Procedural Isomorphism and Lambda Transformation

(Does Peter Love John's Wife?)

Session 3: Logic

John loves his wife. So does Peter. Therefore, John and Peter share a property. But which one? There are two options. (1) Loving John's wife. Then John and Peter love the same woman (and there is trouble on the horizon). (2) Loving one's own wife. Then, unless they are married to the same woman, John loves one woman and Peter loves another woman (and both are exemplary husbands). To provide a principled answer, I am going to present a theory of synonymy (in terms of quasi-identity, or isomorphism, between logical procedures), λ-conversion, and loss of information.

In my procedural semantics I construe the meaning of an expression as an abstract procedure encoded by the expression. Meanings are structured, because these procedures specify particular constituent steps that are to be executed in order to obtain the object (if any) beyond the language to which the expression belongs. Procedural semantics is more fine-grained than an extensional semantics, including the intensional semantics of possible-world semantics, because within a procedural framework we can easily distinguish between merely equivalent and synonymous expressions. Expressions can be equivalent by denoting one and the same object, yet in different ways. In this case they are not synonymous. Thus my semantics is *hyperintensional*, and the key tenet of my procedural, i.e., hyperintensional semantics is this. Synonymous expressions encode one and the same procedure, which can be taken as an explication of Frege's *mode of presentation*. Such a hyperintensional approach is highly expressive. It meets the philosophical adequacy constraints pertaining to a logically acceptable semantic theory, such as meanings being structured and context-invariant, as well as heeding the principles of compositionality and transparency.

So far, so good. However, a fundamental problem arises. Whereas in possible-world semantics (excluding impossible worlds), meanings are individuated extensionally (thus in a precise, if crude, manner), in procedural semantics we face the problem of identity of procedures. The problem of individuation of meanings was of extreme importance to Church. He criticized Carnap's attempt to define intensional isomorphism and opted for synonymous isomorphism. Church's attempts are well summarized in Anderson (1998, p. 162) as Alternatives (0), (1), (1'), and (2). Accordingly, meanings are identical if the corresponding λ-terms are

(0) synonymous isomorphic

(1) λ-convertible

(1’) λ- and η-convertible

(2) logically equivalent

Alternative (2) was rejected already by Carnap and was also too coarse-grained for Church, of course. Alternative (1) is the transitive closure of α- and β-convertibility. (1’) adds to those η-convertibility. (0) is the transitive closure including α-convertibility and the substitution of terms for those constants that are postulated to be synonymous with a given constant.

The notation of my procedural semantics is also based on the typed λ-calculus. However, there is an important difference: my λ-terms do not denote functions – instead they denote procedures (generalized algorithms) producing functions. My background theory is Tichý’s Transparent Intensional Logic. The abstract procedures that are assigned to expressions as their context-invariant meanings are defined as TIL *constructions*. The problem is that constructions are too much of a good thing: they are too fine-grained for the purposes of a procedural semantics for natural language, at least. Thus we are facing a problem similar to Church’s. Our goal is to define the relation of procedural isomorphism over the set of constructions (of a particular order). In this paper I consider two variants of this definition, namely Alternatives (½) and (¾). Alternative (½) includes α- and η-convertibility. Alternative (¾) adds to these two a restricted form of β-convertibility. I am going to show that in this way we eliminate the minor difference between constructions that differ only in which variables they λ-bind.

This paper presents arguments for the unacceptability of unrestricted β-convertibility as a criterion for the identity of procedures. As is well known, β-convertibility is not, in general, an equivalent transformation in the logic of partial functions, of which TIL is an example. But I will show that even in those cases where two constructions are mutually β-convertible the reduced construction can be semantically weaker than the unreduced one. Thus even an equivalent β-reduction can yield a loss of semantic, analytic information. (More on analytic information can be found in Author, 2010). At the same time I define a restricted form of β-reduction that is immune to the loss. As a result, I present a definition of procedural isomorphism as a criterion of synonymy that is based on Alternative (¾). Accordingly, expressions are synonymous if their meanings are procedurally isomorphic, which means that the constructions assigned to them are transferable using α-, η- and restricted β-conversion.

So does Peter love John’s wife, or does Peter love his own wife? With the logical machinery at hand, I analyse both readings, namely the strict and sloppy one, and demonstrate how to avoid the loss of information consisting in *which* reading was the original one.

References

AUTHOR. Paper. *Journal of Philosophical Logic*.

Anderson, C. A. 1998. Alonzo Church’s contributions to philosophy and intensional logic. *The Bulletin of Symbolic Logic* 4: 129-171.

Church, A. 1954. Intensional isomorphism and identity of belief. *Philosophical Studies* 5: 65-73.

Church, A. 1993. A revised formulation of the logic of sense and denotation. Alternative (1).