Centrum voor Wiskunde en Informatica

REPORT *RAPPORT*

A path-based variable-free system for predicate logic

D. Ben-Shalom

Computer Science/Department of Software Technology

Report CS-R9444 July 1994

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.
SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

A Path-based Variable-Free System for Predicate Logic

Dorit Ben-Shalom[1,2]
dorit@cognet.ucla.edu

[1] *CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

[2] *Department of Linguistics, UCLA*

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

Abstract

Inspired by natural language, the paper suggests a variable-free system for predicate logic. The system generalizes the implicit variables of modal logic, interpreted as instructions to access elements of the current computational path.

CR Subject Classification (1991): I.2.7
Keywords and Phrases: natural language semantics, path, variable
Note: I would like to acknowledge the financial support of project NF 102/62-356 ('Structural and Semantic Parallels between Natural Languages and Programming Languages'), funded by the Netherlands Organization for Scientific Research (N.W.O). I would like to thank Johan van Benthem, Jan van Eijck, Wilfried Meyer Viol, and Kees Vermeulen for helpful comments on earlier drafts.

1. Introduction

The idea that variables should not be part of semantic representations of natural language is not new (see, for example, Purdy (1991), and Sanchez (1991)). It rests on the simple observation that, unlike verbs or nouns, variables are not visible in natural language expressions. But the issue of variables in natural language semantics acquires a new meaning in the context of first-order dynamic logic (Harel (1984)), and the related natural language formalisms of DRT (Heim(1982), Kamp (1981)) and DPL (Groenendijk and Stokhof (1991)). This is so because in these logics variables are not merely combinatory devices that enable quantification. In addition, they are names for the registers of a random access machine that is being programmed. The basic instruction in first-order dynamic logic is the assignment $x := t$, changing the current content of the x register to the current value of the term t. DRT and DPL use this kind of dynamic mechanism to model aspects of natural language anaphora: an antecedent like *a man* introduces a register x whose value represents a man. An anaphor like *he* can use the register name x to retrieve that representation. Nevertheless, the original objection against explicit variables in natural language semantics still stands.

In particular, suppose one accepts the assumption that natural language semantics involves registers which are dynamically updated. Even so it is possible to reject the assumption that these registers are accessed by explicit names like x (cf. Vermeulen (1991)). As an alternative, we suggest an implicit, contextually dependent access method, which treats the current registers as a sequence, or a path. The basic instruction is $x_i := t$, where x_i is the i-th register back in the current path. To illustrate the idea in its simplest form, the present paper applies it to predicate logic itself, rather than to any particular

Report CS-R9444
ISSN 0169-118X
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

dynamic system. In this setting, the implicit access method can be viewed as a generalization of the implicit access method of modal logic, where the only accessible element is the last member of the current path. Technically, this allows for a particularly simple comparison between modal logic and predicate logic, now seen as a type of a modal system.

The paper is organized as follows: Section 2 is an intuitive discussion of implicit variables as path instructions. Section 3 defines the modal language used to express formulas of predicate logic. Section 4 defines a semantics for this modal language in terms of sequence models. Sections 5 and 6 present two simple connections between modal logic and predicate logic that become apparent from this perspective: A proof theoretic connection between Enderton's (1972) axiom system for predicate logic and $S5$ in Section 5, and a model theoretic connection between partial isomorphism and modal bisimulations in Section 6. Section 7 compares the path-based system to Predicate-Functor Logic (Quine (1971), Kuhn (1983)). It concludes with a brief discussion of a somewhat similar idea in Dekker (1994).

Notations *ϕ, ψ, χ denote formulas. i, j, k denote natural numbers. $\mathcal{M}$ denotes a Kripke model. $\mathcal{L}$ denotes a first order language with no function symbols. $\mathcal{D}$ denotes a Tarski model with domain D.*

2. Implicit Variables and Path Instructions

As is well-known, variables in modal logic are implicit. For example, a formula like $\Box\Box p$ can be thought of as expressing a first-order formula like $\forall y(xRy \rightarrow \forall z(yRz \rightarrow Pz))$. One possible perspective about the semantics of implicit variables is in terms of *paths*. From this perspective, a modal formula is evaluated at a point, which is viewed as a path of length 1. Each $\Box$ operator is viewed as extending the current path by one element to the right, and each propositional letter is viewed as checking a property of the last element of the current path. For example, if $\Box\Box p$ is evaluated at a point $<s_1>$, then the subformula $\Box p$ is evaluated at sequences of type $<s_1, s_2>$, and p checks the p property of the last member s_3 of sequences of type $<s_1, s_2, s_3>$. Under this view, implicit variables are instructions to access elements of paths. Concretely, an instruction of the form x_i accesses the i-th element from the right of the current path. So the only instruction used by standard modal logic is x_1. Along the same lines, predicate logic can be thought of as a modal system which makes use of the full range of instructions x_i for any positive natural number i. As in modal logic, these instructions play the role of both bound and free variables, depending on their 'modal depth' within the formula. For example, in the formula $\Box p \rightarrow p$, the first instance of an implicit x_1 is bound, whereas the second is free. Semantically, an instruction plays the role of a free variable if the element it accesses is in the sequence at which the formula is evaluated, rather than being added to it through path extensions.

3. Language

Let $\mathcal{L}$ be a first order language without function symbols, defined by:

$$\phi ::= \bot \mid p \mid \phi_1 \rightarrow \phi_2 \mid \forall x_j \phi$$

where p is an atom (by which we mean an atomic formula or an equality).

We assume that the variables of $\mathcal{L}$ are of the form x_i, where i is a positive natural number.

$\mathcal{L}^\circ$, *the modal language determined by* $\mathcal{L}$ is defined by:

$$\phi ::= \bot \mid p \mid \phi_1 \rightarrow \phi_2 \mid \Box\phi$$

where p is an atom of $\mathcal{L}$. (But recall that an expression like x_1 in $\mathcal{L}^\circ$ has the semantics of a path instruction rather than that of a variable).

We define the function $^\circ$, a recursive translation from $\mathcal{L}$ to $\mathcal{L}^\circ$. What $^\circ$ does is replace the variables of $\mathcal{L}$ with the appropriate path instructions of $\mathcal{L}^\circ$. For example, the $\mathcal{L}$ formula $\forall x_3 \forall x_4 R(x_3, x_4, x_5)$ is translated by $\Box\Box R(x_2, x_1, x_7)$, and $\forall x_3 \forall x_4 R(x_4, x_3, x_5)$ as $\Box\Box R(x_1, x_2, x_7)$.

Notation: For $\phi \in \mathcal{L}$, $[+1]\phi$ is the result of replacing each x_i in ϕ with x_{i+1}.

$^\circ$ is defined as follows:

$$\bot^\circ =_{def} \bot$$

$$p^\circ =_{def} p$$

$$(\phi_1 \rightarrow \phi_2)^\circ =_{def} (\phi_1)^\circ \rightarrow (\phi_2)^\circ$$

$$(\forall x_j \phi)^\circ =_{def} \Box([x_1/x_{j+1}][+1]\phi)^\circ$$

For example,

$$(\forall x_3 \forall x_4 R(x_3, x_4, x_5))^\circ$$

$$= \Box([x_1/x_{3+1}][+1]\forall x_4 R(x_3, x_4, x_5))^\circ$$

$$= \Box([x_1/x_{3+1}]\forall x_5 R(x_4, x_5, x_6))^\circ$$

$$= \Box(\forall x_5 R(x_1, x_5, x_6))^\circ$$

$$= \Box\Box([x_1/x_{5+1}][+1]R(x_1, x_5, x_6))^\circ$$

$$= \Box\Box([x_1/x_{5+1}]R(x_2, x_6, x_7))^\circ$$

$$= \Box\Box(R(x_2, x_1, x_7))^\circ$$

$$= \Box\Box R(x_2, x_1, x_7)$$

$^\circ$ is onto $\mathcal{L}^\circ$ and 1-1 up to alphabetic variants. In other words, if $[\phi]$ is the set of alphabetic variants of ϕ, then for all $\psi \in [\phi]$, $\psi^\circ = \phi^\circ$, and for all $\psi \notin [\phi]$, $\psi^\circ \neq \phi^\circ$. For example, $\Box\Box R(x_2, x_1, x_7)$ is also the translation of $\forall x_1 \forall x_2 R(x_1, x_2, x_5)$.

When comparing $\mathcal{L}$ and $\mathcal{L}^\circ$ it is convenient to have definitions for substitution in $\mathcal{L}^\circ$ and for free and bound occurrences of path instructions.

Substitution is easy. For c a constant,

$$[c/x_i]\bot =_{def} \bot$$

$$[c/x_i]p =_{def} \text{ the } \mathcal{L} \text{ atom } [c/x_i]p$$

$$[c/x_i](\phi_1 \rightarrow \phi_2) =_{def} [c/x_i]\phi_1 \rightarrow [c/x_i]\phi_2$$

$$[c/x_i]\Box\phi =_{def} \Box[c/x_{i+1}]\phi$$

$$[x_j/x_i]\bot =_{def} \bot$$

$$[x_j/x_i]p =_{def} \text{ the } \mathcal{L} \text{ atom } [x_j/x_i]p$$

$$[x_j/x_i](\phi_1 \rightarrow \phi_2) =_{def} [x_j/x_i]\phi_1 \rightarrow [x_j/x_i]\phi_2$$

$$[x_j/x_i]\Box\phi =_{def} \Box[x_{j+1}/x_{i+1}]\phi$$

To define when a path instruction is free, we define the modal depth of a formula within another formula. For example, the modal depth of the first occurrence of p in $\Box p \rightarrow p$ is 1, whereas the modal depth of the second occurrence of p is 0.

Let ϕ be be a formula of $\mathcal{L}^\circ$, and ψ an occurrence of a subformula of ϕ. *$md(\psi, \phi)$, the modal depth of ψ in ϕ* is defined by:

$$md(\psi, \bot) =_{def} 0$$

$$md(\psi, p) =_{def} 0$$

$$md(\psi, \phi_1 \rightarrow \phi_2) =_{def} \begin{cases} md(\psi, \phi_1) & \text{if } \psi \text{ occurs in } \phi_1 \\ md(\psi, \phi_2) & \text{if } \psi \text{ occurs in } \phi_2 \end{cases}$$

$$md(\psi, \Box\phi) =_{def} md(\psi, \phi) + 1$$

Definition 1 *An occurrence of a path instruction x_j in $\phi \in \mathcal{L}^\circ$ is free if $j > d$, where d is the modal depth of the atom containing that occurrence in ϕ.*

For example, thinking about $\Box p \rightarrow p$ as $\Box p(x_1) \rightarrow p(x_1)$, the first occurrence of x_1 is bound and the second is free. Note that the usual restrictions on substitution in predicate logic are not needed, since no free variables get 'captured'. For example, $[x_1/x_3]\Box p(x_4) = \Box p(x_2)$.

4. Semantics

Notations *Let S be a set. The set of finite sequences over S is called S^*, and sequences are written as strings. I.e., $<s_1, \ldots, s_n>$ is written as $s_1 \ldots s_n$. If $\sigma = s_1 \ldots s_n$ we write $length(\sigma) = n$.*

Definition 2 *Let $\mathcal{D}$ be a model for $\mathcal{L}$, with domain D, and $\delta \in D^*$. If $\delta = d_n \ldots d_1$ we write $|\phi|^{\mathcal{D},\delta}$ for the truth value of ϕ in $\mathcal{D}$ under the partial assignment $\delta : \{x_1, \ldots, x_n\} \rightarrow D$ defined by $\delta(x_i) = d_i$. (note the reversal of order).*

$\mathcal{D}^\circ$, the Kripke Model determined by $\mathcal{D}$, is defined by:

$$\mathcal{D}^\circ =_{def} <W_{\mathcal{D}^\circ}, R_{\mathcal{D}^\circ}, V_{\mathcal{D}^\circ}>, \text{ where}$$

$$W_{\mathcal{D}^\circ} = D^*$$

$$R_{\mathcal{D}^\circ} = \{<\delta, \delta d> \mid \delta \in D^*, d \in D\}$$

$$V_{\mathcal{D}^\circ}(p)(\delta) = |p|^{\mathcal{D},\delta}$$

So $V_{\mathcal{D}^\circ}(p)(\delta)$ is undefined iff p contains an x_i with $i > length(\delta)$. The definition of satisfaction in $\mathcal{D}^\circ$ is a standard strict (or weak Kleene) interpretation of the undefined truth value U:

$$|\bot|^{\mathcal{D}^\circ,\delta} = \mathsf{F}$$

$$|p|^{\mathcal{D}^\circ,\delta} = V_{\mathcal{D}^\circ}(p)(\delta)$$

$$|\phi_1 \rightarrow \phi_2|^{\mathcal{D}^\circ,\delta} = \begin{cases} \mathsf{U} & |\phi_1|^{\mathcal{D}^\circ,\delta} = \mathsf{U} \text{ or } |\phi_2|^{\mathcal{D}^\circ,\delta} = \mathsf{U} \\ \mathsf{F} & |\phi_1|^{\mathcal{D}^\circ,\delta} = \mathsf{T} \text{ and } |\phi_2|^{\mathcal{D}^\circ,\delta} = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$$

$$|\Box\phi|^{\mathcal{D}^\circ,\delta} = \begin{cases} \mathsf{U} & |\phi|^{\mathcal{D}^\circ,\delta'} = \mathsf{U} \text{ for some } \delta R\delta' \\ \mathsf{T} & |\phi|^{\mathcal{D}^\circ,\delta'} = \mathsf{T} \text{ for all } \delta R\delta' \\ \mathsf{F} & \text{otherwise} \end{cases}$$

If undefinedness in $|\phi|^{\mathcal{D},\delta}$ is also subject to a weak Kleene interpretation, one can prove the following lemma:

Lemma 1 $|\phi^\circ|^{\mathcal{D}^\circ,\delta} = |\phi|^{\mathcal{D},\delta}$

proof:
By induction on the total number of $\rightarrow$ and $\forall$ in ϕ.
$n = 0$: $\phi = \bot$: obvious. $\phi = p$: by the definition of $V_{\mathcal{D}^\circ}$.
$n = n' + 1$: $\phi = \phi_1 \rightarrow \phi_2$: $|(\phi_1 \rightarrow \phi_2)^\circ|^{\mathcal{D}^\circ,\delta} = \mathsf{U}$ iff $|\phi_1^\circ|^{\mathcal{D}^\circ,\delta} = \mathsf{U}$ or $|\phi_2^\circ|^{\mathcal{D}^\circ,\delta} = \mathsf{U}$ iff $|\phi_1|^{\mathcal{D},\delta} = \mathsf{U}$ or $|\phi_2|^{\mathcal{D},\delta} = \mathsf{U}$ by the induction hypothesis, iff there is a free x_k in either ϕ_1 or ϕ_2 that is not in the domain of δ, iff $|\phi_1 \rightarrow \phi_2|^{\mathcal{D},\delta} = \mathsf{U}$. The argument for T is similar.
$\phi = \forall x_j\phi'$: $|([x_1/x_{j+1}][+1]\phi')^\circ|^{\mathcal{D}^\circ,\delta d} = \mathsf{U}$ for some δd iff $|[x_1/x_{j+1}][+1]\phi'|^{\mathcal{D},\delta d} = \mathsf{U}$ for some δd, by the induction hypothesis. Since x_1 must be in the domain of δd this may happen iff there is a free x_k in ϕ' with $k \neq j$ that is not in the domain of δ, iff $|\forall x_j\phi'|^{\mathcal{D},\delta} = \mathsf{U}$. $|([x_1/x_{j+1}][+1]\phi')^\circ|^{\mathcal{D}^\circ,\delta d} = \mathsf{T}$ for all δd iff $|[x_1/x_{j+1}][+1]\phi'|^{\mathcal{D},\delta d} = \mathsf{T}$ for all δd by the induction hypothesis, iff $|\phi'|^{\mathcal{D},\delta''} = \mathsf{T}$ for all δ'' which differ from δ at most in the value for x_j, iff $|\forall x_j\phi'|^{\mathcal{D},\delta} = \mathsf{T}$. □

Intuitively, $\mathcal{D}^\circ$ is an explicit representation of the possible 'paths' in $\mathcal{D}$. In this respect, it is reminiscent of the *unraveling* construction in modal logic (Sahlqvist (1975)).

Definition 3 *Let $\mathcal{M} = <M, R, V>$ be a Kripke model. Its unraveling $\mathcal{M}^*$ is defined by:*

$\mathcal{M}^* =_{def} <W_{\mathcal{M}^*}, R_{\mathcal{M}^*}, V_{\mathcal{M}^*}>$, *where*

$$W_{\mathcal{M}^*} = \{\delta d \mid \delta \in W^*, d \in W\}$$

$$R_{\mathcal{M}^*} = \{<\delta d, \delta dd'> \mid \delta \in W^*, dRd'\}$$

$$V_{\mathcal{M}^*}(p)(\delta d) = V(p)(d)$$

Proposition 1 *For every modal formula ϕ, $d \in W$, $\delta \in W^*$, $\mathcal{M}^*, \delta d \models \phi$ iff $\mathcal{M}, d \models \phi$.*

proof. By induction on the construction of ϕ. □

This proposition will be used in Section 5.

We will also need a notion of entailment for $\mathcal{L}^\circ$.

Definition 4 *A formula of $\phi \in \mathcal{L}^\circ$ is a sentence if it does not contain any free variables.*

The truth value of a sentence ϕ is always T or F, and for every $\mathcal{D}^\circ$,δ, and δ', $|\phi|^{\mathcal{D}^\circ,\delta} = |\phi|^{\mathcal{D}^\circ,\delta'}$. We write $\mathcal{D}^\circ \models \phi$ if $|\phi|^{\mathcal{D}^\circ,\delta} = \mathsf{T}$ for all δ.

To overcome the partiality of the assignments, we define entailment for formulas in terms of entailment for sentences. The intuition is a familiar one: free variables are treated as 'parameters', i.e., as constants relative to an assignment. So let $\{c_i\}_{i\in\omega}$ be a set of new constants, and $[c_i/x_i]\psi$ be the result of applying the set of substitutions $\{[c_i/x_i]\}_{i\in\omega}$ to ψ.

Definition 5 *Let $\Gamma \cup \{\phi\}$ be a set of $\mathcal{L}^\circ$ formulas. $\Gamma \models \phi$ iff for every model $\mathcal{D}$ for $\mathcal{L} \cup \{c_i\}_{i\in\omega}$, if $\mathcal{D}^\circ \models [c_i/x_i]\gamma$ for all $\gamma \in \Gamma$, then $\mathcal{D}^\circ \models [c_i/x_i]\phi$.*

Proposition 2 $\Gamma^\circ \models \phi^\circ$ *iff* $\Gamma \models \phi$.

5. Axiomatics

Since this logic is a simple variant of predicate logic, it can be axiomatized by a simple translation of a deduction system for predicate logic. For example, here is a translation of the Hilbert-style system in Enderton (1972), with generalization as an inference rule.

Notation: For $\phi \in \mathcal{L}^\circ$, $[d+1/d]\phi$ is the result of replacing every free path instruction x_i with the path instruction x_{i+1}. And similarly for $[d-1/d]\phi$.

PL°:

Axioms:

Boolean tautologies

Distribution

$$\Box(\phi \to \phi') \to (\Box\phi \to \Box\phi')$$

Quantifier

$$\phi \to \Box[d+1/d]\phi$$

$$\Box\phi \to [t/x_0][d-1/d]\phi$$

Equality

$$x_i = x_i$$

$$x_i = x_j \to (p \to p')$$

where p' is obtained from p by replacing some occurrences of x_i by x_j.

Inference rules:

Modus ponens

$$\phi \to \phi', \phi \quad / \quad \phi'$$

Generalization

$$\vdash \phi \quad / \quad \vdash \Box[x_1/x_i][d+1/d]\phi$$

Theorem 1 *PL° is sound and complete with respect to the class of $\mathcal{D}^\circ$ models.*

proof:
Soundness may be checked directly. (The quantifier rules are sound without side conditions because the substitutions they contain can not make any free variable bound). Completeness follows automatically from the completeness of Enderton's system with generalization as an inference rule. For if $\Gamma \models \phi$ then since $^\circ$ is onto, for every $\psi \in \Gamma \cup \{\phi\}$ there is a predicate logic formula ψ' such that $\psi'^\circ = \psi$, and by Proposition 2, $\{\gamma' \mid \gamma \in \Gamma\} \models \phi'$. So if $\phi_1, \ldots, \phi_n$ is a derivation of ϕ' from $\{\gamma' \mid \gamma \in \Gamma\}$, then $\phi_1^\circ, \ldots, \phi_n^\circ$ is a derivation of ϕ from Γ: if ϕ_i is a predicate logic axiom, then ϕ_i° is one of the axioms above, if ϕ_i is derived from ϕ_j and ϕ_k by modus ponens, then ϕ_i° is derived by modus ponens from ϕ_j° and ϕ_k°, and similarly for generalization. □

Let $\mathcal{L}$ be a first-order language with some unary predicates, and recall the discussion in Section 2. Syntactically, one can think of the standard modal language (ML) as that fragment of $\mathcal{L}^\circ$ where all predicates are unary, there are no equalities, and the only path instruction is x_1.[1] Semantically, the sequence models of $\mathcal{L}^\circ$ resemble unravelled universal Kripke models (Kripke models where $R = W \times W$). And the modal deduction system $S5$ is sound and complete with respect to the class of universal Kripke models. This state of affairs invites a question about the relation between the modal translation of Enderton's system and $S5$. As it happens, it is easy to prove the following simple connection:

Definition 6 *$\vdash_{PL^\circ|ML} \phi$ iff there is a derivation $\phi_1 \ldots \phi_n$ of PL°, where $\phi_n = \phi$ is derived from axioms, and $\forall i$, $1 \leq i \leq n$, $\phi_i \in ML$.*

Theorem 2 *For every $\phi \in ML$, $\vdash_{PL^\circ|ML} \phi$ iff $\vdash_{S5} \phi$.*

proof:
$\Rightarrow$
Suppose $\vdash_{PL^\circ|ML} \phi$. $\vdash_{S5} \phi$ since PL° is sound and $S5$ is complete: Restricted to ML, the models of $\mathcal{L}^\circ$ are the unravelings of universal models for ML fragment of $\mathcal{L}^\circ$. So by Proposition 1, if ϕ were false at some point of a universal model, it would be false in a model of $\mathcal{L}^\circ$.
$\Leftarrow$
Conversely, we show that every $S5$ derivation has a corresponding derivation in $PL^\circ \mid ML$. Every boolean tautology and distribution axiom of $S5$ corresponds to a boolean tautology or distribution axiom of $PL^\circ \mid ML$. Modus ponens is a rule of $PL^\circ \mid ML$, and necessitation corresponds to generalization with $i = 2$. Reflexivity ($\Box\phi' \to \phi'$) corresponds to the second quantifier axiom with $t = x_1$ and transitivity ($\Box\phi' \to \Box\Box\phi'$) corresponds to the first quantifier axiom with $\phi = \Box\phi'$. Finally, symmetry ($\phi' \to \Box\Diamond\phi'$) can be derived by a contraposition of $\Box\neg\phi' \to \neg\phi'$ followed by an application of the first quantifier axiom with $\phi = \Diamond\phi'$. □

[1]The restriction to unary predicates is not essential here, since if there is only one path instruction then every n-ary predicate can be traded for a unary one.

Concretely, suppose the rules of PL° are restricted such that their assumptions and conclusions are in ML. Similarly, the axioms of PL° are restricted to those instances in ML. The result is a system that is very much like $S5$: It consists of the boolean tautologies, distribution, modus ponens, necessitation and reflexivity, but instead of transitivity and symmetry together, the single restricted axiom scheme: $\phi \rightarrow \Box\phi$ if every p in ϕ is in the scope of at least one $\Box$.

6. Bisimulations

The basic notion of structural similarity for models of a modal language is that of a bisimulation. The corresponding basic notion for models of a first order language is that of a partial isomorphism. The theorem below shows that the modal system for predicate logic in this paper is natural in the sense that in there these two notions coincide. (Cf. Fernando (1992) for a similar result using a fragment of first-order dynamic logic).

Definition 7 *$\mathcal{M}^s$, the submodel of $\mathcal{M}$ generated by s, is defined by:*

$\mathcal{M}^s =_{def} <W^s, R^s, V^s>$, *where*

$W^s = \{s' \mid sR^is',\ 0 \leq i\}$,

$R^s = R \cap (W^s \times W^s)$,

$V^s(p) = V(p) \cap W^s$

Definition 8 *$\mathcal{M}^s_n$, the orbit of depth n around s in $\mathcal{M}$, is defined by:*

$\mathcal{M}^s_n =_{def} <W^s_n, R^s_n, V^s_n>$, *where*

$W^s_n = \{s' \mid sR^is',\ 0 \leq i \leq n\}$,

$R^s_n = R \cap (W^s_n \times W^s_n)$,

$V^s_n(p) = V(p) \cap W^s_n$

Definition 9 *A relation C between the domains of two Kripke models $\mathcal{M} = <W, R, V>$ and $\mathcal{M}' = <W', R', V'>$ is a bisimulation if it satisfies the following conditions:*

If sCs', then s, s' have the same values for propositional letters

If sCs' and sRt, then there exists a point t' such that $s'R't$ and tCt'

analoguously in the other direction.

Definition 10 *Let $\mathcal{D}$ be a model for $\mathcal{L}$, $c_1, \ldots, c_n$ new constants. $(\mathcal{D}, d_1 \ldots d_n)$ is the model for $\mathcal{L} \cup \{c_1, \ldots, c_n\}$ that extends $\mathcal{D}$ by sending each c_i to $d_i \in D$.*

Definition 11 *$<d_1 \ldots d_n, d'_1 \ldots d'_n>$ is a local isomorphism between $\mathcal{D}$ and $\mathcal{D}'$ iff $(\mathcal{D}, d_1 \ldots d_n)$ and $(\mathcal{D}', d'_1 \ldots d'_n)$ satisfy the same quantifier-free sentences.*

Definition 12 *At each step $i > 0$ of the Infinite Ehrenfeucht game between $\mathcal{D}$ and $\mathcal{D}'$, the first player (Di) chooses an element of either D or D'. The second player (Sy) chooses an element of the other model. Let d_i be the element of D chosen in the i-th step, and similarly for d'_i. Sy wins if for every i, $<d_1 \dots d_i, d'_1 \dots d'_i>$ is a local isomorphism between $\mathcal{D}$ and $\mathcal{D}'$. Sy has a winning strategy if he wins every game between $\mathcal{D}$ and $\mathcal{D}'$.*

Definition 13 *$\mathcal{D}$ and $\mathcal{D}'$ are partially isomorphic iff the Sy player has a winning strategy for the infinite Ehrenfeucht game between $\mathcal{D}$ and $\mathcal{D}'$.*

The relevant notion of bisimulation here is a 3-valued one. In other words, if C is a bisimulation and sCs', then for every atom p, the truth value of p at s is the same value of p at s': T for T, F for F, and U for U.

Lemma 2 *$\delta \in W_{\mathcal{D}^\circ}$ and $\delta' \in W_{\mathcal{D}'^\circ}$ have the same truth values for atoms iff $<\delta, \delta'>$ is a local isomorphism between $\mathcal{D}$ and $\mathcal{D}'$.*

proof:
Suppose $length(\delta) < length(\delta') = n$. Then the atom $x_n = x_n$ is defined at δ' but undefined at δ. Thus δ and δ' must have the same length, say k. By Lemma 1, δ and δ' have the same truth values for atoms iff $|p|^{\mathcal{D},\delta} = |p|^{\mathcal{D}',\delta'}$ for every atom p of $\mathcal{L}$ with variables in $\{x_1, \dots, x_k\}$. □

Theorem 3 *There is a non-empty bisimulation between $\mathcal{D}^\circ$ and $\mathcal{D}'^\circ$ iff $\mathcal{D}$ and $\mathcal{D}'$ are partially isomorphic.*

proof:
$\Rightarrow$
Suppose there is a non-empty bisimulation between $\mathcal{D}^\circ$ and $\mathcal{D}'^\circ$. Then w.l.g there is a bisimulation Z that connects the roots of $\mathcal{D}^\circ$ and $\mathcal{D}'^\circ$, since the structure of the models is repetitive and the restriction of a local isomorphism is a local isomorphism.
The *Sy* player of the infinite Ehrenfeucht game can use Z as a winning strategy as follows: By induction on n, *Sy* can play in such a way that the position reached after the n-th step of the game is in Z:
$n = 0$: Z connects the two roots.
$n = n' + 1$: By the induction hypothesis, the current position $<\delta, \delta'>$ is in Z. If the n-th move of *Di* is an element d of D then it determines a δd s.t. $\delta R_{\mathcal{D}^\circ} \delta d$. So *Sy* can choose an element d' of D' s.t. $\delta' R_{\mathcal{D}'^\circ} \delta' d'$, which exists since Z is a bisimulation. Similarly if *Di* chooses an element d' of D'.
$\Leftarrow$
Suppose $\mathcal{D}$ and $\mathcal{D}'$ are partially isomorphic. The winning strategy of *Sy* determines a bisimulation Z as follows:

$Z =_{def}$ the set of positions $<\delta, \delta'>$

reached in games in which *Sy* uses his winning strategy

Z is clearly non-empty. Suppose $<\delta, \delta'>$ is in Z and $\delta R_{\mathcal{D}^\circ} \delta d$. d can be seen as a move of *Di*, so *Sy* can use his winning strategy for choosing a d' s.t. $\delta' R_{\mathcal{D}'^\circ} \delta' d'$. Similarly if $\delta' R_{\mathcal{D}'^\circ} \delta' d'$ for some d' in $\mathcal{D}'^\circ$. □

A completely analogous argument shows that the *finite* Ehrenfeucht game of n rounds corresponds to a bisimulation between $(\mathcal{D}^{\circ})_n$ and $(\mathcal{D}'^{\circ})_n$ that connects the two roots.
The Ehrenfeucht-Fräissé Theorem can be thus restated as follows:

Theorem 4 *If $\mathcal{L}$ is finite, then $(\mathcal{D}, \delta)$ and $(\mathcal{D}', \delta')$ satisfy the same sentences of $\mathcal{L}$ of rank n or less iff there is a bisimulation between $(\mathcal{D}^{\circ})^{\delta}_n$ and $(\mathcal{D}'^{\circ})^{\delta'}_n$ that connects δ and δ'.*

7. Discussion

The system presented in this paper is a variable-free system for predicate logic, based on sequence models. In these respects, it resembles Predicate-Functor Logic (Quine (1971), Kuhn (1983)). There are, however, at least three major differences between these systems.

The first difference has to do with what is meant by 'variable-free'. The goal of Predicate-Functor Logic is to eliminate bound variables altogether, by use of functors. The goal of the system in this paper is to avoid using explicit names for either free or bound variables, but no attempt is made to eliminate the variables themselves. The second difference has to do with how the sequences of the sequence models are manipulated. Both systems quantify over a fixed position at the end of each sequence, and manipulate the coordinates of the sequences to make this suffice. But the main tool of Predicate-Functor Logic is permutation, whereas the main tool of the present system is the use of projection functions. The third difference has to do with 'autonomy', in the sense of the relation between the new system and existing logics. Because of its philosophical goals, Predicate-Functor Logic is set up to be 'autonomous' from prior logical systems. The present system, on the other hand, is a relatively simple extention of standard modal logic.

Some of these differences can be glimpsed from the forms of the formulas themselves. For example, here is a predicate logic formula, its Predicate-Functor translation from Quine (1971), and its modal translation (using the usual abbreviations, e.g. $\Diamond =_{def} \neg\Box\neg$).

$$\exists x_2(\neg(x_2 = x_3) \cap F(x_3, x_2, x_3) \cap F(x_1, x_2, x_3))$$

$$]SSp_7p_4(-I \times F^3 \times F^3)x_3x_3x_3x_1x_3$$

$$\Diamond(\neg(x_1 = x_4) \cap F(x_4, x_1, x_4) \cap F(x_2, x_1, x_4))$$

We conclude with a brief discussion of Dekker's (1994) version of DPL, of which we became aware when about to finish the paper. Dekker's system resembles the current system in that projection functions are used to replace variables. But unlike the current system, they are used for dynamic rather than 'static' quantification: the predicate logic part of Dekker's system is a standard system with variables.

In Dynamic Predicate Logic (Groenendijk and Stokhof (1991)), dynamic quantification is employed to model aspects of natural language anaphora. As in first-order dynamic logic, registers are named by variables, and are accessed by an explicit access method: The existential quantifier in the translation of an expression like *a boy* introduces a register, named by an explicit variable like x. Subsequent pronouns like *he* make use of the value of this register by employing a free occurrence of x. The following is a typical (simplified) translation.

A boy likes a girl. She admires him

$$\exists x(Bx \wedge \exists y(Gy \wedge Lxy)) \wedge Ayx$$

Instead of this explicit access method, Dekker suggests an implicit method based on projection functions: An information state is an n-ary relation between individuals of a model $\mathcal{D}$. An existential formula like $\exists x Bx$ transforms an n-ary relation s into the $n+1$-ary relation $\{\delta d \mid \delta \in s \text{ and } d \text{ is a boy }\}$. A pronoun uses the values in an information state by employing a projection function p_i that accesses the i-th last coordinate of the tuples of the state. The example above has the following simplified translation:

$$\exists x(Bx \wedge \exists y(Gy \wedge Lxy)) \wedge Ap_2p_1$$

Beyond the mere technical similarity between Dekker's ideas and those of the present paper, there is an opportunity here for a generalization about the role of pronouns in natural language. Ben-Shalom (in progress) uses a variable-free system to model bound anaphora in natural language. As in the present paper, quantification is achieved by a modal system with path instructions over sequence models. In particular, the accessibility relation i extends tuples by their i-th last coordinate. The following are typical translations.

A boy likes a girl

$$\Diamond_{boy}\Diamond_{girl}like$$

A girl admires herself

$$\Diamond_{girl}\Box_1 admire$$

Writing i_s for the accessibility relation that accesses the i-th last coordinate of static tuples, and i_d for the accessibility relation accessing the i-th last coordinates of dynamic ones, one can compare the following translations:

A boy likes a girl. She admires him

$$\Diamond_{boy}\Diamond_{girl}like \wedge \Box_{1\_d}\Box_{2\_d}admire$$

A boy likes a girl. She admires herself

$$\Diamond_{boy}\Diamond_{girl}like \wedge \Box_{1\_d}\Box_{1\_s}admire$$

In other words, pronouns can be argued to be interpreted by projection functions whether they have static or dynamic antecedents. The difference is confined to which tuples are accessed: those of the static or the dynamic current state.

References

Ben-Shalom, D.: in progress, *Semantic Trees*, Ph.D. thesis, UCLA, Los Angeles

Dekker, P.: 1994, Predicate logic with anaphora, ms., University of Amsterdam

Enderton, H.: 1972, *A Mathematical Introduction to Logic*, Academic Press

Fernando, T.: 1992, Transition systems and dynamic semantics, technical report CS-R9217, CWI, Amsterdam

Groenendijk, J. and Stokhof, M.: 1991, Dynamic Predicate Logic, *Linguistics and Philosophy* 14, 39 – 100

Harel, D.: 1984, Dynamic logic, in D. Gabbay et al (eds.), *Handbook of Philosophical Logic* vol. 2, Reidel

Heim, I.: 1982, *The Semantics of Definite and Indefinite Noun Phrases*, Ph.D. thesis, University of Massachusetts, Amherst, Published in 1989 by Garland, New York

Kamp, H.: 1981, A theory of truth and semantic representation, in J. Groenendijk et al (eds.), *Formal Methods in the Study of Language*, 277 – 322, Mathematical Centre, Amsterdam

Kuhn, S.: 1983, An axiomatization of Predicate Functor Logic, *Notre Dame Journal of Formal Logic* 24, 233 – 241

Purdy, W.: 1991, A logic for natural language, *Notre Dame Journal of Formal Logic* 32, 409 – 425

Quine, W.V.: 1971, Algebraic logic and predicate functors, in Fenstad (ed.), *Proceedings of the Second Scandinavian Logic Symposium*, 309 – 316, North Holland, Amsterdam

Sahlqvist, H.: 1975, Completeness and correspondence in the first and second order semantics for modal logics, in *Proceedings of the Third Scandinavian Logic Symposium*, North Holland, Amsterdam

Sanchez, V.: 1991, *Studies on Natural Logic and Categorial Grammar*, Ph.D thesis, University of Amsterdam, Amsterdam

Vermeulen, C.F.M.: 1991, Merging without mystery: variables in dynamic semantics, O.T.S. working paper CL-91-003, Utrecht University