Discussion of Differences between the Embedding and PLM

Daniel Kirchner

March 9, 2017

1 Terms and Variables

The Principia explicitly distinguishes between terms and variables for all primitive types. Furthermore, it postulates (29.2)[2, p. 191], that for every term τ, which is not a definite description, there exists (a variable) β that is equal to τ: $\exists\,\beta\ (\beta = \tau)$. Thereby, any denoting term can be substituted for any variable (the substitution of identicals is an axiom). The only terms that may not denote are definite descriptions.

Mapping this distinction between terms and variable and the Principia's notion of non-denoting terms to the (non-partial) functional logic of Isabelle/HOL, where generally every free variable symbol can be substituted by any term of the same type, is not trivial. (In fact, there is ongoing work to investigate whether the embedding of free logic in HOL as studied in the papers by Christoph and Dana Scott could be fruitfully combined with the work presented here; the solution presented here has been worked out independently).

The solution in the current embedding is as follows: First, it does not explicitly distinguish between relation variables and relation terms and drops the corresponding axiom (29.2)[2, p. 191], which implicitly holds in our context (note that this is the case, although the equality is not primitive, but defined: the statement $\exists\,\beta.\ \beta = \tau$ for one-place relations for instance expands to $\exists\,\beta.\ \Box(\forall\,x.\ \{\!|x^P,\beta|\!\} \equiv \{\!|x^P,\tau|\!\})$. This still has to be proven, but without the distinction between variables and terms it suffices to show that $\Box(\forall\,x.\ \{\!|x^P,\tau|\!\} \equiv \{\!|x^P,\tau|\!\})$ which follows from the remaining axioms). Consequently, the additional precondition $\exists\,\beta\ (\beta = \tau)$ in axiom (29.1)[2, p. 190] is dropped as well. To address the issue of possibly non-denoting definite descriptions, the embedding distinguishes between the types ν and κ. Roughly speaking, the type ν corresponds to PLMs individual variables, whereas the type κ corresponds to the Principia's individual terms. Constructs of type ν always denote (individuals), whereas objects of type κ may contain definite descriptions that may not denote. The condition under which an object of type κ denotes is internally stored/remembered by an annotation (Boolean flag). More precisely, the type κ is internally represented as a tuple of a Boolean and an object of type ν). The decoration -^P is used to represent only objects of κ that denote (internally x^P maps x which is of type ν to an object of type κ representing the tuple $(\mathit{True},\ x)$). Consequently, any theorem of the Principia that uses individual variables can be represented in the embedding using a variable of type ν decorated by -^P (see also the section about axiom and theorem schemata below).

In order to be able to substitute denoting definite descriptions for an expression like x^P, the axiom *cqt-5-mod* assures the following:

$\mathit{SimpleExOrEnc}\ \psi \Longrightarrow [[\psi\ x \rightarrow (\exists\, \alpha.\ \alpha^P = x)]]$

$\mathit{SimpleExOrEnc}\ \psi$ is an inductive predicate that is *True* if and only if ψ is a simple exemplification or encoding formula. In the functional setting this means that ψ is a function from κ to o (the type of propositions) that is either the exemplification of an n-place relation by its argument (among other arbitrary objects for $n > 1$) or an encoding expression in its argument. *cqt-5-mod* therefore assures that an object of type κ can be substituted for an expression of the form x^P, if it is contained in a true exemplification or encoding expression. The axiom itself is a logical consequence of the original axioms (29.2) and (29.5)[2, p. 191].

One might think that dropping the additional precondition in axiom (29.1)[2, p. 190] constitutes a problem for the embedding, as now any formula that is true for all individuals can directly be instantiated for a definite description. This is not the case, though. The embedding does not define quantification for the type κ, but only for the type ν. Therefore, a function φ in the expression $\forall\ x\ .\ \varphi\ x$ cannot be a function from κ to o, but only from ν to o. The statement *forall x it holds that x exemplifies F* is represented by $\forall\ x\ .\ (\!|F,x^P|\!)$ in the embedding and can only be instantiated for definite descriptions that can be substituted for an expression of the form x^P, i.e. for definite descriptions that denote.

Consequently, the modified axioms of quantification in the embedding are equivalent to, resp. correspond to, the original axioms (29)[2, p. 191].

The embedding could easily be modified to include a similar distinction for relation terms as well. The equivalent of the -^P decoration for relations would then internally be the identity, as relation terms always denote. As this would introduce more complexity to the embedding and would not change its logical consequences, we decided not to include such a distinction in the embedding.

However, the combination with a proper embedding of free logic (see above) seems an interesting opportunity for future work. First investigations suggest that such a combination may make it possible to drop the current distinction between the types κ and ν and the adjustment of the axioms.

2 Propositional Formulas and Lambda Expressions

The Principia explicitly distinguishes between propositional formulas and formulas that may contain encoding subformulas. As outlined in [1] there is no trivial solution for reproducing this distinction in the context of a functional logic. The embedding only uses one primitive type o for propositions, and an expression of type o *may* contain encoding subformulas. The issue that arises here is that naively allowing lambda-expressions to contain encoding subformulas in combination with axiom (39) leads to inconsistencies. The solution to this problem lies in the observation that any propositional formula as defined in the Principia (i.e. any formula *not* containing encoding subformulas), can be represented by a function acting on *urelements* in the Aczel-model of the theory, rather than a function acting on *individuals*. Only encoding subformulas depend on the actual individuals, whereas all other expressions (i.e. exemplification subformulas) only depend on the urelements corresponding to the individuals.

This way the lambda expressions of the embedded logic can be represented by lambda expressions in the meta-logic as: $(\boldsymbol{\lambda} x.\ (\varphi\ x)) = (\lambda u.\ \varphi\ (\upsilon\nu\ u))$

Here, x is an individual object of type ν and $\upsilon\nu$ maps an urelement to some (undefined) individual in its preimage. This way φ is a function acting on individuals (of type ν) and can thereby represent the matrix of any lambda-expressions of Principia. In the meta-logic this function is converted to a function acting on urlements, though, so the expression $\boldsymbol{\lambda} x.\ \varphi\ x$ only implies *being x, such that there exists some y that is mapped to the same urelement as x, and it holds that $\varphi\ y$*. Conversely, only *for all y that are mapped to the same urelement as x it holds that $\varphi\ y$* is a sufficient condition to conclude that x exemplifies $\boldsymbol{\lambda} x.\ \varphi\ x$.

As propositional formulas only depend on urelements, however, the resulting lambda-expressions can accurately represent the lambda-expressions of the Principia. Moreover, using the construction described above, lambda-expressions that do contain encoding subformulas do not lead to inconsistencies.

It is interesting to note that the embedding suggests, that the restrictions on lambda-expressions in the Principia could in general be extended in a consistent way: instead of restricting lambda-expressions to propositional formulas entirely, it would be sufficient to disallow the occurrence of the *bound variables* of the lambda-expression in an encoding subformula to avoid inconsistencies.

The expression $\boldsymbol{\lambda} x.\ (\!|F,x^P|\!)\ \boldsymbol{\&}\ \{\!|y^P,\ G|\!\}$ can be formulated in the embedding and $(\!|\boldsymbol{\lambda} x.\ (\!|F,x^P|\!)\ \boldsymbol{\&}\ \{\!|y^P,\ G|\!\},\ z^P|\!)$ is equivalent to $(\!|F,z^P|\!)\ \boldsymbol{\&}\ \{\!|y^P,\ G|\!\}$ as one would expect. Still these kinds of expressions are not part of PLM.

3 Theorem and Axiom Schemata

As already mentioned above, in the logic of Isabelle/HOL generally every free variable symbol can be substituted by any term of the same type. Stating an axiom or theorem containing a free variable symbol (e.g. of type o) therefore implicitly asserts that the statement is true for all objects of the same type (e.g. all propositions). Axiom (21.1)[2, p. 186]) for instance can therefore simply be stated as the following:

$[[\varphi \rightarrow (\psi \rightarrow \varphi)]]$

This automatically asserts that all expressions of type o can be substituted for φ or ψ, so the statement itself is in fact already the complete axiom schema.

Consequently the equivalence of alphabetic variants as well as α-Conversion implicitly hold in the context of Isabelle/HOL and don't have to be explicitly stated.

Note that stating an axiom or theorem for a decorated variable x^P as described in the previous sections, only allows objects of type ν to be substituted for x. To substitute a definite description it first has to be assured that $\exists\ x\ .\ (x^P) = \boldsymbol{\iota} x\ .\ \varphi\ x$, as the type of $\boldsymbol{\iota} x.\ \varphi\ x$ is κ which is different from the type ν.

4 Modally-Strict Proofs

The deductive system PLM described in the Principia distinguishes between theorems that are *modally-strict* and theorems that are not *modally-strict*. A theorem is modally-strict if it can be derived from other modally-strict theorems or any of the axioms that are not necessitation-averse. Consequently, if a formula is a modally-strict theorem, then the same formula prefixed with the box-operator is a theorem of PLM (the corresponding meta-rule in PLM is called *RN*). Conversely, if $\Box\varphi$ is a theorem of PLM, this does *not* imply that φ is a modally-strict theorem (see the remark about the converse of RN (52)[2, p. 213]).

The embedding on the other hand explicitly models the modal logic of the theory with a primitive notion of possible worlds (i.e. Kripke semantics). The regular axioms are stated to be true in all possible worlds and therefore their necessitations are implicitly true, as the box-operator is semantically defined to mean truth in all possible worlds. The necessitation-averse axiom on the other hand is stated to be true only in the designated actual world, from which its necessitation is therefore not derivable.

Consequently, modally-strict theorems can be stated and proven to be true for an *arbitrary* possible world, whereas non-modally-strict theorems are stated and proven to be true for the actual world.

In this representation, however, in contrast to PLM, the converse of *RN* becomes true: If $\Box\varphi$ is proven as a theorem (i.e. proven to be true in the designated actual world), then by the semantics of the box operator it follows that φ is true for an arbitrary possible world, which is how modally-strict theorems are stated in the embedding.

However, in Isabelle/HOL all dependencies necessary to prove a theorem are explicitly stated in its proof and we explicitly refrain from stating or using the converse of *RN* (although automation suffers due to this restriction). All theorems that are derived from the deductive system in the embedding therefore still correspond to modally-strict theorems in PLM.

Using the meta-logic directly it would be possible to prove that theorems hold for an arbitrary possible world, that are not modally-strict theorems in PLM, though.

This is not a flaw of the embedding per se, though. The notion of modal-strictness in PLM is purely proof-theoretical and based on the derivability of a theorem from other theorems. As the embedding explicitly gives all dependencies necessary to derive each theorem, it thereby exactly provides the information necessary to classify a theorem to be modally-strict or not. Semantically on the other hand, there is no equivalent to the distinction between modally-strict and non-strict theorems, so there is no way to judge whether a theorem is modally-strict solely based on its semantic truth evaluation in general.

References

[1] P. E. Oppenheimer and E. N. Zalta. Relations versus functions at the foundations of logic: Type-theoretic considerations. *Journal of Logic and Com-*

putation, (21):351374, 2011.

[2] E. N. Zalta. Principia logico-metaphysica. http://mally.stanford.edu/principia.pdf. [Draft/Excerpt; accessed: October 28, 2016].