ψ). The formula (n, ϕ) will export $n + e(\phi)$ anaphoric elements, for $e(\phi)$ measures the number of new referents that are introduced by ϕ. In order to ensure that all of these can be absorbed by (m, ψ), we have to assume that $n + e(\phi) \leq m$. Making the assumption $n + e(\phi) \leq m$ into a presupposition boils down to the statement that if $n + e(\phi) > m$ then it is vacuously *false* that (m, ψ) follows from (n, ϕ). These considerations lead to the following formal definition of logical consequence for L:

Definition 12 (L Consequence)

1. $(n, \phi) \models (m, \psi) \;:\Longleftrightarrow\; n + e(\phi) \leq m$ *and for all* $\mathcal{M}, \sigma, \tau$*: if* ${}_\sigma[\![n, \phi]\!]^{\mathcal{M}}_\tau$ *then there are* θ, ρ *with* $\tau \sqsubseteq \theta$ *and* ${}_\theta[\![m, \psi]\!]^{\mathcal{M}}_\rho$.
2. $(n, \phi) \,=\!\!|\, (m, \psi) \;:\Longleftrightarrow\; n + e(\phi) \leq m$ *and there are* $\mathcal{M}, \sigma, \tau$ *with* ${}_\sigma[\![n, \phi]\!]^{\mathcal{M}}_\tau$ *such that for no* θ *with* $\tau \sqsubseteq \theta$ *is there a* ρ *with* ${}_\theta[\![m, \psi]\!]^{\mathcal{M}}_\rho$.

This consequence relation is truly dynamic in that it allows carrying anaphoric links from premiss to conclusion. For example: from 'a man walks and he talks' it follows that 'he talks':

$$(0, \exists; M1; W1; T1) \models (1, T1).$$

The following lemma shows that L consequence has a very desirable property.

Lemma 13 (Transitivity) *For all* $(n, \phi), (m, \psi), (k, \chi) \in L$*:*

If $(n, \phi) \models (m, \psi)$ *and* $(m, \psi) \models (k, \chi)$ *then* $(n, \phi) \models (k, \chi)$.

Proof. Suppose $(n, \phi) \models (m, \psi)$ and $(m, \psi) \models (k, \chi)$, and assume ${}_\sigma[\![n, \phi]\!]^{\mathcal{M}}_\tau$. We have to show that there are θ and ρ with $\tau \sqsubseteq \theta$ and ${}_\theta[\![k, \chi]\!]^{\mathcal{M}}_\rho$.

By $(n, \phi) \models (m, \psi)$ and the assumption there are $\tau' \sqsupseteq \tau$ and θ with ${}_{\tau'}[\![m, \psi]\!]^{\mathcal{M}}_\theta$. From this and $(m, \psi) \models (k, \chi)$ we get $\theta' \sqsupseteq \theta$ and ρ with ${}_{\theta'}[\![k, \chi]\!]^{\mathcal{M}}_\rho$. By incrementality and by transitivity of $\sqsubseteq$, we get $\tau \sqsubseteq \theta'$ and we are done. □

One of the problems with the dynamic consequence relation of DPL [11] is the fact that it is not transitive, as witnessed by Van Benthem's example:

Suppose a man owns a house. Then he owns a garden.
Suppose a man owns a garden. Then he sprinkles it.
BUT NOT: Suppose a man owns a house. Then he sprinkles it.

The example makes a point, not about this piece of natural language reasoning, but about the DPL rendering of it:

$$\begin{array}{l} \exists x; Mx; \exists y; Hy; Oxy \models_{dpl} \exists z; Gz; Oxz. \\ \underline{\exists z; Gz; Oxz \models_{dpl} Sxz.} \\ \exists x; Mx; \exists y; Hy; Oxy \not\models_{dpl} Sxz. \end{array}$$

The point of the example is that it shows that the DPL consequence relation $\models_{dpl}$ is not transitive. Here $\models_{dpl}$ is defined by: $\phi \models_{dpl} \psi$ iff for all $\mathcal{M}, \sigma, \sigma'$: if $\mathcal{M}, \sigma, \sigma' \models_{dpl} \phi$ then there is some σ'' with $\mathcal{M}, \sigma', \sigma'' \models_{dpl} \psi$.

The example can also be brought to bear on DRT, where it serves to illustrate that DRT consequence is not transitive either:

$$\frac{\begin{array}{|c|}\hline x, y \\ \hline Mx \\ Hy \\ Oxy \\ \hline \end{array} \models_{drt} \begin{array}{|c|}\hline z \\ \hline Gz \\ Oxz \\ \\ \hline \end{array} \qquad \begin{array}{|c|}\hline z \\ \hline Gz \\ Oxz \\ \\ \hline \end{array} \models_{drt} \begin{array}{|c|}\hline \\ \hline Sxz \\ \\ \\ \hline \end{array}}{\begin{array}{|c|}\hline x, y \\ \hline Mx \\ Hy \\ Oxy \\ \hline \end{array} \not\models_{drt} \begin{array}{|c|}\hline \\ \hline Sxz \\ \\ \\ \hline \end{array}}$$

If we use L-formulas, the situation looks better. The translation of Van Benthem's example now runs:

$$\frac{(0, \exists; M1; \exists; H2; O1\ 2) \models (2, \exists; G3; O1\ 3) \quad (2, \exists; G3; O1\ 3) \models (3, S1\ 3)}{(0, \exists; M1; \exists; H2; O1\ 2) \models (3, S1\ 3)}$$

This is indeed a valid argument, for the consequence notion of L is indeed transitive.

Note that in the definition of valid consequence for L *existential padding* is used to provide an antecedent for the index 3. The conclusion should be read as:

> Suppose a man owns a house. Then there is a thing which he sprinkles.

This is of course the conclusion one would expect.

The following proposition shows that we can always choose to make existential padding explicit:

Proposition 14 *For all $(n, \phi), (m, \psi) \in L$ with $n + e(\phi) \leq m$:*

$(n, \phi) \models (m, \psi)$ iff $(n, \phi) \models (n + e(\phi), \exists^k; \psi)$, where $k = m - (n + e(\phi))$.

Proof. Induction on k. $\square$

6. A Calculus for Incremental Dynamic Reasoning

In this section, we will give a set of sequent deduction rules for incremental dynamic reasoning. We postpone the treatment of equality to Section 9.

We will write sequents as $(n, \phi) \Longrightarrow (m, \psi)$, where $\Longrightarrow$ is the sequent separator. Note that $(n, \phi) \Longrightarrow (m, \bot)$, for any $m \geq n + e(\phi)$, expresses that (n, ϕ) is inconsistent.

In the calculus we are about to present, we need some further notation for substitutions, in addition to $[{}^{m}_{+k}]$. Recall that $[{}^{m}_{+k}]$ is the substitution that replaces every index $n > m$ by $n + k$. This is useful to create room for k new indices starting from position $m+1$. What we also need is an operation that removes a gap after the substitution of a referent for an $\exists$ that binds position m. The operation for this is $[{}^{m}_{-1}]$; this replaces every index $n > m$ by $n - 1$. Finally, $[k/m]$ has the usual meaning: replace index m by k everywhere. We abbreviate $[{}^{m}_{-1}][k/m]$ as $[k/m]^{-}$ ('substitute k for m and close the gap'). And that's all we need.

In the calculus, we use C, with and without subscripts, as a variable over contexts (formula lists composed with ;, including the empty list). We extend the function e to contexts by stipulating that $e(C) = 0$ if C is the empty list. Substitution is extended to contexts in a similar way. In the rules below we will use T as an abbreviation of formulas ϕ with $e(\phi) = 0$ (T for *Test* formula).

Structural Rules
Test Axiom

$$\frac{}{(n, T) \Longrightarrow (n, T)}\ (n, T) \in L$$

Soundness of Test Axiom If ${}_{\sigma}[\![n, T]\!]^{\mathcal{M}}_{\tau}$ then $\sigma = \tau$ (because T is a test) and therefore ${}_{\tau}[\![n, T]\!]^{\mathcal{M}}_{\tau}$. Thus, $(n, T) \models (n, T)$.

Transitivity Rule

$$\frac{(n, \phi) \Longrightarrow (m, \psi) \qquad (m, \psi) \Longrightarrow (k, \chi)}{(n, \phi) \Longrightarrow (k, \chi)}$$

Soundness of Transitivity Rule This was established in Lemma 13.

Test Contraction Rules

$$\frac{(n, C_1T; TC_2) \Longrightarrow (m, \phi)}{(n, C_1TC_2) \Longrightarrow (m, \phi)} \qquad \frac{(n, \phi) \Longrightarrow (m, C_1T; TC_2)}{(n, \phi) \Longrightarrow (m, C_1TC_2)}$$

Note that contraction does in general not hold for formulas which are not tests. For instance, $\exists; \exists$ puts two elements on the stack, $\exists$ only one.
There is also a rule for left weakening. Due to the format where ; serves as the concatenation operator for formulas, the rule for ; left does double duty as an antecedent weakening rule. See below. Succedent weakening would be the step from $(n, \phi) \Longrightarrow (m, \psi)$ to $(n, \phi) \Longrightarrow (m, \neg(\neg\psi; \neg\chi))$. This is taken care of by the negation rules.

Soundness of Test Contraction Rules Immediate from the fact that $[\![n, T]\!]^{\mathcal{M}} = [\![n, T; T]\!]^{\mathcal{M}}$.

Test Swap Rules

$$\frac{(n, C_1T_1; T_2C_2) \Longrightarrow (m, \phi)}{(n, C_1T_2; T_1C_2) \Longrightarrow (m, \phi)} \qquad \frac{(n, \phi) \Longrightarrow (m, C_1T_1; T_2C_2)}{(n, \phi) \Longrightarrow (m, C_1T_2; T_1C_2)}$$

Soundness of Test Swap Rules Immediate from the fact that $[\![n, T1; T_2]\!]^{\mathcal{M}} = [\![n, T_2; T_1]\!]^{\mathcal{M}}$.

∃ Swap Rules

$$\frac{n, C_1 T; \exists\ C_2 \Longrightarrow (m, \phi)}{n, C_1 \exists; ([^{k}_{+1}]T)C_2 \Longrightarrow (m, \phi)}\ k = n + e(C_1)$$

$$\frac{(n, \phi) \Longrightarrow (m, C_1 T; \exists\ C_2)}{(n, \phi) \Longrightarrow (m, C_1 \exists; ([^{k}_{+1}]T)C_2)}\ k = m + e(C_1)$$

These rules allow us to pull ∃ leftward through a test T, provided we increment the appropriate indices in T.
Pulling ∃ through a test T in the opposite direction is allowed in those cases where ∃ does not bind anything in T. Now we must adjust T by decrementing the appropriate indices:

$$\frac{(n, C_1 \exists; T C_2) \Longrightarrow (m, \phi)}{(n, C_1([^{k}_{-1}]T); \exists\ C_2) \Longrightarrow (m, \phi)}\ k = n + e(C_1),\ k+1 \text{ not in } T$$

$$\frac{(n, \phi) \Longrightarrow (m, C_1 \exists; T C_2)}{(n, \phi) \Longrightarrow (m, C_1([^{k}_{-1}]T); \exists\ C_2)}\ k = m + e(C_1),\ k+1 \text{ not in } T$$

Soundness of ∃ Swap Rules Soundness of the rules for moving ∃ to the left follows from the fact that $[\![k, T; \exists]\!]^{\mathcal{M}} = [\![k, \exists; [^{k}_{+1}]T]\!]^{\mathcal{M}}$.
Soundness of the rules for moving ∃ to the right follows from the fact that if index $k+1$ does not occur in T, then $[\![k, \exists; T]\!]^{\mathcal{M}} = [\![k, ([^{k}_{-1}]T); \exists]\!]^{\mathcal{M}}$.

Context Rules
Memory Shift Rules

$$\frac{(n, \phi) \Longrightarrow (m, \psi)}{(n+1, [^{n}_{+1}]\phi) \Longrightarrow (m+1, [^{n}_{+1}]\psi)} \qquad \frac{(n, \phi) \Longrightarrow (m, \psi)}{(n, \phi) \Longrightarrow (m+1, [^{m}_{+1}]\psi)}$$

Soundness of Memory Shift Rules Memory shift on lefthand side: If ${}_{\sigma}[\![n, \phi]\!]^{\mathcal{M}}_{\sigma}\hat{}\tau$ then for all $a \in M$, ${}_{\sigma\hat{}a}[\![n+1, [^{n}_{+1}]\phi]\!]^{\mathcal{M}}_{\sigma}\hat{}a\hat{}\tau$. Soundness of memory shift on righthand side is established similarly.

Context Extension

$$\frac{(n, \exists; \phi) \Longrightarrow (m, \psi)}{(n+1, \phi) \Longrightarrow (m, \psi)}$$

The counterpart to the rule of context extension (i.e., context absorption) is the rule for introducing an existential quantifier in the antecedent (see the logical rules below).
What context extension and absorption express is that linking information to an outside context (of which nothing further is known) is equivalent, for all purposes of reasoning, to assuming that your information is existentially quantified over.
This is how one can make sense of a ongoing conversation about an unknown 'he': instead of asking questions of identification that might interrupt the flow of the gossip one simply inserts an existential quantifier and listens to what is being said.

Soundness of Context Extension Follows from the fact that

$${}_{\sigma}[\![n, \exists; \phi]\!]^{\mathcal{M}}_{\tau} \text{ iff for some } a \in M,\ {}_{\sigma\hat{}a}[\![n+1, \phi]\!]^{\mathcal{M}}_{\tau}.$$

Logical Rules
The rule for ∃ Left (the converse of context extension) is a special case of the rule ; Left. See below.

∃ Right

$$\frac{(n,\phi) \Longrightarrow (m, [^k/_{m+1}]^- \psi)}{(n,\phi) \Longrightarrow (m, \exists; \psi)}$$

This format is familiar from the Gentzen format of ∃-right in standard predicate logic. Here is an example application:

$$\frac{(1, R1\ 1; \neg(\exists; S1\ 2)) \Longrightarrow (1, R1\ 1; \neg(\exists; S1\ 2))}{(1, R1\ 1; \neg(\exists; S1\ 2)) \Longrightarrow (1, \exists; R1\ 2; \neg(\exists; S2\ 3))}$$

$R1\ 1; \neg(\exists; S1\ 2)$ equals $[^1/_2]^-(R1\ 2; \neg(\exists; S2\ 3))$, so this is indeed a correct application of the rule.

Soundness of ∃ Right Assume a model $\mathcal{M}$ with input and output assignments σ, τ such that ${}_\sigma[\![n,\phi]\!]^{\mathcal{M}}_\tau$. Then by the soundness of the premiss there is a $\theta \sqsupseteq \tau$ and a ρ with

$${}_\theta[\![m, [^k/_{m+1}]^- \psi]\!]^{\mathcal{M}}_\rho.$$

Let $[\![k]\!]^{\mathcal{M}}_\theta = a$. Then, by the definition of the substitution $[^k/_{m+1}]^-$, ${}_{\theta\hat{\ }a}[\![m+1,\psi]\!]^{\mathcal{M}}_\rho$. It follows that ${}_\theta[\![m, \exists; \psi]\!]^{\mathcal{M}}_\rho$. This proves $n, \phi \models (m, \exists; \psi)$.

; Left and Right

$$\frac{(n+e(\phi),\psi) \Longrightarrow (m,\chi)}{(n,\phi;\psi) \Longrightarrow (m,\chi)} \qquad \frac{(n,\phi) \Longrightarrow (m,\psi) \qquad (n,\phi) \Longrightarrow (m,\chi)}{(n,\phi) \Longrightarrow (m, \psi; [^m_{+e(\psi)}]\chi)}$$

The first of these does double duty as a left weakening rule. Antecedent weakening is always extension on the lefthand side. This is because extension on the righthand-side might affect the stack. Weakening with a test is valid anywhere in the antecedent; the swap rules account for that.
An example application of the rule for ; right is:

$$\frac{(1, R1\ 1) \Longrightarrow (1, \exists; R1\ 2) \qquad (1, R1\ 1) \Longrightarrow (1, \exists; R2\ 1)}{(1, R1\ 1) \Longrightarrow (1, \exists; R1\ 2; \exists; R3\ 1)}$$

Soundness of ; Left Suppose ${}_\sigma[\![n,\phi;\psi]\!]^{\mathcal{M}}_\tau$. Let $\sigma' := \tau[1..e(\phi)]$. Then ${}_{\sigma'}[\![n+e(\phi),\psi]\!]^{\mathcal{M}}_\tau$. By the soundness of the premiss, there are $\theta \sqsupseteq \tau$ and ρ with ${}_\theta[\![m,\chi]\!]^{\mathcal{M}}_\rho$. This establishes $(n,\phi;\psi) \models (m,\chi)$.

Soundness of ; Right Assume ${}_\sigma[\![n,\phi]\!]^{\mathcal{M}}_\tau$. Then by the soundness of the second premiss, there are $\theta \sqsupseteq \tau$ and ρ with ${}_\theta[\![m,\chi]\!]^{\mathcal{M}}_\rho$. By Proposition 10, for any $\theta' \in M^{e(\psi)}$,

$${}_{\theta\hat{\ }\theta'}[\![m+e(\psi), [^m_{+e(\psi)}]\chi]\!]^{\mathcal{M}}_{\theta\hat{\ }\theta'\hat{\ }\rho}$$

By the soundness of the first premiss, combined with Lemma 3, there is a $\theta' \in M^{e(\psi)}$ with

$${}_\theta[\![m,\psi]\!]^{\mathcal{M}}_{\theta\hat{\ }\theta'}.$$

It follows that ${}_\theta[\![m,\psi;[^m_{+e(\psi)}]\chi]\!]^{\mathcal{M}}_{\theta\hat{\ }\theta'\hat{\ }\rho}$. This establishes $(n,\phi) \models (m,\psi;[^m_{+e(\psi)}]\chi)$.

¬ Left and Right

$$\frac{(n,\phi) \Longrightarrow (n+e(\phi),\psi)}{(n,\phi;\neg\psi) \Longrightarrow (m,\bot)}\ m \geq n+e(\phi) \qquad \frac{(n,\phi;\psi) \Longrightarrow (m,\bot)}{(n,\phi) \Longrightarrow (n+e(\phi),\neg\psi)}$$

Soundness of ¬ Left Assume ${}_\sigma[\![n,\phi;\neg\psi]\!]^{\mathcal{M}}_\tau$. Then ${}_\sigma[\![n,\phi]\!]^{\mathcal{M}}_\tau$ and there is no θ with ${}_\tau[\![n+e(\phi),\psi]\!]^{\mathcal{M}}_\theta$. Contradiction with the soundness of the premiss. This establishes $(n,\phi;\neg\psi) \models (m,\bot)$.

Soundness of ¬ Right Assume ${}_{\sigma}[\![n,\phi]\!]^{\mathcal{M}}_{\tau}$. Then by the soundness of the premiss, there is no θ with ${}_{\tau}[\![n+e(\phi),\psi]\!]^{\mathcal{M}}_{\theta}$. This establishes $(n,\phi) \models (n+e(\phi),\neg\psi)$.

Double Negation Rules

$$\frac{(n,\phi) \Longrightarrow (m,\neg\neg\psi)}{(n,\phi) \Longrightarrow (m,\psi)} \qquad\qquad \frac{(n,\phi;\neg\neg\psi) \Longrightarrow (m,\bot)}{(n,\phi;\psi) \Longrightarrow (m,\bot)}$$

Soundness of Double Negation Rules For Double Negation Left, assume ${}_{\sigma}[\![n,\phi]\!]^{\mathcal{M}}_{\tau}$. Then by the soundness of the premiss, for no $\theta \sqsupseteq \tau$ is there a ρ with

$${}_{\theta}[\![m,\neg\psi]\!]^{\mathcal{M}}_{\rho}.$$

In particular, we do not have ${}_{\theta}[\![m,\neg\psi]\!]^{\mathcal{M}}_{\theta}$. Therefore, there is a ρ with

$${}_{\theta}[\![m,\psi]\!]^{\mathcal{M}}_{\rho}.$$

This establishes $(n,\phi) \models (m,\psi)$. The soundness of Double Negation Right is established similarly. This completes the presentation of the calculus. Since we have checked the soundness of the axioms and rules as we went along, we have:

Theorem 15 *The Calculus of Incremental Dynamic Reasoning is sound.*

7. Some Derivable Rules for Incremental Dynamic Reasoning

We derive some extra rules that we need for the completeness reasoning in Section 8.

Proposition 16 (Contradiction Rule) *The following rule is derivable:*

$$\frac{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\neg\chi) \qquad (n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\chi)}{(n,\phi) \Longrightarrow (n+e(\phi),\psi)}$$

Proof. Consider the following derivations:

$$\frac{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\neg\chi) \qquad (n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\chi)}{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\neg\chi;\chi)}\ ;\mathrm{r}$$

$$\cfrac{\cfrac{\cfrac{}{(n+e(\phi),\neg\chi) \Longrightarrow (n+e(\phi),\neg\chi)}\ \text{test axiom}}{(n+e(\phi),\neg\chi;\neg\neg\chi) \Longrightarrow (n+e(\phi),\bot)}\ \neg\mathrm{l}}{(n+e(\phi),\neg\chi;\chi) \Longrightarrow (n+e(\phi),\bot)}\ \mathrm{dn}$$

From these two, by transitivity, we get $(n,\phi,\neg\psi) \Longrightarrow (n+e(\phi),\bot)$. From this, we derive the desired conclusion as follows:

$$\cfrac{\cfrac{\cfrac{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\neg\chi) \qquad (n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\chi)}{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\bot)}\ \text{see above}}{(n,\phi) \Longrightarrow (n+e(\phi),\neg\neg\psi)}\ \neg\mathrm{r}}{(n,\phi) \Longrightarrow (n+e(\phi),\psi)}\ \mathrm{dn}$$

□

Proposition 17 (Cases Rule) *The following rule is derivable:*

$$\frac{(n,\phi;\neg\psi) \Longrightarrow (n+e(\phi),\chi) \qquad (n,\phi;\neg\neg\psi) \Longrightarrow (n+e(\phi),\chi)}{(n,\phi) \Longrightarrow (n+e(\phi),\chi)}$$

Proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{(n,\phi;\neg\psi)\Longrightarrow(n+e(\phi),\chi)}{(n,\phi;\neg\psi;\neg\chi)\Longrightarrow(n+e(\phi),\bot)}\;\neg\mathrm{l}}{(n,\phi;\neg\chi;\neg\psi)\Longrightarrow(n+e(\phi),\bot)}\;\mathrm{swap}}{(n,\phi;\neg\chi)\Longrightarrow(n+e(\phi),\neg\neg\psi)}\;\neg\mathrm{r} \qquad \dfrac{\dfrac{\dfrac{(n,\phi;\neg\neg\psi)\Longrightarrow(n+e(\phi),\chi)}{(n,\phi;\neg\neg\psi;\neg\chi)\Longrightarrow(n+e(\phi),\bot)}\;\neg\mathrm{l}}{(n,\phi;\neg\chi;\neg\neg\psi)\Longrightarrow(n+e(\phi),\bot)}\;\mathrm{swap}}{(n,\phi;\neg\chi)\Longrightarrow(n+e(\phi),\neg\neg\neg\psi)}\;\neg\mathrm{r}}{(n,\phi)\Longrightarrow(n+e(\phi),\chi)}\;\mathrm{contrad}$$

□

Proposition 18 (Ex Falso Rule) *The following rule is derivable:*

$$\dfrac{}{(n,\bot)\Longrightarrow(n,\phi)}$$

Proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(n,\bot)\Longrightarrow(n,\bot)}\;\mathrm{start}}{(n,\neg\phi;\bot)\Longrightarrow(n,\bot)}\;;\ \mathrm{l}}{(n,\bot;\neg\phi)\Longrightarrow(n,\bot)}\;\mathrm{swap}}{(n,\bot)\Longrightarrow(n,\neg\neg\phi)}\;\neg\mathrm{r}}{(n,\bot)\Longrightarrow(n,\phi)}\;\mathrm{dn}$$

□

Proposition 19 (Inconsistency Rule) *The following rule is derivable:*

$$\dfrac{(n,\phi)\Longrightarrow(m,\bot)}{(n,\phi)\Longrightarrow(m,\psi)}$$

Proof.

$$\dfrac{(n,\phi)\Longrightarrow(m,\bot) \qquad \dfrac{}{(m,\bot)\Longrightarrow(m,\psi)}\;\text{ex falso}}{(n,\phi)\Longrightarrow(m,\psi)}\;\mathrm{tr}$$

□

Proposition 20 (Modus Ponens) *The following rule is derivable:*

$$\dfrac{(n,\phi)\Longrightarrow(n+e(\phi),(\neg\neg\psi)\to\chi) \qquad (n,\phi)\Longrightarrow(n+e(\phi),\psi)}{(n,\phi)\Longrightarrow(n+e(\phi),\chi)}$$

Proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(n,\phi\Longrightarrow(n+e(\phi),\neg(\neg\neg\psi;\neg\chi)))}{(n,\phi;\neg\neg(\neg\neg\psi;\neg\chi))\Longrightarrow(n+e(\phi),\bot)}\;\neg\mathrm{l}}{(n,\phi;\neg\neg\psi;\neg\chi)\Longrightarrow(n+e(\phi),\bot)}\;\mathrm{dn}}{(n,\phi;\neg\neg\psi)\Longrightarrow(n+e(\phi),\neg\neg\chi)}\;\neg\mathrm{r}}{(n,\phi;\neg\neg\psi)\Longrightarrow(n+e(\phi),\chi)}\;\mathrm{dn} \qquad \dfrac{\dfrac{(n,\phi)\Longrightarrow(n+e(\phi),\psi)}{(n,\phi;\neg\psi)\Longrightarrow(n+e(\phi),\bot)}\;\neg\mathrm{r}}{(n,\phi;\neg\psi)\Longrightarrow(n+e(\phi),\chi)}\;\mathrm{incons}}{(n,\phi)\Longrightarrow(n+e(\phi),\chi)}\;\mathrm{cases}$$

□

8. Completeness of the Calculus

To establish the completeness of the calculus, assume that $(0,\phi) \not\Longrightarrow (m,\psi)$. If $m < e(\phi)$, the definition of L consequence immediately yields that $(0,\phi) \not\models (m,\psi)$, so we may assume that $m \geq e(\phi)$. By proposition 14, we may also assume without loss of generality that $m = e(\phi)$.

Because of the context extension rule, our assumption that the context is initially empty is harmless. For suppose it is not, and we have that $(n,\phi) \not\Longrightarrow (m,\psi)$ for $n \neq 0$. Then by context extension also $(0,\exists^n;\phi) \not\Longrightarrow (m,\psi)$.

We will construct a countermodel by a slight modification of the standard Henkin construction for the completeness of classical predicate logic. It is convenient to use k for $e(\phi)$ throughout the reasoning that follows. Also, in the following, we extend the language with individual constants.

Definition 21 *A set of L formulas is k-bounded if every member of the set is in L_k (i.e., every formula in the set has the form (k,ϕ)). We use (k,Γ) to refer to k-bounded sets of formulas.*
$\phi \vdash_\Gamma \psi :\Leftrightarrow$ there are $(k,\phi_1),\dots,(k,\phi_n) \in (k,\Gamma)$ with $(0,\phi;\neg\neg\phi_1;\cdots;\neg\neg\phi_n) \Longrightarrow (k,\psi)$.
(k,Γ) is consistent with $(0,\phi)$ if there is a (k,ψ) with $\phi \not\vdash_\Gamma \psi$.
(k,Γ) is negation complete with respect to $(0,\phi)$ if for every $(k,\psi) \in L$ either $\phi \vdash_\Gamma \psi$ or $\phi \vdash_\Gamma \neg\psi$.
(k,Γ) has witnesses for $(0,\phi)$ if for every $(k,\exists;\psi)$ such that $\phi \vdash_\Gamma \exists;\psi$ there is a c for which $(k,\neg\neg\exists\psi \to [^c/_{k+1}]^-\psi) \in (k,\Gamma)$.

Note that in the definition of $\phi \vdash_\Gamma \psi$ the extra premisses from Γ do not extend the 'anaphoric context': the context change potential of the premisses from Γ is blocked off by means of double negation signs.

Proposition 22 *If $\phi \not\vdash_\Gamma \psi$ then at least one of $(k,\Gamma) \cup \{(k,\psi)\}$, $(k,\Gamma) \cup \{(k,\neg\psi)\}$ is consistent with $(0,\phi)$.*

Proof. Use the Cases Rule. □

Let $(k,\exists;\chi_1),\dots$ be a list of all k-bounded formulas of L that start with $\exists$. Let $C_0 := c^0_1,\dots$ be a list of fresh individual constants. Let L_0 be $L(C_0)$ (the result of adding the constants C_0 to L).

$$(k,\Delta_0) := \{(k,\neg\neg\exists\chi_i \to [^{c^0_i}/_{k+1}]^-\chi_i) \mid i \in \mathbb{N}^+\}.$$

Let $(k,\exists\chi^m_1),\dots$ be a list of all k-bounded existential formulas which occur in L_m. Let $C_{m+1} := c^{m+1}_1,\dots$ be a list of fresh individual constants. Let $L_{m+1} := L_m(C_{m+1})$.

$$(k,\Delta_{m+1}) := \{(k,\neg\neg\exists\chi^{m+1}_i \to [^{c^{m+1}_i}/_{k+1}]^-\chi^{m+1}_i) \mid i \in \mathbb{N}^+\}.$$

Let $C := \bigcup_m C_m$, and let (k,Δ) be the set of $L(C)$ formulas given by:

$$(k,\Delta) := \bigcup_m (k,\Delta_m).$$

Proposition 23 *If (k,Γ) consists of $L(C)$ formulas, and $(k,\Gamma) \supseteq (k,\Delta)$, then (k,Γ) has witnesses for $(0,\phi)$.*

Proof. Take some $(k,\exists\psi)$ with $\phi \vdash_\Gamma \exists\psi$. Then $\exists\psi \in L_m$ for some m. So there is some $c \in C$ with $\neg\neg\exists\psi \to [^c/_{k+1}]^-\psi \in \Delta_{m+1}$. So $(k,\neg\neg\exists\psi \to [^c/_{k+1}]^-\psi) \in (k,\Delta) \subseteq (k,\Gamma)$. □

Proposition 24 *If (k,Γ) has witnesses for $(0,\phi)$ and $\phi \vdash_\Gamma \exists\psi$, then there is some $c \in C$ with $\phi \vdash_\Gamma [^c/_{k+1}]^-\psi$.*

Proof. By the fact that Modus Ponens is a derivable rule (Proposition 20). □

Proposition 25 *If (k,Γ) is consistent with $(0,\phi)$ then there is a $(k,\Gamma') \supseteq (k,\Gamma)$ which is consistent with $(0,\phi)$, negation complete with respect to $(0,\phi)$, and has witnesses for $(0,\phi)$.*

Proof. Assume (k,Γ) consistent with $(0,\phi)$. Let $(k,\chi_1),\dots,(k,\chi_i),\dots$ be an enumeration of all k-bounded formulas of the language $L(C)$. Extend (k,Γ) as follows to a (k,Γ') with the required properties.

$$(k,\Gamma_0) := (k,\Gamma)\cup(k,\Delta)$$

$$(k,\Gamma_{m+1}) := \begin{cases} (k,\Gamma_m\cup\{\chi_m\}) & \text{if } (k,\Gamma_m\cup\{\chi_m\}) \text{ consistent with } (0,\phi),\\ (k,\Gamma_m) & \text{otherwise.}\end{cases}$$

$$(k,\Gamma') := (k,\bigcup_m \Gamma_m)$$

$(k,\Gamma')\supseteq(k,\Delta)$, so by Proposition 23 (k,Γ') has witnesses for $(0,\phi)$.
Assume (k,Γ') is inconsistent with $(0,\phi)$. Then some (k,Γ_m) has to be inconsistent with $(0,\phi)$ and contradiction with Proposition 22. So (k,Γ') is consistent with $(0,\phi)$.
Finally, (k,Γ') is negation complete by construction. $\square$

Definition 26 (Canonical Model) *Let (k,Γ) be consistent with $(0,\phi)$, be negation complete with respect to $(0,\phi)$, and have witnesses for $(0,\phi)$. Then $\mathcal{M}_\Gamma=(D,I)$ is defined as follows. $D :=$ the set of natural numbers $\{1,\dots,k\}$ together with the set of constants C occurring in $\Gamma\cup\{\phi\}$. For all terms of the language, let $I(t):=t$. Let $I(P):=\{\langle t_1,\dots,t_k\rangle \mid \phi\vdash_\Gamma Pt_1\cdots t_k)\}$ (where it is given that all the t_i are either constants or indices in the range $1,\dots,k$).*

Lemma 27 (Satisfaction Lemma) *Let (k,Γ) be consistent with $(0,\phi)$, be negation complete with respect to $(0,\phi)$, and have witnesses for $(0,\phi)$. For all k-bounded ξ:*
$\phi\vdash_\Gamma\xi$ iff $\exists\tau$ with ${}_{\langle 1..k\rangle}[\![k,\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle\hat{}\tau}$.

Proof. Induction on the structure of ξ.
$\phi\not\vdash_\Gamma\bot$ by the fact that $(0,\phi)$ is consistent and Γ is consistent with $(0,\phi)$.
$\phi\vdash_\Gamma\exists$ by the fact that, as $(0,\phi)$ is consistent and Γ is consistent with $(0,\phi)$, we have $\phi\vdash_\Gamma\top$, and therefore by $\exists r$, $\phi\vdash_\Gamma\exists;\top$.
$\phi\vdash_\Gamma Pt_1\cdots t_n$ iff $\langle t_1..t_n\rangle\in I(P)$ iff ${}_{\langle 1..k\rangle}[\![k,Pv_1\cdots v_n]\!]^{M_\Gamma}_{\langle 1..k\rangle}$.
$\phi\vdash_\Gamma\neg\xi$ iff (Γ negation complete) $\phi\not\vdash_\Gamma\xi$ iff (i.h.) there is no τ with ${}_{\langle 1..k\rangle}[\![k,\xi]\!]^{M_\Gamma}_\tau$ iff (semantic clause for $\neg$) ${}_{\langle 1..k\rangle}[\![k,\neg\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle}$.
$\phi\vdash_\Gamma\exists\xi$ iff (Γ has witnesses) $\phi\vdash_\Gamma[^c/_{k+1}]^-\xi$ iff (i.h.) there is a τ with ${}_{\langle 1..k\rangle}[\![k,[^c/_{k+1}]^-\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle\hat{}\tau}$ iff ${}_{\langle 1..k\rangle}[\![k,\exists\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle\hat{}c\hat{}\tau}$.
The final case we have to deal with is $A;\xi$, where A ranges over $\bot$, $Pt_1\cdots t_n$, $\neg\xi'$. In this case, we have: $\phi\vdash_\Gamma A;\xi$ iff (A is a test) $\phi\vdash_\Gamma A$ and $\phi\vdash_\Gamma\xi$ iff (i.h. twice, plus the fact that A is a test) ${}_{\langle 1..k\rangle}[\![k,A]\!]^{M_\Gamma}_{\langle 1..k\rangle}$ and there is a τ with ${}_{\langle 1..k\rangle}[\![k,\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle\hat{}\tau}$ iff there is a τ with ${}_{\langle 1..k\rangle}[\![k,A;\xi]\!]^{M_\Gamma}_{\langle 1..k\rangle\hat{}\tau}$. $\square$

Proposition 28 *Let (k,Γ) be consistent with $(0,\phi)$, be negation complete with respect to $(0,\phi)$, and have witnesses for $(0,\phi)$. Then ${}_\epsilon[\![0,\phi]\!]^{M_\Gamma}_{\langle 1..k\rangle}$.*

Proof. Let $\exists^k\phi'$ be the result of applying the rule for moving $\exists$ leftward as many times as necessary to ϕ to ensure that $e(\phi')=0$. Then $(0,\phi)$ and the formula $(0,\exists^k\phi')$ are proof equivalent. Furthermore, we have:

$$\dfrac{\dfrac{\dfrac{}{(k,\phi')\Longrightarrow(k,\phi')}\text{ test axiom}}{(0,\exists^k\phi')\Longrightarrow(k,\phi')}\ \exists\text{l}, k\text{ times}}{(0,\phi)\Longrightarrow(k,\phi')}\ \text{swap rules}$$

Therefore, $\phi \vdash_\Gamma \phi'$, and by the satisfaction lemma, ${}_{\langle 1..k\rangle}[\![k, \phi']\!]^{M_\Gamma}_{\langle 1..k\rangle}$. Also, by definition of the semantics for $\exists$, we have that ${}_\epsilon[\![0, \exists^k]\!]^{M_\Gamma}_{\langle 1..k\rangle}$. By the semantic equivalence of ϕ and $\exists^k\phi'$ we get ${}_\epsilon[\![0, \phi]\!]^{M_\Gamma}_{\langle 1..k\rangle}$. □

Theorem 29 (Completeness) *If* $(n, \phi) \models (m, \psi)$ *then* $(n, \phi) \Longrightarrow (m, \psi)$.

Proof. Assume $(n, \phi) \not\Longrightarrow (m, \psi)$. Without loss of generality we may assume that $(n, \phi) \not\Longrightarrow (k, \psi)$, where $k := n + e(\phi)$, because from $(n, \phi) \Longrightarrow (k, \psi)$ it follows by a suitable number of applications of memory shift on the righthand side that $(n, \phi) \Longrightarrow (m, \psi')$, where ψ' is the result of shifting the 'write registers' of ψ to the right.
By context extension, it follows from $(n, \phi) \not\Longrightarrow (k, \psi)$ that $(0, \exists^n\phi) \not\Longrightarrow (k, \psi)$. Set $\phi' := \exists^n\phi$. Then $k = e(\phi')$, and $\{(k, \neg\psi)\}$ is consistent with $(0, \phi')$. By proposition 25, there is a $(k, \Gamma) \supseteq \{(k, \neg\psi)\}$ which is consistent with $(0, \phi')$, is negation complete with respect to $(0, \phi')$, and has witnesses for $(0, \phi')$. Construct the canonical model and apply the satisfaction lemma to get:

$$ {}_{\langle 1..k\rangle}[\![k, \neg\psi]\!]^{M_\Gamma}_{\langle 1..k\rangle}. $$

By the semantic clause for negation we have that for all τ:

$$ \langle 1..k\rangle, \tau) \notin [\![k, \psi]\!]^{M_\Gamma}. $$

By proposition 28:

$$ {}_\epsilon[\![0, \phi']\!]^{M_\Gamma}_{\langle 1..k\rangle}. $$

This proves $(0, \phi') \not\models (k, \psi)$, i.e., $(0, \exists^n; \phi) \not\models (k, \psi)$, and therefore, $(n, \phi) \not\models (k, \psi)$. □

9. Anaphoric Reasoning with Equality

Anaphoric linking makes extensive use of equality. See Van Eijck [8] for an in-depth analysis of the use of equality in anaphoric descriptions. An anaphoric definite description like *the garden* can be treated as a definiteness quantifier followed by a link to a contextually available index. The translation of *He sprinkles the garden* would then be something like $2, \iota : (3 \doteq 2; G3); S1\,3$. Also, the determiner *an other* often has an implicit anaphoric element. In such cases, the treatment involves non-identity links to contextually available referents. *He met an other woman* gets a translation like $2, \exists; 3 \neq 2; W3; M1\,3$. Below we indicate how to handle equality, while leaving the axiomatisation of definiteness in the present framework for another occasion.
The following rules must be added to the calculus to deal with equality statements (we now assume the presence of a set *Cons* of individual constants):

Reflexivity Axiom

$$ \frac{}{(n, \phi) \Longrightarrow (m, t \doteq t)}\ n + e(\phi) \leq m,\ t \in \text{Cons} \cup \{1, \ldots, m\} $$

Soundness of Reflexivity Axiom The axiom expresses that equality is reflexive.

Substitution Rule

$$ \frac{(n, \phi) \Longrightarrow (m, [{}^{t_1}/{}_{t_2}]\psi)}{(n, \phi; t_1 \doteq t_2) \Longrightarrow (m, \psi)}\ t_1, t_2 \in \text{Cons} \cup \{1, \ldots, m\} $$

Example application:

$$ \dfrac{\dfrac{}{(0, \top) \Longrightarrow (0, a \doteq a)}\,\text{refl}}{(0, a \doteq b) \Longrightarrow (0, b \doteq a)}\,\text{subst} $$

For the correctness of this application, note that $a \doteq a$ is of the form $[^a/_b]b \doteq a$.

$$\dfrac{\dfrac{}{(0, a \doteq b) \Longrightarrow (0, a \doteq b)}\ \text{test axiom}}{(0, a \doteq b; b \doteq c) \Longrightarrow (0, a \doteq c)}\ \text{subst}$$

For the correctness of this application, note that $a \doteq b$ is of the form $[^b/_c]a \doteq c$.

$$\dfrac{\dfrac{\dfrac{}{(3, 1 \doteq 2) \Longrightarrow (3, 1 \doteq 2)}\ \text{test axiom}}{(3, 1 \doteq 2; 2 \doteq 3) \Longrightarrow (3, 1 \doteq 3)}\ \text{subst}}{(0, \exists; \exists; \exists; 1 \doteq 2; 2 \doteq 3) \Longrightarrow (3, 1 \doteq 3)}\ \exists\text{l}, 3\ \text{times}$$

Soundness of the Substitution Rule Assume ${}_\sigma[\![n, \phi; t_1 \doteq t_2]\!]^{\mathcal{M}}_\tau$. Then ${}_\sigma[\![n, \phi]\!]^{\mathcal{M}}_\tau$, and $[\![t_1]\!]^{\mathcal{M}}_\tau = [\![t_2]\!]^{\mathcal{M}}_\tau$. By the soundness of the premiss, there are $\theta \sqsupseteq \tau$ and ρ with ${}_\theta[\![m, [^{t_1}/_{t_2}]\psi]\!]^{\mathcal{M}}_\rho$. Therefore, ${}_\theta[\![m, \psi]\!]^{\mathcal{M}}_\rho$. This shows $(n, \phi; t_1 \doteq t_2) \models (m, \psi)$.

The completeness of the anaphoric calculus with equality is proved by modifying the Henkin construction in the usual way (taking equivalence classes of terms under provable equality as elements of the canonical model).

10. Conclusion

Semanticists sympathetic to DRT do not tend to worry about the top down construction algorithm for DRSs, with a novelty condition to ensure incrementality of interpretation. Those interested in carrying out a Montagovian or Fregean enterprise of building representations for complex constituents out of representations for its components insist on the formulation of a bottom up procedure, however. This has led to the emergence of various dynamic logics intended as rational reconstructions of the DRT programme. Unfortunately, the most well known of these, DPL, has a problem of destructive assignment, and is therefore not the best candidate for a reformulation of DRT in Fregean or Montagovian terms. This flaw is remedied in the present proposal.

When looking at the general picture of dynamic reconstruction proposals for DRT, what may emerge is that there is no single 'best' reconstruction, but that various proposals shed light on different aspects of the dynamics of text processing that all merit study in their own right. The present 'calculus of incremental dynamics' focusses on the abstraction over anaphoric context in reasoning. It gives an explicit account of anaphoric links between premisses and conclusion in reasoning, and is more well behaved than previous DRT calculi because of its match with a transitive consequence relation. To be sure, the present sequent approach to axiomatising dynamic logics can also be used to get still closer to standard DRT, or to axiomatize DPL and its variants: see the proof systems that are given in Van Eijck [9].

To wind up our story we mention some connections to related work. Via the translation to DRT in Section 3 (proposition 7) we have a proof system for (a streamlined version of) DRT. The calculus makes the discipline of using and modifying the anaphoric context and of handling dynamically bound indices fully explicit. It can be viewed as a proof system for 'pure' DRSs, a proof system that avoids the award reference to alphabetic variance in the rules of proof proposed in Kamp and Reyle [17]. The present calculus differs from the earlier proof system for DRT by Saurer [22] in the fact that it does not rely on an implicit translation to FOL.

Finally we mention the connections with Dekker [7], where a similar plea is made for incremental dynamics but the problem of a calculus for reasoning is not addressed, with Visser and Vermeulen [24] and Visser [26, 25], and with Blackburn and Venema [5] and Hollenberg [14]. To see the connection with Hollenberg's equational axioms of dynamic negation and relational composition, note that these are all derivable in the calculus of incremental dynamics (as of course they should be).

Acknowledgements
Thanks to Marco Hollenberg and Albert Visser for stimulating remarks and helpful discussions, to Paul Dekker for valuable suggestions, and to two reviewers of this journal for criticism that spurred me on to considerable improvements.

References

1. P. Aczel. Variable binding. Notes of a talk for Accolade, Plasmolen, November 1996.
2. H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics (2nd ed.)*. North-Holland, Amsterdam, 1984.
3. J. Barwise. Noun phrases, generalized quantifiers and anaphora. In P. Gärdenfors, editor, *Generalized Quantifiers: linguistic and logical approaches*, pages 1–30. Reidel, Dordrecht, 1987.
4. D. Ben-Shalom. A path-based variable-free system for predicate logic. Technical Report CS-R9444, CWI, Amsterdam, July 1994.
5. P. Blackburn and Y. Venema. Dynamic squares. *Journal of Philosophical Logic*, 24:469–523, 1995.
6. N.G. de Bruyn. A survey of the project AUTOMATH. In J.R. Hindley and J.P. Seldin, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 579–606. Academic Press, London, 1980.
7. P. Dekker. Predicate logic with anaphora. In L. Santelmann and M. Harvey, editors, *Proceedings of the Fourth Semantics and Linguistic Theory Conference*, page 17 vv, Cornell University, 1994. DMML Publications.
8. J. van Eijck. The dynamics of description. *Journal of Semantics*, 10:239–267, 1993.
9. J. van Eijck. Axiomatising dynamic logics for anaphora. submitted to Journal of Language and Computation, July 1998.
10. J. van Eijck and H. Kamp. Representing discourse in context. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 179–237. Elsevier, Amsterdam, 1996.
11. J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100, 1991.
12. C.A. Gunter. *Semantics of Programming Languages*. MIT Press, Cambridge, Massachusetts, 1992.
13. I. Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, Amherst, 1982.
14. M. Hollenberg. An equational axiomatisation of dynamic negation and relational composition. *Journal of Logic, Language and Information*, 6:381–401, 1997.
15. H. Kamp. A theory of truth and semantic representation. In J. Groenendijk et al., editors, *Formal*

Methods in the Study of Language. Mathematisch Centrum, Amsterdam, 1981.

16. H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.
17. H. Kamp and U. Reyle. A calculus for first order discourse representation structures. *Journal of Logic, Language and Information*, 5(3–4):297–348, 1996.
18. S. Kuhn. An axiomatisation of predicate functor logic. *Notre Dame Journal of Formal Logic*, 24:233–241, 1983.
19. W.C. Purdy. A logic for natural language. *Notre Dame Journal of Formal Logic*, 32:409–425, 1991.
20. W.V.O. Quine. Variables explained away. In *Selected Logic Papers*. ??, ??
21. V. Sánchez. *Studies on Natural Logic and Categorial Grammar*. PhD thesis, University of Amsterdam, 1991.
22. W. Saurer. A natural deduction system for discourse representation theory. *Journal of Philosophical Logic*, 22(3):249–302, 1993.
23. C.F.M. Vermeulen. Sequence semantics for dynamic predicate logic. *Journal of Logic, Language, and Information*, 2:217–254, 1993.
24. C.F.M. Vermeulen and A. Visser. Dynamic bracketing and discourse representation. Technical Report 131, Utrecht Research Institute for Philosophy, 1995.
25. A. Visser. Dynamic relation logic is the logic of DPL-relations. *Journal of Logic, Language and Information*, 6:441–452, 1997.
26. A. Visser. The design of dynamic discourse denotations. Lecture notes, Utrecht University, February, 1994.