1991

D.J.N. van Eijck, F.-J. de Vries

Dynamic interpretation and Hoare deduction

Computer Science/Department of Software Technology Report CS-R9115 March

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Copyright © Stichting Mathematisch Centrum, Amsterdam

Dynamic Interpretation and Hoare Deduction

Jan van Eijck[1,2] & Fer-Jan de Vries[1]

[1]CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

[2]OTS, Trans 10, 3512 JK Utrecht, The Netherlands

Abstract

This paper presents a dynamic assigment language with η and ι assignment in the style of the dynamic predicate logic of [7], extended with generalized quantifiers. The constructs for η and ι assignment allow a very straightforward analysis of indefinite and definite descriptions in natural language, while the dynamic treatment of generalized quantifiers is intended to cover the dynamic interaction of quantification and description in natural language. The use of this dynamic assignment language (called *DAL*) for natural language analysis, along the lines of [2] and [7], is demonstrated by examples. The medium permits one to treat a wide variety of 'donkey sentences': conditionals with a donkey pronoun in their consequent and quantified sentences with donkey pronouns anywhere in the scope of the quantifier. Our account does not suffer from the so-called proportion problem.

Discussions about the correctness or incorrectness of proposals for dynamic interpretation of language have been hampered in the past by the difficulty of seeing through the ramifications of the dynamic semantic clauses (phrased in terms of input-output behaviour) in non-trivial cases. To remedy this, we supplement the dynamic semantics of our representation language with an axiom system in the style of Hoare. The rules we propose form a deduction system for *DAL* which is proved correct and complete with respect to the semantics. We define the *static meaning* of a *DAL* program π as the weakest condition φ such that π terminates successfully on all states satisfying φ, and we show that our calculus gives a straightforward method for finding static meanings of *DAL* programs.

1985 Mathematics Subject Classification: 03B65, 68Q55, 68S10.
CR Categories: F.3.1, F.3.2, I.2.4, I.2.7.
Keywords and Phrases: semantics of natural language, dynamic interpretation, Hoare logic, knowledge representation languages.

1 Introduction

Our starting point is a dynamic perspective on natural language as proposed in [2] and [7]. For a quick review of the advantages of such a dynamic perspective, consider mini discourse (1), where the hearer is asked to take two different individuals in mind.

(1) *A man walked in. He sat down. Another man walked in.*

Suppose we intend the reading where *he* is anaphorically linked to the earlier indefinite and *another* is anaphorically constrained (to borrow a term from [2]) by that same indefinite. We follow [2] in using superscript indices for antecedents and subscripts for anaphors, to indicate the intention.

(2) *A man^1 walked in. He$_1$ sat down. Another$_1$ man^2 walked in.*

What intuitively happens when discourse (2) is processed can be described as follows. First one (i.e., the processor) is invited to focus on an arbitrary man. Then one is asked to consider a choice of man where that man walked in. Furthermore one is asked to focus on a choice of man where that man sat down as well. Next one is assumed to keep this choice of man in mind, and again to pick a reference to an arbitrary man, but in such a way that that man is different from the first man. Finally one is to consider a second choice of man where that second man walked in.

This account sounds like a piece of imperative programming, which suggests that its meaning can be given in terms of a translation into a programming language. Here is such a translation (tense is ignored), in a language which has the same expressive power as the dynamic predicate logic of [7], but which reveals its imperative programming nature a bit more clearly.

(3) $\eta v_1 : \textit{man}\ v_1;\ \textit{walk-in}\ v_1;\ \textit{sit-down}\ v_1;\ \eta v_2 : (v_2 \neq v_1;\ \textit{man}\ v_2);\ \textit{walk-in}\ v_2.$

The programming language employed in (3) has two kinds of basic statements: assignments and tests. $\eta x : \pi$ is a command to assign an arbitrary individual to x that fulfils requirement π. $R(t_1 \cdots t_n)$ is a test which succeeds on a given input variable state if the values of the terms t_1 through t_n fulfil the condition specified by the relation symbol R, and fails otherwise. Sequences of statements are formed with the sequencing operator ;. The assignments are non-deterministic, which means that semantically a program is not a function from states to states, but a relation between states (or equivalently, a function from states to sets of states). Test statements narrow down the set of output states. A test relates an input state that satisfies it to itself, and an input state that does not satisfy it to nothing at all.

We will fix the meaning for this assign-and-test mini-language by giving a (dynamic) semantics for it. Next, we specify a set of Hoare style rules for it, by way of axiom system. The advantage of doing both of these things is that the Hoare style rules provide a link to notions of static semantics, thus allowing us to take snapshots of truth conditions at various stages in the discourse processing, so to speak. Stated otherwise, the Hoare style rules allow us to consider projections from dynamic logic to static logic, in the sense of [3].

2 Dynamic Assignment Logic: Syntax and Informal Semantics

This section is meant as an introduction to dynamic predicate logic in its undisguised form as an imperative programming language. The ingredients to be introduced here are atomic tests, sequential program composition, indefinite and definite assignments, program implication and program negation. Later on we will add generalized quantifiers. We call the language we are about to present *DAL* (Dynamic Assignment Logic).

In natural language, one does not engage in explicitly bookkeeping with regard to the 'slots' used for keeping track of individuals mentioned in discourse. One just keeps them in mind, and does not confuse them, that is all. To make sure that in *DAL* the slots do not get confused, one might stipulate that new assignments to variables which are already 'active' are forbidden. We will not impose this constraint on the general framework, but we will define a sublanguage of *DAL* programs where the constraint is imposed.

We first define the set of programs of *DAL* and the set **av** of *assignment variables* of a *DAL* program. For simplicity's sake we take the terms of *DAL* to be a set of individual variables V (one might want to add constants and deictic parameters to this, but we will not do so here). C is a set of individual constants

Given a set of terms and a set of relation symbols, the set of *DAL* programs is the smallest set such that the following hold.

1. $\bot$ is a program.
2. If t_1, t_2 are terms, then $t_1 = t_2$ is a program.
3. If R is an n-place relation symbol and $t_1, \ldots, t_n$ are terms, then $R(t_1 \cdots t_n)$ is a program.
4. If π_1 and π_2 are programs then $(\pi_1;\ \pi_2)$ is a program.
5. If π_1 and π_2 are programs then $(\pi_1 \Rightarrow \pi_2)$ is a program.
6. If π is a program, then $\neg\pi$ is a program.
7. If π is a program and x is a variable, then $\eta x : \pi$ is a program.
8. If π is a program and x is a variable, then $\iota x : \pi$ is a program.

Here is the definition of the set $\mathbf{av}(\pi)$ of *assignment variables* of a program π.

1. $\mathbf{av}(\bot) = \emptyset$.
2. $\mathbf{av}(t_1 = t_2) = \emptyset$.
3. $\mathbf{av}(R(t_1 \cdots t_n)) = \emptyset$.
4. $\mathbf{av}((\pi_1;\ \pi_2)) = \mathbf{av}(\pi_1) \cup \mathbf{av}(\pi_2)$.
5. $\mathbf{av}((\pi_1 \Rightarrow \pi_2)) = \emptyset$.
6. $\mathbf{av}(\neg\pi) = \emptyset$.
7. $\mathbf{av}(\eta x : \pi) = \{x\} \cup \mathbf{av}(\pi)$.
8. $\mathbf{av}(\iota x : \pi) = \{x\} \cup \mathbf{av}(\pi)$.

Next we define the sublanguage of *DAL* programs (call it DAL_0), where no new values get assigned to 'active' variables. Here we need simultaneous recursion on π and $\mathbf{av}(\pi)$. It is the smallest set such that the following hold.

1. $\bot$ is a program with $\mathbf{av}(\bot) = \emptyset$.
2. If t_1, t_2 are terms, then $t_1 = t_2$ is a program with $\mathbf{av}(t_1 = t_2) = \emptyset$.
3. If R is an n-place relation symbol and $t_1, \ldots, t_n$ are terms, then $R(t_1 \cdots t_n)$ is a program with $\mathbf{av}(R(t_1 \cdots t_n)) = \emptyset$.
4. If π_1 and π_2 are programs with $\mathbf{av}(\pi_1) \cap \mathbf{av}(\pi_2) = \emptyset$ then $(\pi_1;\ \pi_2)$ is a program with $\mathbf{av}((\pi_1;\ \pi_2)) = \mathbf{av}(\pi_1) \cup \mathbf{av}(\pi_2)$.
5. If π_1 and π_2 are programs with $\mathbf{av}(\pi_1) \cap \mathbf{av}(\pi_2) = \emptyset$ then $(\pi_1 \Rightarrow \pi_2)$ is a program with $\mathbf{av}((\pi_1 \Rightarrow \pi_2)) = \emptyset$.
6. If π is a program, then $\neg\pi$ is a program with $\mathbf{av}(\neg\pi) = \emptyset$.
7. If π is a program and x is a variable with $x \notin \mathbf{av}(\pi)$ then $\eta x : \pi$ is a program with $\mathbf{av}(\eta x : \pi) = \{x\} \cup \mathbf{av}(\pi)$.
8. If π is a program and x is a variable with $x \notin \mathbf{av}(\pi)$ then $\iota x : \pi$ is a program with $\mathbf{av}(\iota x : \pi) = \{x\} \cup \mathbf{av}(\pi)$.

We will follow the usual predicate logical convention of omitting outermost parentheses for readability. Also, it will become evident from the semantic clause for sequential composition that the ; operator is associative. Therefore, we will often take the liberty to write $\pi_1;\ \pi_2;\ \pi_3$ instead of $(\pi_1;\ \pi_2);\ \pi_3$ or $\pi_1;\ (\pi_2;\ \pi_3)$. Also, $t_1 \neq t_2$ will be used to abbreviate $\neg t_1 = t_2$ (cf. example (3)). Finally, we use $\top$ as an abbreviation for $\neg\bot$.

The restrictions on assignment variables will for example rule out (4) as a program of DAL_0 (though not of *DAL*).

(4) $\quad \eta v_1 : \mathit{man}\ v_1;\ \eta v_1 : \mathit{boy}\ v_1.$

The remainder of this section is devoted to an informal account of the semantics of atomic test predicates, implication, negation and union of *DAL* programs, and η and ι assignment. Section 3 will give the formal semantics.

Semantically, what we are interested in is *states*, functions from the set of *DAL* variables to individuals in a model. Semantically, *DAL* programs act as state transformers: a *DAL* program takes an input state and either indicates success by producing an output state or it indicates failure by not producing anything at all. Equivalently, we can view the meaning of a program as a function mapping anyy input state to the set of all possible outputs the program can produce for that input. A program which is a test will on input A either produce output set $\{A\}$ (in case the test succeeds) or output set $\emptyset$ (in case the test fails). Programs which may produce non-singleton sets are non-deterministic; for some inputs there is more than one possible output state. Examples of non-deterministic programs are η assignment programs; the program $\eta x : \pi$ has, on input A, the set of all states which may differ from A in the fact that they have another x value, namely some value that satisfies π.

The program $\bot$ expresses a test which always fails; it is meant to express the same as **if** *true* **then** *fail* **else** *skip* **fi**. In other words: for every input state A, $\bot$ will produce output state $\emptyset$. As was mentioned above, we use $\top$ as an abbreviation for $\neg\bot$. The program $\top$ is a test which always succeeds; in other words, it is meant to express the same as the ALGOL style statement **if** *true* **then** *skip* **else** *fail* **fi**. In other words, for every input state A, $\top$ will produce output set $\{A\}$. Atomic predicates like $t_1 = t_2$ or $R(t_1 \cdots t_n)$ are meant to express tests which may fail; in ALGOL style notation: **if** $R(t_1 \cdots t_n)$ **then** *skip* **else** *fail* **fi**. Again in terms of input output behaviour: If $R(t_1 \cdots t_n)$ evaluates to true in state A, the predicate will have output set $\{A\}$, otherwise the output set will be $\emptyset$.

Programs of the form $(\pi_1 \Rightarrow \pi_2)$ are intended to treat the interplay of natural language implication and descriptions, as in the following examples.

(5) *If a girl*^{1} *has a boyfriend*^{2}, *she*$_1$ *teases him*$_2$.

(6) *If a man*^{1} *admires the king*^{2}, *he*$_1$ *cheers him*$_2$.

Example (5) has the following *DAL* translation.

(7) $(\eta v_1 : \mathit{girl}\ v_1;\ \eta v_2 : \mathit{boyfriend}\ v_2;\ \mathit{has}(v_1, v_2)) \Rightarrow \mathit{teases}(v_1, v_2)$.

To get the semantics right, one has to assume that (7) is true if and only if every output state for the antecedent $\eta v_1 : \mathit{girl}\ v_1;\ \eta v_2 : \mathit{boyfriend}\ v_2;\ \mathit{has}(v_1, v_2)$ will be an appropriate input state for the consequent $\mathit{teases}(v_1, v_2)$ (see [2] or [7]).

Negation should allow one to treat examples like the following, where the negation has scope over an indefinite.

(8) *The manager*^{1} *does not use a PC*^{2}.

This example can be translated into *DAL* as follows:

(9) $\iota v_1 : (\mathit{manager}\ v_1);\ \neg(\eta v_2 : \mathit{pc}\ v_2;\ \mathit{use}(v_1, v_2))$.

To get the semantics right, a negated program should act as a test: $\neg\pi$ should accept (without change) all variable states which cannot serve as input for π, and reject all others. In fact, it will turn out that $\neg\pi$ is definable in terms of $\Rightarrow$ and $\bot$, as $\pi \Rightarrow \bot$.

Definite descriptions can act as anaphors and antecedents at the same time. Discourse (10) provides an example.

(10) *A customer*^{1} *entered. The woman*$^2_1$ *sat down. She*$_2$ *smiled.*

The indices indicate that *the woman* has *a customer* as its antecedent, while at the same time acting itself as antecedent for *she* in the next sentence (and constraining the gender of the pronoun). A *DAL* translation of (10) is given in (11).

(11) $\eta v_1 : \mathit{customer}\ v_1;\ \mathit{enter}\ v_1;\ \iota v_2 : (v_2 = v_1;\ \mathit{woman}\ v_2);\ \mathit{sit\text{-}down}\ v_2;\ \mathit{smile}\ \ v_2$.

The ι assignment in (11) is dependent on the η assignment to variable v_1. With reference to a particular assignment for v_1, the description is unique. Note that the ι assignment to v_2 does indirectly act as a test on the previous η assignment to v_1: this test will weed out η assignments that are inappropriate in the light of the subsequent discourse.

Definite descriptions can also be dependent on each other. Consider the string of characters in (12).

(12) $a\ \hat{A}\,\hat{}\,b\ C$.

Suppose just for an instant that (12) is a state of affairs one is talking about. The state of affairs involves characters and hat symbols (*hats* for short). With reference to (12), it does make sense to talk about *the character with the hat*, although (12) neither has a unique character nor a unique hat. We can, for instance, truthfully assert (13) about (12).

(13) *The character with the hat is a capital.*

The translation into DAL is straightforward:

(14) $\iota v_1 : (character\ v_1;\ \iota v_2 : (hat\ v_2;\ with(v_1, v_2)));\ capital\ v_1$.

Intuitively, the first ι assignment 'tries out' individual characters C until it finds the unique C with the property that a unique hat H for C can be found.

The semantic picture sketched above is still in need of an extra touch. Nothing we have said so far makes clear how the dynamic treatment of definite descriptions is meant to deal with their uniqueness presuppositions. This topic will not be dealt with in this paper. We intend to devote a separate paper to it.

3 Semantics: Formal Definitions

Assume a model $\mathcal{M} = \langle U, I \rangle$, with U a universe of individuals and I an interpretation function for the first order relation symbols of the language. We consider the set $\mathbf{S}$ of all functions $A : V \rightarrow U$. This is the set of *states* for $\mathcal{M}$.

A state A for $\mathcal{M} = \langle U, I \rangle$ determines a valuation $\mathbf{V}_A$ for the terms of the language as follows: f $t \in V$ then $\mathbf{V}_A(t) = A(t)$ (as we take all our terms to be variables, this is all there is to the definition of $\mathbf{V}_A$). If A is a state for $\mathcal{M}$, x a variable and d an element of the universe or $\mathcal{M}$, then $A[x := d]$ is the state for $\mathcal{M}$ which is just like A except for the possible difference that x is mapped to d.

We define a function $[\![\pi]\!]_{\mathcal{M}} : \mathbf{S} \rightarrow \mathcal{P}\mathbf{S}$ by recursion. A, B, C are used as metavariables over states. The function $[\![\pi]\!]_{\mathcal{M}}$ depends on the model $\mathcal{M}$, but for convenience we will often write $[\![\pi]\!]$ rather than $[\![\pi]\!]_{\mathcal{M}}$. The function should be read as: on input state A, π may produce any of the outputs in output state set $[\![\pi]\!](A)$.

1. $[\![\bot]\!](A) = \emptyset$.
2. $[\![R(t_1 \cdots t_n)]\!](A) = \begin{cases} \{A\} & \text{if } \langle \mathbf{V}_A(t_1), \ldots, \mathbf{V}_A(t_n) \rangle \in I(R) \\ \emptyset & \text{otherwise.} \end{cases}$
3. $[\![t_1 = t_2]\!](A) = \begin{cases} \{A\} & \text{if } \mathbf{V}_A(t_1) = \mathbf{V}_A(t_2) \\ \emptyset & \text{otherwise.} \end{cases}$
4. $[\![(\pi_1;\ \pi_2)]\!](A) = \bigcup\{[\![\pi_2]\!](B) \mid B \in [\![\pi_1]\!](A)\}$.
5. $[\![(\pi_1 \Rightarrow \pi_2)]\!](A) = \begin{cases} \{A\} & \text{if for all } B \in [\![\pi_1]\!](A) \text{ it holds that } [\![\pi_2]\!](B) \neq \emptyset \\ \emptyset & \text{otherwise.} \end{cases}$
6. $[\![\neg\pi]\!](A) = \begin{cases} \{A\} & \text{if } [\![\pi]\!](A) = \emptyset \\ \emptyset & \text{otherwise.} \end{cases}$
7. $[\![\eta x : \pi]\!](A) = \bigcup\{[\![\pi]\!](A[x := d]) \mid d \in U\}$.
8. $[\![\iota x : \pi]\!](A) = \begin{cases} [\![\pi]\!](A[x := d]) \text{ for the unique } d \in U & \\ \text{for which } [\![\pi]\!](A[x := d]) \neq \emptyset & \text{if such a } d \text{ exists} \\ \emptyset & \text{otherwise.} \end{cases}$

Truth is defined in terms of input-output behaviour: π is true relative to model $\mathcal{M}$ if there are states A, B for $\mathcal{M}$ such that $B \in [\![\pi]\!]_{\mathcal{M}}(A)$. Two programs π_1, π_2 are equivalent if for every model $\mathcal{M}$ and every state A for $\mathcal{M}$, $[\![\pi_1]\!]_{\mathcal{M}}(A) = [\![\pi_2]\!]_{\mathcal{M}}(A)$.

Dynamic consequence is defined as follows: $\pi_1 \models \pi_2$ if for every model $\mathcal{M}$ and for all states A, B for $\mathcal{M}$: if $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ then there is a state C with $C \in [\![\pi_2]\!]_{\mathcal{M}}(B)$. The various other possible consequence relations for dynamic logic will not concern us in this paper.

The statement $\eta x : \pi$ performs a non-deterministic action, for it sanctions any assignment to x of an individual satisfying π. The statement acts as a test at the same time: in case there are no individuals satisfying π the set of output states for any given input state will be empty. In fact, the meaning of $\eta x : \pi$ can be thought of as a random assignment followed by a test, for $\eta x : \pi$ is equivalent to $\eta x : \top;\ \pi$, or in more standard notation, $x := ?;\ \pi$. It follows immediately from this explanation plus the dynamic meaning of sequential composition that $\eta x : (\pi_1); \pi_2$ is equivalent with $\eta x : (\pi_1;\ \pi_2)$.

The interpretation conditions for ι assignment make clear how the uniqueness condition is handled dynamically. The statement $\iota x : \pi$ consists of a test followed by a deterministic action in case the test succeeds: first it is checked whether there is a unique π; if so, this individual is assigned to x; otherwise the program fails (in other words, the set of output states is empty). Thus we see that the two programs $\iota x : (\pi_1);\ \pi_2$ and $\iota x : (\pi_1;\ \pi_2)$ are not equivalent. The program $\iota x : (\pi_1;\ \pi_2)$ succeeds if there is a unique object d satisfying $\pi_1;\ \pi_2$, while the requirement for $\iota x : (\pi_1);\ \pi_2$ is stronger: there has to be a unique individual d satisfying π_1, and d must also satisfy π_2.

The clause for dynamic implication should take care of the proper treatment of the description *his wife* in example (15).

(15) *If the president is married then his wife will be cross with him.*

To indicate the reading where the possessive pronoun *his* is anaphorically linked to *the president*, we again use indices. The intended reading of (15) is indicated by the following indexing.

(16) *If the president*^{1} *is married then [his*$_1$ *wife]*^{2} *will be cross with him*$_1$.

A suitable translation is the following:

(17) $(\iota v_1 : \textit{president}\ v_1;\ \textit{married}\ v_1) \Rightarrow (\iota v_2 : \textit{wife-of}(v_2, v_1);\ \textit{cross-with}(v_2, v_1))$.

Assuming that *the president* has unique reference, we can say the following. Either the subprogram for *The president is married* will not complete successfully, and then program (17) succeeds, or there will be a unique referent for ι assignment in the subprogram for the consequent, and the subprogram for *his wife will be cross with him* will succeed, provided that the unique referent for v_2 satisfies $\textit{cross-with}(v_2, v_1)$.

4 Adding Quantifiers

Adding quantifiers to *DAL* is relatively straightforward. For convenience, we restrict attention to binary quantifiers. Let $Q_1, Q_2, \ldots$ be a list of binary quantifier symbols. Assume that the interpretation functions of the models $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ are extended with suitable interpretations for these. That is to say, for every Q_i, $I(Q_i)$ is a binary quantifier relation on $\mathcal{P}(D)$, i.e. a relation satisfying the constraints of *extension*, *isomorphy* and *conservativity* (see for example [14]).

A quantifier relation R is *conservative* (or: lives on its first argument) if $R(A, B)$ iff $R(A, A \cap B)$. A quantifier relation satisfies *extension* if adding or deleting individuals from the part of the universe which is outside the extension of the arguments does not affect the relation, i.e., if the relation satisfies $R_E(A, B)$ iff $R'_E(A, B)$, for all E, E' with $E, E' \supseteq A \cup B$.

Because of our reliance on *conservativity*, the analysis provided in this section does not work without further ado for a quantifier like *only*, as in *Only men are chauvinists*. This problem will be addressed in Section 5.

First we extend the syntax of *DAL*. The set of *DAL* programs is the smallest set satisfying the following clauses.

1. – 8. As above, in the definition of *DAL*.

9. If π_1 and π_2 are programs and Q is a quantifier, then $Qx(\pi_1, \pi_2)$ is a program.

For the time being we do not take external dynamic effects of quantifiers into account, i.e. we take our quantifiers to be only internally dynamic. This transpires in the definition of the set of assignment variables of a quantified program.

1. – 8. As above.

9. $\mathbf{av}(Qx(\pi_1, \pi_2)) = \emptyset$.

As for the extended version of DAL_0, the set of DAL_0 programs is the smallest set satisfying the following clauses.

1. – 8. As above, in the definition of DAL_0.

9. If π_1 and π_2 are programs with $x \notin \mathbf{av}(\pi_1)$, $x \notin \mathbf{av}(\pi_2)$ and $\mathbf{av}(\pi_1) \cap \mathbf{av}(\pi_2) = \emptyset$, and Q is a quantifier, then $Qx(\pi_1, \pi_2)$ is a program with $\mathbf{av}(Qx(\pi_1, \pi_2)) = \emptyset$.

Here is the intended semantics.

1. – 8. As above.

$$9.\ [\![Qx(\pi_1,\pi_2)]\!](A) = \begin{cases} \{A\} & \text{if} \\ & \langle\{d \in U \mid [\![\pi_1]\!](A[x := d]) \neq \emptyset\}, \\ & \ \{d \in U \mid [\![\pi_1;\ \pi_2]\!](A[x := d]) \neq \emptyset\}\rangle \\ & \qquad\qquad\qquad\qquad \in I(Q) \\ \emptyset & \text{otherwise.} \end{cases}$$

The semantic clause makes clear that quantifiers, as defined here, do not have an external dynamic effect. Externally, they acts as tests: in case the test succeeds the set of output states has the input state as its only member. The internal dynamic effect of a quantifier does interact with the external dynamic effects of definites and indefinites, however.

To see how the semantics of quantification works in the simplest possible case, let us walk through example program (18).

(18) $\qquad Qx(Sx, Tx)$.

According to the semantic clause, on input state A this program gives $\{A\}$ iff the sets (19) and (20) are in the relation $I(Q)$.

(19) $\qquad \{d \in U \mid [\![Sx]\!](A[x := d]) \neq \emptyset\}$.

(20) $\qquad \{d \in U \mid [\![Sx;\ Tx]\!](A[x := d]) \neq \emptyset\}$.

According to the semantic clause for atomic tests and the definition of the valuation function for terms, the set (19) can be rewritten as (21).

(21) $\qquad \{d \in U \mid d \in I(S)\}$.

In the same way, and using the semantic clause for ; , the set (20) can be rewritten as (22).

(22) $\qquad \{d \in U \mid x \in I(S) \text{ and } x \in I(T)\}$.

Because the Q is assumed to denote a conservative quantifier with first argument as given by (21), (22) can be replaced by (23).

(23) $\qquad \{d \in U \mid x \in I(T)\}$.

Thus we find that (18) is true iff the sets given in (21) and (23) are in the relation denoted by the quantifier. This is the expected result.

Dynamic effects will start to play a rôle if one or both of the argument programs of a quantified program has an external dynamic effect. The traditional 'donkey' examples such as (24) are cases in point (see [6]).

(24) $\qquad$ *Every girl*^{1} *who has a boyfriend*^{2} *teases him*$_2$.

Example (24) has the following *DAL* translation.

(25) $\qquad Q_\forall v_1(Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2, Tv_1v_2)$.

Here $Q_\forall$ denotes the generalized universal quantifier, i.e. the relation of inclusion. Establishing the meaning of examples like (24) by direct reasoning about the operational semantics is awkward, so we will guide the reader through the thicket once more.

According to the semantic clause for quantifier programs, on input state A this program gives $\{A\}$ iff the sets (26) and (27) are in the inclusion relation.

(26) $\qquad \{d \in U \mid [\![Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2]\!](A[v_1 := d]) \neq \emptyset\}$.

(27) $\qquad \{d \in U \mid [\![Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2;\ Tv_1v_2]\!](A[v_1 := d]) \neq \emptyset\}$.

First we reduce (26). Applying the semantic clauses for ; and η assignment and for atomic tests makes clear that (26) describes the same set as (28).

(28) $\{d \in U \mid \text{there is a } d' \in U \text{ such that } d \in I(G), d' \in I(B), \langle d, d' \rangle \in I(H)\}$.

Similarly, application of the semantic clauses for ;, for η assignment and for atomic tests makes clear that (27) describes the same set as (29).

(29) $\{d \in U \mid \text{there is a } d' \in U \text{ such that } d \in I(G), d' \in I(B), \langle d, d' \rangle \in I(H), \langle d, d' \rangle \in I(T)\}$.

Paraphrasing this, we see that the semantic clause for quantified programs entails that translation (25) of (24) is true iff the set of girls who have a boyfriend is included in the set of girls who have a boyfriend and tease that boyfriend. Note that this may still be true if some of the girls have several boyfriends and tease only one of them. This shows that the reading we get for (24) is weaker than the one we got for (5). Of course, as soon as we impose the meaning postulate that every girl who has a boyfriend has precisely one boyfriend, the two translations become equivalent again.

The reader might wonder why example (24) gets a weaker interpretation than expected. The answer is that there turns out to be some latitude as to the exact way of internal dynamic binding of assignment variables from the antecedent program of the quantifier in its consequent program. These issues will be dealt with in the next section.

Readers who find the reasoning about dynamic clauses that is necessary to grasp the meaning of examples like (24) awkward have our full sympathy. Bear with us, and the axiom system to be presented in Section 6 will alleviate your problem.

Because we have used *conservativity*, the above story does not work for the quantifier relation interpreting *only*. *Only P Q* is true iff the set of non-Ps is included in the set of non-Qs, or equivalently, if the set of Qs is included in the set of Ps. Because *only* is not conservative, (30) does not mean the same as (31).

(30) *Only girls*^{1} *who tease a boyfriend*^{2} *lose him*$_2$.

(31) *Only girls*^{1} *who tease a boyfriend*^{2} *tease a boyfriend and lose him*$_2$.

Rather, (30) means something like (32).

(32) *Only girls*^{1} *who tease a boyfriend*^{2} *lose their boyfriend*$_2$.

This suggests that in this case the pronoun is a pronoun of laziness rather than a genuine donkey pronoun. But pronouns in the context of non-conservative quantifiers pose difficult problems, as is also borne out by the following example.

(33) *Only girls*^{1} *who have a boyfriend*^{2} *bring him*$_2$ *to the party.*

On its most salient reading, (33) is true as a matter of course, because it can be paraphrased as (34).

(34) *No girls*^{1} *who don't have a boyfriend*^{2} *will bring him*$_2$ *to the party.*

Interestingly, the paraphrase (34) poses a difficulty for our framework too. The problem is that the variable for *a boyfriend*$_2$ is screened off by the negation operator, so that it is not available anymore at the level where *him*$_2$ looks for an antecedent. To deal with (34) one would again need externally dynamic negation. We leave the problem of non-conservative quantifiers with the remark that it merits further investigation.

The following example, closely related to (24), gives rise to the so-called *proportion problem* in traditional discourse representation theory (see [10] for the basics of discourse representation theory, [11] for an up-to-date formulation, and [9] for details on the proportion problem).

(35) *Most girls*^{1} *who have a boyfriend*^{2} *tease him*$_2$.

The proportion problem may arise in connection with (35) in case there are girls who are naughty enough to have a large number of boyfriends and to tease them all. Accounts which give rise to the proportion problem would handle (35) as a case of quantification over girl-boyfriend pairs. To see how the present proposal fares, consider the translation of (35) in *DAL*.

(36) $\mathbf{most}\ v_1((girl\ v_1;\ \eta v_2 : boyfriend\ v_2;\ have\ (v_1, v_2)), tease\ (v_1, v_2))$.

This is true in state A if there are states B, C such that the sets given in (37) and (38) are in the **most**-relation.

(37) $\{d \in U \mid B \in [\![girl\ v_1;\ \eta v_2 : boyfriend\ v_2;\ have\ (v_1, v_2)]\!](A[v_1 := d])\}$

(38) $\{d \in U \mid B \in [\![girl\ v_1;\ \eta v_2 : boyfriend\ v_2;\ have\ (v_1, v_2);\ tease\ (v_1, v_2)]\!](A[v_1 := d])\}$

The set given by (37) is the set of all girls who have a boyfriend, while the set given by (38) is the set of all girls who have a boyfriend and tease him. The quantification is over girls, as it should be, and not over girl–boyfriend pairs, as in the accounts which give rise to the proportion problem. In other words, this spells out a reading that does not suffer from the proportion problem. Again this is not the only possible reading; an alternative reading will be discussed in the next section.

To show that the treatment of quantification proposed here is different from the treatment proposed in [7], it is enough to show that the approach of [7] suffers from the proportion problem. One of the examples they discuss (o.c. p.81) is, essentially, (39), in the reading which can be paraphrased as (40).

(39) *If a girl*^{1} *has a boyfriend*^{2} *she usually teases him*$_2$.

(40) *In most cases in which a girl*^{1} *has a boyfriend*^{2} *she teases him*$_2$.

To treat this example, Groenendijk and Stokhof reconstruct Lewis' adverb of quantification approach in dynamic predicate logic, by reading the quantifier as a relation between sets of states. Dynamic implication, $\Rightarrow$, would then correspond to $\rightarrow_\forall$, to be interpreted as: for *all* output states A of the antecedent, applying the consequent to A will produce an output. Similarly, the examples with *usually* or *in most cases* are analyzed with $\rightarrow_M$, to be interpreted as: for *most* output states A of the antecedent, applying the consequent to A will produce an output. To see that this account does give rise to the proportion problem, observe that output states of the antecedent *a girl has a boyfriend* where Mary has John as a boyfriend and where the same Mary has Fred as a boyfriend will have to count as different states. The example sentences may have a reading where these should indeed count as different, but the point is that quantification over states makes it impossible to express readings where they should count as the same, as in the reading of (39) which is equivalent to the most salient reading of (35). For such cases, quantification over states does simply lead to incorrect results.

Our approach differs from the approach in [7] precisely in that quantification is always over individuals and never over states. Our reconstruction of (40) would be as follows. Because the quantification is over cases or occasions, we have to add an occasion parameter to the predicates used for translating verb phrases, so *have(x,y,o)* and *tease(x,y,o)* for *x has y at occasion o* and *x teases y at occasion o*, respectively. The *DAL* translation of (40) now becomes:

(41) $\textbf{most}\ o_1(\eta v_1(girl\ v_1;\ \eta v_2 : boyfriend\ v_2;\ have\ (v_1, v_2, o_1)), tease\ (v_1, v_2, o_1))$.

To make this true, on our account, the set of occasions at which a girl has a boyfriend and the set of occasions at which a girl has a boyfriend which she teases must be in the **most** relation.

Of course, on our account there is still a fair amount of latitude as to how (39), (40) and (42) are interpreted.

(42) *If a girl has several boyfriends, she usually teases them.*

But the latitude resides where it belongs, for a margin of uncertainty remains as long as it is unclear what counts as an *occasion*, and it disappears as soon as this is resolved. As soon as we have a model where occasions are fully individuated, our quantificational analysis gives the right meanings. The discussion summarised in [9] of the meanings of 'donkey' examples with *usually* should therefore in our view be re-interpreted as a discussion of factors that might be involved in the individuation of occasions.

5 Weak and Strong Readings of Quantifiers

The semantic clause for quantifier programs $Qx(\pi_1, \pi_2)$ in the previous section takes care of the dynamic binding of the assignment variables of the antecedent program π_1 in the consequent program π_2 by comparing the set of objects for wich π_1 succeeds with the set of objects for which $\pi_1;\ \pi_2$ succeeds. *Prima facie*, there is no compelling reason to use sequential composition of π_1 and π_2 to achieve the desired internal binding effect. Another obvious possibility is the choice $\pi_1 \Rightarrow \pi_2$. We will now explore this alternative. Let us call the quantifier reading presented in the previous section the *weak* reading of the quantifier. The semantic clause where $\Rightarrow$ is substituted for ; gives the *strong* reading of the quantifier. Thus, the clauses for weak and strong readings of quantifiers (which we will distinguish as Q^w and Q^s) run like this:

Weak readings of quantifiers

$$[\![Q^w x(\pi_1,\pi_2)]\!](A) = \begin{cases} \{A\} & \text{if} \\ & \langle\{d \in U \mid [\![\pi_1]\!](A[x := d]) \neq \emptyset\}, \\ & \ \{d \in U \mid [\![\pi_1;\ \pi_2]\!](A[x := d]) \neq \emptyset\}\rangle \\ & \qquad\qquad\qquad\qquad \in I(Q) \\ \emptyset & \text{otherwise.} \end{cases}$$

Strong readings of quantifiers

$$[\![Q^s x(\pi_1,\pi_2)]\!](A) = \begin{cases} \{A\} & \text{if} \\ & \langle\{d \in U \mid [\![\pi_1]\!](A[x := d]) \neq \emptyset\}, \\ & \ \{d \in U \mid [\![\pi_1 \Rightarrow \pi_2]\!](A[x := d]) \neq \emptyset\}\rangle \\ & \qquad\qquad\qquad\qquad \in I(Q) \\ \emptyset & \text{otherwise.} \end{cases}$$

It is easy to show that as long as the antecedent program π_1 does not have an external dynamic effect, there is nothing to choose between the weak and the strong reading of a quantifier. The following proposition refers to *conservativity* and *extension*; See Section 4 for the definitions.

Proposition 1 *If Q is a quantifier satisfying extension and conservativity and* $\mathbf{av}(\pi_1) = \emptyset$, *then the programs $Q^w x(\pi_1,\pi_2)$ and $Q^s x(\pi_1,\pi_2)$ are equivalent.*

Proof: If $\mathbf{av}(\pi_1) = \emptyset$, this means that π_1 is a test. Thus, instead of (43) we can write (44).

(43) $\{d \in U \mid [\![\pi_1;\ \pi_2]\!](A[x := d]) \neq \emptyset\}$.

(44) $\{d \in U \mid [\![\pi_1]\!](A[x := d]) \neq \emptyset \text{ and } [\![\pi_2]\!](A[x := d]) \neq \emptyset\}$.

Let S_1 be the set $\{d \in U \mid [\![\pi_1]\!](A[x := d]) \neq \emptyset\}$, and S_2 the set $\{d \in U \mid [\![\pi_2]\!](A[x := d]) \neq \emptyset\}$. Then the set given by (44) is the set $S_1 \cap S_2$. Thus, the quantifier program $Q^w x(\pi_1,\pi_2)$ holds iff the sets S_1 and $S_1 \cap S_2$ are in the relation $I(Q)$. Because of the conservativity of Q, this is the case iff S_1 and S_2 are in the $I(Q)$ relation.

Similarly, if $\mathbf{av}(\pi_1) = \emptyset$, then instead of (45) we can write (46).

(45) $\{d \in U \mid [\![\pi_1 \Rightarrow \pi_2]\!](A[x := d]) \neq \emptyset\}$.

(46) $\{d \in U \mid [\![\pi_1]\!](A[x := d]) = \emptyset \text{ or } [\![\pi_2]\!](A[x := d]) \neq \emptyset\}$.

Thus, the set given by (46), with respect to some universe U, is the set $(U - S_1) \cup S_2$. Because Q satisfies extension, we may take U to be any set including the set $S_1 \cup S_2$. In particular, if we take U to be $S_1 \cup S_2$, the set $(U - S_1) \cup S_2$ reduces to S_2. Thus, again, the quantifier program succeeds iff S_1 and S_2 are in the $I(Q)$ relation. ■

It is clear from the proof of proposition 1 that the strong reading of quantifiers will get us the right results for non-conservative quantifiers provided they satisfy *extension* and provided their first argument is a test. It turns out that we can handle examples like *Only men are chauvinists* after all; because *only* does satisfy extension, the recipe is simply to rely on the strong reading of the quantifier *only*. Note, however, that non-conservative quantifiers with restriction clauses with a dynamic effect are still beyond our scope.

In Section 4 it transpired that the weak readings of universally quantified donkey sentences were not equivalent to the *if then* versions of these donkey sentences. The availability of strong readings for quantifiers has remedied this situation, as the following proposition shows.

Proposition 2 *For any programs π_1, π_2 with* $\mathbf{av}(\pi_1) \cap \mathbf{av}(\pi_2) = \emptyset$: *if x is a variable such that* $x \notin \mathbf{av}(\pi_1) \cup \mathbf{av}(\pi_2)$, *then $(\eta x : \pi_1) \Rightarrow \pi_2$ is equivalent with $Q^s_\forall x(\pi_1,\pi_2)$.*

Proof: Take an arbitrary model $\mathcal{M}$ and an arbitrary state A for that model, and just check the semantic clauses: $[\![(\eta x : \pi_1) \Rightarrow \pi_2]\!]_{\mathcal{M}}(A) = \{A\}$ iff for all $B \in [\![(\eta x : \pi_1)]\!]_{\mathcal{M}}(A)$ it holds that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$. By the semantic clause for η assignment this is the case iff for all B with the property that for some $d \in U_{\mathcal{M}}$ it is the case that $B \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$, it holds that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$. By quantifier logic, this is equivalent to: for all $d \in U_{\mathcal{M}}$ and for all B, if $B \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$, then $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$.

The semantic clause for strong readings of quantifiers says that $[\![Q^s_\forall x(\pi_1, \pi_2)]\!]_{\mathcal{M}}(A) = \{A\}$ iff all $d \in U_{\mathcal{M}}$ such that $[\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$ have the property that $[\![\pi_1 \Rightarrow \pi_2]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$. By the semantic clause for $\Rightarrow$, this condition is equivalent to: for all $d \in U_{\mathcal{M}}$ such that $[\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$ it is the case that all $B \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$ have the property that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$. Equivalently: for all $d \in U_{\mathcal{M}}$ and for all C it holds that if $C \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$ then is the case that all $B \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$ have the property that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$. By quantifier logic, this in turn is equivalent to: for all $d \in U_{\mathcal{M}}$ and for all B, if $B \in [\![\pi_1]\!]_{\mathcal{M}}(A[x := d])$ then $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$.

Since $(\eta x : \pi_1) \Rightarrow \pi_2$ and $Q^s_\forall x(\pi_1, \pi_2)$ are both tests, and since we have shown that these tests succeed in precisely the same circumstances, we have established the claim that the programs are equivalent. ■

An obvious next question is whether a similar result holds for $Q^w_\exists x(\pi_1, \pi_2)$ and $\eta x : \pi_1;\ \pi_2$. The answer is of course *no*, for the simple reason that $Q^w_\exists x(\pi_1, \pi_2)$ is a test while $\eta x : \pi_1;\ \pi_2$ is not. But there is something more to the question than this. For quantifiers like *every* and *most*, it is not difficult to invent pairs of example sentences where the first member of the pair has to be paraphrased as a weakly read existential quantifier and the second member as a strongly read one (see below for some standard examples from the literature of such pairs for the quantifiers *every* and *most*). It seems to us that it is much harder to find examples of such pairs for the case of *some*. It is not even completely clear to us if the distinction between $Q^w_\exists x(\pi_1, \pi_2)$ and $Q^s_\exists x(\pi_1, \pi_2)$ makes intuitive sense. This observation might be interpreted as evidence for the case that determiners like *some* and *a* are quite special after all, in that they are to be treated in terms of active assignment variables rather than generalized quantifiers. We will not pursue this issue any further here, as it would lead us into the realm of externally dynamic quantification and thus beyond the scope of this paper.

There is extensive discussion in the literature (see e.g. [4] and the references cited therein) of the distinction between weak and strong readings of quantified 'donkey' sentences with quantifiers like *every* and *most*. Sentences like (47), (49) and (51) are given in the literature as cases where the weak reading is appropriate, while sentences like (48), (50) and (52) seem to require the strong reading.

(47) *If a man has a dime, he puts it in the parking meter.*

(48) *If a man owned a slave, he also owned its offspring.*

(49) *Every man who has a dime will put it in the parking meter.*

(50) *Every man who owned a slave also owned its offspring.*

(51) *Most men who have a dime will put it in the parking meter.*

(52) *Most men who owned a slave also owned its offspring.*

For the examples (49), (50), (51) and (52) we get the required contrast from the distinction between the two interpretation rules for quantifiers. For examples (47) and (48), an analysis in terms of occasions is needed to get the required constrast. As long as we stick to the treatment of *if then* clauses in terms of $\Rightarrow$, we only get the strong readings. But of course, a more subtle analysis of (47) and (48) will treat the *if then* construction in terms of universal quantification over cases, along the lines sketched in Section 4. The analysis of (47) and (48) now becomes (53) and (54), respectively.

(53) $Q^w_\forall o_1(\eta v_1(\mathit{man}\ v_1;\ \eta v_2 : \mathit{dime}\ v_2;\ \mathit{have}\ (v_1, v_2, o_1)), \mathit{put\text{-}in\text{-}parking\text{-}meter}\ (v_1, v_2, o_1))$.

(54) $Q^s_\forall o_1(\eta v_1(\mathit{man}\ v_1;\ \eta v_2 : \mathit{slave}\ v_2;\ \mathit{owned}\ (v_1, v_2, o_1)),$
$\iota v_3 : \mathit{offspring\text{-}of}\ (v_3, v_2);\ \mathit{owned}\ (v_1, v_3, o_1))$.

We can paraphrase the semantic condition imposed by (53) as follows: every occasion where there is a man with a dime is an occasion where there is a man with a dime who puts a dime in the parking meter. This is intuitively acceptable. The paraphrase for (54) becomes: every occasion where there is a man with a slave that he owns is an occasion where it holds for every man and every slave owned by that man at that occasion, that the man owns the offspring of that slave. This also seems intuitively

acceptable. If the reader is not convinced that these are indeed the correct paraphrases of the weak and strong quantifier readings for these examples, the sections that follow will provide an easy means to verify these claims.
Other example from the folk-lore are quite easy to handle. Consider examples (55), (56) and (57).

(55) *If a man*^{1} *shares a place with another*$_1$ *man*^{2}*, he*$_1$ *shares the housework with him*$_2$*.*

(56) *Most customers*^{1} *who buy a spark plug*^{2} *buy three other*$_2$ *spark plugs with it*$_2$ *(unless they own a 2CV).*

(57) *If a customer*^{1} *buys a spark plug*^{2}*, he*$_1$ *usually buys three other*$_2$ *spark plugs with it*$_2$ *(unless he owns a 2CV).*

In the case of (55) we get the right results if we analyse the *if then* phrase in terms of weak or strong universal quantification over occasions (some reflection will show that for this example the weak and strong readings are equivalent). The example was invented to argue against a so-called E-type analysis of the pronouns he_1 and him_2 (see [9]), but in the present perspective there is nothing problematic about such symmetric cases at all. On the analysis we propose, sentence (55) is true iff every occasion in which there are two distinct men sharing a place is an occasion in which there are two distinct men sharing a place and also sharing the housework. Equivalently, sentence (55) is true iff every occasion in which there are two distinct men sharing a place is an occasion in which every pair of distinct men sharing a place is such that they also share the housework.

The equivalence of strong and weak readings also holds for example (56). Under the weak reading, example (56) says that the majority of customers (excluding 2CV owners) that buy a spark plug are customers that buy a spark plug plus three other spark plugs. Under the strong reading, example (56) says that the majority of customers (excluding 2CV owners) that buy a spark plug are customers that for every spark plug that they buy, buy three other ones. It is not difficult to see that in this case as well, the weak and strong readings are equivalent. The same holds for the case of (57), although here we have to analyze in terms of quantification over occasions.

6 An Axiom System for Dynamic Interpretation

Discussions about the correctness or incorrectness of proposals for dynamic interpretation of language have been hampered in the past by the difficulty of seeing through the ramifications of the dynamic semantic clauses in non-trivial cases. To remedy this, we supplement the semantics of our representation language with an axiom system in the style of Hoare (see [1] for an overview of this approach). The axioms and proof rules we propose form a deduction system allowing us to prove statements about *DAL* programs; in Section 7, this calculus is proved sound and in Section 8 it is proved complete with respect to the semantics.

Our deductive system for dynamic logic is a hybrid calculus, with statements characterizing variable states, plus two kinds of correctness statements, which we call *universal* and *existential* correctness statements. Thus, the system has three kinds of statements: (i) formulae of a language of first order predicate logic with the same sets of variables and predicate letters as the *DAL* language under consideration, and extended with the same set of generalized quantifiers (call this assertion language L), (ii) triples of the form $\{\varphi\}\ \pi\ \{\psi\}$, where φ, ψ are L-formulae, and π is a *DAL*-program, and (iii) triples of the form $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$, where again φ, ψ are L-formulae, and π is a *DAL*-program.

The statements of the form φ are used for making assertions about variable states A for L with respect to models $\mathcal{M}$ for L. Because the *DAL* language and the assertion language L have the same set of variables, variable states for the *DAL* language are variable states for L. The relation $\mathcal{M} \models \varphi[A]$, for state A verifies φ in $\mathcal{M}$, is defined in the standard way. If φ is a formula of the assertion language L, then $\mathbf{fv}(\varphi)$ is the set of free variables of φ, and if φ is an L formula and x, y are variables then $[y/x]\varphi$ is the result of the substitution of y for all free occurrences of x in φ.

The statements of the form $\{\varphi\}\ \pi\ \{\psi\}$ are *universal* correctness statements. In the terminology of Hoare's logic, they express partial correctness. The statement $\{\varphi\}\ \pi\ \{\psi\}$ expresses that all variable states A (for an arbitrary model $\mathcal{M}$) that satisfy φ have the property that if some variable state B is an output state of π for input state A, then B satisfies φ.

The statements of the form $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ are *existential* correctness statements. In terms of Hoare's logic, they represent the bits one has to add to partial correctness statements to ensure total correctness. The statement $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ expresses that for all input variable states A (for an arbitrary model $\mathcal{M}$) that satisfies φ there is some variable state B satisfying ψ in the set of output states of π.

Because our intuitions about static meaning seem to be much better developed than our intuitions about dynamic meaning, we can, for a large class of natural language sentences, check whether the intuitive meaning of a sentence S corresponds to the meaning of its *DAL* translation π in the following precise sense. Does the intuitive meaning of S precisely describe the set of states for which π terminates successfully? In terms of Hoare's logic, we can describe this set of states by the weakest existential precondition of π with respect to $\top$. What we are looking for is the weakest φ for which the statement $\langle\varphi\rangle\ \pi\ \langle\top\rangle$ is still true. The φ we are looking for has to satisfy the additional condition that it does not contain free occurrences of the assignment variables of π (the members of $\mathbf{av}(\pi)$); φ gives the *static meaning* of the program π.

It may seem that our intention to use the calculus to get from dynamic to static meaning will allow us to get by with just existential correctness statements. To see that this is not so, note that such statements do not allow us to express failure of a program for a given sets of input states. The statement $\langle\varphi\rangle\ \pi\ \langle\bot\rangle$ does *not* express failure of π on input states satisfying φ. Rather, it expresses the fact that for all inputs satisfying φ the program π is guaranteed to produce an output satisfying $\bot$, a statement which is absurd for all non-contradictory φ. Failure of a *DAL* program π on the set of inputs specified by φ, is readily expressed in terms of universal correctness, namely by $\{\varphi\}\ \pi\ \{\bot\}$. It is clear that in order to treat negation of programs and dynamic implication between programs, both universal and existential correctness statements are needed in the calculus.

The meanings of $\{\varphi\}\ \pi\ \{\psi\}$ and $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ are formally specified in terms of the dynamic interpretation function $[\![\cdot]\!]_{\mathcal{M}}$ that was given above plus a satisfaction relation $\mathcal{M} \models \varphi[A]$, to be read as: variable state A for V satisfies φ in $\mathcal{M}$. This notion is defined in the standard way. The notion of $\mathcal{K}$-validity for correctness statements is defined as follows.

$\mathcal{K}$-validity of Correctness Statements

If F has the form φ, where φ is a formula of the assertion language, then $\mathcal{K} \models F$ if $\mathcal{M} \models F[A]$ for all models $\mathcal{M} \in \mathcal{K}$ and all states A for $\mathcal{M}$.

If F has the form $\{\varphi\}\ \pi\ \{\psi\}$, then $\mathcal{K} \models F$ if the following holds. For all models $\mathcal{M} \in \mathcal{K}$, for all states A for $\mathcal{M}$, if $\mathcal{M} \models \varphi[A]$ then for all states $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ it is the case that $\mathcal{M} \models \psi[B]$.

If F has the form $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$, then $\mathcal{K} \models F$ if the following holds. For all models $\mathcal{M} \in \mathcal{K}$, for all states A for $\mathcal{M}$, if $\mathcal{M} \models \varphi[A]$ then there is at least one state $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \psi[B]$.

The atomic predicates of *DAL* act as tests. The following test axioms account for their behaviour.

Test Axioms

$\{\top\}\ \bot\ \{\bot\}$.

$\langle\bot\rangle\ \bot\ \langle\bot\rangle$.

$\{R(t_1 \cdots t_n) \rightarrow \varphi\}\ R(t_1 \cdots t_n)\ \{\varphi\}$.

$\langle R(t_1 \cdots t_n) \wedge \varphi\rangle\ R(t_1 \cdots t_n)\ \langle\varphi\rangle$.

$\{t_1 = t_2 \rightarrow \varphi\}\ t_1 = t_2\ \{\varphi\}$.

$\langle t_1 = t_2 \wedge \varphi\rangle\ t_1 = t_2\ \langle\varphi\rangle$.

The axioms for the program $\bot$ express that $\bot$ always fails. The atomic predicates and the identities each have two axioms. In both cases the first (the axiom for universal correctness) gives the preconditions under which, if the program succeeds, all output states will satisfy φ. The second (the axiom for existential correctness) gives the preconditions which guarantee successful termination, with the postconditions guaranteed by the test.

For purposes of reasoning with the system one needs an oracle rule for the class $\mathcal{K}$ of models that one is interested in (for natural language applications such a class will generally be given by specifying a set of meaning postulates that all members of $\mathcal{K}$ should satisfy).

$\mathcal{K}$ Oracle Rule

Every assertion valid in $\mathcal{K}$ is an axiom.

The well-known consequence rule holds for universal and existential correctness.

Consequence Rules

$$\frac{\varphi \rightarrow \psi \qquad \{\psi\}\ \pi\ \{\chi\} \qquad \chi \rightarrow \xi}{\{\varphi\}\ \pi\ \{\xi\}.} \qquad \frac{\varphi \rightarrow \psi \qquad \langle\psi\rangle\ \pi\ \langle\chi\rangle \qquad \chi \rightarrow \xi}{\langle\varphi\rangle\ \pi\ \langle\xi\rangle.}$$

Next, one needs to specify the meanings of complex programs in the axiomatic framework.

Rules of Composition

$$\frac{\{\varphi\}\ \pi_1\ \{\psi\} \qquad \{\psi\}\ \pi_2\ \{\chi\}}{\{\varphi\}\ (\pi_1;\ \pi_2)\ \{\chi\}.} \qquad \frac{\langle\varphi\rangle\ \pi_1\ \langle\psi\rangle \qquad \langle\psi\rangle\ \pi_2\ \langle\chi\rangle}{\langle\varphi\rangle\ (\pi_1;\ \pi_2)\ \langle\chi\rangle.}$$

Rules of Negation

$$\frac{\{\varphi\}\ \pi\ \{\bot\}}{\langle\varphi \wedge \psi\rangle\ \neg\pi\ \langle\psi\rangle.} \qquad \frac{\langle\varphi\rangle\ \pi\ \langle\top\rangle}{\{\varphi \vee \psi\}\ \neg\pi\ \{\psi\}.}$$

Rules of Implication

$$\frac{\{\varphi\}\ \pi_1\ \{\psi\} \qquad \langle\psi\rangle\ \pi_2\ \langle\top\rangle.}{\langle\varphi \wedge \chi\rangle\ (\pi_1 \Rightarrow \pi_2)\ \langle\chi\rangle.} \qquad \frac{\langle\varphi\rangle\ \pi_1\ \langle\psi\rangle \qquad \{\psi\}\ \pi_2\ \{\bot\}.}{\{\varphi \vee \chi\}\ (\pi_1 \Rightarrow \pi_2)\ \{\chi\}.}$$

Rules of η Assignment

$$\frac{\{\varphi\}\ \pi\ \{\psi\}}{\{\forall x\varphi\}\ \eta x : \pi\ \{\psi\}.} \qquad \frac{\langle\varphi\rangle\ \pi\ \langle\psi\rangle}{\langle\exists x\varphi\rangle\ \eta x : \pi\ \langle\psi\rangle.}$$

In the rules of ι assignment it is convenient to use $\exists! x\varphi$ as an abbreviation for $\exists x \forall y([y/x]\varphi \leftrightarrow y = x)$, where y is a variable which is free for x in φ.

Rules of ι Assignment

$$\frac{\langle\varphi\rangle\ \pi\ \langle\psi\rangle \qquad \{\neg\varphi\}\ \pi\ \{\bot\}}{\langle\exists! x\varphi\rangle\ \iota x : \pi\ \langle\psi\rangle.} \qquad \frac{\langle\varphi\rangle\ \pi\ \langle\top\rangle \qquad \{\neg\varphi\}\ \pi\ \{\bot\} \qquad \{\psi\}\ \pi\ \{\chi\}}{\{\forall x(\forall y([y/x]\varphi \leftrightarrow y = x) \rightarrow \psi)\}\ \iota x : \pi\ \{\chi\}.}$$

Note that in the static description logic the η and ι operators from the dynamic assignment logic are contextually eliminated.

Rules of Quantification: Weak Readings

$$\frac{\langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\psi\rangle\ \pi_1;\ \pi_2\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_1;\ \pi_2\ \{\bot\}}{\langle Qx(\varphi, \psi) \wedge \chi\rangle\ Q^w x(\pi_1, \pi_2)\ \langle\chi\rangle.}$$

$$\frac{\{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_1;\ \pi_2\ \{\bot\} \qquad \langle\psi\rangle\ \pi_1;\ \pi_2\ \langle\top\rangle}{\{Qx(\varphi, \psi) \rightarrow \chi\}\ Q^w x(\pi_1, \pi_2)\ \{\chi\}.}$$

Rules of Quantification: Strong Readings

$$\frac{\langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\psi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_1 \Rightarrow \pi_2\ \{\bot\}}{\langle Qx(\varphi, \psi) \wedge \chi\rangle\ Q^s x(\pi_1, \pi_2)\ \langle\chi\rangle.}$$

$$\frac{\{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_1 \Rightarrow \pi_2\ \{\bot\} \qquad \langle\psi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\top\rangle}{\{Qx(\varphi, \psi) \rightarrow \chi\}\ Q^s x(\pi_1, \pi_2)\ \{\chi\}.}$$

In case we know the quantifier to be $\downarrow$MON, $\uparrow$MON, MON$\downarrow$ or MON$\uparrow$, then in the rules of quantification the first, second, third or fourth premiss, respectively, can be omitted.

Recall that a quantifier Q is $\downarrow$MON (downward monotone in its first argument) if Q is interpreted as a relation R between sets with the property that $R(S_1, S_2)$ and $S_1' \subseteq S_1$ imply $R(S_1', S_2)$. A quantifier Q is $\uparrow$MON (upward monotone in its first argument) if Q is interpreted as a relation R between sets with the property that $R(S_1, S_2)$ and $S_1 \subseteq S_1'$ imply $R(S_1', S_2)$. Similarly, MON$\downarrow$ or MON$\uparrow$ are used for monotonicity in the second argument. The binary universal quantifier $Q_\forall$, for example, is $\downarrow$MON and MON$\uparrow$, while the binary existential quantifier $Q_\exists$ is $\uparrow$MON and MON$\uparrow$.

Incidentally, the rules of quantification demonstrate that the snapshot language for static meaning (the assertion language) has to have essentially the same expressive power as the language for dynamic meaning (the *DAL* language), for we need a static counterpart for every dynamic generalized quantifier.

The above axioms and rules engender a notion of $\mathcal{K}$-derivation, as follows. A $\mathcal{K}$-derivation is a finite sequence of correctness formulae $F_1, \ldots, F_n$ such that for every i, $1 \leq i \leq n$, F_i is a test axiom or a an axiom according to the $\mathcal{K}$ oracle rule, or F_i is the conclusion of an instance of one of the inference rules while the premisses of that rule occur among $F_1, \ldots, F_{i-1}$. A $\mathcal{K}$-derivation $F_1, \ldots, F_n$ is said to be a $\mathcal{K}$-derivation of F_n. F is called *$\mathcal{K}$-derivable* in the proof system if there is a $\mathcal{K}$-derivation of F. Notation: $\mathcal{K} \vdash F$. In the next section, the soundness of this proof system relative to $\mathcal{K}$ will be proved.

7 Soundness of the Calculus

An inference from premisses $F_1, \ldots, F_n$ to conclusion F is called $\mathcal{K}$-valid if $\mathcal{K}$ validity of the premisses implies $\mathcal{K}$ validity of the conclusion. We will now show that the proof system given in the previous section is correct relative to $\mathcal{K}$, i.e. for every correctness statement F:

$\mathcal{K} \vdash F$ implies $\mathcal{K} \models F$.

To prove this, we first show that the axioms are $\mathcal{K}$-valid, and next that the inference rules preserve $\mathcal{K}$-validity. The soundness result then follows by induction on the length of derivations.

Test Axioms The soundness of the test axioms for $\perp$ is clear from the definition of K validity. For the universal axiom for $R(t_1 \cdots t_n)$ the reasoning is as follows. Assume a model and a state $\mathcal{M}, A$ such that (58).

(58) $\mathcal{M} \models R(t_1 \cdots t_n) \rightarrow \varphi[A]$.

There are two possibilities. If $\mathcal{M} \models \neg R(t_1 \cdots t_n)[A]$, then, by the semantic clause for $R(t_1 \cdots t_n)$, the output state set will be empty, so trivially every member of this set will satisfy φ. If $\mathcal{M} \models R(t_1 \cdots t_n)[A]$ then, by (58), $\mathcal{M} \models \varphi[A]$. In this case, by the semantic clause for $R(t_1 \cdots t_n)$, $[\![R(t_1 \cdots t_n)]\!]_{\mathcal{M}}(A) = \{A\}$, so again every member of the output set does satisfy φ.

For the existential axiom for $R(t_1 \cdots t_n)$, assume (59).

(59) $\mathcal{M} \models \varphi \wedge R(t_1 \cdots t_n)[A]$.

It follows immediately from the semantic clause for $R(t_1 \cdots t_n)$ that (60).

(60) $[\![R(t_1 \cdots t_n)]\!]_{\mathcal{M}}(A) = \{A\}$.

So there is a member of the output set which does satisfy φ. The reasoning for the universal and existential test axioms for $t_1 = t_2$ is similar.

$\mathcal{K}$ Oracle Rule The axioms generated by the $\mathcal{K}$ oracle rule are valid by definition.

Consequence Rules The soundness proof of the first consequence rule is standard (and trivial). We prove the soundness of the second consequence rule. Assume (61), (62) and (63).

(61) $\mathcal{K} \models \varphi \rightarrow \psi$.

(62) $\mathcal{K} \models \langle\psi\rangle\ \pi\ \langle\chi\rangle$.

(63) $\mathcal{K} \models \chi \rightarrow \xi$.

We have to show $\mathcal{K} \models \langle\varphi\rangle\ \pi\ \langle\xi\rangle$. Take a model $\mathcal{M} \in \mathcal{K}$ and a state A for $\mathcal{M}$ such that (64).

(64) $\mathcal{M} \models \varphi[A]$.

From (61) and (64) it follows that (65).

(65) $\mathcal{M} \models \psi[A]$.

From (65) and (62) it follows that there is a state $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \chi[B]$. From this and (63) it follows that there is a state $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \xi[B]$.

Rules of Composition For the first rule of composition, assume (66) and (67).

(66) $\mathcal{M} \models \{\varphi\}\ \pi_1\ \{\psi\}[A]$.

(67) $\mathcal{M} \models \{\psi\}\ \pi_2\ \{\chi\}[A]$.

What we have to show is $\mathcal{M} \models \{\varphi\}\ \pi_1;\ \pi_2\ \{\chi\}[A]$, so we assume $\mathcal{M} \models \varphi[A]$. Suppose some state $C \in [\![\pi_1;\ \pi_2]\!]_{\mathcal{M}}(A)$. Then there is some state $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ such that $C \in [\![\pi_1]\!]_{\mathcal{M}}(B)$. It follows from (66) and (67) that $\mathcal{M} \models \chi[C]$.

For the second rule of composition, assume (68) and (69).

(68) $\mathcal{M} \models \langle\varphi\rangle\ \pi_1\ \langle\psi\rangle[A]$.

(69) $\mathcal{M} \models \langle\psi\rangle\ \pi_2\ \langle\chi\rangle[A]$.

What we have to show is $\mathcal{M} \models \langle\varphi\rangle\ \pi_1;\ \pi_2\ \langle\chi\rangle[A]$, so we assume $\mathcal{M} \models \varphi[A]$. From (68) it follows that there is some state $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \psi[B]$. From (69) it follows that there is some state $C \in [\![\pi_2]\!]_{\mathcal{M}}(B)$ with $\mathcal{M} \models \chi[C]$, which is what we had to prove.

Rules of Negation If we consider $\neg\pi$ as an abbreviation for $\pi \Rightarrow \bot$, we can derive the rules of negation as follows. Assume (70).

(70) $\{\varphi\}\ \pi\ \{\bot\}$.

One of the test axioms for $\bot$ gives us (71).

(71) $\langle\bot\rangle\ \bot\ \langle\top\rangle$.

Application of the first rule of implication to (70) and (71) gives (72).

(72) $\langle\varphi \wedge \psi\rangle\ \pi \Rightarrow \bot\ \langle\psi\rangle$.

This takes care of the first rule of negation. For the second rule, assume (73).

(73) $\langle\varphi\rangle\ \pi\ \langle\top\rangle$.

One of the test axioms for $\bot$ gives (74).

(74) $\{\top\}\ \bot\ \{\bot\}$.

Applying the second rule of implication to (73) and (74) gives (75).

(75) $\{\neg\varphi \rightarrow \psi\}\ \pi \Rightarrow \bot\ \{\psi\}$.

This takes care of the second rule of negation.

Rules of Implication For the first rule of implication, assume (76), (77) and (78).

(76) $\mathcal{M} \models \varphi \wedge \chi[A]$.

(77) $\mathcal{M} \models \{\varphi\}\ \pi_1\ \{\psi\}[A]$.

(78) $\mathcal{M} \models \langle\psi\rangle\ \pi_2\ \langle\top\rangle[A]$.

We have to show (79).

(79) $\mathcal{M} \models \langle\varphi \wedge \chi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\chi\rangle[A]$.

To show (79), we have to establish the fact that there is some state $B \in [\![\pi_1 \Rightarrow \pi_2]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \varphi[B]$. In view of (76), we are done if we can show that $[\![\pi_1 \Rightarrow \pi_2]\!]_{\mathcal{M}}(A) = \{A\}$. For this it suffices, by the definition of the semantics for $\pi_1 \Rightarrow \pi_2$, to show that for all $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ it holds that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$. But this follows immediately from (77) and (78).

For the second rule of implication, assume (80), (81) and (82).

(80) $\mathcal{M} \models \varphi \vee \chi[A]$.

(81) $\mathcal{M} \models \langle\varphi\rangle\ \pi_1\ \langle\psi\rangle[A]$.

(82) $\mathcal{M} \models \{\psi\}\ \pi_2\ \{\bot\}[A]$.

We have to show (83).

(83) $\mathcal{M} \models \{\varphi \vee \chi\}\ \pi_1 \Rightarrow \pi_2\ \{\chi\}[A]$.

Because $[\![\pi_1 \Rightarrow \pi_2]\!]$ is a test, we are done if we dan show that if $[\![\pi_1 \Rightarrow \pi_2]\!]_\mathcal{M} \neq \emptyset$, then $\mathcal{M} \models \chi[A]$. This is the case iff either $[\![\pi_1 \Rightarrow \pi_2]\!]_\mathcal{M} = \emptyset$ or $\mathcal{M} \models \chi[A]$. In view of the semantic clause for $\Rightarrow$ this is the case if either there is some state $B \in [\![\pi_1]\!]_\mathcal{M}(A)$ with $[\![\pi_2]\!]_\mathcal{M}(B) = \emptyset$, or $\mathcal{M} \models \chi[A]$. From (80) it follows that either $\mathcal{M} \models \varphi[A]$ or $\mathcal{M} \models \chi[A]$. In the second case we are done. In the first case, it follows from (81) and (82) that there is indeed a state $B \in [\![\pi_1]\!]_\mathcal{M}(A)$ with $[\![\pi_2]\!]_\mathcal{M}(B) \neq \emptyset$, which is what was to be shown.

Rules of η Assignment For the first rule of η assignment, assume (84) and (85).

(84) $\mathcal{K} \models \{\varphi\}\ \pi\ \{\psi\}$.

(85) $\mathcal{M} \models \forall x \varphi[A]$.

We have to show (86).

(86) $\mathcal{M} \models \{\forall x \varphi\}\ \eta x : \pi\ \{\psi\}[A]$.

Assume that state $B \in [\![\eta x : \pi]\!]_\mathcal{M}(A)$. Then $B \in [\![\pi]\!]_\mathcal{M}(A[x := d])$ for some $d \in U$. It follows from (85) that $\mathcal{M} \models \varphi[A[x := d]]$, so we derive from (84) that $\mathcal{M} \models \psi[B]$.

For the second rule of η assignment, assume (87) and (88).

(87) $\mathcal{K} \models \langle\varphi\rangle\ \pi\ \langle\psi\rangle$.

(88) $\mathcal{M} \models \exists x \varphi[A]$.

We have to show (89).

(89) $\mathcal{M} \models \langle\exists x \varphi\rangle\ \eta x : \pi\ \langle\psi\rangle[A]$.

From (88) it follows that there is some $d \in U$ with $\mathcal{M} \models \varphi[A[x := d]]$, while from (87) it follows that there is some state $B \in [\![\pi]\!]_\mathcal{M}(A[x := d])$ with $\mathcal{M} \models \psi[B]$. By the semantic clause for η assignment it follows, then, that $B \in [\![\eta x : \pi]\!]_\mathcal{M}(A)$.

Rules of ι Assignment For the first rule of ι assignment, assume (90) and (91).

(90) $\mathcal{K} \models \langle\varphi\rangle\ \pi\ \langle\psi\rangle$.

(91) $\mathcal{K} \models \{\neg\varphi\}\ \pi\ \{\bot\}$.

Suppose (92).

(92) $\mathcal{M} \models \exists! x \varphi[A]$.

We have to show that there is some $B \in [\![\iota x : \pi]\!]_\mathcal{M}(A)$ with $\mathcal{M} \models \psi[B]$. From (92) it follows that there is a $d \in U$ such that (93), and for all $d' \in U, d' \neq d$, (94).

(93) $\mathcal{M} \models \varphi[A[x := d]]$.

(94) $\mathcal{M} \models \neg\varphi[A[x := d']]$.

Because of (90) and (93), $[\![\pi]\!]_\mathcal{M}(A[x := d]) \neq \emptyset$ and moreover, there is a $B \in [\![\pi]\!]_\mathcal{M}(A[x := d])$ with $\mathcal{M} \models \psi[B]$. Because of (91) and (94), $[\![\pi]\!]_\mathcal{M}(A[x := d']) = \emptyset$ for all $d' \neq d$. Thus, by the semantic clause for ι assignment, $[\![\iota x : \pi]\!]_\mathcal{M}(A) \neq \emptyset$ and moreover $B \in [\![\iota x : \pi]\!]_\mathcal{M}(A)$.

For the second rule of ι assignment, assume (95) and (96).

(95) $\mathcal{K} \models \langle\varphi\rangle\ \pi\ \langle\top\rangle$.

(96) $\mathcal{K} \models \{\neg\varphi\}\ \pi\ \{\bot\}$.

(97) $\mathcal{K} \models \{\psi\}\ \pi\ \{\chi\}$.

Suppose (98).

(98) $\mathcal{M} \models \forall x(\forall y([y/x]\varphi \leftrightarrow y = x) \rightarrow \psi)[A]$.

We have to show: if $B \in [\![\iota x : \pi]\!]_{\mathcal{M}}(A)$ then $\mathcal{M} \models \chi[B]$. From (98) we have for all $d \in U$:

(99) *If* $\mathcal{M} \models \forall y(([y/x]\varphi \leftrightarrow y = x) \rightarrow \psi)[A[x := d]]$, *then* $\mathcal{M} \models \chi[A[x := d]]$.

Suppose $B \in [\![\iota x : \pi]\!]_{\mathcal{M}}(A)$. Then we can derive two facts from this and the semantic clause for ι assignment, namely (100) and (101).

(100) *There is a unique* $d \in U$ *with* $[\![\pi]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$.

(101) *For the unique* $d \in U$ *satisfying (100) it holds that* $B \in [\![\pi]\!]_{\mathcal{M}}(A[x := d])$.

From (95) and (96) we derive (102):

(102) $[\![\pi]\!]_{\mathcal{M}}(A) \neq \emptyset$ *iff* $\mathcal{M} \models \varphi[A]$, *for all* $\mathcal{M}, A, f$.

From (102) and (100):

(103) *There is a unique* $d \in U$ *with* $\mathcal{M} \models \varphi[A[x := d]]$.

From (103):

(104) $\mathcal{M} \models \forall y([y/x]\varphi \leftrightarrow y = x)[A[x := d]]$.

From (104) and (99):

(105) $\mathcal{M} \models \psi[A[x := d]]$.

From (105), (101) and (97): $\mathcal{M} \models \chi[B]$.

Rules of Quantification We only treat the rules for weak readings, as the treatment of the rules for strong readings is completely similar. For the first rule of quantification, assume (106), (107), (108), and (109).

(106) $\mathcal{K} \models \langle\varphi\rangle\ \pi_1\ \langle\top\rangle$.

(107) $\mathcal{K} \models \{\neg\varphi\}\ \pi_1\ \{\bot\}$.

(108) $\mathcal{K} \models \langle\psi\rangle\ \pi_1;\ \pi_2\ \langle\top\rangle$.

(109) $\mathcal{K} \models \{\neg\psi\}\ \pi_1;\ \pi_2\ \{\bot\}$.

Suppose (110).

(110) $\mathcal{M} \models Qx(\varphi, \psi) \wedge \chi[A]$.

We have to establish (111).

(111) $\mathcal{M} \models \langle Qx(\varphi, \psi)\rangle\ Q^w x(\pi_1, \pi_2)\ \langle\chi\rangle[A]$.

We are done if we can show that $[\![Q^w x(\pi_1, \pi_2)]\!]_{\mathcal{M}} = \{A\}$. From (110) it follows that the sets $\{d \in U \mid \mathcal{M} \models \varphi[A[x := d]]\}$ (call this set S_1) and $\{d \in U \mid \mathcal{M} \models \psi[A[x := d]]\}$ (call this set S_2) are in the relation $I(Q)$ (where I is the interpretation function of $\mathcal{M}$). Suppose $\mathcal{M} \models \varphi[A[x := d]]$. Then it follows from (106) that $[\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$. Thus we have (112).

(112) $S_1 \subseteq \{d \mid [\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$.

Suppose conversely that $[\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$. Then it follows from (107) that $\mathcal{M} \not\models \neg\varphi[A[x := d]]$, and therefore that $\mathcal{M} \models \varphi[A[x := d]]$. Thus we have (113).

(113) $\{d \mid [\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\} \subseteq S_1$.

Similarly, we get from (108) that (114). and from (109) that (115).

(114) $S_2 \subseteq \{d \mid [\![\pi_1;\ \pi_2]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$.

(115) $\{d \mid [\![\pi_1;\ \pi_2]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\} \subseteq S_2$.

From (112) and (113) we have (116), and from (114) and (115) we have (117).

(116) $S_1 = \{d \mid [\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$.

(117) $S_2 = \{d \mid [\![\pi_1;\ \pi_2]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$.

It follows from (116), (117) and the semantic clause for quantifier programs that $[\![Q^w x(\pi_1, \pi_2)]\!]_{\mathcal{M}} = \{A\}$.

Now suppose Q is $\downarrow$MON. Then, with S_1, S_2 as above, $S_1' \subseteq S_1$ implies $\langle S_1', S_2\rangle \in I(Q)$. Thus, (113) and (117) are now enough to conclude that $[\![Q^w x(\pi_1, \pi_2)]\!]_{\mathcal{M}} = \{A\}$, and we do not need the first premiss. For the cases where Q is $\uparrow$MON, MON$\downarrow$ or MON$\uparrow$ the reasoning is similar.

For the second rule of quantification, again assume (106), (107), (108), and (109), but now suppose (118).

(118) $\mathcal{M} \models Qx(\varphi, \psi) \rightarrow \chi[A]$.

We can express (118) equivalently as (119).

(119) $\mathcal{M} \models \neg Qx(\varphi, \psi) \vee \chi[A]$.

It follows from (119) that either the sets S_1, S_2, defined as above, are *not* in relation $I(Q)$ or $\mathcal{M} \models \chi[A]$. By the same reasoning as above, the sets S_1, S_2 are in the relation $I(Q)$ iff program $Q^w x(\pi_1, \pi_2)$ accepts state A. Thus, by the semantic clause for quantifier programs, either $[\![Q^w x(\pi_1, \pi_2)]\!]_{\mathcal{M}}(A) = \emptyset$ or $\mathcal{M} \models \chi[A]$. This is what we had to show.

Again suppose Q is $\downarrow$MON. Then it follows from $\langle S_1, S_2\rangle \notin I(Q)$ and $S_1 \subseteq S_1'$ that $\langle S_1', S_2\rangle \notin I(Q)$. Thus we only need (112) to clinch the argument, and the premiss (107) is superfluous. Similarly for the other cases.

We have now proved the following theorem.

Theorem 3 (Soundness) *If* $\mathcal{K} \vdash F$ *then* $\mathcal{K} \models F$.

8 Completeness of the Calculus

Suppose we establish $\mathcal{K} \vdash \langle\varphi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\varphi\rangle$ for some φ with $\mathcal{K} \models \varphi$. Then it follows by the soundness of the calculus that $\mathcal{K} \models \langle\varphi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\varphi\rangle$, and by the $\mathcal{K}$ validity of φ that $\mathcal{M} \models \pi_1[A]$ implies $\mathcal{M} \models \pi_2[A]$, for all $\mathcal{M} \in \mathcal{K}$ and all states A for $\mathcal{M}$ defined for φ. In other words, the proof system can be considered as an axiomatisation of the notion of dynamic consequence, relative to classes of models $\mathcal{K}$. To see that the proof system is powerful enough we also have to establish its completeness relative to $\mathcal{K}$. For this we need the concepts of the weakest universal precondition and the weakest existential precondition of a *DAL* program and a formula of the assertion language.

The weakest universal precondition of a *DAL* program π and an L formula ψ is the L formula φ for which the following holds: $\mathcal{M} \models \varphi[A]$ iff for all $B \in [\![\pi]\!]_{\mathcal{M}}(A)$, it holds that $\mathcal{M} \models \psi[B]$ (for arbitrary $\mathcal{M}$). The weakest existential precondition of a *DAL* program π and an L formula ψ is the L formula φ for which the following holds: $\mathcal{M} \models \varphi[A]$ iff there is a $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \psi[B]$ (for arbitrary $\mathcal{M}$).

Note that it follows immediately from these definitions that the weakest universal precondition of a program π and an L formula ψ equals the negation of the weakest existential precondition of π and $\neg\psi$. This is because for all $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ it holds that $\mathcal{M} \models \psi[B]$ is equivalent to: there is no $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ for which $\mathcal{M} \models \neg\psi[B]$. This equivalence means that either of the two notions would suffice for what follows. For practical purposes, however, it is convenient to use both weakest universal and weakest existential preconditions, so we will define functions for both.

It is not obvious at first sight that the weakest universal and existential precondition of a *DAL* program and an L formula always exist (as formulas of L), so we have to show that this is indeed the case. We will inductively define functions $\mathbf{wup}(\pi, \psi)$ and $\mathbf{wep}(\pi, \psi)$ of which we will then show that they express the weakest universal precondition, respectively the weakest existential precondition of π and ψ.

$\mathbf{wup}(\bot, \psi) = \top$.
$\mathbf{wep}(\bot, \psi) = \bot$.
$\mathbf{wup}(R(t_1 \cdots t_n), \psi) = R(t_1 \cdots t_n) \rightarrow \psi$.
$\mathbf{wep}(R(t_1 \cdots t_n), \psi) = R(t_1 \cdots t_n) \wedge \psi$.

$\mathbf{wup}(t_1 = t_2), \psi) = t_1 = t_2 \rightarrow \psi.$
$\mathbf{wep}(t_1 = t_2), \psi) = t_1 = t_2 \wedge \psi.$
$\mathbf{wup}(\pi_1;\ \pi_2, \psi) = \mathbf{wup}(\pi_1, \mathbf{wup}(\pi_2, \psi)).$
$\mathbf{wep}(\pi_1;\ \pi_2, \psi) = \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \psi)).$
$\mathbf{wup}(\neg\pi, \psi) = \mathbf{wep}(\pi, \top) \vee \psi.$
$\mathbf{wep}(\neg\pi, \psi) = \mathbf{wup}(\pi, \bot) \wedge \psi.$
$\mathbf{wup}(\pi_1 \Rightarrow \pi_2, \psi) = \mathbf{wep}(\pi_1, \mathbf{wup}(\pi_2, \bot)) \vee \psi.$
$\mathbf{wep}(\pi_1 \Rightarrow \pi_2, \psi) = \mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top)) \wedge \psi.$
$\mathbf{wup}(\eta x : \pi, \psi) = \forall x \mathbf{wup}(\pi, \psi).$
$\mathbf{wep}(\eta x : \pi, \psi) = \exists x \mathbf{wep}(\pi, \psi).$
$\mathbf{wup}(\iota x : \pi, \psi) = \forall x(\forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x) \rightarrow \psi).$
$\mathbf{wep}(\iota x : \pi, \psi) = \exists x(\forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x) \wedge \psi).$
$\mathbf{wup}(Q^w x(\pi_1, \pi_2), \psi) = Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \top))) \rightarrow \psi.$
$\mathbf{wep}(Q^w x(\pi_1, \pi_2), \psi) = Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \top))) \wedge \psi.$
$\mathbf{wup}(Q^s x(\pi_1, \pi_2), \psi) = Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))) \rightarrow \psi.$
$\mathbf{wep}(Q^s x(\pi_1, \pi_2), \psi) = Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))) \wedge \psi.$

To show that these functions indeed give the weakest universal and existential preconditions of a program and a formula, a case by case check is necessary. The cases of atomic programs are checked directly, and induction is used to check the cases of complex programs.

Atomic Test Programs We only treat the case of $R(t_1 \cdots t_n)$, the cases of $\bot$ and $t_1 = t_2$ being similar. A state A for model $\mathcal{M}$ satisfies the weakest existential precondition of a test $R(t_1 \cdots t_n)$ and a formula ψ by definition of weakest existential precondition iff there is a state B with $B \in [\![R(t_1 \cdots t_n)]\!]_{\mathcal{M}}(A)$ and $\mathcal{M} \models \psi[B]$. By the semantic clause for atomic tests, this is the case iff $\mathcal{M} \models R(t_1 \cdots t_n)[A]$ and $\mathcal{M} \models \psi[A]$. This in turn is the case iff $\mathcal{M} \models R(t_1 \cdots t_n) \wedge \psi[A]$. Thus, the weakest existential precondition of $R(t_1 \cdots t_n)$ and ψ is $R(t_1 \cdots t_n) \wedge \psi$.

A state A for model $\mathcal{M}$ satisfies the weakest universal precondition of a test $R(t_1 \cdots t_n)$ and a formula ψ by definition of weakest universal precondition iff for every state B with $B \in [\![R(t_1 \cdots t_n)]\!]_{\mathcal{M}}(A)$ it holds that $\mathcal{M} \models \psi[B]$. By the semantic clause for atomic tests, this is the case iff it holds that if $\mathcal{M} \models R(t_1 \cdots t_n)[A]$ then $\mathcal{M} \models \psi[A]$. This in turn is the case iff $\mathcal{M} \models R(t_1 \cdots t_n) \rightarrow \psi[A]$. Thus, the weakest universal precondition of $R(t_1 \cdots t_n)$ and ψ is $R(t_1 \cdots t_n) \rightarrow \psi$.

Composition Left to the reader.

Negation A state A for model $\mathcal{M}$ satisfies the weakest existential precondition of a program $\neg\pi$ and a formula ψ by definition of weakest existential precondition iff there is a state B with $B \in [\![\neg\pi]\!]_{\mathcal{M}}(A)$ and $\mathcal{M} \models \psi[B]$. By the semantic clause for negation this is the case iff $[\![\pi]\!]_{\mathcal{M}}(A) = \emptyset$ and $\mathcal{M} \models \psi[A]$. By the induction hypothesis this is the case iff $\mathcal{M} \models \mathbf{wup}(\pi, \bot) \wedge \psi[A]$. Thus, the weakest existential precondition of $\neg\pi$ and ψ is $\mathbf{wup}(\pi, \bot) \wedge \psi$.

A state A for model $\mathcal{M}$ satisfies the weakest universal precondition of a program $\neg\pi$ and a formula ψ by definition of weakest universal precondition iff for all states B with $B \in [\![\neg\pi]\!]_{\mathcal{M}}(A)$ it holds that $\mathcal{M} \models \psi[B]$. By the semantic clause for negation this is the case iff it holds that either $[\![\pi]\!]_{\mathcal{M}}(A) = \emptyset$ or $\mathcal{M} \models \psi[A]$. By the induction hypothesis this is the case iff either $\mathcal{M} \models \mathbf{wep}(\pi, \top)[A]$ or $\mathcal{M} \models \psi[A]$. This is the case iff $\mathcal{M} \models \mathbf{wep}(\pi, \top) \vee \psi[A]$. Thus, the weakest universal precondition of $\neg\pi$ and ψ is $\mathbf{wep}(\pi, \top) \vee \psi$.

Implication A state A for model $\mathcal{M}$ satisfies the weakest existential precondition of a program $\pi_1 \Rightarrow \pi_2$ and a formula ψ by definition of weakest existential precondition iff there is a state B with $B \in [\![\pi_1 \Rightarrow \pi_2]\!]_{\mathcal{M}}(A)$ and $\mathcal{M} \models \psi[B]$. By the semantic clause for implication this is the case iff for all B with $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ it holds that $[\![\pi_2]\!]_{\mathcal{M}}(B) \neq \emptyset$, and $\mathcal{M} \models \psi[A]$. By the induction hypothesis this is the case iff $\mathcal{M} \models \mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))[A]$ and $\mathcal{M} \models \psi[A]$. This is the case iff $\mathcal{M} \models \mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top)) \wedge \psi[A]$. Thus, the weakest existential precondition of $\pi_1 \Rightarrow \pi_2$ and ψ is $\mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top)) \wedge \psi$.

A state A for model $\mathcal{M}$ satisfies the weakest universal precondition of a program $\pi_1 \Rightarrow \pi_2$ and a formula ψ by definition of weakest universal precondition iff for all states B with $B \in [\![\pi_1 \Rightarrow \pi_2]\!]_{\mathcal{M}}(A)$ it holds that $\mathcal{M} \models \psi[B]$. By the semantic clause for implication this is the case iff either there is some state B with $B \in [\![\pi_1]\!]_{\mathcal{M}}(A)$ and $[\![\pi_2]\!]_{\mathcal{M}}(B) = \emptyset$, or $\mathcal{M} \models \psi[A]$. By the induction hypothesis this is the case iff either $\mathcal{M} \models \mathbf{wep}(\pi_1, \mathbf{wup}(\pi_2, \bot))[A]$ or $\mathcal{M} \models \psi[A]$. This is the case iff $\mathcal{M} \models \mathbf{wep}(\pi_1, \mathbf{wup}(\pi_2, \bot)) \vee \psi[A]$. Thus, the weakest universal precondition of $\pi_1 \Rightarrow \pi_2$ and ψ is $\mathbf{wep}(\pi_1, \mathbf{wup}(\pi_2, \bot)) \vee \psi$.

η **Assignment** Left to the reader.

ι **Assignment** A state A for model $\mathcal{M}$ satisfies the weakest existential precondition of a program $\iota x : \pi$ and a formula ψ by definition of weakest existential precondition iff there is a state B with $B \in [\![\iota x : \pi]\!]_{\mathcal{M}}(A)$ and $\mathcal{M} \models \psi[B]$. By the semantic clause for ι assignment, this is the case iff there is a unique d for which $[\![\pi]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$, and moreover for this particular d there is a state B with $B \in [\![\pi]\!]_{\mathcal{M}}(A[x := d])$ and $\mathcal{M} \models \psi[B]$. By the induction hypothesis this is the case iff for some x, both $\mathcal{M} \models \forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x)[A]$ and $\mathcal{M} \models \mathbf{wep}(\pi, \psi)[A]$. This is the case iff (120).

(120) $\quad \mathcal{M} \models \exists x(\forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x) \wedge \mathbf{wep}(\pi, \psi))[A]$.

Because in any model, for any state, and for any ψ, $\mathbf{wep}(\pi, \psi)$ implies $\mathbf{wep}(\pi, \top)$, we see that (120) is equivalent to (121).

(121) $\quad \mathcal{M} \models \exists x(\forall y([y/x]\mathbf{wep}(\pi, \psi) \leftrightarrow y = x))[A]$.

Thus, the weakest existential precondition of $\iota x : \pi$ and ψ is $\exists! x : \mathbf{wep}(\pi, \psi)$.

A state A for model $\mathcal{M}$ satisfies the weakest universal precondition of a program $\iota x : \pi$ and a formula ψ by definition of weakest universal precondition iff for all states B with $B \in [\![\iota x : \pi]\!]_{\mathcal{M}}(A)$ it holds that $\mathcal{M} \models \psi[B]$. By the semantic clause for ι assignment, this is the case iff it holds that if there is a unique d with $[\![\pi]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset$, then any state B with $B \in [\![\pi]\!]_{\mathcal{M}}(A[x := d])$ will have $\mathcal{M} \models \psi[B]$. By the induction hypothesis this is the case iff for any x, $\mathcal{M} \models \forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x)[A]$ implies $\mathcal{M} \models \mathbf{wup}(\pi, \psi)$. This is the case iff $\mathcal{M} \models \forall x(\forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x) \rightarrow \mathbf{wup}(\pi, \psi))[A]$. Thus, the weakest universal precondition of $\iota x : \pi$ and ψ is $\forall x(\forall y([y/x]\mathbf{wep}(\pi, \top) \leftrightarrow y = x) \rightarrow \mathbf{wup}(\pi, \psi))$.

Quantification We only give the reasoning for the weakest existential precondition of weak readings. A state A for model $\mathcal{M}$ satisfies the weakest existential precondition of a program $Q^w x(\pi_1, \pi_2)$ and a formula ψ by definition of weakest existential precondition iff there is a state B with $B \in [\![Q^w x(\pi_1, \pi_2)]\!]_{\mathcal{M}}(A)$ and $\mathcal{M} \models \psi[B]$. By the semantic clause for quantification, this is the case iff the sets given by (122) and (123) are in the relation $I(Q)$ and moreover $\mathcal{M} \models \psi[A]$.

(122) $\quad \{d \mid [\![\pi_1]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$

(123) $\quad \{d \mid [\![\pi_1;\ \pi_2]\!]_{\mathcal{M}}(A[x := d]) \neq \emptyset\}$

By the induction hypothesis and the semantic rule for ; , this is the case iff the sets given by (124) and (125) are in relation $I(Q)$, and moreover $\mathcal{M} \models \psi[A]$.

(124) $\quad \{d \mid \mathcal{M} \models \mathbf{wep}(\pi_1, \top)[A]\}$

(125) $\quad \{d \mid \mathcal{M} \models \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \top))[A]\}$

By the semantic properties of Q, this in turn is the case iff (126).

(126) $\quad \mathcal{M} \models Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \top))) \wedge \psi$.

Thus, the weakest existential precondition of $Q^w x(\pi_1, \pi_2)$ and ψ is $Qx(\mathbf{wep}(\pi_1, \top), \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \top))) \wedge \psi$.

This completes the proof that the definitions of **wep** and **wup** are indeed adequate, in other words we have established the following.

Lemma 4 (wep/wup adequacy) *For all $\mathcal{M} \in \mathcal{K}$, all states A for $\mathcal{M}$, all $\psi \in L$, and all $\pi \in DAL$:*

$\mathcal{M} \models \mathbf{wup}(\pi, \psi)[A]$ *iff it holds for all $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ that $\mathcal{M} \models \psi[B]$.*

$\mathcal{M} \models \mathbf{wep}(\pi, \psi)[A]$ *iff there is a $B \in [\![\pi]\!]_{\mathcal{M}}(A)$ with $\mathcal{M} \models \psi[B]$.*

Next, we show that for all $\pi \in DAL$ and for all $\psi \in L$, $\mathcal{K} \vdash \langle \mathbf{wep}(\pi, \psi) \rangle\ \pi\ \langle \psi \rangle$ and $\mathcal{K} \vdash \{\mathbf{wup}(\pi, \psi)\}\ \pi\ \{\psi\}$. Again, we proceed by way of an induction argument.

Atomic Test Programs We only prove the result for the case of $R(t_1 \cdots t_n)$, the other atomic cases being similar. Assume $\mathcal{K} \models \langle \mathbf{wep}(R(t_1 \cdots t_n), \psi) \rangle \ R(t_1 \cdots t_n) \ \langle \psi \rangle$. By the definition of **wep** this is equivalent to $\mathcal{K} \models \langle R(t_1 \cdots t_n) \wedge \psi \rangle \ R(t_1 \cdots t_n) \ \langle \psi \rangle$. By the existential test axiom, (127).

(127) $\quad \mathcal{K} \vdash \langle R(t_1 \cdots t_n) \wedge \psi \rangle \ R(t_1 \cdots t_n) \ \langle \psi \rangle$.

For the universal case, assume $\mathcal{K} \models \{\mathbf{wup}(R(t_1 \cdots t_n), \psi)\} \ R(t_1 \cdots t_n) \ \{\psi\}$. By the definition of **wup**, this is equivalent to $\mathcal{K} \models \{R(t_1 \cdots t_n) \rightarrow \psi\} \ R(t_1 \cdots t_n) \ \{\psi\}$. By one of the test axioms, (128).

(128) $\quad \mathcal{K} \vdash \{R(t_1 \cdots t_n) \rightarrow \psi\} \ R(t_1 \cdots t_n) \ \{\psi\}$.

Composition Assume $\mathcal{K} \models \langle \mathbf{wep}(\pi_1;\ \pi_2, \psi) \rangle \ \pi_1;\ \pi_2 \ \langle \psi \rangle$. By the definition of **wep**, this is equivalent to $\mathcal{K} \models \langle \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \psi)) \rangle \ \pi_1;\ \pi_2 \ \langle \psi \rangle$. By the **wep** adequacy lemma, (129) and (130).

(129) $\quad \mathcal{K} \models \langle \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \psi)) \rangle \ \pi_1 \ \langle \mathbf{wep}(\pi_2, \psi) \rangle$.

(130) $\quad \mathcal{K} \models \langle \mathbf{wep}(\pi_2, \psi) \rangle \ \pi_2 \ \langle \psi \rangle$.

By the induction hypothesis, (129) and (130) yield (131) and (132).

(131) $\quad \mathcal{K} \vdash \langle \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \psi)) \rangle \ \pi_1 \ \langle \mathbf{wep}(\pi_2, \psi) \rangle$.

(132) $\quad \mathcal{K} \vdash \langle \mathbf{wep}(\pi_2, \psi) \rangle \ \pi_2 \ \langle \psi \rangle$.

Applying the existential composition rule to (131) and (132) gives (133).

(133) $\quad \mathcal{K} \vdash \langle \mathbf{wep}(\pi_1, \mathbf{wep}(\pi_2, \psi)) \rangle \ \pi_1;\ \pi_2 \ \langle \psi \rangle$.

The reasoning for universal assertions about weakest universal preconditions of programs of the form $\pi_1;\ \pi_2$ is similar.

Negation Assume $\mathcal{K} \models \langle \mathbf{wep}(\neg\pi, \psi) \rangle \ \neg\pi \ \langle \psi \rangle$. By the definition of **wep**, this is equivalent to $\mathcal{K} \models \langle \mathbf{wup}(\pi, \bot) \wedge \psi \rangle \ \neg\pi \ \langle \psi \rangle$. By the **wup** adequacy lemma, (134)

(134) $\quad \mathcal{K} \models \{\mathbf{wup}(\pi, \bot)\} \ \pi \ \{\bot\}$.

By the induction hypothesis, (134) yields (135).

(135) $\quad \mathcal{K} \vdash \{\mathbf{wup}(\pi, \bot)\} \ \pi \ \{\bot\}$.

An application of one of the negation rules to (135) gives (136).

(136) $\quad \mathcal{K} \vdash \langle \mathbf{wup}(\pi, \bot) \wedge \psi \rangle \ \neg\pi \ \langle \psi \rangle$.

The reasoning for universal assertions about weakest universal preconditions for programs of the form $\neg\pi$ is similar.

Implication Assume $\mathcal{K} \models \langle \mathbf{wep}(\pi_1 \Rightarrow \pi_2, \psi) \rangle \ \pi_1 \Rightarrow \pi_2 \ \langle \psi \rangle$. By the definition of **wep**, this is equivalent to $\mathcal{K} \models \langle (\mathbf{wup}(\pi, \mathbf{wep}(\pi_2, \top)) \wedge \psi) \rangle \ \pi_1 \Rightarrow \pi_2 \ \langle \psi \rangle$. By the **wup** and **wep** adequacy lemma, (137) and (138).

(137) $\quad \mathcal{K} \models \{\mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))\} \ \pi_1 \ \{\mathbf{wep}(\pi_2, \top)\}$.

(138) $\quad \mathcal{K} \models \langle \mathbf{wep}(\pi_2, \top) \rangle \ \pi_2 \ \langle \top \rangle$.

By the induction hypothesis, (137) yields (139) and (138) yields (140).

(139) $\quad \mathcal{K} \vdash \{\mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))\} \ \pi_1 \ \{\mathbf{wep}(\pi_2, \top)\}$.

(140) $\quad \mathcal{K} \vdash \langle \mathbf{wep}(\pi_2, \top) \rangle \ \pi_2 \ \langle \top \rangle$.

Now apply the first rule of implication to (139), (140):

(141) $\quad \mathcal{K} \vdash \{\mathbf{wup}(\pi_1, \mathbf{wep}(\pi_2, \top))\} \ \pi_1 \Rightarrow \pi_2 \ \{\psi\}$.

The reasoning for universal assertions about weakest universal preconditions for programs of the form $\pi_1 \Rightarrow \pi_2$ is similar.

η **Assignment** Assume $\mathcal{K} \models \langle \mathbf{wep}(\eta x : \pi, \psi) \rangle\ \eta x : \pi\ \langle \psi \rangle$. By the definition of **wep** this is equivalent to $\mathcal{K} \models \langle \exists x \mathbf{wep}(\pi, \psi) \rangle\ \eta x : \pi\ \langle \psi \rangle$. By the **wep** adequacy lemma, (142).

(142) $\mathcal{K} \models \langle \mathbf{wep}(\pi, \psi) \rangle\ \pi\ \langle \psi \rangle$.

By the induction hypothesis, (142) yields (143).

(143) $\mathcal{K} \vdash \langle \mathbf{wep}(\pi, \psi) \rangle\ \pi\ \langle \psi \rangle$.

An application of the existential η assignment rule to (143) gives (144).

(144) $\mathcal{K} \vdash \langle \exists x \mathbf{wep}(\pi, \psi) \rangle\ \eta x : \pi\ \langle \psi \rangle$.

The reasoning for universal assertions about weakest universal preconditions for programs of the form $\eta x : \pi$ is similar.

ι **Assignment and Quantification** Left to the reader.

We have now established the following lemma.

Lemma 5 (wep/wup derivability) *For all $\pi \in DAL$ and for all $\psi \in L$,*

$$\mathcal{K} \vdash \langle \mathbf{wep}(\pi, \psi) \rangle\ \pi\ \langle \psi \rangle \textit{ and } \mathcal{K} \vdash \{ \mathbf{wup}(\pi, \psi) \}\ \pi\ \{ \psi \}.$$

The rest of the proof of the completeness result is very easy. We want to show that $\mathcal{K} \models F$ implies $\mathcal{K} \vdash F$. In case F equals φ for some $\varphi \in L$, $\mathcal{K} \models \varphi$ implies $\mathcal{K} \vdash \varphi$ by the $\mathcal{K}$ oracle rule. For the case where F equals $\langle \varphi \rangle\ \pi\ \langle \psi \rangle$ the reasoning is as follows. Assume (145).

(145) $\mathcal{K} \models \langle \varphi \rangle\ \pi\ \langle \psi \rangle$.

By the **wep** adequacy lemma it follows from (145) that (146).

(146) $\mathcal{K} \models \varphi \to \mathbf{wep}(\pi, \psi)$.

From this, by the $\mathcal{K}$ oracle rule, (147).

(147) $\mathcal{K} \vdash \varphi \to \mathbf{wep}(\pi, \psi)$.

From the wep/wup derivability lemma we have (148).

(148) $\mathcal{K} \vdash \langle \mathbf{wep}(\pi, \psi) \rangle\ \pi\ \langle \psi \rangle$.

From (147) and (148) by an application of the existential consequence rule, (149).

(149) $\mathcal{K} \vdash \langle \varphi \rangle\ \pi\ \langle \psi \rangle$.

By similar reasoning we derive from (150) that (151).

(150) $\mathcal{K} \models \{ \varphi \}\ \pi\ \{ \psi \}$.
(151) $\mathcal{K} \vdash \{ \varphi \}\ \pi\ \{ \psi \}$.

This completes the proof of the final result of this section.

Theorem 6 (Completeness) *If $\mathcal{K} \models F$ then $\mathcal{K} \vdash F$.*

9 Use of the Calculus

The discussion of weakest preconditions in Section 8 will have made clear that the rules of the calculus are given in a format particularly suited to the calculation of weakest universal and existential preconditions. To find the weakest existential precondition of a program π and a formula ψ, proceed as follows. Start with the conclusion $\langle ? \rangle\ \pi\ \langle \top \rangle$, and apply the rules of the calculus working backwards, thus decomposing π. This will eventually produce a formula φ to fill the ? slot. Thus, as explained in Section 6, the calculus allows us to calculate the static meanings of (the *DAL* translations of) natural language example sentences. As a first example, we calculate the static meaning of the conditional donkey sentence given in (5), with *DAL* translation (7), which is repeated here for convenience.

(152) $(\eta x : Gx;\ \eta y : By;\ Hxy) \Rightarrow Txy.$

We want to find the weakest precondition φ such that (153).

(153) $\langle\varphi\rangle\ (\eta x : Gx;\ \eta y : By;\ Hxy) \Rightarrow Txy\ \langle\top\rangle.$

According to the rule for implication, we are done if we can find the weakest ψ such that $\langle\psi\rangle\ Txy\ \langle\top\rangle$ and then calculate the weakest φ such that (154).

(154) $\{\varphi\}\ \eta x : Gx;\ \eta y : By;\ Hxy\ \{\psi\}.$

It follows from one of the test axioms plus the existential consequence rule that ψ equals Txy. Thus, we are done if we can find the weakest φ such that (155).

(155) $\{\varphi\}\ \eta x : Gx;\ \eta y : By;\ Hxy\ \{Txy\}.$

Two applications of the universal rule for composition and two applications of the universal rule for η assignment give the end result.

(156) $\{\forall x(Gx \rightarrow \forall y(By \rightarrow (Hxy \rightarrow Txy)))\}\ \eta x : Gx;\ \eta y : By;\ Hxy\ \{Txy\}.$

From the fact that the procedure calculates the weakest precondition under which (152) can succeed it follows that $\forall x(Gx \rightarrow \forall y(By \rightarrow (Hxy \rightarrow Txy)))$ is the static meaning of this program.

Next we show that $\Rightarrow$ can be defined in terms of ; and $\neg$, by deriving two rules for $\neg(\pi_1;\ \neg\pi_2)$ which are identical to the rules for dynamic implication. To derive the first rule of implication, assume (157) and (158).

(157) $\{\varphi\}\ \pi_1\ \{\psi\}.$

(158) $\langle\psi\rangle\ \pi_2\ \langle\top\rangle.$

From (158) we get (159) by the second negation rule.

(159) $\{\psi\}\ \neg\pi_2\ \{\bot\}.$

Next, applying the first composition rule to (157) and (159) gives (160).

(160) $\{\varphi\}\ \pi_1;\ \neg\pi_2\ \{\bot\}.$

An application of the first negation rule to (160) gives the desired result.

(161) $\langle\varphi \wedge \chi\rangle\ \neg(\pi_1;\ \neg\pi_2)\ \langle\chi\rangle.$

For the second rule of implication, start with (162) and (163).

(162) $\langle\varphi\rangle\ \pi_1\ \langle\psi\rangle.$

(163) $\{\psi\}\ \pi_2\ \{\bot\}.$

Applying the first rule of negation to (163) gives (164).

(164) $\langle\psi\rangle\ \neg\pi_2\ \langle\psi\rangle.$

The second consequence rule allows us to derive (165) from (164).

(165) $\langle\psi\rangle\ \neg\pi_2\ \langle\top\rangle.$

The second rule of composition can be applied to (162) and (165), giving (166).

(166) $\langle\varphi\rangle\ \pi_1;\ \neg\pi_2\ \langle\varphi\rangle.$

The second rule of consequence yields (167).

(167) $\langle\varphi\rangle\ \pi_1;\ \neg\pi_2\ \langle\top\rangle.$

The second negation rule gives the desired result.

(168) $\{\varphi \vee \chi\} \neg(\pi_1; \neg\pi_2) \{\chi\}$.

Next, we derive the static meaning of (35) (under its weak reading), the translation of which is repeated here for convenience.

(169) $\mathbf{most}^w\ v_1(Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2, Tv_1v_2)$.

We want to find the weakest precondition φ for which (170).

(170) $\langle\varphi\rangle\ \mathbf{most}^w\ v_1(Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2, Tv_1v_2)\ \langle\top\rangle$.

In view of the first quantifier rule, we know that φ equals $\mathbf{most}^w\ v_1(\psi, \chi)$, where ψ and χ are given by (171), (172), (173), and (174).

(171) $\langle\psi\rangle\ Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2\ \langle\top\rangle$.

(172) $\{\neg\psi\}\ Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2\ \{\bot\}$.

(173) $\langle\chi\rangle\ Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2;\ Tv_1v_2\ \langle\top\rangle$

(174) $\{\neg\chi\}\ Gv_1;\ \eta v_2 : Bv_2;\ Hv_1v_2;\ Tv_1v_2\ \{\bot\}$.

Note that (172) and (174) are only there to guarantee that ψ and χ are *weakest* existential preconditions of the given programs with respect to $\top$. The quantifier rule says that the rule in this case would still hold if we omit (174) (in virtue of the fact that **most** is MON$\uparrow$), but of course then there is no guarantee anymore that $\mathbf{most}\ v_1(\psi, \chi)$ expresses the *weakest* existential precondition of (169) with respect to $\top$. However, if we take care not to use the consequence rules we calculate weakest preconditions anyway, so then we can omit (172) and (174) and still arrive at the weakest existential precondition of (169) with respect to $\top$.

Application of the rules for η assignment and composition give the following value for ψ:

(175) $Gv_1 \wedge \exists v_2(Bv_2 \wedge Hv_1v_2)$.

(176) $Gv_1 \wedge \exists v_2(Bv_2 \wedge Hv_1v_2 \wedge Tv_1v_2)$.

Thus, we arrive at the following static meaning for (169):

(177) $\mathbf{most}\ v_1(Gv_1 \wedge \exists v_2(Bv_2 \wedge Hv_1v_2), Gv_1 \wedge \exists v_2(Bv_2 \wedge Hv_1v_2 \wedge Tv_1v_2))$.

One final remark on the fact that our calculus is geared to finding preconditions, given a program and an output condition. We hope to have demonstrated the usefulness of this in the above examples. However, one might also be interested in calculating postconditions (which of course is possible with the calculus). It is straightforward to check the following. For a program π which is a test, calculating the weakest existential precondition of π with respect to $\top$ is equivalent to calculating the strongest universal postcondition of π with respect to $\top$. For programs which are not tests, this equivalence breaks down, but in such cases there is a different reason for being interested in postconditions. The strongest universal postcondition with respect to $\top$ of a program π which is not a test will have free occurrences of precisely those variables that are available for external dynamic binding in programs π' following π.

10 Dynamic Logic and (Non) Monotonicity

It is possible to characterize the difference between the languages *DAL* and DAL_0 in terms of the axioms of our calculus, by adding the following rule to the calculus to make it applicable to DAL_0 programs.

Monotonicity Rule

$$\frac{\{\varphi\}\ \pi_1\ \{\psi\}}{\{\varphi\}\ \pi_1;\ \pi_2\ \{\psi\}}, \text{ provided } \mathbf{fv}(\varphi) = \emptyset.$$

This rule can be proved sound by induction on the complexity of π_2. In the basic case π_2 has the form of a basic test, let us say π_2 equals $R(t_1 \cdots t_n)$. Then certainly one can safely conclude from $\{\varphi\}\ \pi_1\ \{\psi\}$ to $\{\varphi\}\ \pi_1;\ R(t_1 \cdots t_n)\ \{\psi\}$, because of the fact that $R(t_1 \cdots t_n)$ is a test and the fact that the conclusion is a conditional statement.

In the induction step, all DAL_0 constructs which form test programs pose no difficulty, again because of the fact that the conclusion of the Monotonicity Rule is a conditional statement. Thus, the only programming constructs that could cause trouble are sequential composition and η and ι assignment.

Assume that π_2 has the form $\pi'; \pi''$. Then we can use the induction hypothesis to conclude from $\{\varphi\}\ \pi_1\ \{\psi\}$ to $\{\varphi\}\ \pi_1;\ \pi'\ \{\psi\}$, and use the induction hypothesis once more to conclude from this to $\{\varphi\}\ \pi_1;\ \pi';\ \pi''\ \{\psi\}$.

Assume that π_2 has the form $\eta x : \pi'$. Also assume $\{\varphi\}\ \pi_1\ \{\psi\}$, with $\mathbf{fv}(\varphi) = \emptyset$. We have to show that one can safely conclude $\{\varphi\}\ \pi_1;\ \eta x : \pi'\ \{\psi\}$. First observe that the induction hypothesis gives us: from $\{\varphi\}\ \pi_1\ \{\psi\}$ one can conclude $\{\varphi\}\ \pi_1;\ \pi'\ \{\psi\}$. Thus, there is some χ with $\{\varphi\}\ \pi_1\ \{\chi\}$ and $\{\chi\}\ \pi'\ \{\psi\}$. Moreover, since $\mathbf{fv}(\varphi) = \emptyset$, we may assume $\mathbf{fv}(\chi) \subseteq \mathbf{av}(\pi_1)$ (use a simple induction argument). Now it follows from the fact that program $\pi_1;\ \eta x : \pi'$ is a DAL_0 program that $x \notin \mathbf{fv}(\chi)$. Thus, $\exists x \chi$ is equivalent to χ, as the quantification over x is vacuous. Thus we can conclude from $\{\varphi\}\ \pi_1\ \{\chi\}$ to $\{\varphi\}\ \pi_1\ \{\exists x \chi\}$. From $\{\chi\}\ \pi'\ \{\psi\}$, derive with the universal rule for η assignment that $\{\exists x \chi\}\ \pi'\ \{\psi\}$. Then the universal composition rule allows us to conclude that $\{\varphi\}\ \pi_1;\ \eta x \pi'\ \{\psi\}$. The reasoning for the case of ι assignment is similar. This completes the proof of the soundness of the Monotonicity Rule for DAL_0 programs.

We can use the monotonicity rule to infer from (178) that (179).

(178) $\langle\varphi\rangle\ \pi_1 \Rightarrow \pi_2\ \langle\top\rangle$.

(179) $\langle\varphi\rangle\ (\pi_1;\ \pi_1') \Rightarrow \pi_2\ \langle\top\rangle$.

Conditions: the precondition formula φ should not have any free variables, and the programs in (178) and (179) should be DAL_0 programs.

Since $\mathcal{M} \models \pi_1 \Rightarrow \pi_2[A]$ iff it holds that if $\mathcal{M} \models \pi_1[A]$ then $\mathcal{M} \models \pi_2[A]$, this shows that dynamic consequence for DAL_0 programs is monotonic (provided no program variable is used without previous assignment). It should be noted that all DAL translations of natural language examples in this paper are indeed DAL_0 programs where no program variables are used without previous assignment.

11 Conclusion

In this paper we have demonstrated the potential of the use of tools from programming language semantics for the semantics of natural language. While η and ι assignment can in principle be decomposed in random assignment with subsequent testing, we have two reasons for preferring the treatment we gave. In the first place, extending the treatment of definite descriptions with an account of their presuppositions (which is an obvious next move in the framework we have presented) would make a decomposition of ι assignment impossible or at least very impractical. Secondly, and more importantly, ι and η assignments are to be preferred over a decomposition in terms of random assignment plus subsequent testing because the introduction of an individual by a definite or an indefinite noun phrase gets an exact counterpart in the dynamic translation language. In other words, the real merit of ι and η assignments is that they allow faithfulness to linguistic form.

Instead of the universal and existential Hoare-style correctness statements that we employed we might have used the toolkit of dynamic logic (cf. [8]). Replacing $\{\varphi\}\ \pi\ \{\psi\}$ by $\varphi \rightarrow [\pi]\psi$ and $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ by $\varphi \rightarrow \langle\pi\rangle\psi$ is all there is to such a change. Still, we prefer our notation, for several reasons. In the first place, it is less cluttered than the dynamic logic notation. Next, the full expressive power of dynamic logic is not needed for our purposes, so it seems wiser to choose a tool that fits the requirements more precisely. Finally, the static $\{\varphi\}$ and $\langle\varphi\rangle$ statements can be used as comments to annotate DAL programs, thus providing proof outlines for deriving the static meanings of programs.

A set-up where quantifiers, negations, and implications act as tests, and where descriptions are virtually the only externally dynamic elements, will be very strict about which anaphoric links are possible. The framework as it has been set up above does correctly predict that the anaphoric links suggested by the indices in (180) are out.

(180) **No boy*^{1} *owned a car*^{2}. *He*$_1$ *washed it*$_2$.

The reason for this is our assumption that quantifiers act externally as tests. The binary quantifier that translates *no* acts externally as a test, so after the processing of the first sentence of discourse (180) we are back with the initial state, and the two pronouns in the second sentence will not get proper referents.

It is well-known, however, that relaxations are necessary to deal with the external dynamic effects of symmetric quantifiers and with various cases of so-called subordination (see for example [12] and [9]). Our next self-imposed task is to extend and modify the framework to deal with such cases.

Acknowledgement

This paper has benefitted from helpful comments by Krzysztof Apt, Reinhard Muskens, Martin Stokhof, Johan van Benthem and Albert Visser.

References

[1] K.R. Apt, 'Ten Years of Hoare's Logic: A Survey—Part I', *ACM Transactions on Programming Languages and Systems*, Vol. 3, No. 4, October 1981, 431–483.

[2] J. Barwise, 'Noun Phrases, Generalized Quantifiers and Anaphora', in P. Gärdenfors (ed.), *Generalized Quantifiers / Linguistic and Logical Approaches*, 1–29, Reidel, Dordrecht, 1987.

[3] J. van Benthem, 'General Dynamics', ITLI report, Amsterdam, 1990 (to appear in *Theoretical Linguistics*).

[4] Gennaro Chierchia, 'Anaphora and Dynamic Logic', in M. Stokhof, J. Groenendijk & D. Beaver (eds.), *Quantification and Anaphora I*, Edinburgh 1991 (DYANA deliverable R2.2.A), 37–78.

[5] G. Evans, 'Pronouns', in: *Linguistic Inquiry*, 11, 337-362.

[6] P.T. Geach, *Reference and Generality / An Examination of Some Medieval and Modern Theories* , Cornell University Press, Ithaca & London, 1962 (Third revised edition: 1980)

[7] J. Groenendijk & M. Stokhof, 'Dynamic Predicate Logic', *Linguistics and Philosophy*, 14, 1991, 39–100.

[8] D. Harel, 'Dynamic Logic', in D. Gabbay & F. Guenthner, *Handbook of Philosophical Logic, Vol. II*, Reidel, Dordrecht, 1984, 497–604.

[9] I. Heim, 'E-Type Pronouns and Donkey Anaphora', *Linguistics and Philosophy*, 13, 1990, 137–177.

[10] H. Kamp, 'A Theory of Truth and Semantic Representation', in Groenendijk e.a. (eds.), *Formal Methods in the Study of Language*, Mathematisch Centrum, Amsterdam, 1981.

[11] H. Kamp & U. Reyle, *From Discourse to Logic*, Manuscript, Institute for Computational Linguistics, University of Stuttgart, 1990.

[12] C. Roberts, 'Modal Subordination and Pronominal Anaphora in Discourse', *Linguistics and Philosophy*, 12, 1989, 683–721.

[13] B. Russell, 'On Denoting', *Mind*, 14, 1905, 479–493.

[14] D. Westerståhl, 'Quantifiers in Formal and Natural Languages', in Gabbay & Guenthner (eds.), *Handbook of Philosophical Logic*, Vol. IV, Reidel, Dordrecht, 1989, 1–131.