Predicate Logic with Flexibly Binding Operators and Natural Language Semantics*

Peter Pagin and Dag Westerståhl

Sometime 1992

Abstract

A new formalism for predicate logic is introduced, with a non-standard method of binding variables, which allows a compositional formalization of certain anaphoric constructions, including 'donkey sentences' and cross-sentential anaphora. A proof system in natural deduction format is provided, and the formalism is compared with other accounts of this type of anaphora, in particular Dynamic Predicate Logic.

Keywords: anaphora, compositionality, DPL, dynamic semantics, variable-binding

1 INTRODUCTION

There appears to be broad consensus about the view that certain natural language constructions with anaphoric pronouns cannot be compositionally formalized in predicate logic, at least not in any reasonable way. Prime examples are so-called donkey sentences and simple cross-sentence anaphora. Accounting for these phenomena was a main motivation behind the introduction of alternative semantic frameworks such as Discourse Representation Theory (DRT, Kamp 1981) and Dynamic Predicate Logic (DPL, Groenendijk and Stokhof 1991).

We wish to challenge this received view, by presenting a new form of predicate logic, called Predicate logic with Flexibly binding Operators (PFO), and by showing that the aforementioned linguistic phenomena, as well as many other familiar constructions with pronouns and quantifiers, can be straightforwardly formalized in PFO in a perfectly compositional way.

The alternative formalisms replace the semantics of predicate logic with some more or less *dynamic* semantics. As to syntax, DPL uses the formulas of predicate logic, but employs a non-standard form of variable-binding. PFO, on the other hand, retains essentially the standard semantics, but uses a slightly different syntax, with yet another mode of binding variables. It will be seen

*Both authors have been supported by grants from the Swedish Council for Research in the Humanities and Social Sciences.

that PFO, in contrast with the other formalisms, is really a version of predicate logic.

We begin (section 2) by briefly delineating the notion of compositionality at stake in the claim that donkey sentences etc. cannot be compositionally formalized in predicate logic. Then we present PFO (section 3), and show, mainly by means of examples, how it handles the relevant linguistic phenomena (section 4). PFO is in fact a simple and natural system which may have some interest of its own. We demonstrate its intertranslatability with standard predicate logic (section 5), and we exhibit what logic looks like in this format (sections 6 and 7). Then we are in a position to compare PFO, PL and DPL w.r.t. their respective logical consequence relations, the form of their truth definitions, and their dynamic/non-dynamic properties (section 8). The paper ends with a compositional version of PFO semantics, and some general remarks on the idea of compositionality and on the relative merits of PFO and DPL for natural language semantics (section 9). An intuitionistic version of PFO is presented in an appendix.

The variable-binding mechanism of standard predicate logic (PL) has at least the following features: It uses *unary variable-binding operators*, it is *selective* in the sense that the operators explicitly indicate the variable to be bound, and it *binds from the inside out*, in that an occurrence of a variable is always bound by the 'nearest' operator occurrence indicating that variable.

These features are, in fact, largely independent of each other. For example, we could change the PL mechanism so that it binds from the outside in instead, but otherwise remains the same. This variant of PL would have a simpler notion of bound variable (an occurrence of x is bound by an operator occurrence $\forall x$ iff it is within the scope of that operator occurrence), but the truth definition would be more complex than in PL.

Varying 'binding priorities' as in the previous paragraph is non-standard, but certain other variations are familiar. Logic with generalized quantifiers uses non-unary variable-binding operators, which apply to more than one formula (and which may bind more than variable in each formula), but the mechanism is still selective and works from the inside and out. In DPL, the usual selective unary operators are used, but variable-binding is neither from the inside out nor from the outside in; the 'binding priorities' are rather more complex.

PFO has the following variable-binding features:

(i) The variable-binding operators are *binary.* Besides being well suited to natural language quantification, this allows exploitation of the analogy between existential quantification and conjunction, and between universal quantification and implication:[1] in fact, PFO fuses sentential and variable-binding operators and permits a formulation where the only symbols used, in addition to non-logical symbols, variables and identity, are

$$\bot\,,\ [\,,\ ]\,.$$

[1]The intuitionistic version of PFO (cf. the Appendix) also uses the analogy between existential quantification and disjunction.

(ii) Binding in PFO is *unselective*, in that all variables which are common to both immediate subformulas of a quantified formula, become bound in the quantified formula.

(iii) The 'binding priority' of PFO is *from the outside in*: every occurrence of a variable x occurring in both immediate subformulas ϕ and ψ of a quantified formula becomes bound, regardless of whether that occurrence was free or already bound in ϕ (or ψ) taken by itself; previous bindings are thus in a sense cancelled.

As a consequence, vacuous quantification cannot occur in PFO. Likewise, no variable can occur both free and bound in a formula, nor be quantified more than once in it.

Remark: We are not aware of any formalism that looks like PFO, although it has features in common with other systems.[2] Binary quantification is the point of departure for applications of generalized quantifiers to natural language, and it is rather clear that the use of the binary existential and universal quantifier eliminates the need for sentential operators (except $\perp$). The idea of unselective quantification was used for so-called adverbs of quantification in Lewis (1975).

2 COMPOSITIONALITY AND ANAPHORA

In this section we state the principle of compositionality with enough precision to be able to illustrate the force, *and* the limitation, of this requirement with respect to the relevant linguistic constructions. The principle will be further discussed in section 9.

The compositionality principle asserts that the meaning of a complex expression is a function of the meaning of its parts. If meanings are embodied as expressions in a formal language, one can ensure compositionality by requiring some structural similarity between the natural and the formal language. Montague gave precise conditions on *translation functions* between two languages, the point being that if F is a primitive syntactic rule in L_1, say with two arguments of certain syntactic categories, there is a corresponding, possibly derived, syntactic rule G in L_2, with corresponding categories for arguments and value, such that the translation function h is a *homomorphism* w.r.t. these operations:

$$(H)\quad h(F(\alpha,\beta)) = G(h(\alpha), h(\beta))$$

(provided both languages are 'disambiguated').[3] The primitive syntactic rules can be used to define the notion of a *constituent* of an expression in the language. (H) will then imply a corresponding *constituency principle*:

(C) *If* α *is a consituent of* β *in* L_1, $h(\alpha)$ *is a constituent of* $h(\beta)$ *in* L_2.

[2] Cf., however, the Postscript.

[3] In "Universal grammar", cf. Montague (1974), ch. 7, section 5.

For our purposes here it suffices to consider constituents of sentences which are themselves sentences (formulas).[4] Consider the familiar examples

(1) If Pedro owns a donkey he is rich

(2) If Pedro owns a donkey he beats it

These sentences appear to have the same structure: a conditional with an existential sentence as antecedent. The following translation of (1) into PL preserves this structure:

(1′) $\exists y(\mathit{donkey}(y) \wedge \mathit{owns}(p, y)) \rightarrow \mathit{rich}(p)$

But no corresponding procedure works for (2): if we replace $\mathit{rich}(p)$ by $\mathit{beats}(p, y)$ in (1′), the result is not a sentence, and if we extend the scope of the existential quantifier to $\mathit{beats}(p, y)$, the resulting sentence has the wrong meaning. Of course, every logic student can formalize (2) by

(2′) $\forall y((\mathit{donkey}(y) \wedge \mathit{owns}(p, y)) \rightarrow \mathit{beats}(p, y))$

but the problem is to do it compositionally; (2′) does not have the formalizations of the subsentences of (2) as subformulas.

The argument depends on two assumptions: (a) that the two sentences should be translated by the same 'if-then'-rule, and (b) that indefinite noun phrases like *a donkey* should be rendered with the existential quantifier. Both assumptions seem entirely reasonable.

Apparently, then, when we formalize (2) in PL, (C) is violated. Note, however, that the argument also relies on the actual syntactic rules (formation rules) of PL. To see this, consider the following alternative version PL′: The formation rules for atomic formulas, negations, conjunctions, disjunctions, existential and universal quantifications are as usual, but there are two implication symbols $\rightarrow$ and $\Rightarrow$, and an extra quantifier symbol $\forall\!\forall$, with the following new formation rules:

$$G_1(\phi, \psi) = \begin{cases} \forall\!\forall x(\phi_0 \rightarrow \psi), & \text{if } \phi = \exists x \phi_0 \\ \phi \rightarrow \psi, & \text{otherwise} \end{cases}$$

$$G_2(\phi, \psi) = \begin{cases} \phi \rightarrow \psi, & \text{if } \phi = \exists x \phi_0 \\ \phi \Rightarrow \psi, & \text{otherwise.} \end{cases}$$

Semantically, $\forall\!\forall$ is treated just as ordinary universal quantification, and $\rightarrow$ and $\Rightarrow$ both as ordinary material implication. Now, it is not hard too see that the following holds. *First*, this system is just a version of ordinary predicate logic; the syntax is a bit akward but the semantics is the same. *Second*, the syntax of PL′ is still 'disambiguated': each formula has a unique analysis (that is why we introduced $\forall\!\forall$ and $\Rightarrow$). *Third*, however, (1) and (2) can both be correctly

[4] However, in another paper we will show that PFO style variable binding can also be used with a formalization which is fully compositional, i.e., also at the subsentential level.

formalized in this system using G_1, and the formalization is compositional in the sense that (H) holds.[5]

Examples like this illustrate the familiar notion that "it is always possible to satisfy compositionality by simply adjusting the syntactic and/or semantic tools one uses, unless that is, the latter are constrained on independent grounds".[6] The point is that it is not sufficient to merely claim one has a version of predicate logic which handles donkey sentences compositionally. To be of interest, the formalism must possess other virtues, unlike the ad hoc PL′. We hope it will be clear that the PFO formalism, to be presented in the next section, indeed has such additional virtues.

Besides donkey sentences, cross sentence anaphora creates problems for compositional translation into PL. For example,

(3) A farmer walks. He whistles.

(3′) $\exists x((\mathit{farmer}(x) \wedge \mathit{walks}(x)) \wedge \mathit{whistles}(x))$

Here we have a *discourse* or *text*, but the first sentence does not correspond to a subformula of the formalization.

3 THE PFO FORMALISM

PFO has two binary operators $[\cdot,\cdot]$ and $(\cdot,\cdot)$ corresponding to universal quantification (implication) and existential quantification (conjunction), respectively. For example,

(Px, Qy)	translates as	$Px \wedge Qy$,
$[Px, Qy]$	translates as	$Px \rightarrow Qy$,
(Px, Qx)	translates as	$\exists x(Px \wedge Qx)$,
$[Px, Qx]$	translates as	$\forall x(Px \rightarrow Qx)$.

Likewise (cf. (ii) in section 1),

$(Pxyz, Qyzuv)$	translates as	$\exists y \exists z(Pxyz \wedge Qyzuv)$,
$[Px, (Qy, Rxy)]$	translates as	$\forall x(Px \rightarrow \exists y(Qy \wedge Rxy))$.

However,

$[Px, (Qx, Rxy)]$	translates as	$\forall x(Px \rightarrow (Qy \wedge Rxy))$.

[5] Suppose a is *Pedro owns a donkey*, and (1) and (2) are construed as $F(a, \mathit{he\ is\ rich})$ and $F(a, \mathit{he\ is\ beats\ it})$, respectively. Assuming that $h(a) = \exists y(\mathit{donkey}(y) \wedge \mathit{owns}(p, y))$, letting F correspond to G_1, and slurring over the treatment of the pronouns involved, we get, using (H), the truth-functionally correct translations $\forall\forall y((\mathit{donkey}(y) \wedge \mathit{owns}(p, y)) \rightarrow \mathit{rich}(p))$ and $\forall\forall y((\mathit{donkey}(y) \wedge \mathit{owns}(p, y)) \rightarrow \mathit{beats}(p, y))$. The constituency principle (C) also holds: note, for example, that $\exists y\phi$ is a constituent of $\forall\forall y(\phi \rightarrow \psi)$ in PL′.

[6] Groenendijk and Stokhof (1991), p. 93. An example of an independent constraint might be that the (formula) syntax, in contrast with the one for PL′, be *context free* and that the notion of a constituent be *standard* (e.g., that a constituent of an expression is a substring of that expression; cf. the previous footnote). PFO satisfies this constraint.

Quantification starts from the outside; since x is common to Px and (Qx, Rxy) it is quantified, but then it cannot be quantified again in (Qx, Rxy), so the latter becomes a conjunction. We now give the formal definitions.

3.1 Syntax

Non-logical symbols, variables, the *identity symbol* $=$ and the *absurdity symbol* $\bot$ are as usual. *Terms* and *atomic formulas* are defined in the ordinary way. *Formulas* are defined inductively as follows:

DEFINITION 3.1

(a) Atomic formulas are *formulas*.

(b) If ϕ and ψ are *formulas* then so are (ϕ, ψ) and $[\phi, \psi]$.

The notion of a *subformula* of a formula is defined in the obvious way. Free and bound variable occurrences are introduced by first giving an inductive definition of what it means for an occurrence of a variable x to be *surface bound* (*depth bound*) in ϕ.

DEFINITION 3.2

(a) No occurrence of x is *surface bound* or *depth bound* in an atomic formula.

(b) An occurrence of x is *surface bound* in (ϕ, ψ) if x occurs in both ϕ and ψ. Similarly for $[\phi, \psi]$.

(c) An occurrence of x is *depth bound* in (ϕ, ψ) if it is not *surface bound* in (ϕ, ψ), but is *surface bound* in a subformula of (ϕ, ψ). Similarly for $[\phi, \psi]$.

DEFINITION 3.3 An occurrence of x is *bound* in ϕ if it is either surface or depth bound in ϕ, otherwise it is *free* in ϕ. A formula is a *sentence* if no variable occurs free in it.

The notions of free and bound variables in PFO differ from those of ordinary predicate logic in the following respect:

PROPOSITION 3.1 *If one occurrence of x is free (surface bound, depth bound) in ϕ, then so are all other occurrences of x in ϕ.*

Proof: By induction on formulas. The atomic case is clear. Now suppose x_0 is an occurrence of x in (ϕ, ψ) (the case of $[\phi, \psi]$ is entirely similar).
Case 1: x_0 is free in (ϕ, ψ). Then x_0 is free in ϕ but x does not occur at all in ψ (or *vice versa*). By induction hypothesis, all occurrences of x are free in ϕ, and it follows that the same holds for (ϕ, ψ).
Case 2: x_0 is surface bound in (ϕ, ψ). Then x occurs in both ϕ and ψ, so, by definition, all occurrences of x are surface bound in (ϕ, ψ).
Case 3: x_0 is depth bound in (ϕ, ψ). Then x does not occur at all in ψ and there is a subformula (ϕ_0, ψ_0) or $[\phi_0, \psi_0]$ of ϕ such that x_0 is in this subformula and x occurs in both ϕ_0 and ψ_0 (or vice versa with ϕ and ψ interchanged).

Assume (ϕ_0, ψ_0) is the largest such subformula. If x_1 is another occurrence of x in (ϕ, ψ), hence in ϕ, then x_1 has to be in ϕ_0 or ψ_0, for otherwise there would be a larger subformula of ϕ with the above property. Hence, x_1 is depth bound in (ϕ, ψ). $\Box$

Given a formula ϕ, we let Var_ϕ be the set of variables occurring in ϕ, and $Free_\phi$ ($Sbound_\phi$, $Dbound_\phi$) the set of free (surface bound, depth bound) variables in ϕ. By Proposition 3.1, the latter three sets form a partition of Var_ϕ. Also, let $Bound_\phi = Sbound_\phi \cup Dbound_\phi$. We have

$$Sbound_{(\phi,\psi)} = Sbound_{[\phi,\psi]} = Var_\phi \cap Var_\psi$$

$$Sbound_{(\phi,\psi)} = Sbound_{[\phi,\psi]} = (Bound_\phi \setminus Var_\psi) \cup (Bound_\psi \setminus Var_\phi)$$

$$Free_{(\phi,\psi)} = Free_{[\phi,\psi]} = (Free_\phi \setminus Var_\psi) \cup (Free_\psi \setminus Var_\phi)$$

Call a variable x *quantified at* a subformula ψ of ϕ, if x is surface bound in ψ but does not occur in ϕ outside of ψ. Every bound variable of ϕ is quantified at a unique subformula of ϕ (for example, if x is surface bound in ϕ, it is quantified at ϕ in ϕ). But if ϕ in turn is a subformula of θ, x may be quantified at ψ in ϕ although not quantified at ψ in θ. This is the sense in which quantifications in subformulas can be *cancelled* in formulas of PFO.

3.2 Semantics

A *model* $\mathcal{M}$ consists, as usual, of a nonempty set M and an interpretation function assigning suitable denotations to non-logical symbols. Let Var be the set of variables. An *M-assignment* is a function f from Var to M. We must define what it means for f to *satisfy* a formula ϕ in $\mathcal{M}$. As we saw, quantifications in subformulas of ϕ may be cancelled in ϕ. This means that the usual ternary satisfaction relation cannot be defined directly by an induction going 'from inside and out'. However, we can first define inductively a satisfaction relation between *four* things: an assignment f, a formula ϕ, a model $\mathcal{M}$, and a subset X of Var; X is to be a set of 'marked' variables, which cannot be quantified again in subformulas of ϕ.

Given $\mathcal{M}$ and an M-assignment f, the *value* $t^{\mathcal{M},f}$ of a term t is defined as usual. If $a_1, \ldots, a_k \in M$ then $f(x_i/a_i)_{1\leq i\leq k}$ is the assignment which is like f except that a_i is assigned to x_i, $1 \leq i \leq k$. Also, $\{x_i/a_i\}_{1\leq i\leq k}$ stands for any M-assignment which assigns a_i to x_i. (The subscript with the condition $1 \leq i \leq k$ will usually be omitted.)

DEFINITION 3.4 Let $\mathcal{M}$, f and X be as above.

(a) $\mathcal{M}, X \models_f Pt_1 \ldots t_n \Longleftrightarrow \langle t_1^{\mathcal{M},f}, \ldots, t_n^{\mathcal{M},f} \rangle \in P^{\mathcal{M}}$

(b) $\mathcal{M}, X \models_f (t_1 = t_2) \Longleftrightarrow t_1^{\mathcal{M},f} = t_2^{\mathcal{M},f}$

(c) Not $\mathcal{M}, X \models_f \bot$

Let $(\mathit{Var}_\phi \cap \mathit{Var}_\psi) \setminus X = \{x_1, \ldots, x_k\}$.

(d) $\mathcal{M}, X \models_f (\phi, \psi) \Longleftrightarrow$ there are $a_1, \ldots, a_k \in M$ such that
$\mathcal{M}, X \cup \{a_1, \ldots, a_k\} \models_{f(x_i/a_i)} \phi$ and $\mathcal{M}, X \cup \{a_1, \ldots, a_k\} \models_{f(x_i/a_i)} \psi$

(e) $\mathcal{M}, X \models_f [\phi, \psi] \Longleftrightarrow$ for all $a_1, \ldots, a_k \in M$,
if $\mathcal{M}, X \cup \{a_1, \ldots, a_k\} \models_{f(x_i/a_i)} \phi$ then $\mathcal{M}, X \cup \{a_1, \ldots, a_k\} \models_{f(x_i/a_i)} \psi$.

DEFINITION 3.5 $\mathcal{M} \models_f \phi \Longleftrightarrow \mathcal{M}, \emptyset \models_f \phi$.

Note that if $(\mathit{Var}_\phi \cap \mathit{Var}_\psi) \setminus X$ is empty, i.e., if ϕ and ψ have no variables in common, or if their common variables are all in X, then (ϕ, ψ) behaves as conjunction, and $[\phi, \psi]$ as material implication. One easily establishes

LEMMA 3.2 *If f and g are M-assignments which agree on $\mathit{Free}_\phi \cup X$, then $\mathcal{M}, X \models_{\mathrm{f}} \phi$ iff $\mathcal{M}, X \models_{\mathrm{g}} \phi$.*

COROLLARY 3.3 *If ϕ is a sentence and f, g any two M-assignments, then $\mathcal{M} \models_{\mathrm{f}} \phi$ iff $\mathcal{M} \models_{\mathrm{g}} \phi$.*

Thus, the following definition makes sense.

DEFINITION 3.6 If ϕ is a sentence, $\mathcal{M} \models \phi$ iff for some f, $\mathcal{M} \models_f \phi$.

We state another lemma for later use; it follows almost directly from the truth definition:

LEMMA 3.4 *If $\mathrm{Var}_\phi \setminus X = \mathrm{Var}_\phi \setminus Y$, then $\mathcal{M}, X \models_{\mathrm{f}} \phi$ iff $\mathcal{M}, Y \models_{\mathrm{f}} \phi$.*

As an example of how the truth definition works, let us spell out the truth conditions for the sentence $[Px, (Qy, Rxy)]$. This sentence is true in $\mathcal{M}$ iff for some f, $\mathcal{M} \models_f [Px, (Qy, Rxy)]$, and we have

$\mathcal{M} \models_f [Px, (Qy, Rxy)]$

$\Longleftrightarrow \mathcal{M}, \emptyset \models_f [Px, (Qy, Rxy)]$

$\Longleftrightarrow$ for all $a \in M$, if $\mathcal{M}, \{x\} \models_{f(x/a)} Px$ then $\mathcal{M}, \{x\} \models_{f(x/a)} (Qy, Rxy)$

$\Longleftrightarrow$ for all $a \in P^{\mathcal{M}}$ there is $b \in M$ such that $\mathcal{M}, \{x, y\} \models_{f(x/a, y/b)} Qy$ and $\mathcal{M}, \{x, y\} \models_{f(x/a, y/b)} Rxy$

$\Longleftrightarrow$ for all $a \in P^{\mathcal{M}}$ there is $b \in Q^{\mathcal{M}}$ such that $\langle a, b \rangle \in R^{\mathcal{M}}$, so the sentence is equivalent to $\forall x(Px \rightarrow \exists y(Qy \wedge Rxy))$. For $[Px, (Qx, Rxy)]$, on the other hand, since $x \in \{x\}$, the last two lines become, with $f(y) = b$,

$$\iff \text{for all } a \in P^{\mathcal{M}},\ \mathcal{M}, \{x\} \models_{f(x/a)} Qx \text{ and } \mathcal{M}, \{x\} \models_{f(x/a)} Rxy$$

$$\iff \text{for all } a \in P^{\mathcal{M}},\ a \in Q^{\mathcal{M}} \text{ and } \langle a, b\rangle \in R^{\mathcal{M}}.$$

This is the truth condition of the formula $\forall x(Px \rightarrow (Qx \wedge Rxy))$.

Negation can be introduced as usual:

$$\neg\phi \overset{\text{def}}{=} [\phi, \bot]$$

It follows that

$$\mathcal{M}, X \models_f \neg\phi \iff \text{not } \mathcal{M}, X \models_f \phi$$

The truth definition gives the formula (ϕ, ψ) the same truth conditions as $\neg[\phi, \neg\psi]$ (whether or not ϕ and ψ have common variables), so we may use only $[\ ,\ ]$, $\bot$ and the comma as logical symbols in PFO. The comma could also be eliminated, by writing

$$[\phi][\psi]$$

instead of $[\phi, \psi]$ (cf. (i) at the end of section 1). Having the comma saves on the number of parentheses, though.

4 FORMALIZATION IN PFO

We shall not give precise formalization rules here, but the basic ideas can be made clear from examples. In these examples we sometimes indicate intermediate steps from an English sentence to its formalization, using expressions that mix English and PFO syntax.

(1) Pedro owns a donkey

translates as

(1′) (*donkey y* , *p owns y*)

Likewise,

(2) A farmer owns a donkey

first goes into

(*farmer x* , *x owns a donkey*)

and the same rule that takes us from (1) to (1′) then gives

(2′) (*farmer x* , (*donkey y* , *x owns y*))

Similarly, from

(3) Every man loves a woman

we get the usual two formalizations

(3′) $[man\ x\ ,\ (woman\ y\ ,\ x\ loves\ y)]$

(3″) $(woman\ y\ ,\ [man\ x\ ,\ x\ loves\ y])$

Next, consider

(4) If Pedro owns a donkey he is rich

We translate first to

$[Pedro\ owns\ a\ donkey\ ,\ he\ is\ rich]$

and then as before to

(4′) $[(donkey\ y\ ,\ p\ owns\ y)\ ,\ rich\ p]$

which is the desired implication, with an existential quantification in the antecedent. Now, exactly the same procedure works for

(5) If Pedro owns a donkey he beats it

namely, first

$[Pedro\ owns\ a\ donkey\ ,\ he\ beats\ it]$

and then

(5′) $[(donkey\ y\ ,\ p\ owns\ y)\ ,\ p\ beats\ y]$

This is a universal quantification, with conjunction in the antecedent, as desired.[7] Likewise, by these rules

(6) If a farmer owns a donkey he beats it

translates into

(6′) $[(farmer\ x\ ,\ (donkey\ y\ ,\ x\ owns\ y))\ ,\ x\ beats\ y]$

where both x and y are universally quantified. Note that (2) is a subsentence of (6), and (2′) a subformula of (6′).

Now consider some examples with relative clauses.

(7) Every farmer who owns a donkey is rich.

Here are the translation steps:

$[farmer\ who\ owns\ a\ donkey\ x\ ,\ rich\ x]$

[7] In (4) and (5) we have, for simplicity, treated "he" just as a substitute for the proper name.

$[(farmer\ x\ ,\ x\ owns\ a\ donkey)\ ,\ rich\ x]$

(7′) $[(farmer\ x\ ,\ (donkey\ y\ ,\ x\ owns\ y))\ ,\ rich\ x]$

Again, exactly the same steps translate

(8) Every farmer who owns a donkey beats it

as

(8′) $[(farmer\ x\ ,\ (donkey\ y\ ,\ x\ owns\ y))\ ,\ x\ beats\ y]$

i.e., (6′), which is the desired result.

In general, NP's of the form "a(n) N" like the conjunction "and" both translate with (ϕ, ψ), whereas NP's "every N" and "if (then)" translate with $[\phi, \psi]$. If the antecedent ϕ of the latter has the form (χ, θ), and the consequent ψ has variables in common with it, these variables become universally quantified.[8]

So far, no distinction between indefinite and quantified NP's w.r.t. formalization has been made. However, only indefinite NP's in the antecedent of a conditional allow 'donkey anaphora'. If we apply the rules hinted at by the previous examples to a universally quantified NP in that position we may get wrong results:

(9) Pedro owns every donkey

(9′) $[donkey\ y\ ,\ p\ owns\ y]$

(10) If Pedro owns every donkey he is rich

(10′) $[[donkey\ y\ ,\ p\ owns\ y]\ ,\ rich\ p]$

(11) If Pedro owns every donkey he beats it

(11?) $[[donkey\ y\ ,\ p\ owns\ y]\ ,\ p\ beats\ y]$

(9′) and (10′) are correct formalizations, but (11?) is not (it is equivalent to $\forall y(donkey\ y \rightarrow p\ owns\ y) \rightarrow p\ owns\ y)$).

Since PFO is a variant of predicate logic, no distinction corresponding to the one between indefinite and quantified NP's is built into it. Instead, the distinction can be made at the level of the formalization rules. Suppose f is the formalization function induced by these rules. We assume that the sentences in the fragment have been disambiguated w.r.t. scope dependencies and anaphoric relations by means of some suitable *indexing*. Call the thus indexed sentences *readings*. f is then a function from readings to PFO formulas. As indicated

[8] First-order PFO fails to give sentences such as *If I have a coin in my pocket I will put it in the meter* the preferred 'weak' donkey reading, just as DPL or (first-order) DRT. One way to account for such readings is within an extended framework allowing generalized quantifiers and plurals. We would then universally quantify over sets of all coins in my pocket, and in the consequent there would be a choice between saying that I put *every* element in such a set in the meter (strong reading), or *some* element in the set (weak reading).

above, the rules also produce and operate on syntactic objects in between (indexed) sentences and PFO formulas. For convenience, we include those among the readings too.

Now, suppose that, in formalization, a formula ϕ with quantified x is embedded in a larger formula. When can x be 'reused' outside of ϕ (thereby cancelling the quantification)? Here is a first answer. Call x *f, S-quantified at ϕ* if S is a reading with $f(S) = \phi$ and some N is such that (i) a noun phrase Det+N has wide scope in S, and (ii) $f(\mathrm{N}+x)$ is a subformula of ϕ. Next, for ϕ in the range of f, call x *f-quantified in ϕ* if there is a subformula ψ of ϕ (not necessarily proper) and a reading S' such that x is f, S'-quantified at ψ. For example, y is f-quantified in (*man x* , [*donkey y* , *x owns y*]), since it is f, S'-quantified at [*donkey y* , *x owns y*], where S' is *x owns every donkey*$_2$ (which appears in the formalization of *a man*$_1$ *owns every donkey*$_2$). Now we can state the following restriction:

(*) *A variable f-quantified in* $[\phi, \psi]$ *is not to be used outside* $[\phi, \psi]$.

The effect of (*) is to rule out certain readings, thus restricting the domain and range of f. In particular, it forbids anaphoric interpretations of pronoun occurrences in certain readings. It rules out (11?) but permits

(11′) [[*donkey y* , *p owns y*] , *p beats z*]

(from the reading *if Pedro*$_1$ *owns every donkey*$_2$ *he*$_1$ *beats it*$_3$). For another familiar example, consider

(12) Every donkey loves a farmer who cares for it

A simple analogy with (3) would seem to give us the two formalizations

(12′) [*donkey x* , ((*farmer y* , *y cares for x*) , *x loves y*)]

(12?) ((*farmer y* , *y cares for x*) , [*donkey x* , *x loves y*])

but only (12′) is correct, since *a farmer* cannot have wide scope in (12) unless *it* is anaphoric to something outside the sentence. (*) rules out (12?) but permits

(12″) ((*farmer y* , *y cares for z*) , [*donkey x* , *x loves y*])

Likewise, given that $\neg\phi$ abbreviates $[\phi, \bot]$, (*) rules out the formalization

[¬(*student x* , *sleeps x*) , *tired x*]

of the sentence

(13) If no student sleeps he is tired

which is correct—(13) cannot in fact be read anaphorically.

Next, consider a Bach-Peters case, where the definite NP gets a Russellian analysis:

(14) A pilot who sighted it downed the Mig that chased him

Here are some translation steps:

$(pilot\ who\ sighted\ it\ x\ ,\ x\ downed\ the\ Mig\ that\ chased\ him)$

$((pilot\ x\ ,\ x\ sighted\ it)\ ,\ (the\ Mig\ that\ chased\ him = y\ ,\ x\ downed\ y))$

$((pilot\ x\ ,\ x\ sighted\ it)\ ,\ (((Mig\ y\ ,\ y\ chased\ him)\ ,\ [(Mig\ z\ ,\ z\ chased\ him)\ ,\ z = y])\ ,\ x\ downed\ y))$

where we have left the pronouns in place. Now, replacing *it* by x and *him* by y gives the Bach-Peters reading. We could also replace either one, or both, by new variables; this gives the (perfectly reasonable) readings where the pronouns are anaphoric to something outside the sentence.[9]

The same kind of formalization works if we replace *a pilot* in (14) by *the pilot*, or *the Mig* by *a Mig*. On the other hand,

(15) Every pilot who sighted it downed every Mig that chased him

should have no reading where *every Mig that chased him* binds *it*. This follows from (*); first we get

$[(pilot\ x\ ,\ x\ sighted\ it)\ ,\ [(Mig\ y\ ,\ y\ chased\ him)\ ,\ x\ downed\ y]]$

and now (*) prevents replacement of *it* by y, but permits replacing *him* by x. We can, however, get the reasonable formalization

(15′) $[(pilot\ x\ ,\ x\ sighted\ z)\ ,\ [(Mig\ y\ ,\ y\ chased\ x)\ ,\ x\ downed\ y]]$

Disjunctive sentences ϕ *or* ψ can be formalized with the operator $\{\cdot, \cdot\}$, defined as

$$\{\phi, \psi\} \stackrel{\text{def}}{=} [\neg\phi, \psi]$$

We can then uniformly formalize sentences with disjunctive noun phrases by means of this operator and existential generalization[10], as shown in the following steps:

(16) Bill owns a horse or a donkey

$(x = a\ horse\ or\ x = a\ donkey,\ Bill\ owns\ x)$

$(\{x = a\ horse\ ,\ x = a\ donkey\}\ ,\ Bill\ owns\ x)$

(16′) $(\{(horse\ y\ ,\ x = y)\ ,\ (donkey\ z\ ,\ x = z)\}\ ,\ Bill\ owns\ x)$

[9]Similar readings were possible for a number of the previous example sentences, though we didn't bother to mention them. Bach-Peters sentences appear to be a problem for DPL, due to the essential 'left to right direction' of that system.

[10]The following analysis resulted as a response to a question by Lauri Karttunen.

This method slightly complicates formalizations, but applies to definite and indefinite NP disjuncts alike (e.g., *the horse or the donkey*, *the horse or a donkey*) and also allows a uniform treatment of indefinites, since no special clauses are needed for their occurrences in disjunctive NPs. The treatment of donkey anaphora also comes out as desired:

(17) If Bill owns a horse or a donkey, then he beats it

(17′) $[(\{(horse\ y\ ,\ x = y)\ ,\ (donkey\ z\ ,\ x = z)\}\ ,\ Bill\ owns\ x)\ ,\ Bill\ beats\ x]$

The formalization is compositional. The difference in availability of anaphoric readings between this case and

(18) If Bill owns a horse or Bill owns a donkey, then he beats it

is brought out. Not only can no variable can replace '?' in

(18′) $[\{(horse\ x\ ,\ Bill\ owns\ x)\ ,\ (donkey\ y\ ,\ Bill\ owns\ y)\}\ ,\ Bill\ beats\ ?]$

and capture an anaphoric relation to both indefinite noun phrases, but, because of the definition of $\{\cdot, \cdot\}$, the choice of x or y as substituend is also a violation of (*). Note also that (*) forbids replacing the last occurrence of x in (17′) by y or z, again as desired, since there is no reading of (17) on which "it" is anaphoric just to "a horse" or just to "a donkey".

In PFO, generic and non-generic readings of sentences like *It don't mean a thing if it ain't got that swing* get the same formalization, which allows a straightforward compositional analysis e.g. of *If it's not in Dutch, then it don't mean a thing if it ain't got that swing.* Note that in DPL, like in PL (and DRT), a formalization of (the generic reading of) the first sentence cannot occur as a subformula of a formalization of the generic reading of the second.

Finally, consider simple cross sentence anaphora of the following kind:

(19) Pedro owns a donkey. He feeds it. It frightens a girl. She hates it.

In predicate logic we formalize such a text as a conjunction. This is particularly easy in PFO: just add on the relevant formulas using the operation $(\cdot, \cdot)$:

(19′) $((((donkey\ y\ ,\ p\ owns\ y)\ ,\ p\ feeds\ y)\ ,\ (girl\ z\ ,\ y\ frightens\ z))\ ,\ z\ hates\ y)$

(19′) is logically equivalent to

(19″) $\exists y \exists z ((((donkey\ y \wedge p\ owns\ y) \wedge p\ feeds\ y) \wedge (girl\ z \wedge y\ frightens\ z)) \wedge z\ hates\ y)$

but every 'conjunct' in (19) corresponds to a subformula of (19′), which is not the case for (19″)). Some of the conjuncts in (19) introduce individuals which are referred to by pronouns in later conjuncts. In ordinary predicate logic the existential quantification must then be 'raised' to a position where its scope includes those later conjuncts. No such 'raising' is required in PFO.

As a last example of the use of (*), note that it predicts the fact that in

(20) A student didn't arrive. He was tired.

(21) Not every student arrived. He was tired.

the pronoun can be anaphoric in (20) but not in (21) (even though the initial sentences of (20) and (21) are logically equivalent).[11]

5 PFO AND PREDICATE LOGIC

Consider, for simplicity, PFO with $(\cdot,\cdot)$ as a defined operator, and predicate logic PL with $=$, $\bot$, $\rightarrow$, $\forall$ as primitive logical symbols. For a PL-formula ϕ, $FV(\phi)$ is the set of free variables in ϕ.

To translate from PFO to PL we simply follow the definition of satisfaction for PFO.

DEFINITION 5.1 For a PFO-formula ϕ and $X \subseteq \mathit{Var}$, define inductively the PL-formula $\phi^{+,X}$ by

(a) $\phi^{+,X} = \phi$, if ϕ is atomic,

(b) $[\phi,\psi]^{+,X} = \forall x_1 \ldots \forall x_n(\phi^{+,X\cup\{x_1,\ldots x_n\}} \rightarrow \psi^{+,X\cup\{x_1,\ldots x_n\}})$,
where $x_1,\ldots,x_n$ are the elements of $\mathit{Sbound}_{[\phi,\psi]} \setminus X$ (in some fixed order).

DEFINITION 5.2 $\phi^+ = \phi^{+,\emptyset}$

A straightforward induction shows

PROPOSITION 5.1 $\mathcal{M}, X \models_{\mathrm{f}} \phi \Longleftrightarrow \mathcal{M} \models^{PL}_{\mathrm{f}} \phi^{+,X}$

Clearly, the same variables occur in ϕ and $\phi^{+,X}$. Concerning the *free* variables, we have the following

PROPOSITION 5.2 *If $\mathit{Free}_\phi \subseteq X \subseteq \mathit{Var}_\phi$, then $FV(\phi^{+,X}) = X$. In particular, if ϕ is a sentence, so is ϕ^+.*

Proof: It is clear from Definition 5.1 that $X \subseteq FV(\phi^{+,X})$. Suppose $z \in \mathit{Var}_\phi \setminus X$. Then $z \in \mathit{Bound}_\phi$ and there is a unique largest subformula $[\psi,\theta]$ of ϕ such that $z \in \mathit{Sbound}_{[\psi,\theta]}$. Translating from the 'outside', we arrive at $[\psi,\theta]$ and must form $[\psi,\theta]^{+,Y}$ for some Y such that $X \subseteq Y$. But the variables in $Y \setminus X$ are surface bound in larger subformulas of ϕ than $[\psi,\theta]$. Therefore,

[11]There are, however, some (more or less convincing) counterexamples to (*) involving negation, which can be accommodated by allowing double negation elimination before applying (*). We also note that with this rule PFO can handle Barbara Partee's example

Either there is no bathroom in this house or it is in a funny place

which creates problems for DPL and DRT. Given that *either ϕ or ψ* is rendered by $[\neg\phi,\psi]$, and *there is no χ which is θ* by $\neg(\chi,\theta)$, we get the formalization $[[[(Bx,Hx),\bot],\bot],Fx]$, which, after a double negation elimination, becomes $[(Bx,Hx),Fx]$, which is permitted by (*).

$z \in \mathit{Sbound}_{[\psi,\theta]} \setminus Y$, and z is universally quantified at that step in the translation. Thus, z is bound in $\phi^{+,X}$. $\square$

COROLLARY 5.3 *When ϕ is a PFO-sentence, $\mathcal{M} \models^{PFO} \phi$ iff $\mathcal{M} \models^{PL} \phi^+$.*

In fact, it is not hard to see that $^+$ is an *injection* from the set of PFO-formulas into the set of PL-formulas.

Translating in the opposite direction is slightly more interesting.

DEFINITION 5.3 If ϕ is a PL-formula, define the PFO-formula ϕ^* inductively as follows:

(a) $\phi^* = \phi$, if ϕ is atomic

(b) $(\phi \to \psi)^* = [\phi^*, \psi^*]$

(c) $(\forall x \phi)^* \;=\; [x = x \;,\; \psi^*]$

Note that $(Px \to Qx)^* = [Px, Qx]$, which is equivalent to $\forall x(Px \to Qx)$, so free variables can become bound in this translation, and the meaning of formulas is not in general preserved. However, if we 'mark' the free variables of ϕ, meaning will be preserved for *strict* formulas:

DEFINITION 5.4 A PL-formula is called *strict*, if (i) no variable occurs both free and bound in it, (ii) all quantifiers use distinct variables, and (iii) there is no vacuous quantification.

Every PL-formula is of course logically equivalent to a strict PL-formula. In PFO, strictness is built into the syntax (Proposition 3.1).

PROPOSITION 5.4 *For a strict PL-formula ϕ, $\mathcal{M} \models^{PL}_{\mathrm{f}} \phi$ iff $\mathcal{M}, FV(\phi) \models_{\mathrm{f}} \phi^*$.*

Examples like $\forall x Px \to Qx$ and $\forall x Px \to \forall x Qx$ show that the restriction to strict PL- formulas is necessary.

A variable occurs in ϕ iff it occurs in ϕ^*. Also, if ϕ is strict, $\mathit{Free}_{\phi^*} \subseteq FV(\phi)$. Thus, if ϕ is a sentence, so is ϕ^*, and we get

COROLLARY 5.5 *If ϕ is a strict PL-sentence, then $\mathcal{M} \models^{PL} \phi$ iff $\mathcal{M} \models^{PFO} \phi^*$.*

Proof of Proposition 5.4: By induction. The case of atomic formulas is clear. Consider first a formula $\forall x \phi$. By strictness, x is free in ϕ, and $FV(\phi) = FV(\forall x \phi) \cup \{x\}$. We have

$$\mathcal{M} \models^{\mathrm{PL}}_{f} \forall x \phi$$

$\iff$ for all $a \in M$, $\mathcal{M} \models^{\mathrm{PL}}_{f(a/x)} \phi$

$\Longleftrightarrow$ for all $a \in M$, $\mathcal{M}, FV(\phi) \models_{f(a/x)} \phi^*$ (ind. hypothesis)

$\Longleftrightarrow$ for all $a \in M$, $\mathcal{M}, FV(\phi) \models_{f(a/x)} x = x$ implies $\mathcal{M}, FV(\phi) \models_{f(a/x)} \phi^*$

$\Longleftrightarrow$ $\mathcal{M}, FV(\forall x\phi) \models_f [x = x\ ,\ \phi^*]$ (since x occurs in ϕ^*)

Now consider a formula $\phi \to \psi$. We have

(1) $\mathcal{M}, FV(\phi \to \psi) \models_f [\phi^*, \psi^*] \Longleftrightarrow$
$\mathcal{M}, FV(\phi \to \psi) \models_f \phi^*$ implies $\mathcal{M}, FV(\phi \to \psi) \models_f \psi^*$

For, let z be any surface bound variable in $[\phi^*, \psi^*]$. Then z occurs in both ϕ and ψ, hence it is *free* in both ϕ and ψ, by the assumption of strictness. So (1) follows from the definition of satisfaction. Now, $FV(\phi \to \psi) = FV(\phi) \cup FV(\psi)$, and we observe next that

(2) $Var_{\phi^*} \setminus FV(\phi) = Var_{\phi^*} \setminus (FV(\phi) \cup FV(\psi))$

since any variable in $Var_{\phi^*} \setminus FV(\phi)$ is bound in ϕ, hence does not occur at all in ψ. From (2), it follows by Lemma 3.4 that

$$\mathcal{M}, FV(\phi \to \psi) \models_f \phi^* \Longleftrightarrow \mathcal{M}, FV(\phi) \models_f \phi^*$$

and similarly for ψ^*. Hence, using the induction hypothesis and (1),

$$\mathcal{M}, FV(\phi \to \psi) \models_f [\phi^*, \psi^*] \Longleftrightarrow \mathcal{M} \overset{\text{PL}}{\models_f} \phi \to \psi$$

and the result is proved. □

Note that $(\forall x Px)^* = (x = x \to Px)^*$, so the function * is not one-one. But this is the only type of exception. In fact, one easily proves that * is a *bijection* from the set of PL-formulas (strict or not) with no subformulas of the form $x = x \to \psi$ to the set of PFO- formulas.

Remark 1: The prohibition of vacuous quantification in strict formulas is because if x does not occur in ϕ, it will be bound in $\forall x\phi$ but free in $[x = x\ ,\ \phi^*]$. However, if we define a translation function ** as in Definition 5.3 except that (c) is changed to

$$(\forall x\phi)^{**} = [x = x\ ,\ [x = x\ ,\ \phi^{**}]],$$

vacuous quantification can be permitted in strict PL-formulas; Proposition 5.4 and Corollary 5.5 will still hold for **.

Remark 2 (On the complexity of the PFO syntax in the Chomsky hierarchy): Just as for PL, the set of PFO-formulas is context free. It might be thought that the set of PFO-*sentences* should be simpler than the set of PL- sentences, since in order to decide whether an occurrence of x in a PFO formula ϕ is free or not, one needs only check if there is an occurrence of x in another atomic formula: if so, x is bound, otherwise it is free. Nevertheless, it follows from the Pumping Lemma for context free languages (Hopcroft and Ullman, p. 125 ff.) that this set is not context free.

To see this, let, for some fixed non-logical vocabulary, $L_{PFO}(L_{PL})$ be the corresponding set of PFO- (PL-)sentences. There is no number k such that any $s \in L_{PFO}$ of length $\geq k$ is decomposable into $uvwzy$ such that

(i) length$(vwz) \leq k$

(ii) length$(vz) > 0$

(iii) $uv^iwz^iy \in L_{PFO}$ for $i \geq 0$

For assume there is such a k and let s be $(x_n = x_m, x_n = x_m)$, where n and m indicate index strings the lengths of which are different and exceed k. Clearly the only substrings of length $\leq k$ we can iterate as in (iii) while preserving *formulahood* are substrings of the index strings. But these iterations will have to be made either within a single index string or within index strings of different variables, since occurrences of the same variable are too far apart (cf. clause (i)). Hence, already one iteration will destroy sentencehood. □

Let $L^{**} = \{\phi^{**} : \phi \in L_{PL}\}$. L^{**} is a fragment of L_{PFO} with the same expressive power as L_{PFO} itself. For any $\phi \in L_{PL}$, ϕ^{+**} is a sentence of L^{**} with the same meaning (in the sense of section 9). L^{**} is structurally rather like L_{PL}: every variable first occurs in subformulas $(x = x, (x = x, \ldots)$ or $[x = x, [x = x, \ldots]$ before any other occurrences to the right, just as occurrences in $\forall x$ or $\exists x$ in L_{PL}, and, as in L_{PL}, there need be no further occurrences. We conjecture that L_{PL} and L^{**} have the same complexity, i.e., can be generated by the same kind of grammars. Indeed, the *indexed grammar* for L_{PL} (with $=$ as only primitive predicate) given in Marsh and Partee (1984) can be straightforwardly modified into a grammar for L^{**}. Marsh and Partee conjecture that the fragment of L_{PL} which results by eliminating sentences with vacuous quantification is not an indexed language, and for partly the same reasons we believe that L_{PFO} itself is of greater complexity than L^{**}. (*End of remark*)

If ϕ is a PFO-sentence, ϕ^{+} is strict, so we have

$$\mathcal{M} \overset{\text{PFO}}{\models} \phi \Longleftrightarrow \mathcal{M} \overset{\text{PFO}}{\models} \phi^{+*}$$

although usually ϕ and ϕ^{+*} are distinct. Likewise, if ϕ is a strict PL-sentence,

$$\phi \leftrightarrow \phi^{*+}$$

is valid.

In conclusion, there are satisfaction-preserving translations for *sentences* between PFO and the strict fragment of PL. Furthermore, if we stipulate that free variables in PL-formulas are 'marked' in the PFO truth definition, these translations are satisfaction-preserving for *formulas* as well.

6 DEFINABILITY AND CONSEQUENCE IN PFO

The last remark in the previous section indicates what model theory in PFO looks like: notions involving sentences are as usual, whereas for formulas we have to use 'marking' of variables. For example, $R \subseteq M^n$ is $\mathcal{M}$-*definable in* PL iff there is a PL-formula ϕ with $FV(\phi) = \{x_1, \ldots, x_n\}$ such that for all $a_1, \ldots, a_n \in M$,

$$\langle a_1, \ldots, a_n \rangle \in R \Longleftrightarrow \mathcal{M} \overset{\text{PL}}{\underset{\{x_i/a_i\}}{\models}} \phi$$

Correspondingly, we say that $R \subseteq M^n$ is $\mathcal{M}$-*definable in* PFO iff there is a PFO-formula ϕ and $X = \{x_1, \ldots, x_n\}$ with $\mathit{Free}_\phi \subseteq X \subseteq \mathit{Var}_\phi$ such that for all $a_1, \ldots, a_n \in M$,

$$\langle a_1, \ldots, a_n \rangle \in R \Longleftrightarrow \mathcal{M}, X \underset{\{x_i/a_i\}}{\models} \phi$$

PROPOSITION 6.1 *$R \subseteq M^n$ is $\mathcal{M}$-definable in PL iff it is $\mathcal{M}$- definable in PFO.*

Proof: Suppose R is $\mathcal{M}$-definable in PL by a formula ϕ. We may assume that ϕ is strict. But then it follows from Proposition 5.4 that R is $\mathcal{M}$-definable in PFO by ϕ^* with $X = FV(\phi)$, since clearly $\mathit{Free}_{\phi^*} \subseteq X \subseteq \mathit{Var}_{\phi^*}$. Now suppose R is $\mathcal{M}$-definable in PFO by a formula ψ with a set of 'marked' variables X. It follows from Propositions 5.1 and 5.2 that R is $\mathcal{M}$-definable in PL by $\psi^{+,X}$. $\square$

As another example, consider the model theoretic notion of an *elementary extension*: $\mathcal{M} \prec_{PL} \mathcal{N}$ iff $\mathcal{M} \subseteq \mathcal{N}$ and for every PL-formula ϕ and every M-assignment f, if $\mathcal{M} \overset{\text{PL}}{\underset{f}{\models}} \phi$ then $\mathcal{N} \overset{\text{PL}}{\underset{f}{\models}} \phi$. Here is the corresponding PFO version: $\mathcal{M} \prec_{PFO} \mathcal{N}$ iff $\mathcal{M} \subseteq \mathcal{N}$ and for every PFO-formula ϕ, every $X \subseteq \mathit{Var}_\phi$, and every M-assignment f, if $\mathcal{M}, X \underset{f}{\models} \phi$ then $\mathcal{N}, X \underset{f}{\models} \phi$. Again, it follows from Propositions 5.1 and 5.4 that $\mathcal{M} \prec_{PL} \mathcal{N}$ iff $\mathcal{M} \prec_{PFO} \mathcal{N}$.

The notion of *logical consequence*, on the other hand, involves only sentences. Let Γ be a set of PFO-sentences and ϕ a PFO-sentence:

DEFINITION 6.1 $\Gamma \overset{\text{PFO}}{\models} \phi$ iff no model makes all the sentences in Γ true and ϕ false.

Clearly,

$$\Gamma \overset{\text{PFO}}{\models} \phi \Longleftrightarrow \Gamma^+ \overset{\text{PL}}{\models} \phi^+$$

(where $\Gamma^+ = \{\psi^+ : \psi \in \Gamma\}$). Likewise, if Γ is a set of strict PL-sentences and ϕ a strict PL-sentence, then

$$\Gamma \overset{\text{PL}}{\models} \phi \Longleftrightarrow \Gamma^* \overset{\text{PFO}}{\models} \phi^*$$

7 NATURAL DEDUCTION

PFO allows a rather elegant formulation of natural deduction, with just one introduction rule and one elimination rule, plus rules for negation and identity. Here deducibility involves *sentences*, so we shall need to assume that there is always a sufficient number of individual constants around to perform instantiations. In the two rules below, $[\phi, \psi]$ is assumed to be a sentence with $Var_\phi \cap Var_\psi = \{x_1, \dots, x_n\}$, and $\phi(t_1, \dots, t_n)$ is the result of simultaneously replacing *all* occurrences (free or bound) of x_i in ϕ by t_i. The rules are presented in the usual informal way. $\phi^\dagger$ marks that the assumption ϕ has been discharged (killed).

INTR

$$\begin{array}{c} \phi(c_1, \dots, c_n)^\dagger \\ \vdots \\ \psi(c_1, \dots, c_n) \\ \hline [\phi, \psi] \end{array}$$

where $\{x_1, \dots, x_n\} = Var_\phi \cap Var_\psi$ and $c_1, \dots, c_n$ do not occur in ϕ, ψ or open assumptions in the derivation of $\psi(c_1, \dots, c_n)$, except $\phi(c_1, \dots, c_n)$.

ELIM

$$\frac{\phi(t_1, \dots, t_n) \qquad [\phi, \psi]}{\psi(t_1, \dots, t_n)}$$

where $\{x_1, \dots, x_n\} = Var_\phi \cap Var_\psi$ and $t_1, \dots, t_n$ are closed terms.

In addition, we have the rule for (classical) negation:

NEG

$$\begin{array}{c} \neg\phi^\dagger \\ \vdots \\ \bot \\ \hline \phi \end{array}$$

It is easy to show (in the usual way) that NEG needs only be stated for atomic ϕ. Only the identity rules remain. Let t, t_1, t_2 be closed terms.

ID-axiom $\quad t = t$

ID-rule

$$\frac{t_1 = t_2 \qquad \phi}{\phi'}$$

where ϕ' results from ϕ by replacing some occurrences of t_1 by t_2

When ϕ and ψ have no common variables, INTR and ELIM amount to the usual introduction and elimination rules for implication. They also yield the familiar forms of introduction and elimination of universal quantification as *derived rules*, here called ∀-INTR and ∀-ELIM.

∀-INTR

$$\frac{[\phi(c_1, \ldots, c_n), \psi(c_1, \ldots, c_n)]}{[\phi, \psi]}$$

where $\{x_1, \ldots, x_n\} \subseteq Var_\phi \cap Var_\psi$ and $c_1, \ldots, c_n$ do not occur in ϕ, ψ or open assumptions in the derivation of $[\phi(c_1, \ldots, c_n), \psi(c_1, \ldots, c_n)]$.

∀-ELIM

$$\frac{[\phi, \psi]}{[\phi(t_1, \ldots, t_n), \psi(t_1, \ldots, t_n)]}$$

where $\{x_1, \ldots, x_n\} \subseteq Var_\phi \cap Var_\psi$ and $t_1, \ldots, t_n$ are closed terms.

Note that we need only instantiate *some* of the common variables in ∀-ELIM. Here is a derivation of ∀-INTR. Let $Var_\phi \cap Var_\psi = \{x_1, \ldots, x_n, y_1, \ldots, y_k\}$ and let $d_1, \ldots, d_k$ be new individual constants.

$$\dfrac{\dfrac{\phi(c_1, \ldots, c_n, d_1, \ldots, d_k)^\dagger \qquad [\phi(c_1, \ldots, c_n, y_1, \ldots, y_k), \psi(c_1, \ldots, c_n, y_1, \ldots, y_k)]}{\psi(c_1, \ldots, c_n, d_1, \ldots, d_k)}\ \text{ELIM}}{[\phi(x_1, \ldots, x_n, y_1, \ldots, y_k), \psi(x_1, \ldots, x_n, y_1, \ldots, y_k)]}\ \text{INTR}$$

∀-ELIM is similar.

Let $\Gamma \overset{\text{PFO}}{\vdash} \phi$ mean that there is a derivation of ϕ with open assumptions in Γ, and write $\phi_1, \ldots, \phi_k \overset{\text{PFO}}{\vdash} \phi$ instead of $\{\phi_1, \ldots, \phi_k\} \overset{\text{PFO}}{\vdash} \phi$. The superscript will often be omitted. We now give some properties of this deducibility relation.

If ϕ is any formula with $Var_\phi = \{x_1, \ldots, x_n\}$, and $c_1, \ldots, c_n$ are new constants,

$$\frac{\phi(c_1, \ldots, c_n)^\dagger}{[\phi, \phi]}$$

is an instance of INTR, which proves (1):

(1) $\vdash [\phi, \phi]$, for any formula ϕ

(2) If $Var_\phi \cap Var_\psi = Var_\psi \cap Var_\theta = Var_\phi \cap Var_\theta$, then $[\phi,\psi],[\psi,\theta] \vdash [\phi,\theta]$.

(3) If $Var_\phi \cap Var_\psi = \emptyset$, then $\Gamma \cup \{\phi\} \vdash \psi \Longleftrightarrow \Gamma \vdash [\phi,\psi]$.

The rules INTR and ELIM have corresponding derived rules for the operator $(\cdot,\cdot)$:

$$\text{(,)-INTR} \quad \frac{[\phi(t_1,\ldots,t_n),\psi(t_1,\ldots,t_n)]}{[\phi,\psi]}$$

where $\{x_1,\ldots,x_n\} = Var_\phi \cap Var_\psi$ and $t_1,\ldots,t_n$ are closed terms.

$$\text{(,)-ELIM} \quad \frac{(\phi,\psi) \qquad \begin{array}{c}\phi(c_1,\ldots,c_n)^\dagger \quad \psi(c_1,\ldots,c_n)^\dagger\\ \vdots\\ \theta\end{array}}{\theta}$$

where $\{x_1,\ldots,x_n\} = Var_\phi \cap Var_\psi$ and $c_1,\ldots,c_n$ do not occur in ϕ, ψ or θ, nor in open assumptions in the derivation of θ, except possibly $\phi(c_1,\ldots,c_n)$ and $\psi(c_1,\ldots,c_n)$.

Here is a derivation of (,)-ELIM:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\phi(c_1,\ldots,c_n)^\dagger \quad \psi(c_1,\ldots,c_n)^\dagger\\ \vdots\\ \theta\end{array} \qquad [\theta,\bot]^\dagger}{\bot}\,\text{ELIM}}{[\phi(c_1,\ldots,c_n),\bot]}\,\text{INTR}}{[\phi,[\psi,\bot]]}\,\text{INTR} \qquad [[\phi,[\psi,\bot]],\bot]}{\bot}\,\text{ELIM}}{\theta}\,\text{NEG}$$

(note that $[[\phi,[\psi,\bot]],\bot] = (\phi,\psi)$). From these rules we can get the following:

(4) If (ϕ,ψ) is a sentence then $(\phi,\psi) \vdash (\psi,\phi)$.

(5) Under the same conditions on variables as in ∀-ELIM,

$$\exists\text{-INTR} \qquad \frac{(\phi(t_1,\ldots,t_n),\psi(t_1,\ldots,t_n))}{(\phi,\psi)}$$

(6) If $\{x_1,\ldots,x_n\} \subseteq \mathit{Var}_\phi \cap \mathit{Var}_\psi$ and $c_1,\ldots,c_n$ are not in ϕ, ψ, θ or in any open assumption other than $(\phi(c_1,\ldots,c_n),\psi(c_1,\ldots,c_n))$, then

$$\exists\text{-ELIM} \qquad \frac{(\phi,\psi) \qquad \begin{array}{c}(\phi(c_1,\ldots,c_n),\psi(c_1,\ldots,c_n))^\dagger\\ \vdots \\ \theta\end{array}}{\theta}$$

(7) If $\mathit{Var}_\phi \cap \mathit{Var}_\psi = \emptyset$, then $\phi,\psi \vdash (\phi,\psi)$. (∧-INTR)

(8) If $\mathit{Var}_\phi \cap \mathit{Var}_\psi = \emptyset$, then $(\phi,\psi) \vdash \phi$. (∧-ELIM.1)

(9) If $\mathit{Var}_\phi \cap \mathit{Var}_\psi = \emptyset$, then $(\phi,\psi) \vdash \psi$. (∧-ELIM.2)

These examples should make it plausible that our rules are sufficient for PFO. That this is indeed the case follows from the

THEOREM 7.1 (Completeness Theorem) *If* $\Gamma \overset{PFO}{\models} \phi$ *then* $\Gamma \overset{PFO}{\vdash} \phi$*.*

This can be proved from the completeness theorem for PL via the translations described in section 5, but it is easier to give Henkin's argument directly for PFO. Here is a brief outline: It suffices to show that every consistent set of sentences has a model, where Γ is *consistent* if not $\Gamma \overset{\mathrm{PFO}}{\vdash} \bot$. Call Γ *full*, if whenever $(\phi,\psi) \in \Gamma$ with $\mathit{Var}_\phi \cap \mathit{Var}_\psi = \{x_1,\ldots,x_n\}$, $(\phi(c_1,\ldots,c_n),\psi(c_1,\ldots,c_n)) \in \Gamma$ for some $c_1,\ldots,c_n$ in the corresponding language. One shows as usual (making use of ∃-ELIM) that every consistent Γ can be extended to a maximally consistent and full set Σ in a language obtained by adding an infinite set C of individual constants. From Σ the Henkin model $\mathcal{M}_\Sigma$ is constructed in the standard way. One verifies by induction that

(10) If $\mathit{Free}_\psi \subseteq \{x_1,\ldots,x_n\}$, f is an assignment, and $c_1,\ldots,c_n \in C$, then
$$\mathcal{M}_\Sigma, \{x_1,\ldots,x_n\} \models_{f(x_i/[c_i])} \psi(x_1,\ldots,x_n) \Longleftrightarrow \mathcal{M}_\Sigma \overset{\mathrm{PFO}}{\models} \psi(c_1,\ldots,c_n)$$

(where $[c]$ is the equivalence class of c under the usual equivalence relation on the set of individual constants). Using this, together with the fullness and maximal consistency of Σ (hence closure under deductive consequences, in particular ∃-INTR), the usual inductive proof of

(11) For all sentences ϕ, $\phi \in \Sigma \iff \mathcal{M}_\Sigma \overset{\text{PFO}}{\models} \phi$

goes through, and the proof is complete. □

8 PFO, PL AND DPL

We have said that PFO, in contrast with DPL, is basically a variant of PL. In this section we elaborate briefly on this claim, mainly by looking a bit closer at the respective 'truth definitions' in these systems.

The first thing to note, however, is that the claim *cannot* be substantiated by means of the usual model theoretic notion of *expressive power*. Two logics are equivalent in this sense if for each sentence in the first logic there is a sentence in the second with the same class of models, and vice versa. The translations in section 5 show that this is indeed the case for PL and PFO, but similar translations exist between DPL, or DRT, and PL. In terms of the expressive power of sentences, PL, DPL, DRT and PFO are all *equivalent*.

Beyond expressive power, there is no accepted general standard for comparing logics. A natural suggestion, though, is to look at the properties of the respective notions of *logical consequence*. Such a comparison substantiates our claim. It is well known that the (preferred) notion of logical consequence in DPL is quite different from the one in PL. For example, it is not reflexive or transitive. Furthermore, although the relation $\overset{\text{DPL}}{\models}$ is axiomatizable, no explicit and natural axiomatization exists to date, in spite of efforts to find one. On the other hand, as we saw in section 7, the relation $\overset{\text{PFO}}{\models}$ is easily axiomatized, by a straightforward adaption of a familiar axiomatization of $\overset{\text{PL}}{\models}$. We tentatively conclude that the relations of logical consequence in PL and PFO are essentially the *same*, whereas the logical consequence relation of DPL would appear to differ in some important way.

To sharpen the comparison between our three systems we need to look at their respective basic satisfaction relations. Here the essential difference between DPL and PL, one supposes, is that the former but not the latter is *dynamic*. To see how PFO fares in this respect, it is worth while considering for a moment precisely what this dynamism consists in.

The usual PL semantics associates (given a model) with each formula a set of assignments, whereas the usual DPL semantics associates a binary relation between assignments with each formula. Let us say we have a semantics in terms of *truth conditions* in the first case, and in terms of *input/output conditions* in the second.

Making the dynamic/static distinction in terms of input/output conditions vs. truth conditions is standard (cf., for example, van Benthem 1991). But of course the mere *existence* of a semantics of one of these kinds for a formalism tells us nothing about its dynamic nature. For, each semantics ($\overset{4}{\models}$) in terms of input/output conditions *induces* a truth conditional semantics ($\overset{3}{\models}$):

(1) $\mathcal{M} \overset{3}{\underset{f}{\models}} \phi \iff \exists g\, \mathcal{M}, g \overset{4}{\underset{f}{\models}} \phi.$

This is how DPL gets its truth conditional semantics. And conversely, each truth conditional semantics ($\overset{3}{\models}$) *trivially* generates a semantics in terms of input/output conditions ($\overset{4}{\models}$) such that (1) holds: just let $\mathcal{M}, g \overset{4}{\underset{f}{\models}} \phi$ iff $f = g$ and $\mathcal{M} \overset{3}{\underset{f}{\models}} \phi$.

Thus, both DPL and PL can in principle be viewed from a dynamic *and* a static *perspective*. The point of saying that DPL but not PL is dynamic is rather that the input/output semantics for DPL is fundamental in some sense: it is the intended semantics, it has a recursive definition, in contrast with the induced truth conditional semantics, etc., and correspondingly for PL.

One may think a dynamic perspective on PL is pointless. The trivial input/output conditions mentioned above certainly add nothing of interest. And if the PL syntax is equipped with a *real* input/output semantics, the result, it might seem, is DPL, not PL. Actually, things are a little more complex than that. This can be seen from the following alternative input/output semantics for PL, which lies between the trivial semantics and the one given by DPL.

Take the definition of the standard 4-place satisfaction relation for DPL, with $\neg$, $\wedge$, and $\exists$ as primitive, and just change the clause for $\wedge$ to intersection instead of composition:

DEFINITION 8.1

(i) $\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} Pt_1 \ldots t_n \iff f = g$ and $\langle t_1^{\mathcal{M},f}, \ldots, t_n^{\mathcal{M},f} \rangle \in P^{\mathcal{M}}$

(ii) $\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} \neg\phi \iff f = g$ and for no h, $\mathcal{M}, h \overset{\mathrm{d}}{\underset{f}{\models}} \phi$

(iii) $\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} \phi \wedge \psi \iff f = g$ and $\exists p\, \mathcal{M}, p \overset{\mathrm{d}}{\underset{f}{\models}} \phi$ and $\exists q\, \mathcal{M}, q \overset{\mathrm{d}}{\underset{f}{\models}} \psi$

(iv) $\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} \exists x\phi \iff \exists h(f[x]h$ and $\mathcal{M}, g \overset{\mathrm{d}}{\underset{h}{\models}} \phi)$

($f[x]h$ means that the assignments f and h differ at most on the variable x).

Recall that the DPL semantics is like this except that (iii) is replaced by

(iii′) $\mathcal{M}, g \overset{\mathrm{DPL}}{\underset{f}{\models}} \phi \wedge \psi \iff \exists h(\mathcal{M}, h \overset{\mathrm{DPL}}{\underset{f}{\models}} \phi$ and $\mathcal{M}, g \overset{\mathrm{DPL}}{\underset{h}{\models}} \phi)$

But even with (iii), it makes *some* sense to think of Definition 8.1 as an input/output semantics, because of clause (iv). We have $\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} \exists xPx$ iff $f[x]g$ and $g(x) \in P^{\mathcal{M}}$ just as in DPL. Likewise, the clause for negation is the same as in DPL, and hence the non-duality of $\exists$ and $\forall$ in terms of input/output conditions:

$$\mathcal{M}, g \overset{\mathrm{d}}{\underset{f}{\models}} \forall x\phi \iff f = g \text{ and } \forall h(f[x]h \Rightarrow \mathcal{M}, h \overset{\mathrm{d}}{\underset{f}{\models}} \phi)$$

However, even though this is a non-trivial dynamic semantics, *it is still a semantics for PL, and not for some other system.* This is due to two things: (i) the standard syntactic notion of free/bound variables in PL corresponds to the semantic notion of free/bound variables implicit in Definition 8.1; (ii) the truth conditional semantics induced by Definition 8.1 is the ordinary PL semantics. The latter claim follows from the (easily proved)

PROPOSITION 8.1 $\mathcal{M} \models^{PL}_{\mathrm{f}} \phi \iff \exists g\, \mathcal{M}, g \models^{d}_{\mathrm{f}} \phi$

Although PL can be seen in this not-completely-trivial dynamic perspective, its truth conditional semantics is, of course, the fundamental one: it does not need a dynamic counterpart but 'stands on its own'. But the above semantics locates the *precise* point where DPL differs from PL from a dynamic perspective: in treating conjunction as composition and not intersection. Perhaps this, then, is the essential characteristic of a dynamic logic.

Now, what about PFO? It would seem that its truth conditional semantics is not fundamental: only the 4-place satisfaction relation was recursively defined in section 3.2. On the other hand, conjunction is not treated as composition. Also, the semantics associates with each formula not a relation between assignments but a relation between an assignment and a variable set—hardly an input/output relation. Our distinctions are perhaps not yet sharp enough to settle the matter. However, we shall now point to one way to settle it, although we do not claim it is the final word on the issue.

It turns out, perhaps surprisingly, that PFO does have a recursive truth conditional semantics, provided we make one small generalization of the previous set-up. The generalization is to *allow partial assignments*, i.e., functions from subsets of *Var* (including $\emptyset$) to universes of models. This is a rather insignificant move, easily carried through for PL and DPL. One only needs to see to it that $\mathcal{M} \models^{\mathrm{PL}}_{f} \phi$ (or $\mathcal{M}, g \models^{\mathrm{DPL}}_{f} \phi$) implies that the free variables of ϕ (in the respective sense) are in $dom(f)$. There are in fact two options here: either let $\mathcal{M} \models^{\mathrm{PL}}_{f} \phi$ be *undefined* when $FV(\phi) \not\subseteq dom(f)$, or let it be *false* then. For definiteness, choose the last option, which is perhaps simplest.[12] Clearly, this is a trivial modification.

Now do the same for PFO, but this time add a *trick*: we also let $dom(f)$ be the set of 'marked' variables. This is feasible since free variables can always be treated as 'marked'—they are certainly not to be quantified. In addition, we put the 'marked' bound variables in $dom(f)$. The value of f for these is immaterial; the point is that f is defined for them, and undefined for the (surface-bound) variables that we do quantify.

Let f, g,... be partial assignments in what follows. If $dom(f) \cap X = \emptyset$, let $f \subseteq_X h$ mean that h extends f to X, i.e., $dom(h) = dom(f) \cup X$ and $h|dom(f) = f$. The revised truth definition is quite simple:

[12]But note that now, ϕ and $\neg\phi$ can both be false: the usual logical laws only hold when all relevant free variables are in the domain of the considered assignment.

DEFINITION 8.2

(i) $\mathcal{M} \models'_f Pt_1 \ldots t_n \iff Var_{Pt_1 \ldots t_n} \subseteq dom(f)$ and $\langle t_1^{\mathcal{M},f}, \ldots, t_n^{\mathcal{M},f} \rangle \in P^{\mathcal{M}}$

Let $(Var_\phi \cap Var_\psi) \setminus dom(f) = X$

(ii) $\mathcal{M} \models'_f (\phi, \psi) \iff Free_{(\phi,\psi)} \subseteq dom(f)$ & $\exists h \supseteq_X f$ ($\mathcal{M} \models'_h \phi$ and $\mathcal{M} \models'_h \psi$)

(iii) $\mathcal{M} \models'_f [\phi, \psi] \iff Free_{[\phi,\psi]} \subseteq dom(f)$ and $\forall h \supseteq_X f$ ($\mathcal{M} \models'_h \phi \Rightarrow \mathcal{M} \models'_h \psi$) [13]

Thus, if $x \in dom(f)$, x is not quantified. That is why we can *extend* f to the variables which are to be quantified; we do not, as with total assignments, need to *change* the value of f for those variables.

That this is just the old PFO notion of satisfaction in a new guise follows from the next result, which is not hard to prove.

PROPOSITION 8.2 *If f is total, $\mathcal{M}, X \models^{PFO}_{\mathrm{f}} \phi$ iff $\mathcal{M} \models'_{f|(\mathrm{Free}_\phi \cup X)} \phi$.*

So here we have a recursive truth conditional semantics for PFO. Furthermore, we can equip PFO with an input/output semantics just as for PL: Consider the PFO syntax with $\neg$ and $(\cdot, \cdot)$ as primitive operators:

DEFINITION 8.3

(i) $\mathcal{M}, g \models^{\mathrm{d}'}_f Pt_1 \ldots t_n \iff Var_{Pt_1 \ldots t_n} \subseteq dom(f)$ & $f \subseteq g$ & $\langle t_1^{\mathcal{M},f}, \ldots, t_n^{\mathcal{M},f} \rangle \in P^{\mathcal{M}}$

(ii) $\mathcal{M}, g \models^{\mathrm{d}'}_f \neg\phi \iff Free_\phi \subseteq dom(f)$ & $f \subseteq g$ & $Var_\phi \subseteq dom(g)$ & $\forall h \neg \mathcal{M}, h \models^{\mathrm{d}'}_f \phi$

(iii) $\mathcal{M}, g \models^{\mathrm{d}'}_f (\phi, \psi) \iff Free_{(\phi,\psi)} \subseteq dom(f)$ & $\exists h \supseteq_X f(\mathcal{M}, g \models^{\mathrm{d}'}_h \phi$ & $\mathcal{M}, g \models^{\mathrm{d}'}_h \psi)$, where $X = (Var_\phi \cap Var_\psi) \setminus dom(f)$.[14]

Again, it can be shown that the induced truth conditional semantics is the one we just gave:

PROPOSITION 8.3 $\mathcal{M} \models'_{\mathrm{f}} \phi \iff \exists g\ \mathcal{M}, g \models^{d'}_{\mathrm{f}} \phi$

[13] Although this is a truth conditional semantics in the present sense, it differs in certain respects from the corresponding PL semantics. In PL, if $f|FV(\phi) = g|FV(\phi)$, it follows (also when f,g are partial) that f satisfies ϕ in $\mathcal{M}$ iff g satisfies ϕ in $\mathcal{M}$. In PFO, on the other hand, one must assume $f|Var_\phi = g|Var_\phi$ to obtain the corresponding conclusion. Related to this is the fact that, whereas in PL the *truth* of ϕ in $\mathcal{M}$ can be defined as the satisfaction of ϕ in $\mathcal{M}$ by *any* assignment, in PFO we must require ϕ to be satisfied by the *empty* assignment (or by any assignment which is undefined for all variables in ϕ).

[14] This definition is related to Definition 8.2 as Definition 8.1 is related to the standard definition of satisfaction in PL, modulo some differences due to the use of partial assignments. For example, the condition '$f = g$' in the PL case is here replaced by '$f \subseteq g$'.

In conclusion, these results offer a quick way to substantiate the claim that PFO is essentially non-dynamic, just as PL, although there exist non-trivial dynamic perspectives on both systems. We suspect, however, that this is only one side of the coin. To get to the heart of the (non-)dynamic nature of DPL, PFO, PL, and similar systems, one should also compare in more general terms their respective *variable-binding* mechanisms. After all, DPL syntax is characterized by a very particular notion of variable-binding, and the same holds for PFO. It seems to us that there is more to say about the interplay between variable-binding and dynamics, but that is the subject of another paper.

9 PFO AND COMPOSITIONALITY

What is the *meaning* of a PFO formula? A natural suggestion is to include in semantic interpretation precisely as much as it takes to make the language compositional. In PL, we get a *truth value* of a formula ϕ, relative to a model $\mathcal{M}$ and an M-assignment f. That is, $[\![\phi]\!]_{\mathcal{M},f}$ is 1 or 0. The argument f cannot be dropped. In PFO, yet another argument, a set of variables X, is needed to ensure compositionality. The resulting generalization of the notion of truth conditions is, we feel, quite natural.

The recursive definition of the meaning function $[\![\phi]\!]$ for a formal language is often not given compositionally, but it can sometimes be rewritten in such a fashion. For example, the standard truth definition for PL is not compositional, but may be rewritten so that, for example, to the rule which from 'x' and 'ϕ' forms '$\exists x\phi$' corresponds the rule which from $[\![x]\!]$ and $[\![\phi]\!]$ gives the function which, for $\mathcal{M}$ and f, yields the value 1 iff $[\![\phi]\!]_{\mathcal{M},f(a/x)} = 1$ for some $a \in M$.

It is not entirely obvious that the truth definition for PFO in section 3 can be similarly rewritten, since the variables quantified over in, say, (ϕ, ψ), are not given as arguments to the formation rule. Nevertheless, this can be done, as we now indicate. As in section 8, the key is to allow *partial* M-assignments, but this time they are used in conjunction with the variable sets, to obtain a truly compositional semantics for PFO.

Let a model $\mathcal{M}$ be fixed. For a partial f and a term t, $t^{\mathcal{M},f}$ is now defined iff each variable in t is in $dom(f)$, and for $a_1, \ldots, a_n \in M$, $\{x_1, \ldots, x_n\} \subseteq dom(f(x_i/a_i))$ (even if $x_i \notin dom(f)$). We also let the meaning functions $[\![\phi]\!]$ be partial, so that $[\![\phi]\!]_{X,f}$ is defined iff $Free_\phi \cup (Var_\phi \cap X) \subseteq dom(f)$ (we omit the subscript $\mathcal{M}$). It is then straightforward to check that if $(Var_\phi \cap Var_\psi) \setminus X = \{x_1, \ldots, x_n\}$, $a_1, \ldots, a_n \in M$, $f' = f(x_i/a_i)$ and $X' = X \cup \{x_1, \ldots, x_n\}$, then $[\![(\phi, \psi)]\!]_{X,f}$ is defined iff $[\![\phi]\!]_{X',f'}$ and $[\![\psi]\!]_{X',f'}$ are both defined. This ensures that the switch to partial assignments in no way makes PFO partial; indeed the logic is just as before and we are only allowing that assignments may be undefined for irrelevant variables. Now, we can easily see that

$[\![\phi]\!]_{\emptyset,f}$ is defined iff $Free_\phi \subseteq dom(f)$

$[\![\phi]\!]_{Var,f}$ is defined iff $Var_\phi \subseteq dom(f)$

(*Var* is the set of all variables), and this allows us to recover Free_ϕ and Var_ϕ, which is the point of the present modification of the truth definition. Thus, let $\mathcal{K}$ be the class of PFO type meaning functions, i.e., partial functions from pairs (X, f) to $\{0, 1\}$, and let, for $G \in \mathcal{K}$,

$$\mathbf{F}_G = \cap\{dom(f) : G(\emptyset, f) \text{ is defined}\}$$

$$\mathbf{V}_G = \cap\{dom(f) : G(\mathit{Var}, f) \text{ is defined}\}$$

Also, for $X \subseteq \mathit{Var}$ and $G, H \in \mathcal{K}$, let

$$\mathbf{D}_{G,H,X} = (\mathbf{F}_G \setminus \mathbf{V}_H) \cup (\mathbf{F}_H \setminus \mathbf{V}_G) \cup ((\mathbf{V}_G \cup \mathbf{V}_H) \cap X)$$

(cf. the characterization of $\mathit{Free}_{(\phi,\psi)}$ at the end of section 3.1). Now, corresponding to the formation rule for $(\cdot, \cdot)$, we can define a function $\mathbf{O}_{(,)} : \mathcal{K} \times \mathcal{K} \longrightarrow \mathcal{K}$ as follows:

$$\mathbf{O}_{(,)}(G, H)(X, f) \quad \begin{cases} \text{def.} & \Longleftrightarrow \quad \mathbf{D}_{G,H,X} \subseteq dom(f) \\ = 1 & \Longleftrightarrow \quad \exists a_1, \ldots, a_n \in M\ (G(X', f') = H(X', f') = 1), \\ & \qquad \text{where } (\mathbf{V}_G \cap \mathbf{V}_H) \setminus X = \{x_1, \ldots, x_n\}, \\ & \qquad f' = f(x_i/a_i) \text{ and } X' = X \cup \{x_1, \ldots, x_n\} \\ = 0 & \Longleftrightarrow \quad \text{defined and} \neq 1 \end{cases}$$

Then, $\mathbf{O}_{(,)}(G, H)([\![\phi]\!], [\![\psi]\!]) = [\![(\phi, \psi)]\!]$. The function for $[\cdot, \cdot]$ is defined analogously, and we have a compositional mapping from syntactic to semantic structures.

There are few general arguments for the desirability of a compositional semantics. One such argument proceeds from the premiss that a language be learnable. As is generally recognized, however, this premiss only yields the conclusion that the meaning of a complex expression be effectively computable from the meaning of its parts and the mode of composition. Of course, a compositional semantics provides a particularly simple and elegant way of computing meanings of complex expressions, and for this reason the requirement of learnability has some relevance for the desirability of a compositional semantics, but the argument still falls far short of conclusiveness.

In the case of donkey sentences we see another, stronger, argument. A compositional treatment of (2) in section 2, the paradigmatic donkey sentence, allows us to avoid postulating an ambiguity in "a donkey", sometimes treating it as existential and sometimes as a universal quantifier. This is a good reason, since no such ambiguity is intuitively perceived.

In classical formulations of DRT (such as Kamp, 1981), no such ambiguity is postulated. On the other hand an additional level of semantic representations, discourse representation structures, is introduced, where "a donkey" is represented as a discourse referent (a free variable) together with a condition (that of being a donkey). That is, we have here two distinct notions of semantic content, one given by discourse representation structures and the other by the traditional notion of truth conditions, the latter being related to the former via embedding conditions.

At this point, we believe, a more general and more significant reason for compositionality emerges, namely, the requirement of a single, uniform notion of semantic content. The point of formal semantics is, we take it, to provide a theoretical account of linguistic practice. The reason for including semantics as part of such an account, i.e., for postulating semantic constructs, is the high degree of context *in*dependence in the way structured utterances contribute to our understanding of *speakers*, and to the expression of their thoughts and beliefs. We take semantic content to be a formal counterpart to this informal notion of the context invariant contribution of syntactic objects to the understanding of speakers. Now, traditionally the fundamental syntactic category is the sentence. The sentence has been thought of as the basic unit for depicting reality, since sentences are the smallest units that can be *evaluated* as right or wrong in this respect. It has also been regarded as the unit for articulating thoughts. These two functions of sentences can be seen as providing basic requirements on semantic concepts. Some, like Davidson, have thought that one range of semantic concepts can account for both functions, and some, like Frege, have thought otherwise. In both cases, however, the sentence is fundamental, and semantic descriptions of smaller units are to be evaluated according to their adequacy in contributing to descriptions of sentences. Seen in this light, the requirement of compositionality, at sentence level, reflects the desirability of a certain kind of explanation of semantic concepts: the semantic content of a sentence should both be specified by concepts explainable in the desired way and provide for its contribution to semantic contents of larger units—larger sentences or texts.

From this it is seen on the one hand that the requirement of compositionality has a basic conceptual motivation, and on the other hand that compositionality is not to be achieved at all costs. A compositional theory which uses a notion of sentence content for which no other justification can be given than that it provides a compositional treatment, does not automatically rank higher than a non-compositional alternative. Our problem is, of course, that traditional notions of sentence meaning, such as that of *truth conditions*, may be well justifiable but do not provide a strictly compositional treatment of the anaphoric constructions considered in this paper. The question is then whether there are natural extensions of the notion of truth conditions which allow a compositional analysis of these problematic kinds of linguistic context dependence. Specifically, does the suggested interpretation of PFO formulas have this property?

We believe so. Think of a set X of variables relevant for the evaluation of a formula ϕ, relative to a sequence (and a model), *in a context* as registering the impact of that context on the evaluation. Relative to a linguistic context the membership of a variable x in X indicates the occurrence of x elsewhere in the context, and thus that this variable is quantified in some clause superordinate to ϕ. This corresponds to certain anaphoric relations between noun phrases (in and, respectively, outside the natural language sentence formalized by ϕ). Relative to a non-linguistic context the membership of a variable x in X indicates some deictic interpretation of a corresponding pronoun, thus prohibiting a reading according to which x is quantified in ϕ. Conversely, for a given X we can think of the function $[\![\phi]\!]_{\mathcal{M}}(X, \cdot)$ from assignments to truth values as yielding the information about how ϕ contributes to or modifies the content of an embedding discourse in some non-linguistic context, the relevant factors of which are registered in X. If we modified our account by requiring that free variables in ϕ be included in the relevant sets X then we could say that the special case where X is the empty set, corresponding to our traditional notion of truth conditions, is simply the special case of zero context dependence (linguistic as well as non-linguistic).

We thus think we are justified in claiming that our notion of semantic content is a natural and legitimate generalization of the traditional notion of truth conditions; truth conditions in context, so to speak. But this notion of truth conditions in context is clearly similar in spirit e.g. to Heim's notion of file change potential (Heim, 1982) or Groenendijk and Stokhof's notion of (truth conditions *cum*) embedding conditions. What we have claimed on behalf of the PFO notion of semantic content could also, we believe, with justice be claimed on behalf of corresponding notions of DPL and varieties of DRT (perhaps in particular the compositional version in Zeevat, 1989).

Thus it seems that a comparative evaluation based on purely conceptual considerations will be inconclusive. Rather it will have to be based on more technical merits and demerits. We shall end here by making some comments on the relative advantages of PFO and DPL. A first point is that the binary quantification mode of PFO allows more faithful formalization of English quantified sentences than formalisms using the standard $\forall$ and $\exists$. On the other hand, the binding mechanisms of DPL are sometimes closer to natural English than those of PFO. Several features of natural English which are accounted for in DPL itself can only be accounted for in PFO by way of restrictions on variable choice in formalizations. For instance, the non-equivalence of *A man walks. He talks.* with *He talks. A man walks.* is directly reflected in DPL, whereas in PFO only restrictions on variable choice (in this case essentially Heim's novelty- familiarity conditions) tell us that ((*talks x* , (*man x, walks x*)) is not a formalization of the second text, despite the fact that ((*man x* , *walks x*) , *talks x*) is a formalization of the first.[15] Likewise, the unavailability of a directly anaphoric reading of *If every man walks, then he is tired* is directly accounted for by the binding mechanisms of DPL, whereas again in PFO we must rely on restrictions (cf. (*) of section 4). Nothing is, of course, lost, since the restrictions are part of the natural language semantics, but it is clear that in some cases the difference in meaning between a formula and what it formalizes is greater in PFO than in DPL.

This difference cuts both ways, however. Restrictions can easily be modified and thus allow greater flexibility. Cataphoric constructions, like Bach-Peters sentences (cf. (14), section 4) can in this way be accounted for in PFO, but the binding mechanisms of DPL block a straightforward compositional analysis. The same holds of examples like that of Partee (footnote 11, section 4), or the sentence *It don't mean a thing if it ain't got that swing*, discussed in section 4. Ultimately, however, a comparative evaluation must take into account the

[15] This observation was prompted by questions by Hans Kamp and Paul Dekker.

potential both of generalizing applications of the systems, e.g. for treating tenses, and of generalizing the systems themselves, e.g. of applying the basic binding principles in treating generalized quantifiers.

A AN INTUITIONISTIC VERSION OF PFO

PFO does not have a primitive disjunction, but the operator $\{\cdot,\cdot\}$ defined by

$$\{\phi, \psi\} \stackrel{\text{def}}{=} [\neg\phi, \psi]$$

appears to be what is needed for formalization of disjunction in natural language; cf. section 4. Note that $\{\phi, \psi\}$ means that *for all* $x_1, \ldots, x_n$, ϕ *or* ψ , where $x_1, \ldots, x_n$ are the variables common to ϕ and ψ. To get an intuitionistic version of PFO, however, it turns out that we need to fuse *existential* quantification and disjunction into one operator. Thus, we introduce the operator

$$\langle\phi, \psi\rangle$$

meaning that *for some* $x_1, \ldots, x_n$, ϕ *or* ψ (i.e., with a corresponding clause added to the PFO truth definition), and with the following introduction and elimination rules:

$\langle,\rangle$-INTR $\quad \dfrac{\phi(t_1, \ldots, t_n)}{\langle\phi, \psi\rangle} \qquad \dfrac{\psi(t_1, \ldots, t_n)}{\langle\phi, \psi\rangle}$ where $\{x_1, \ldots, x_n\} = Var_\phi \cap Var_\psi$ and $t_1, \ldots, t_n$ are closed terms.

$\langle,\rangle$-ELIM $\quad \dfrac{\langle\phi, \psi\rangle \quad \begin{array}{c}\phi(c_1, \ldots, c_n)^\dagger \\ \vdots \\ \theta\end{array} \quad \begin{array}{c}\psi(c_1, \ldots, c_n)^\dagger \\ \vdots \\ \theta\end{array}}{\theta}$ where $\{x_1, \ldots, x_n\} = Var_\phi \cap Var_\psi$ and $c_1, \ldots, c_n$ do not occur in ϕ, ψ or θ, nor in open assumptions in the derivation of θ, except possibly $\phi(c_1, \ldots, c_n)$ and $\psi(c_1, \ldots, c_n)$.

In classical PFO, $\langle\cdot,\cdot\rangle$ is definable in the following sense: for all ϕ, ψ with $\{x_1, \ldots, x_n\} = Var_\phi \cap Var_\psi$,

$$\langle\phi, \psi\rangle \text{ and } (x_1 = x_1, (\ldots, (x_n = x_n, \{\phi, \psi\}) \ldots)) \text{ are equivalent}$$

(i.e., satisfied by the same models, assignments and variable sets).[16]

[16] In contrast with $\{\phi, \psi\}$, $\langle\phi, \psi\rangle$ is not, it seems, uniformly definable in PFO, i.e., definable

Intuitionistic PFO, PFO_I, has a natural deduction system consisting of the introduction and elimination rules for $[\cdot,\cdot]$, $(\cdot,\cdot)$, $\langle\cdot,\cdot\rangle$, the identity rules, and the intuitionistic absurdity rule,

$$\mathrm{NEG}_I \qquad \frac{\bot}{\phi}$$

in place of the classical one. That this is really a system for intuitionistic first-order logic can be shown proof-theoretically. Let $\vdash^{\mathrm{PI}}$ be the derivability relation of PFO_I, and $\vdash^{\mathrm{SI}}$ the derivability relation for standard intuitionistic predicate logic. Extend the translations $^+$ and * of section 5 so that, where $\{x_1,\ldots,x_n\} = (\mathit{Var}_\phi \cap \mathit{Var}_\psi) \setminus X$, $\langle\phi,\psi\rangle^{+,X} = \exists x_1 \ldots \exists x_n(\phi^{+,X\cup\{x_1,\ldots,x_n\}} \vee \psi^{+,X\cup\{x_1,\ldots,x_n\}})$, and $(\phi \vee \psi)^* = \langle\phi^*,\psi^*\rangle$. Let $\phi \approx \psi$, where ϕ, ψ are PL_I-formulas, mean that ϕ is strict and results from ψ by bound variable changes and elimination of vacuous quantifications, and let $\Gamma \approx \Delta$ mean that $\approx$ is a 1-1 relation between the sets Γ and Δ. Then we can show that for each derivation in the one system there is a corresponding derivation in the other system, in the following sense:

THEOREM A.1

(a) *If* $\Gamma \vdash^{PI} \phi$*, then* $\Gamma^+ \vdash^{SI} \phi^+$

(b) $\Gamma \vdash^{SI} \phi$ *iff there are* $\Gamma_0 \approx \Gamma$ *and* $\phi_0 \approx \phi$ *such that* $\Gamma_0 \vdash^{SI} \phi_0$

(c) *If* Γ,ϕ *are strict and* $\Gamma \vdash^{SI} \phi$*, then* $\Gamma^* \vdash^{PI} \phi^*$

(d) *If* $\Gamma^{+*} \vdash^{PI} \phi^{+*}$*, then* $\Gamma \vdash^{PI} \phi$

Outline of proof: (a): First, recall that for a PFO-formula $\phi = \phi(x_1,\ldots,x_n)$ with $x_1,\ldots,x_n$ among its (free or bound) variables, $\phi(c_1,\ldots,c_n)$ is the result of replacing *all* occurrences of $x_1,\ldots,x_n$ by $c_1,\ldots,c_n$, respectively. For a PL-formula ϕ, let $\phi[x_1,\ldots,x_n/c_1,\ldots,c_n]$ be the result of replacing all *free* occurrences of $x_1,\ldots,x_n$ by $c_1,\ldots,c_n$, respectively. We need the following fact, for any PFO-formula $\phi = \phi(x_1,\ldots,x_n,y_1,\ldots,y_m)$:

(1) $\phi^{+,\{x_1,\ldots,x_n,y_1,\ldots,y_m\}}[x_1,\ldots,x_n/c_1,\ldots,c_n] = \phi(c_1,\ldots,c_n,y_1,\ldots,y_m)^{+,\{y_1,\ldots,y_m\}}$

---

from a fixed scheme. The distinction between uniformly and non-uniformly definable operators is characteristic of the PFO style of variable-binding. In PL, for example, a binary quantifier Q (of type $\langle 1,1\rangle$) is definable iff there is a PL-formula $\theta = \theta(P_1,P_2)$ with two unary predicate symbols P_1 and P_2 such that

$$\models Qx(P_1x, P_2x) \leftrightarrow \theta(P_1,P_2)$$

Then, for any formulas ϕ, ψ with at most x free,

$$\models Qx(\phi,\psi) \leftrightarrow \theta(\phi,\psi)$$

so the definition for atomic formulas gives a uniform defining scheme.

Note that $x_1, \ldots, x_n, y_1, \ldots, y_m$ have only free occurrences in $\phi^{+,\{x_1,\ldots,x_n,y_1,\ldots,y_m\}}$. (1) is easily proved by induction.

Now, (a) is proved by induction over the number of steps in the derivation of ϕ from Γ. When this number is 0, ϕ is either an identity axiom or an assumption ($\Gamma = \{\phi\}$); in both cases (a) holds for ϕ. The induction steps proceeds by cases, each case defined by the deduction rule applied in the last step of the derivation. Here we only consider the case of $\langle,\rangle$-ELIM. The other cases are similar.

Thus, suppose we have a derivation in PFO_I from $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ to θ of the form

$$\dfrac{\begin{matrix}\Gamma_1 \\ \\ \langle\phi,\psi\rangle\end{matrix} \qquad \begin{matrix}\Gamma_2 \quad \phi(c_1,\ldots,c_n)^{\dagger} \\ \vdots \\ \theta\end{matrix} \qquad \begin{matrix}\Gamma_3 \quad \psi(c_1,\ldots,c_n)^{\dagger} \\ \vdots \\ \theta\end{matrix}}{\theta}$$

with $\{x_1,\ldots,x_n\} = \mathit{Var}_\phi \cap \mathit{Var}_\psi$. By induction hypothesis, we have $\Gamma_1^+ \vdash^{\mathrm{SI}} \langle\phi,\psi\rangle^+$, $\Gamma_2^+ \cup \{\phi(c_1,\ldots,c_n)^+\} \vdash^{\mathrm{SI}} \theta^+$, and $\Gamma_3^+ \cup \{\psi(c_1,\ldots,c_n)^+\} \vdash^{\mathrm{SI}} \theta^+$. Then we also have

$$\dfrac{\begin{matrix} \\ \\ \phi(c_1,\ldots,c_n)^+ \vee \psi(c_1,\ldots,c_n)^+\end{matrix} \qquad \begin{matrix}\Gamma_2^+ \quad \phi(c_1,\ldots,c_n)^{+\dagger} \\ \vdots \\ \theta^+\end{matrix} \qquad \begin{matrix}\Gamma_3^+ \quad \psi(c_1,\ldots,c_n)^{+\dagger} \\ \vdots \\ \theta^+\end{matrix}}{\theta^+}$$

in SI. Call this derivation $\mathbf{\Pi}_0$. $\phi(c_1,\ldots,c_{n-1},x_n)^{+,\{x_n\}}[x_n/c_n] = \phi(c_1,\ldots,c_n)^+$, by (1), and similarly for ψ. Thus, we get by $\exists$-elimination in SI

$$\dfrac{\exists x_n(\phi(c_1,\ldots,c_{n-1},x_n)^{+,\{x_n\}} \vee \psi(c_1,\ldots,c_{n-1},x_n)^{+,\{x_n\}}) \qquad \mathbf{\Pi}_0}{\theta^+}$$

where the assumption $\phi(c_1,\ldots,c_n)^+ \vee \psi(c_1,\ldots,c_n)^+$ is discharged at the final step. Call this derivation $\mathbf{\Pi}_1$. Repeating this step we finally obtain a derivation

$$\frac{\exists x_1 \ldots \exists x_n(\phi^{+,\{x_1,\ldots,x_n\}} \vee \psi^{+,\{x_1,\ldots,x_n\}}) \qquad \mathbf{\Pi}_{n-1}}{\theta^+}$$

in SI, with open assumptions in $\Gamma_2^+ \cup \Gamma_3^+ \cup \{\exists x_1 \ldots \exists x_n(\phi^{+,\{x_1,\ldots,x_n\}} \vee \psi^{+,\{x_1,\ldots,x_n\}})\}$. But $\exists x_1 \ldots \exists x_n(\phi^{+,\{x_1,\ldots,x_n\}} \vee \psi^{+,\{x_1,\ldots,x_n\}}) = \langle \phi, \psi \rangle^+$. Add the derivation from Γ_1^+ to $\langle \phi, \psi \rangle^+$, and we have a derivation from $\Gamma_1^+ \cup \Gamma_2^+ \cup \Gamma_3^+$ to θ^+ in SI.

(b): This is a known fact.

(c): If $\Gamma \vdash^{\mathrm{SI}} \phi$, then there is a *normal* derivation of ϕ from Γ in SI (in the sense of Prawitz, 1965, ch. IV). By the Subformula Property, each formula in this derivation is a subformula of either ϕ or some formula in Γ. Since ϕ and Γ are strict, it follows that each formula in the derivation is strict. We show by induction over the length of the normal derivation that $\Gamma^* \vdash^{\mathrm{PI}} \phi^*$. The following two cases suffice for illustration:

(i) Suppose we have a normal derivation of ϕ from Γ where the last step is an $\rightarrow$-elimination from ψ and $\psi \rightarrow \phi$ to ϕ. Since ψ and $\psi \rightarrow \phi$ are strict, we have by induction hypothesis $\Gamma^* \vdash^{\mathrm{PI}} \psi^*$ and $\Gamma^* \vdash^{\mathrm{PI}} (\psi \rightarrow \phi)^*$. Now $(\psi \rightarrow \phi)^* = [\psi^*, \phi^*]$, and since $\psi \rightarrow \phi$ is a strict sentence, ϕ^* and ψ^* have no common variables. Hence, by [,]-ELIM, $\Gamma^* \vdash^{\mathrm{PI}} \phi^*$.

(ii) Suppose the last step is instead a $\forall$-introduction of $\forall x \phi$ from $\phi[x/c]$. By induction hypothesis, $\Gamma^* \vdash^{\mathrm{PI}} \phi[x/c]^*$. Since $\forall x \phi$ is strict, x has only free occurrences in ϕ, and it follows that $\phi[x/c]^* = \phi^*(c)$. Thus, by [,]-INTR, $[x = x\ ,\ \phi^*]$, i.e., $(\forall x \phi)^*$, is derivable in PFO_I.

(d): One proves by induction over the complexity of ϕ that

$$\phi \vdash^{\mathrm{PI}} \phi^{+*} \text{ and } \phi^{+*} \vdash^{\mathrm{PI}} \phi$$

from which (d) follows. □

POSTSCRIPT: After the first version of this paper was written, Östen Dahl and, independently, Filip Widebäck suggested that we compare PFO with the system of implicit quantification used by C. S. Peirce in his *existential graphs*—they thought the two styles of variable-binding would be rather similar. They were right. Peirce's original works are not easily penetrable, and the recent applications of existential graphs made by some workers in Artificial Intelligence (Sowa, 1984) are not presented from a logical point of view. However, Zeman (1967) in his *system of implicit quantification* (IQ) gives a modernized version of Peirce's ideas.[17] A quick account of IQ will show the similarities with PFO.

[17] Victor Sanchez drew our attention to Zeman's paper.

Zeman does not apply IQ to natural language semantics, nor does he give a recursive truth definition for it.[18] But informally the system is easy to understand.

The syntax is very simple: just the usual atomic formulas, and the Boolean connectives $\neg$ and $\wedge$. Other connectives are defined, e.g., $\phi \rightarrow \psi$ is an abbreviation of $\neg(\phi \wedge \neg\psi)$.

All variables in a formula are implicitly quantified (thus, there are no free variables) according to the following principle. If x occurs in ϕ, the *scope* of x in ϕ is the smallest subformula of ϕ containing *all* the occurrences of x in ϕ. Now insert $\exists x$ immediately before the scope of x in ϕ. Do this for all variables in ϕ (in some fixed order). The result is a PL-sentence with the same meaning as ϕ.

To see how this works we can consider how some typical anaphoric sentences of the present paper would be formalized in IQ. *If Pedro owns a donkey he is rich* becomes

$$(Dy \wedge Opy) \rightarrow Rp$$

i.e., $\neg((Dy \wedge Opy) \wedge \neg Rp)$. The scope of y is $(Dy \wedge Opy)$, so the sentence means $\exists y(Dy \wedge Opy) \rightarrow Rp$. For *If Pedro owns a donkey he is beats it*, on the other hand, we get

$$(Dy \wedge Opy) \rightarrow Bpy$$

where y has the scope $(Dy \wedge Opy) \wedge \neg Bpy$, so the PL-translation is $\neg\exists y((Dy \wedge Opy) \wedge \neg Rp)$, i.e., $\forall y((Dy \wedge Opy) \rightarrow Bpy)$. Note how the existential quantification of y in $(Dy \wedge Opy)$ is cancelled (and turned into universal quantification) in $(Dy \wedge Opy) \rightarrow Bpy$, by the occurrence of y in Bpy, much as in PFO.

Likewise, *Every farmer who owns a donkey beats it* would be formalized as

$$(Fx \wedge (Dy \wedge Oxy)) \rightarrow Bxy$$

which indeed means $\forall x \forall y((Fx \wedge (Dy \wedge Oxy)) \rightarrow Bxy)$. Finally, a text like *A farmer walks. He whistles.* can be formalized as

$$(Fx \wedge Wx) \wedge WHx$$

With respect to the features of variable-binding discussed in section 1, the IQ system does not really use variable-binding operators but is completely *implicit*. So the notion of selectiveness does not apply, but binding is *from the outside in* as in PFO.

The variable-binding mechanism of IQ is thus quite similar to PFO, but a difference is that there are no free variables in IQ. However, if we change IQ so that variables whose scope is an atomic formula are *not* quantified, the resulting mechanism is essentially that of PFO.

[18] Most of his paper deals with establishing that the translations from IQ to PL, and from PL to IQ, preserve derivability (he introduces a system of derivability rules for IQ). This would follow directly (by the completeness of PL) from the fact that they preserve logical consequence, which is easy to establish, once a proper truth definition has been given.

So in a sense, the PFO style of variable-binding is implicit in Peirce's work. Our contribution in this paper, then, is to present it in a modern format, to provide it with a compositional truth definition and a natural deduction proof system, and to apply it to natural language, in particular to certain constructions with quantification and anaphora.

Department of Philosophy
Stockholm University

References

van Benthem, J., 1991, "Logic and the flow of information", to appear in D. Prawitz *et al.*, *Proceedings of the 9th International Congress of Logic, Methodology, and Philosophy of Science*, North-Holland, Amsterdam. [Preprint as CSLI Report No. 91-160, Stanford University.]

Groenendijk, J., and Stokhof, M., 1991, "Dynamic predicate logic", *Linguistics and Philosophy* **14**, 39-100.

Heim, I., 1982, *The Semantics of Definite and Indefinite Noun Phrases*, Ph. D. Thesis, Univ. of Massachusetts.

Hopcroft, J., and Ullman, J., 1979, *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, Reading.

Kamp. H., 1981, "A theory of truth and semantic representation", in J. Groenendijk *et al.* (eds.), *Formal Methods in the Study of Language*, Math. Centre, Amsterdam, 277-322. [Reprinted in J. Groenendijk *et al.* (eds.), 1984, *Truth, Interpretation and Information*, Foris, Dordrecht, 1-41.]

Lewis, D., 1975, "Adverbs of quantification", in E. Keenan (ed.), *Formal Semantics of Natural Language*, Cambridge University Press, Cambridge, 3-15.

Montague, R., 1974, *Formal Philosophy*, ed. R. M. Thomason, Yale University Press, New Haven.

Marsh, W., and Partee, B., 1984, "How non-context free is variable binding?", in M. Cobler *et al.* (eds.), *Proceedings of the West Coast Conference on Formal Linguistics, Vol. 3*, 179-190.

Prawitz, D., 1965, *Natural Deduction*, Almqvist & Wiksell, Stockholm.

Sowa, J., 1984, *Conceptual Structures. Information Processing in Man and Machine*, Addison-Wesley, Reading, Mass.

Zeevat, H., 1989, "A compositional approach to discourse representation theory", *Linguistics and Philosophy* **12**, 95-131.

Zeman, J., 1967, "A system of implicit quantification", *The Journal of Symbolic Logic* **32**, 480-504.