FLEXIBLE VARIABLE-BINDING AND MONTAGUE GRAMMAR

Peter Pagin and Dag Westerståhl
Department of Philosophy, Stockholm University

1 Introduction

In [5] we introduced a version of predicate logic (PFO), with a new method of variable-binding, designed to handle some familiar anaphoric constructions in natural language compositionally. The idea to adapt a classical logical formalism to obtain a compositional account of certain linguistic binding phenomena is also the basis of Groenendijk and Stokhof's DPL [3], but their adaptation is different. DPL preserves the formulas of predicate logic, but changes the variable-binding mechanism and uses a dynamic semantics in the sense that the semantic value of a formula is a relation between assignments rather than a set of assignments. PFO on the other hand employs a new kind of formulas, with yet another variable-binding mechanism, but keeps essentially the standard semantics. The switch to a dynamic semantics is a substantial departure from classical predicate logic (PL), and we argued in [5] that whereas PFO is just a variant of PL, DPL is not.

The main interest of PFO, we feel, is that it is so simple and departs so little from PL. We will not here discuss the need for or the implications of a dynamic semantics (cf. [4] for a recent discussion). Regardless of this it is of interest to see how far one can handle the relevant anaphoric phenomena with a very slight change in the formal language.

PFO and DPL stop at sentence level. To obtain full compositionality à la Montague, one needs to extend the formalism to something like (intensional) type theory—Montague's IL. In this paper we present such an extension of PFO, called TFO. Here the difference with DPL becomes more drastic. The extension of the latter to Dynamic Montague Grammar (DMG) in [2] is not straightforward, since the dynamics and the DPL style variable-binding do not extend to the λ operator. Instead, explicit dynamic mechanisms are added to the formalism, via devices that were formerly used to handle intensions but now take on this further role. The extension of PFO to TFO, on the other hand, proceeds smoothly once one realizes how to do it. The basic PFO recipe for variable-binding works in the type theory too. The intensional apparatus has the same role as in IL. The expressive power is the same. The only thing that needs to be reviewed with some care is conversion—this is a syntactic phenomenon and TFO does have a different syntax than IL.

After some background on PFO style variable-binding we present the syntax and semantics of TFO, and then discuss conversion, in particular the Church-Rosser property and normalization. Finally we show how a standard Montague style translation into TFO handles familiar examples of donkey anaphora and cross sentence anaphora in a compositional way.

The paper is to be regarded as an extended abstract of a more detailed presentation. In particular, some definitions and all proofs have been omitted.

2 Background

The characteristics of PFO style variable-binding are that it is *unselective* and *reverses the binding order*: from the outside in, rather than the usual order from the inside out. Instead of standard connectives and quantifiers two binary variable-binding operators $[\cdot,\cdot]$ and $(\cdot,\cdot)$ are used, corresponding to universal quantification (implication) and existential quantification (conjunction), respectively. These bind (unselectively) *all common* variables of the two arguments, whether these variables were already bound in the arguments or not—such bindings are thus 'cancelled' (binding from the outside in). Details are given in [5], but a few sample formulas and their PL counterparts suffice to convey the idea:

$$\begin{array}{ll} [Ax, By] & Ax \rightarrow By \\ [Ax, Bx] & \forall x(Ax \rightarrow Bx) \\ (Axy, Bxyz) & \exists x \exists y(Axy \wedge Bxyz) \\ [(Axy, Bx), Czu] & \exists x(Axy \wedge Bx) \rightarrow Czu \\ [(Axy, Bx), Dzx] & \forall x(Axy \wedge Bx \rightarrow Dzx) \end{array}$$

In PFO the sentence

(1) If a man encounters a lion he runs from it

can be translated as

(1′) $[(Mx, (Ly, Exy)), Rxy]$

and

(2) A man walks. He talks.

as

(2′) $((Mx, Wx), Tx)$

Both translations are compositional at sentence level.

The reverse binding order of PFO necessitates an adjustment of the usual inductive definition of satisfaction. One way to do this is to let a set of variables X be an argument of the satisfaction relation. When you start evaluating a formula this set is usually empty, but quantified variables are successively put in the set as subformulas, subformulas of subformulas, etc., are evaluated, to prevent variables from being quantified again. That is, the satisfaction relation

$$\mathcal{M}, X \models_f \phi$$

between an assignment f, a formula ϕ, a model $\mathcal{M}$, and a set X of variables is defined so that the variables in X are never quantified. The ordinary ternary satisfaction relation is then obtained by letting $X = \emptyset$.

Before moving on to TFO we make two brief comments. The first is that it is primarily the reverse variable-binding of PFO, rather than the unselectivity, that makes anaphoric sentences come out right. To see this, consider a formalism which is exactly as PL *except* that variable-binding goes from the outside in. In such a system (1) could be translated

(1″) $\forall x \forall y(\exists x(Mx \wedge \exists y(Ly \wedge Exy)) \rightarrow Rxy)$

This has as subformulas the translations of *a man encounters a lion* and *he runs from it*, so it is compositional at sentence level. And, because of the reverse variable-binding, it gets the correct meaning—the existential quantifiers are 'cancelled' because they are within the scope of corresponding universal quantifiers.

Still, (1″) is rather ugly, and it is not quite obvious what the rule for 'if-then' would look like. The unselective PFO operators yield a more elegant system and more natural translation rules, and we will continue to use them in TFO.

Our second comment is that whereas with universal and existential quantification, selectivity vs. unselectivity is mere matter of style, this is not so with other quantifiers. In effect, unselective variable-binding allows quantification over *finite sequences* of individuals, which is an increase of expressive power with many generalized quantifiers. For example, suppose we add to PFO an operator $[_m\cdot,\cdot]_m$ corresponding to the determiner *most*. To evaluate a sentence $[_m\phi,\psi]_m$ relative to a model $\mathcal{M}$, we find the variables common to ϕ and ψ, say $x_1,\ldots,x_n$. Let R (S) be the set of n-tuples $(a_1,\ldots,a_n)$ such that the assignment of a_i to x_i satisfies ϕ (ψ) in $\mathcal{M}$. Then $[_m\phi,\psi]_m$ is true in $\mathcal{M}$ iff $|R \cap S| > |R - S|$. This PFO style generalized quantifier is stronger than the ordinary selective quantifier **most** which binds one variable only in each formula.[1] In the terminology of generalized quantifier theory, if you add a monadic generalized quantifier to PFO, you also obtain all the *resumptions* of that quantifier, because of the unselective variable-binding.

3 TFO

The terms of TFO are the same as those in IL, except that we use the two PFO operators instead of connectives and quantifiers. The *types* are the usual ones: basic types e and t, and complex types (a,b) and (s,a). So the *terms* T_a of type a are defined inductively as follows.

[1] A proof of this fact can be found in [6].

Definition 3.1 (TFO syntax)

(a) Variables and constants of type a are in T_a.

(b) $\perp \in T_t$.

(c) If $t, u \in T_a$ then $(t = u) \in T_t$

(d) If $\phi, \psi \in T_t$ then $[\phi, \psi], (\phi, \psi) \in T_t$.

(e) If $t \in T_{(a,b)}$ and $u \in T_a$ then $tu \in T_b$.

(f) If $t \in T_b$ and x is a variable of type a then $(\lambda x.t) \in T_{(a,b)}$.

(g) If $t \in T_a$ then ${}^\wedge t \in T_{(s,a)}$.

(h) If $t \in T_{(s,a)}$ then ${}^\vee t \in T_a$.

(i) If $\phi \in T_t$ then $\Box\phi \in T_t$.

Thus, we use the unselective PFO operators as well as the selective λ operator in the syntax. However, for *both* operators, the binding direction is from the outside in. Roughly: let an *x-binder*, for a variable x of any type, be a term of the form $(\lambda x.u)$, or of the form $[\phi, \psi]$ or (ϕ, ψ) where x occurs in both ϕ and ψ. Then an occurrence of x in t is bound by the *outermost* x-binder in t within whose scope it occurs. For example, the term

$$(Ax, Bx)$$

(with A and B constants of type (e, t)) expresses, as in PFO, that $A \cap B \neq \emptyset$. But in

$$\lambda.x(Ax, Bx)$$

the PFO binding is not in force, and the term denotes the set $A \cap B$. And the λ binding can in turn be 'cancelled' by a PFO binding further out, as in

$$((\lambda.x(Ax, Bx))y\ ,\ Cx)$$

which expresses that $A \cap B \cap C \neq \emptyset$. Note that since the λ binding is not in force here, the application of the λ term to y has no (semantic) effect.

A precise syntactic definition of binding in TFO is easily given, but the above examples should make the idea clear. The meaning of TFO terms is given by the next definition, which with each term t, model $\mathcal{M}$, possible word w, $\mathcal{M}$-assignment f, and set of variables X associates a denotation $[\![t]\!]_{\mathcal{M},w,f,X}$. As usual, $\mathcal{M}$ consists of a domain $D = D_e$, a set of possible worlds W and an interpretation function I. $D_t = \{0, 1\}$, and the domain is lifted to a function domain D_a for each type a in the usual way. I assigns functions from W to D_a to constants of type a, and an $\mathcal{M}$-assignment assigns a value in D_a to each variable of type a. If t is of type a, $[\![t]\!]_{\mathcal{M},w,f,X} \in D_a$.

Modulo the variable-set X, the clauses in the definition below are exactly the same as for IL, except the two clauses (d) and (f) dealing with variable-binding operators.

Definition 3.2 (TFO semantics)

(a) If x is a variable and C a constant of type a, then $[\![x]\!]_{\mathcal{M},w,f,X} = f(x)$ and $[\![C]\!]_{\mathcal{M},w,f,X} = I(C)(w)$.

(b) $[\![\perp]\!]_{\mathcal{M},w,f,X} = 0$.

(c) $[\![t = u]\!]_{\mathcal{M},w,f,X} = 1$ iff $[\![t]\!]_{\mathcal{M},w,f,X} = [\![u]\!]_{\mathcal{M},w,f,X}$.

(d) Suppose $(\mathit{Var}_\phi \cap \mathit{Var}_\psi) - X = \{x_1, \ldots, x_n\}$, where x_i is of type a_i. Then

$$[\![(\phi, \psi)]\!]_{\mathcal{M},w,f,X} = 1 \text{ iff there are } d_1 \in D_{a_1}, \ldots, d_n \in D_{a_n} \text{ such that}$$
$$[\![\phi]\!]_{\mathcal{M},w,f(x_i/d_i),X\cup\{x_1,\ldots,x_n\}} = [\![\psi]\!]_{\mathcal{M},w,f(x_i/d_i),X\cup\{x_1,\ldots,x_n\}} = 1.$$

Similarly for $[\phi, \psi]$, except that universal quantification and implication is used.

(e) $[\![tu]\!]_{\mathcal{M},w,f,X} = [\![t]\!]_{\mathcal{M},w,f,X}([\![u]\!]_{\mathcal{M},w,f,X})$.

(f) If $(\lambda x.t)$ is of type (a, b), then, for all $d \in D_a$,

$$[\![(\lambda x.t)]\!]_{\mathcal{M},w,f,X}(d) = \begin{cases} [\![t]\!]_{\mathcal{M},w,f,X} & \text{if } x \in X \\ [\![t]\!]_{\mathcal{M},w,f(x/d),X\cup\{x\}} & \text{if } x \notin X \end{cases}$$

(g) If $w' \in W$, then $[\![{}^\wedge t]\!]_{\mathcal{M},w,f,X}(w') = [\![t]\!]_{\mathcal{M},w',f,X}$.

(h) $[\![{}^\vee t]\!]_{\mathcal{M},w,f,X} = [\![t]\!]_{\mathcal{M},w,f,X}(w)$.

(i) $[\![\Box\phi]\!]_{\mathcal{M},w,f,X} = 1$ iff for all $w' \in W$, $[\![\phi]\!]_{\mathcal{M},w',f,X} = 1$

We also define
$$[\![t]\!]_{\mathcal{M},w,f} = [\![t]\!]_{\mathcal{M},w,f,\emptyset}$$

It is rather clear that TFO has the same expressive power as IL. To translate from TFO to IL, define for each TFO term t and each set X of variables an IL term $t^{+,X}$ inductively following Definition 3.1, distributing over the operators except in clauses (d) and (e) which read, respectively,

$$[\phi, \psi]^{+,X} = \forall x_1 \ldots \forall x_n(\phi^{+,X\cup\{x_1,\ldots,x_n\}} \to \psi^{+,X\cup\{x_1,\ldots,x_n\}})$$

(with $x_1, \ldots, x_n$ as in Definition 3.2 (d); $(\phi, \psi)^{+,X}$ is similar),

$$(\lambda x.t)^{+,X} = (\lambda x.t^{+,X\cup\{x\}})$$

It then follows that $[\![t]\!]_{\mathcal{M},w,f,X} = [\![t^{+,X}]\!]_{\mathcal{M},w,f}$, so in particular

$$[\![t]\!]_{\mathcal{M},w,f} = [\![t^{+,\emptyset}]\!]_{\mathcal{M},w,f}$$

To translate in the other direction simply note, first, that logical symbols $\forall$, $\wedge$, etc. can be eliminated in IL, second, that every IL term is equivalent to a *strict* term, i.e., one where no variable is both free and bound, nor quantified more than once, and third, that strict IL terms without logical symbols are also TFO terms and moreover mean the same in both systems.

4 Conversion

The fact that TFO has both selectively and unselectively binding operators makes conversion a little more complex than in IL. Also, the reverse binding direction means that we should keep track of the set of variables X in conversion. Let $\rightsquigarrow_X$ be the conversion relation relative to X between a redex and the result of performing β conversion. Here are some case where β conversion can *not* be performed: *Unless* $y \in X$,

$$(\lambda x.[Ax, Bx])y \not\rightsquigarrow_X [Ay, By]$$

$$(\lambda x.[Ax, By])y \not\rightsquigarrow_X [Ay, By]$$

$$(\lambda x.[Az, (\lambda y.(By, Cx))z])y \not\rightsquigarrow_X [Az, (\lambda y.(By, Cy))z]$$

The reason, of course, is that the usual constraint that no new bindings must be created by the substitution is violated. It is just that such bindings can arise in more ways than one in TFO. In fact, they may arise in three ways, illustrated by the above examples. Therefore, the most straightforward approach to conversion in TFO is to formulate explicitly the variable constraints on conversion and then verify that conversion under these constraints is sound.

Of course, there is another kind of constraint on β conversion, due to the presence of intensional operators. But these constraints are exactly as in IL. They lead to the failure of the Church-Rosser property—a failure which can be overcome by treating s as a regular type of possible words (cf., for example, [1], ch. 5). Thus, for simplicity, and to bring out the characteristic binding features of TFO, we restrict attention in this section to the *extensional part* of TFO. That is, we only consider terms as defined by Definition 3.1 (a)–(f). Hence possible worlds are not needed, and the denotation of a term t can be written $[\![t]\!]_{\mathcal{M},f,X}$.

If t and u are terms and x is a variable, let

$$[x/u]t$$

be the result of replacing *all* occurrences of x in t by u.[2]

The next definition gives the condition for this substitution to be permissible.

Definition 4.1 $P(x, t, u, X)$ is the conjunction of (a)–(c) below:

(a) if x is PFO-bound in t then $Var_u \subseteq X$

(b) if x occurs in one component of a subterm (ϕ, ψ) or $[\phi, \psi]$ of t and $y \in Var_u$ in the other, then $y \in X$

(c) if x occurs in a subterm $(\lambda y.v)$ of t where $y \in Var_u$ then $y \in X$.

[2]Here "occurrence" is to be taken in its literal sense, *except* that x is *not* taken to occur in 'λx'.

Now we say that

$$t \Rightarrow_X t'$$

if t' results from t by performing one β conversion: replacing an occurrence of a subterm $(\lambda x.u)v$ of t for which $P(x, u, v, Z)$ holds by $[x/v]u$, where Z results by adding to X the variables which become bound in the semantic evaluation process from t to the subterm $(\lambda x.u)v$. The following can now be proved.

Proposition 4.1 *If $t \Rightarrow_X t'$, then $[\![t]\!]_{\mathcal{M},f,X} = [\![t']\!]_{\mathcal{M},f,X}$.*

This approach to β conversion is a quite simple extension of the ordinary one, but it has one problem: the Church-Rosser property fails. Here is an example. We have

$$(\lambda z.((\lambda x.(Ax, Bz))z))y \Rightarrow_\emptyset (\lambda z.(Az, Bz))y$$

and

$$(\lambda z.((\lambda x.(Ax, Bz))z))y \Rightarrow_\emptyset (\lambda x.(Ax, By))y$$

but there is no way to continue the reduction to a common term. Note that changing bound variables (α conversion) will not help. In fact, whereas in ordinary λ calculus bound variables can always be chosen so that the variable constraint on substitution is satisfied, this is not so for TFO and the constraint $P(x, t, u, X)$.

In view of this one may try the following alternative approach. First, require, for $t \Rightarrow_X t'$, that the free variables of t are elements of X. This can always be achieved. More precisely, if Y is the set of free variables of t, one can α convert t to a term t_0 such that

$$[\![t]\!]_{\mathcal{M},f,X} = [\![t_0]\!]_{\mathcal{M},f,X\cup Y}$$

for all $\mathcal{M}$ and f.[3] Second, for such terms one may redefine substitution, incorporating α conversions in a way that always makes substitution permissible. Then there is no need for the condition P.

In the example above, both $(\lambda z.(Az, Bz))y$ and $(\lambda x.(Ax, By))y$ convert to (Ay, By) relative to a set of variables containing y. In general, it seems to us that the Church-Rosser property would hold with this approach to conversions. We hope to present the details in the full version of the paper.

Even if Church-Rosser fails on our first approach to conversion, normal forms always exist. This follows from the

Theorem 4.2 (Strong Normalization Theorem)
For every term t and every set of variables X: Every $\Rightarrow_X$-chain starting with t is finite.

[3]In PFO, this holds without any α conversion, since no variable can occur both free and bound in a PFO formula. But TFO does not have the latter property.

5 Montague grammar

A few examples will suffice to illustrate how a fragment of English, containing sentences with donkey anaphora, can be compositionally translated into TFO. To begin we again skip intensions, since they are unimportant for the first examples, and are handled in exactly the same way as in IL. Also, the analysis trees will be implicit—they are just as in ordinary Montague grammar, except that we shall use *indexing* to indicate anaphoric links. Let * be the translation function. Also, let '$\Rightarrow$' stand for (repeated) applications of $\Rightarrow_\emptyset$ as well as standard meaning postulates and intension-extension cancellations.

$$a_i^\star = \lambda Y.\lambda X.(Yx_i, Xx_i)$$

$$man^\star = M$$

$$walk^\star = W$$

$$(a_i\ man)^\star = a_i^\star man^\star = (\lambda Y.\lambda X.(Yx_i, Xx_i))M \Rightarrow \lambda X.(Mx_i, Xx_i)$$

$$(a_i\ man\ walks)^\star = (a_i^\star man^\star) walk^\star \Rightarrow (Mx_i, Wx_i)$$

$$he_i^\star = \lambda X.Xx_i$$

The next example, a donkey sentence, also illustrates a global constraint that the translation must satisfy: Always choose distinct bound variables in the translations. Otherwise unwanted binding may occur, due to the reverse binding order.

$$(if\ a_i\ man\ walks\ he_i\ sings)^\star = [(a_i\ man)^\star walk^\star, (he_i\ sings)^\star]$$

$$\Rightarrow [(\lambda X.(Mx_i, Xx_i))W\ ,\ (\lambda Y.Yx_i)S] \Rightarrow [(Mx_i, Wx_i), Sx_i]$$

The following examples use the PTQ meaning postulates and notation for extensional transitive verbs.

$$(a_1\ man\ encounters\ a_2\ lion)^\star = (a_1\ man)^\star(encounters^\star(a_2\ lion)^\star)$$

$$\Rightarrow (\lambda X.(Mx_1, Xx_1))(E(\lambda Y.(Lx_2, Yx_2))) \Rightarrow (Mx_1, (Lx_2, E_*(x_1, x_2)))$$

$$(if\ a_1\ man\ encounters\ a_2\ lion\ he_1\ runs\ from\ it_2)^\star$$

$$\Rightarrow [(Mx_1, (Lx_2, E_*(x_1, x_2)))\ ,\ R_*(x_1, x_2)]$$

Finally, we point out that we will also get donkey sentences with attitude verbs. For example,

$$(if\ a_1\ man\ believes\ he_1\ owns\ a_2\ donkey\ he_1\ wants\ to\ beat\ it_2)^\star$$

$$\Rightarrow [\,(Mx_1, BEL(x_1,^\wedge(Dx_2, O_*(x_1,x_2))))\;,\; WANT(x_1,^\wedge B_*(x_1,x_2))\,]$$

Again we get universal quantification over (men and) donkeys. The intuitive *de dicto* reading of this sentence certainly appears to universally quantify over *something*. The best candidate seems to be so called objects of thought. The reading can hardly be captured by any informal analysis that would give wide scope to an attitude verb. Hence examples such as this present a rather strong case for a quantificational treatment and for an ontology of objects of thought.

References

[1] Gamut, L.T.F., 1991, *Logic, Language and Meaning*, vol. 2, U Chicago Press, Chicago.

[2] Groenendijk, J., and Stokhof, M.: 1990, Dynamic Montague grammar, in L. Kálmán and L. Pólos (eds.), *Papers from the Second Symposium on Logic and Language*, Akadémiai Kiadó, Budapest, 3–48.

[3] Groenendijk, J., and Stokhof, M.: 1991, Dynamic predicate logic, *Linguistics and Philosophy* 14, 39–100.

[4] Israel, D.: 1994, The very idea of a dynamic semantics: an overview from the underground, this volume.

[5] Pagin, P., and Westerståhl, D.: 1993, Predicate logic with flexibly binding operators and natural language semantics, *Journal of Logic, Language and Information* 2, 89–128.

[6] Westerståhl, D.: 1989, Quantifiers in formal and natural languages, in D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, vol. IV, Reidel, Dordrecht, 1–131.