Towards an Extensional Calculus of Hyperintensions

Marie Duží

VSB-Technical University, Ostrava

Abstract: In this paper I describe an extensional logic of hyperintensions, viz. Tichý's Transparent Intensional Logic (TIL). TIL preserves transparency and compositionality in all kinds of context, and validates quantifying into all contexts, including intensional and hyperintensional ones. The received view is that an intensional (let alone hyperintensional) context is one that fails to validate transparency, compositionality, and quantifying-in; and vice versa, if a context fails to validate these extensional principles, then the context is 'opaque', that is non-extensional. We steer clear of this circle by defining extensionality for hyperintensions presenting functions, functions (including possible-world intensions), and functional values. The main features of our logic are that the senses of expressions remain invariant across contexts and that our ramified type theory enables quantification over any logical objects of any order into any context. The syntax of TIL is the typed lambda calculus; its semantics is based on a procedural redefinition of, inter alia, functional abstraction and application. The only two non-standard features of our logic are a hyperintension called Trivialization and a four-place substitution function (called Sub) defined over hyperintensions. Using this logical machinery I propose rules of existential generalization and substitution of identicals into the three kinds of context.

Keywords. Quantifying-in, extensional/intensional/hyperintensional context, transparency, ramified type theory, transparent intensional logic, extensional rules for three kinds of context.

1 Introduction

In this paper I introduce basic fundamentals of an extensional logic of hyperintensions developed within *procedural semantics* of Transparent Intensional Logic (TIL). Only an extensional logic will validate ex-

tensional principles like the rule of existential generalization or Leibniz's law of substitution of identicals. The cornerstone of TIL approach is that we avail ourselves of rich ontology organised into an infinite bi-dimensional hierarchy of types. We assign to terms and expressions occurring in hyperintensional contexts the very same meaning that we assign to those very same terms and expressions when occurring in intensional and extensional contexts. As a result of this top-down approach, the extensional logical rules apply indiscriminately to all contexts. The upside of our top-down approach is that referential transparency and compositionality of meaning are preserved throughout, together with semantic innocence, since we have no recourse to reference shift. At no point do we invoke contextualist epicycles to somehow create a second semantics for 'non-extensional' contexts. The perceived downside would be that we revise the prevalent extensionalist semantic theory of terms and expressions, in that we universalize Frege's semantics earmarked for *Sinn*-sensitive contexts to all contexts. Be that as it may, it is strength of our solution that it is emphatically not tailor-made specifically for validating extensional principles. Instead it is just yet another application of a large-scale background theory and our solutions are principled and not *ad hoc*.

The paper is organised as follows. In Section 2 I briefly summarise the history of the development of logical semantics from Frege *via* so-called syntactic and model-theoretic turn up to procedural or algorithmic turn. Section 3 introduces the core of TIL so that to describe logical machinery needed in the main Section 4 where basic principles of extensional logic of hyperintensions are introduced.

2 Historical Background

Going back to the history we encounter Frege who was (to the best of my knowledge) the first who were developing formal semantics. In (1892) Frege introduced the well-known semantic schema assigning to expressions their *sense* (*Sinn*) and *denotation* (*Bedeutung*). Wishing to save compositionality, Frege made the semantics of an expression depend on the linguistic context in which it is embedded. According to Frege an expression names its *Bedeutung* (extension) in ordinary contexts and *Sinn* (intension) in oblique contexts.[1] Frege, in an attempt

[1] There is another defect in Frege's semantics; extension of a sentence is its truth-value. Yet in case of empirical sentences, the truth-value depends on

to save compositionality, had recourse to contextualism. The price he paid is too high, indeed. No expression can denote an object, unless a particular kind of context is provided.[2] Yet such a solution is far from being natural. There are cases of real ambiguity, witness homonymous expressions. But would anybody say that 'The author of *Waverley*' were another such a case of homonymy? Hardly; unless, of course, their intuitions had been warped by Fregean contextualism. Furthermore, expressions can be embedded within other expressions to various degrees; consider the sentence

> "Charles knows that Tom believes that the author of *Waverley* is a poet."

The expression 'The author of *Waverley*' should now denote the 'normal' sense of the 'normal sense' of itself. Adding still further layers of embedding sets off an infinite hierarchy of senses, which is to say that 'The author of *Waverley*' has the potential of being infinitely ambiguous. This seems plain wrong, and is first and foremost an awkward artefact of Fregean semantics (see also Duží – Materna 2010, or Duží – Jespersen – Materna 2010, §1.5).

The second half of the last century can be characterized as a linguistic turn in semantics. We were developing systems of particular logics which are characterized by a language with a precisely defined syntax and a model set-theoretic semantics. The main goal of building such a system is to find a subset of sentences of the language, axioms of the theory, which characterize a given area under scrutiny and apply proper rules of inference in order to mechanically derive consequences of the axioms. If the system has a model, then it is consistent, and all we are interested in is manipulating symbols. Therefore, *linguistic or syntactic turn.*

Says David Kaplan:

> During the Golden Age of Pure Semantics we were developing a nice homogenous theory, with language, meanings, and entities of

---

contingent facts, which is not a matter of logical semantics. See also Klement (2002).

[2] It is also remarkable that Frege in his effort to save compositionality postulated that even expressions lacking a standard denotation would have a conventional one. Thus he actually broke the principle of compositionality. I am grateful for this remark to Marian Zouhar.

> the world each properly segregated and related one to another in rather smooth and comfortable ways. This development probably came to its peak in *Carnap*'s Meaning and Necessity (1947). Each *designator* has both an intension and an extension. Sentences have truth-values as extensions and propositions as intensions, predicates have classes as extensions and properties as intensions, terms have individuals as extensions and individual concepts as intensions …. The intension of a compound is a function of the intensions of the parts and similarly the extension (except when intensional operators appear). There is great beauty and power in this theory. But there remained some nagging doubts: proper names, demonstratives, and quantification into intensional contexts. (Kaplan 1990, 13-14)

The mainstream in this direction was *Possible World Semantics* (PWS). Kripke characterizes this semantics as follows:

> We define a proposition (…) as a mapping whose domain is K [a logical space of possible worlds] and whose range is the set {T, F}. (Intuitively, a proposition is something that can be true or false in each world; and (…) we identify propositions that are strictly equivalent, i.e. have the same truth-value in each world. (…) Notice that each proposition determines a unique set of worlds (the set of all worlds mapped into T), and that conversely each set of worlds determines a proposition (its 'characteristic function'). Thus a proposition could just as well have been defined simply as a subset of K. (Kripke 1963, §5.3)

Possible-world intensions are extensionally individuated and the PWS semantics is a *logic of intensions*, in particular the *model-theoretic* (hence set-theoretic) *theory of modalities*. Yet its individuation of meaning is too crude (up to equivalence), and thus it is not apt to solve the notoriously well-known problem of the analysis of belief sentences. Carnap in (1947) says that modal sentences like "It is necessary that *P*" are intensional with respect to the clause *P*. However, sentences about belief like "John believes that *P*" are *neither intensional nor extensional* with respect to *P*. He also criticised Frege's 'naming method' (now we would say the denotational semantics), because then we multiply the names *ad infinitum*, and we end up with the antinomy of naming. For Carnap, extension is not a matter of *logical semantics* because it is a matter of factual knowledge. Prior for the meaning is an *intension* independent of contingent facts that uniquely determines the extension (if any), but not *vice versa*.

In order to solve the problem of belief sentences, Carnap tried to define a stronger relation between expressions than L-equivalence that might rightly calibrate the identity of meaning (i.e. *synonymy*). He defined inductively the relation of *intensional isomorphism* on the set of sentences. Roughly, two sentences S and P are intensionally isomorphic if they are L-equivalent and each designator (either simple or composed) that is a constituent of S is L-equivalent to the respective designator of P. Thus sentences S and P have the same intensional structure if they are composed *in the same way* from designators with the same intensions. In my opinion, all these Carnap's tenets and philosophical desiderata are plausible and it might seem that he succeeded in defining the subject of beliefs, knowledge, convictions, etc. Moreover, his definition is independent of the language and syntactic structure in which this subject is encoded. So far so good; yet Carnap's method has been criticized by Alonzo Church (1954). Church's argument is based on two principles. First, it is Carnap's principle of tolerance (which itself is, of course, desirable), and second, which is less desirable, this principle makes it possible to introduce into a language *syntactically simple* expressions as definitional abbreviations of *semantically complex* expressions (for instance, in English 'fortnight' standing for 'a period of fourteen days').

Thus we can introduce into a language *primitive* symbols P and Q in this way: P is the set of natural numbers that are less than the number 3. Q is the set of natural numbers n for which there are natural numbers x, y, z such that $x^n + y^n = z^n$. But then P and Q are L-equivalent (because they denote the same set of numbers) and also intensionally isomorphic because they have no other constituent designators but themselves. Yet it is easy to believe that $\exists n\ (Qn \wedge \neg Pn)$ without believing that $\exists n\ (Pn \wedge \neg Pn)$.[3] Church proposes a *synonymous isomorphism*: all the mutually corresponding designators must be not only L-equivalent but also synonymous, where the synonymy of syntactically simple designators must be *postulated* as a semantic base of a language. We can postulate any convention for introducing these synonymous abbreviations, but as soon as we postulate the meaning of a constant it becomes valid and cannot be changed by another convention.

Since the late 60-s of the last century many logicians strived for *hyperintensional semantics* and *structured meanings* (see also Fox – Lappin

[3] For the proof of Fermat's theorem is difficult to discover (written in 1954).

2001). The structured character of meaning was urged by David Lewis in (1972), where non-structured intensions are generated by finite, ordered trees. This idea of 'tree-like' meanings obviously influenced George Bealer's idea of 'intensions of the second kind' in his (1982). The idea of structured meanings was propagated also by M.J. Cresswell who defines structured meanings as ordered *n*-tuples (see Creswell 1975; 1985). That this is far from being a satisfactory solution is shown in Tichý (1994), Jespersen (2003) and also Bealer (2004). In brief, these tuples are again set-theoretic entities structured at most from a mereological point of view, by having elements or parts (though one balks at calling elements 'parts', since sets, including tuples, are not complexes). Besides, tuples are of the wrong making to serve as truth-bearers and objects of attitudes, since a tuple cannot be true or be known, hoped, etc., to be true. Simply, tuples are 'flat' from the procedural or algorithmic point of view. The *way* of combining particular parts together is missing here.

In 1994 Moschovakis comes with an idea of *meaning as algorithm*. The meaning of a term *A* is "an (abstract, not necessarily implementable) algorithm which computes the denotation of *A*" (2006, 27; see also 1994).[4] Moschovakis outlines his conception thus:

> The starting point … [is] the insight that a correct understanding of programming languages should explain the relation between a program and the algorithm it expresses, so that the basic interpretation scheme for a programming language is of the form
>
> program $P \rightarrow$ algorithm(P) $\rightarrow$ den(P).
>
> It is not hard to work out the mathematical theory of a suitably abstract notion of algorithm which makes this work; and once this is done, then it is hard to miss the similarity of the above schema with the basic Fregean scheme for the interpretation of a natural language,
>
> term $A \rightarrow$ meaning(A) $\rightarrow$ den(A).
>
> This suggested at least a formal analogy between algorithms and meanings which seemed worth investigating, and proved after some work to be more than formal: when we view natural language with a programmer's eye, it seems almost obvious that we can represent the meaning of

[4] Moschovakis' notion of algorithm borders on being too permissive, since algorithms are normally understood to be effective. (See Cleland 2002 for discussion.)

> a term *A* by the algorithm which is expressed by *A* and which computes its denotation. (Moschovakis 2006, 42)

Yet much earlier, in (1968) and (1969) Pavel Tichý formulated the idea of *procedural semantics.* Thus, for instance, a sentence encodes an *instruction* how in any possible world at any time to execute the abstract *procedure* expressed by the sentence as its meaning, i.e., to evaluate the truth-conditions of the sentence. He developed a logical framework known today as *Transparent Intensional Logic* (TIL). In modern jargon, TIL belongs to the paradigm of *structured meaning*. However, Tichý does not reduce structure to set-theoretic sequences, as do Kaplan and Cresswell. Nor does Tichý fail to explain how the sense of a molecular term is determined by the senses of its atoms and their syntactic arrangement, as Moschovakis objects to 'structural' approaches in (2006, 27).

3 Foundations of TIL

From the formal point of view, TIL is a hyperintensional, partial typed λ-calculus.[5] A main feature of the λ-calculus is its ability to systematically distinguish between functions and functional values. An additional feature of TIL is its ability to systematically distinguish between functions and modes of presentation of functions and modes of presentation of functional values. The TIL operation known as *Closure* is the very *procedure* of presenting or forming or obtaining or *constructing* a function; the TIL operation known as *Composition* is the very *procedure* of *constructing* the value (if any) of a function at an argument. Compositions and Closures are both multiple-step procedures, or *constructions*, that operate on input provided by two one-step constructions, which figure as sub-procedures of Compositions and Closures, namely *variables* and so-called *Trivializations*. Characters such as '*x*', '*y*' '*z*' are words denoting variables, which construct the respective values that an assignment function has accorded to them. The linguistic counterpart of a Trivialization is a constant term always picking out the same object. In order to operate on *X*, *X* needs to be grabbed first. Trivialization is such a one-step grabbing mechanism. Similarly, in order to talk about China (in non-demonstrative and non-indexical English discourse) we need to name China, most simply by using the constant

[5] For details on TIL see in particular Tichý (1988, 2004) and Duží et al. (2010).

'China'. Trivialization is important in what follows, because in order to substitute one sub-construction for another inside a construction it is crucial to be able to grab those three individual constructions.

The logical core of TIL is its notion of *construction* and its *type hierarchy*, which divides into a ramified type theory and a simple type theory. The ramified type hierarchy organizes all higher-order objects, which are all constructions, as well as all functions with domain or range in constructions. The simple type hierarchy organizes first-order objects, which are non-constructions like extensions (individuals, numbers, sets, etc.), possible-world intensions (functions from possible worlds) and their arguments and values. The relevant definitions decompose into three parts. Firstly, simple types of order 1 are defined by Definition 1. Secondly, constructions of order n, and thirdly, types of order $n + 1$.

Definition 1 (*types of order 1*) Let B be a *base*, where a base is a collection of pair-wise disjoint, non-empty sets. Then:

(i) Every member of B is an elementary *type of order 1 over B*.

(ii) Let $\alpha, \beta_1, \ldots, \beta_m$ ($m > 0$) be types of order 1 over B. Then the collection $(\alpha\, \beta_1 \ldots \beta_m)$ of all m-ary partial mappings from $\beta_1 \times \ldots \times \beta_m$ into α is a functional *type of* order *1 over B*.

(iii) Nothing is a *type of order 1 over B* unless it so follows from (i) and (ii).

Remark. For the purposes of natural-language analysis, we are currently assuming the following base of ground types, which is part of the ontological commitments of TIL:

ο: the set of truth-values {**T, F**};
ι: the set of individuals (constant universe of discourse);
τ: the set of real numbers (doubling as temporal continuum);
ω: the set of logically possible worlds (logical space).

Definition 2 (*construction*)

(i) The *variable x* is a *construction* that constructs an object O of the respective type dependently on a valuation v: x *v*-constructs O.

(ii) *Trivialization*: Where X is an object whatsoever (an extension, an intension or a *construction*), ${}^0X$ is the *construction Trivialization.* It constructs X without any change.

(iii) The *Composition* $[X\ Y_1 \ldots Y_m]$ is the following *construction*. If X *v-constructs* a function f of type $(\alpha\, \beta_1 \ldots \beta_m)$, and $Y_1, \ldots, Y_m$ *v-construct* entities $B_1, \ldots, B_m$ of types $\beta_1, \ldots, \beta_m$, respectively, then the *Composition* $[X\ Y_1 \ldots Y_m]$ *v-constructs* the value (an entity, if any,

of type α) of f on the tuple argument $\langle B_1, \ldots, B_m\rangle$. Otherwise the *Composition* $[X\ Y_1 \ldots Y_m]$ does not *v-construct* anything and so is *v-improper*.

(iv) The *Closure* $[\lambda x_1 \ldots x_m\ Y]$ is the following *construction*. Let $x_1, x_2, \ldots, x_m$ be pair-wise distinct variables *v*-constructing entities of types $\beta_1, \ldots, \beta_m$ and Y a construction *v*-constructing an α-entity. Then $[\lambda x_1 \ldots x_m\ Y]$ is the *construction* λ-*Closure* (or *Closure*). It *v-constructs* the following function f of the type $(\alpha\ \beta_1 \ldots \beta_m)$. Let $v(B_1/x_1, \ldots, B_m/x_m)$ be a valuation identical with v at least up to assigning objects $B_1/\beta_1, \ldots, B_m/\beta_m$ to variables $x_1, \ldots, x_m$. If Y is $v(B_1/x_1, \ldots, B_m/x_m)$-improper (see iii), then f is undefined on $\langle B_1, \ldots, B_m\rangle$. Otherwise the value of f on $\langle B_1, \ldots, B_m\rangle$ is the α-entity $v(B_1/x_1, \ldots, B_m/x_m)$-constructed by Y.

(v) The *Single Execution* ${}^1X$ is the *construction* that either *v*-constructs the entity *v*-constructed by X or, if X is not itself a construction or X is *v*-improper, ${}^1X$ is *v*-improper.

(vi) The *Double Execution* ${}^2X$ is the following *construction*. Where X is any entity, the *Double Execution* ${}^2X$ is *v-improper* (yielding nothing relative to v) if X is not itself a construction, or if X does not *v*-construct a construction, or if X *v*-constructs a *v*-improper construction. Otherwise, let X *v*-construct a construction Y and Y *v*-construct an entity Z: then ${}^2X$ *v*-constructs Z.

(vii) Nothing is a *construction*, unless it so follows from (i) through (vi).

Definition 3 (*ramified hierarchy of types*)

$\mathbf{T}_1$ *(types of order 1)*. See Definition 1.

$\mathbf{C}_n$ *(constructions of order n)*

(i) Let x be a variable ranging over a type of order n. Then x is a *construction of order n over B*.

(ii) Let X be a member of a type of order n. Then ${}^0X$, ${}^1X$, ${}^2X$ are *constructions of order n over B*.

(iii) Let $X, X_1, \ldots, X_m$ $(m > 0)$ be constructions of order n over B. Then $[X\ X_1 \ldots X_m]$ is a *construction of order n over B*.

(iv) Let $x_1, \ldots x_m, X$ $(m > 0)$ be constructions of order n over B. Then $[\lambda x_1 \ldots x_m\ X]$ is a *construction of order n over B*.

(v) Nothing is a *construction of order n over B* unless it so follows from $\mathbf{C}_n$ (i)-(iv).

$\mathbf{T}_{n+1}$ *(types of order n + 1)* Let $*_n$ be the collection of all constructions of order n over B. Then

(i) $*_n$ and every type of order n are *types of order n* + 1.
(ii) If $0 < m$ and $\alpha, \beta_1, \ldots, \beta_m$ are types of order $n + 1$ over B, then $(\alpha\ \beta_1\ \ldots\ \beta_m)$ (see T_1 ii)) is a *type of order n* + 1 *over B*.
(iii) Nothing is a *type of order n + 1 over B* unless it so follows from (i) and (ii).

Empirical languages incorporate an element of *contingency* that non-empirical ones lack. Empirical expressions denote *empirical conditions* that may, or may not, be satisfied at some empirical index of evaluation. Non-empirical languages have no need for an additional category of expressions for empirical conditions. We model these empirical conditions as *possible-world intensions.* Intensions are entities of type $(\beta\omega)$: mappings from possible worlds to an arbitrary type β. The type β is frequently the type of the *chronology* of α-objects, i.e., a mapping of type $(\alpha\tau)$. Thus α-intensions are frequently functions of type $((\alpha\tau)\omega)$, abbreviated as '$\alpha_{\tau\omega}$'. I will typically say that an index of evaluation is a world/time pair $\langle w, t\rangle$. *Extensional entities* are entities of a type α where $\alpha \neq (\beta\omega)$ for any type β.

Examples of frequently used intensions are: *propositions* of type $o_{\tau\omega}$ (denoted by non-indexical sentences), *properties* of individuals of type $(o\iota)_{\tau\omega}$ (denoted by predicates or nouns like 'being happy', 'being a mathematician'), binary *relations-in-intension* between individuals of type $(o\iota\iota)_{\tau\omega}$ (usually denoted by verbs like 'admire', 'kick'), individual *offices* or *roles* of type $\iota_{\tau\omega}$ (denoted by definite descriptions like 'pope', 'the president of CR', 'the first man to run 100 m under 9 s'), binary relations-in-intension between individuals and hyperintensions of type $(o\iota*_n)_{\tau\omega}$ (denoted by attitudinal verbs like 'believe', 'know', 'calculate', etc.).

The method of explicit intensionalization and temporalization encodes constructions of possible-world intensions directly in the logical syntax.[6] Where w ranges over ω and t over τ, the following logical form essentially characterizes the logical syntax of empirical language:

---

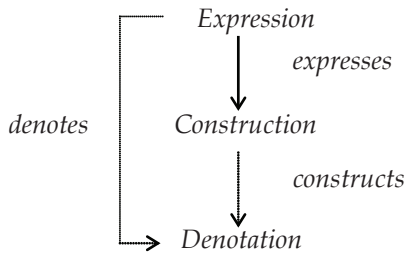
[6] This is one of the issues in which TIL is superior compared to Montague's Intensional Logic (IL). Montague applies only implicit intensionalisation; due to the lack of variables ranging over possible worlds (and times) IL does not validate the Church-Rosser 'diamond' property. Moreover, λ-abstraction, modalities and times must be imitated by special operators. As a result the law of universal instantiation, lambda conversion (β-rule) and Leibniz's Law do not generally hold in IL. For a critical comparison of TIL and Montague's IL see Duží et al. (2010, § 2.4.3).

$\lambda w \lambda t$ […w….t…]. If the Composition […w….t…] v-constructs an entity of type α, then the Closure itself constructs a function of type $((\alpha\tau)\omega)$, or $\alpha_{\tau\omega}$ for short, i.e. an α-intension.

Logical objects like *truth-functions* and *quantifiers* are extensional: $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication) are of type $(o o o)$, and $\neg$ (negation) of type $(o o)$. *Quantifiers* $\forall^\alpha$, $\exists^\alpha$ are type-theoretically polymorphous total functions of type $(o(o\alpha))$, for an arbitrary type α, defined as follows. The *universal quantifier* $\forall^\alpha$ is a function that associates a class A of α-elements with **T** if A contains all elements of the type α, otherwise with **F**. The *existential quantifier* $\exists^\alpha$ is a function that associates a class A of α-elements with **T** if A is a non-empty class, otherwise with **F**.

Notational conventions: Below all type indications will be provided outside the formulae in order not to clutter the notation. Furthermore, 'X/α' means that an object X is (a member) of type α. '$X \rightarrow_v \alpha$' means that the type of the object v-constructed by X is α. We write '$X \rightarrow \alpha$' if what is v-constructed does not depend on a valuation v. Throughout, it holds that the variables $w \rightarrow_v \omega$ and $t \rightarrow_v \tau$. If $C \rightarrow_v \alpha_{\tau\omega}$ then the frequently used Composition [[C w] t], which is the extensionalization of the α-intension v-constructed by C, will be encoded as 'C_{wt}'. When applying truth-functions, identities $=^\alpha/(o\alpha\alpha)$, arithmetic operations and relations $>^\alpha$, $<^\alpha$, we will often use an infix notation without Trivialization and without indicating the type of a function. Instead of '$[^0\exists^\alpha \lambda x\, B]$', '$[^0\forall^\alpha \lambda x\, B]$' ($x \rightarrow_v \alpha$; $B \rightarrow_v o$) we will often write '$\exists x B$', '$\forall x B$'. Thus, for instance, $[^0\forall^\tau \lambda x\, [^0\supset [^0=^\tau x\ ^0 0]\ [^0\forall^\tau \lambda y\, [^0=^\tau [^0+ x\ y]\ y]]]]$ will be encoded as $\forall x\, [[x = {}^0 0] \supset \forall y\, [[x + y] = y]]$.

To summarize, our neo-Fregean semantic schema, which applies to all contexts, is this:



The most important relation in this schema is between an expression and its meaning (a construction). We can investigate *a priori* what (if anything) a construction constructs and what is entailed by it. Once a

construction is explicitly given as a result of logical analysis, the entity (if any) it constructs is already implicitly given, whereas it requires inquiry *a posteriori* to establish the reference of an empirical term at a given world/time pair. As a limiting case, the logical analysis may reveal that the construction fails to construct anything because it is improper. And if the construction is not improper, the denotation can be either a first-order object (i.e. a non-construction) or a lower-order construction. Intensional constructions (constructions of objects of type (βω)) are always proper, since they always construct an intension (including degenerate ones, which return no values at all or always the same value). In linguistic terms, every word whose sense is an intensional construction has a denotation, but will lack a reference at some or all ⟨*w*, *t*⟩ pairs, in case its denotation (a partial function) fails to return a value. This applies to, *inter alia*, 'The pope', 'The first man to run 100 m under 9 s', 'The Evening Star', or 'John's wife'.

Definition 4 (*free and bound variables*) Let *C* be a construction with at least one occurrence of a variable ξ.

(i) Let *C* be ξ. Then the *occurrence of ξ in C* is *free*.
(ii) Let *C* be ^{0}X. Then every *occurrence of ξ in C* is ^{0}*bound* ('Trivialization-bound').
(iii) Let *C* be [λx_1…x_n *Y*]. Any *occurrence of ξ* in *Y* that is one of x_i, $1 \leq i \leq n$, is *λ-bound* in *C* unless it is ^{0}*bound in Y*. Any *occurrence of ξ in Y* that is neither ^{0}*bound* nor *λ-bound* in *Y* is *free* in *C*.
(iv) Let *C* be [*X* X_1…X_n]. Any *occurrence of ξ* that is *free*, ^{0}*bound*, *λ-bound* in one of *X*, X_1,…,X_n is, respectively, *free*, ^{0}*bound*, *λ-bound in C*.
(v) Let *C* be ^{1}*X*. Then any *occurrence of ξ* that is *free*, ^{0}*bound*, *λ-bound* in *X* is, respectively, *free*, ^{0}*bound*, *λ-bound in C*.
(vi) Let *C* be ^{2}*X*. Then any *occurrence of ξ* that is *free*, *λ-bound* in a constituent of *C* is, respectively, *free*, *λ-bound in C*. If an occurrence of ξ is ^{0}*bound* in a constituent ^{0}*D* of *C* and this occurrence of *D* is a constituent of *X*′ *v*-constructed by *X*, then if the occurrence of ξ is free, λ-bound in *D* it is *free*, *λ-bound in C*. Otherwise, any other occurrence of ξ in *C* is ^{0}*bound* in *C*.
(vii) An *occurrence* of ξ is *free*, *λ-bound*, ^{0}*bound* in *C* only due to (i)-(vi).

A construction with at least one occurrence of a free variable is an *open construction*. A construction without any free variables is a *closed construction*.

Definition 5 (*v-congruent and equivalent constructions*) Let C, $D/*_n \rightarrow \alpha$ be constructions, and $\approx_v/(o*_n *_n)$, $\approx/(o*_n *_n)$ binary relations between constructions of order n. Using infix notation without Trivialization, $C \approx_v D$, $C \approx D$, we define: C, D are *v-congruent*, $C \approx_v D$, iff either C and D v-construct the same α-entity, or both C and D are v-improper; C, D are *equivalent*, $C \approx D$, iff C, D are v-congruent for all valuations v.

Corollaries. If C, D are *identical*, then C, D are equivalent, but not *vice versa.* If C, D are equivalent, then C, D are v-congruent, but not *vice versa.*

If meanings of expressions E_1, E_2, that is the constructions expressed by them, are merely v-congruent, we will say that E_1, E_2 are *co-referential.* If meanings of expressions E_1, E_2 are equivalent, we will say that E_1, E_2 are *co-denotational* or *equivalent.*

The next notion we need to define is that of *synonymy*. Our notion of synonymy is defined in terms of *procedural isomorphism*. The term 'procedural isomorphism' is a nod to Carnap's *intensional isomorphism* and Church's *synonymous isomorphism*. Church's Alternatives (0) and (1) leave room for additional Alternatives.[7] One would be Alternative (½), another Alternative (¾). The former includes α- and η-conversion while the latter adds a restricted β-conversion. If we must choose, we would prefer Alternative (¾) to soak up those differences between β-transformations that concern only λ-bound variables and thus (at least appear to) lack natural-language counterparts.

One reason for excluding unrestricted β-conversion is the well-known fact that β-conversion is not an equivalent transformation in logics boasting *partial functions*, such as TIL. Another reason is that occasionally even β-equivalent constructions have different natural-language counterparts; witness the difference between attitude reports *de dicto* vs. *de re*. Thus for instance, if a, b are individuals, the difference between "a believes that b is happy" and "b is believed by a to be happy" is just the difference between β-equivalent meanings. If attitudes are construed as in possible-world semantics, i.e. as relations to intensions (rather than to hyperintensions), the former (*de dicto*) receives the analysis

$$\lambda w \lambda t\, [{}^0Believe_{wt}\, {}^0a\, \lambda w \lambda t\, [{}^0Happy_{wt}\, {}^0b]]$$

---

[7] For Church's Alternatives see Anderson (1998).

while the latter (*de re*) receives the analysis

$$\lambda w \lambda t\, [\lambda x\, [{}^0Believe_{wt}\ {}^0a\ \lambda w \lambda t\, [{}^0Happy_{wt}\ x]]\ {}^0b]$$

Types: $Happy/(o\iota)_{\tau\omega}$; $x \rightarrow_v \iota$; $a, b/\iota$; $Believe/(o\iota o_{\tau\omega})_{\tau\omega}$.

The *de dicto* variant is the β-equivalent contractum of the *de re* variant. Both variants are equivalent because they construct one and the same proposition, the two sentences denoting the same proposition. Yet they denote this proposition in *different ways*, thus they are not synonymous. The equivalent β-reduction leads here to a *loss of analytic* information, namely loss of information about *which* of the two ways, or constructions, has been used to construct this proposition.[8] In this case the loss seems to be harmless, though, because there is only one, unambiguous way to β-expand the *de dicto* version into its equivalent *de re* variant.[9]

However, unrestricted equivalent β-reduction sometimes yields a loss of analytic information that cannot be restored by β-expansion. The well-known example is the sentence "John loves his wife and so does Tom". This sentence is ambiguous between two readings, sloppy and strict. On its sloppy reading John and Tom share the property of each loving their own wife (and both are exemplary husbands). On the strict reading they share the property of loving John's wife (and there are troubles on the horizon). And these are two distinct properties. Thus there are two distinct analyses of "John loves his wife":

Strict: $\lambda w \lambda t\, [\lambda x\, [{}^0Love_{wt}\ x\ [{}^0Wife\_of_{wt}\ {}^0John]\ {}^0John]$
Sloppy: $\lambda w \lambda t\, [\lambda x\, [{}^0Love_{wt}\ x\ [{}^0Wife\_of_{wt}\ x]\ {}^0John]$

But an unrestricted β-reduction turns these two redexes into one and the same contractum:

$$\lambda w \lambda t\, [{}^0Love_{wt}\ {}^0John\ [{}^0Wife\_of_{wt}\ {}^0John]]$$

---

8 For the notion of analytic information, see Duží (2010) and Duží et al. (2010, §5.4).

9 In general, *de dicto* and *de re* attitudes are not equivalent, but logically independent. Consider "*a* believes that the pope is not the pope" and "*a* believes of the pope that he is not the pope". The former, *de dicto*, variant makes *a* deeply irrational and most likely is not a true attribution, while the latter, *de re*, attribution is perfectly reasonable and most likely the right one to make. In TIL the *de dicto* variant is not an equivalent β-contractum of the *de re* variant due to the partiality of the office $Pope/\iota_{\tau\omega}$.

A piece of analytic information has been lost and using the contractum one does not know which property should be applied to Tom.[10]

The *restricted* version of *equivalent* β-conversion we have in mind consists in substituting free variables for λ-bound variables of the same type, and will be called β_r*-conversion*. For instance, we see little reason to differentiate semantically or logically between "*b* is believed by *a* to be happy" and "*b* has the property of being believed by *a* to be happy".[11] The latter sentence expresses

$$\lambda w \lambda t\, [\lambda w' \lambda t'\, \lambda x\, [{}^0Believe_{w't'}\, a\, \lambda w \lambda t\, [{}^0Happy_{wt}\, x]]_{wt}\, b]$$

This is merely a β_r-expanded form of

$$\lambda w \lambda t\, [\lambda x\, [{}^0Believe_{wt}\, a\, \lambda w \lambda t\, [{}^0Happy_{wt}\, x]]\, b]$$

Thus we define:

Definition 6 (*procedurally isomorphic constructions: Alternative* (¾)) Let *C*, *D* be constructions. Then *C*, *D* are *a-equivalent* iff they differ at most by deploying different λ-bound variables. *C*, *D* are *η-equivalent* iff one arises from the other by η-reduction or η-expansion. *C*, *D* are β_r*-equivalent* iff one arises from the other by β_r-reduction or β_r-expansion. *C*, *D* are *procedurally isomorphic*, denoted '${}^0C \approx {}^0D$', $\approx/(o*_n *_n)$, iff there are closed constructions $C_1,\ldots,C_m$, $m \geq 1$, such that ${}^0C = {}^0C_1$, ${}^0D = {}^0C_m$, and all C_i, C_{i+1} $(1 \leq i < m)$ are either α-, η- or β_r-equivalent.

Hence we advocate for the restricted β-conversion; yet β-conversion is the fundamental rule for computing the value of the function *v*-constructed by $[\lambda x\, Y]$ at an argument *v*-constructed by a construction *C*. Its (unrestricted) version 'by name' is this (where $Y(C/x)$ is the result of correct substitution of a construction *C* for *x* in *Y*):

$$[[\lambda x\, Y]\, C] \vdash Y(C/x)$$

---

[10] For the solution of this problem see Duží – Jespersen (to appear).

[11] This is not to say we see no reason at all not to differentiate. For instance, it could be argued that one thing is to believe that *a* is happy and another is to believe that *a* has the property of being happy, because the latter at least appears to presuppose that the believer have the additional conceptual resources to master the notion of property. Thus a proper calibration of procedural isomorphism is still an open problem and it can depend on the area under scrutiny. More discussion on procedural isomorphism can be found in Jespersen (2010).

Due to compositionality, if C is v-improper the Composition $[[\lambda x\, Y]\, C]$ is v-improper as well. But if Y is itself a Closure then it is never v-improper.[12] Thus it may happen that the right-hand side is not equivalent to the left-hand side. For this reason we restrict the rule to C being a variable which is never v-improper.

But we do need a general rule of the λ-calculus for computing the value of a function. Fortunately, it turns out to be feasible to formulate a generally valid computational rule. A distinction familiar from programming languages based on the λ-calculus holds the key to the solution. The invalid rule above is moulded on the programming technique of calling a sub-procedure C by name: the sub-procedure itself is substituted for the 'local variable' x in the 'procedure body' Y. It is well-known among programmers that this technique can have undesirable side-effects, unlike the technique of calling a sub-procedure by value.[13] The idea is simple: execute the sub-procedure C first, and then – *provided this execution does not fail* – substitute the construction of the result ('pass by the value') for x.[14]

The substitution method comes with two special functions.[15] The polymorphous function *Sub* of type $(*_n *_n *_n *_n)$ operates on constructions as follows. When applied to constructions C_1, C_2, C_3, *Sub* returns as its value the construction D that is the result of the correct (i.e. collisionless) substitution of C_1 for C_2 in C_3. For instance, the result of the Composition $[{}^0Sub\ {}^{00}John\ {}^0x\ {}^0[{}^0Wife\_of_{wt}\ x]]$ is the Composition $[{}^0Wife\_of_{wt}\ {}^0John]$. The likewise polymorphous function *Tr* returns as its value the Trivialization of its argument. Thus the result of $[{}^0Tr\ {}^0John]$ is ${}^0John$. If the variable x ranges over ι, the Composition $[{}^0Tr\ x]$ *v(John/x)*-constructs ${}^0John$. Note one essential distinction between the function *Tr* and the construction Trivialization. Whereas the variable x is *free* in $[{}^0Tr\ x]$,

---

12 See Definition 2, iii) and iv).

13 A recent reference to the distinction between 'call by name' and 'call by value' is Pierce (2002, 56-57). See also, for instance, Hyde (1996, Ch. 11) or Plotkin (1975).

14 For conversion by name, see Claim 2.5 and the subsequent proof in Duží et al. (2010, 267-268); for conversion by value, see Claim 2.6 and the subsequent proof in (ibid., 269-270). For the general strategy (inspired by programming languages) of distinguishing between succeeding, failing, and aborting with error, see also Van Eijck – Francez (1995).

15 Sub is introduced in Tichý (1988, 75) and Tr at (ibid., 68).

the Trivialization ^0x *binds* the variable x by constructing just x independently of valuation.[16]

For simplicity's sake, we introduce the TIL translation of the rule of β-conversion by value in its simplified version for unary functions (generalization to n-ary functions is obvious):

$$[[\lambda x\ Y]\ C]\ \vdash\ {}^2[{}^0Sub\ [{}^0Tr\ C]\ {}^0x\ {}^0Y]$$

Note that the Composition on the right-hand side must undergo Double Execution. Provided C is v-proper, it v-constructs an entity, say e. Then the result of the first step (the substitution $[{}^0Sub\ [{}^0Tr\ C]\ {}^0x\ {}^0Y]$) is the *construction* $Y(e/x)$. The resulting construction must then be executed in order to obtain the value of the function v-constructed by $[\lambda x\ Y]$ at the argument e. Hence, *Double* Execution. Otherwise, if C is v-improper, the substitution fails to construct anything, because due to the compositionality constraint the whole Composition $[{}^0Sub\ [{}^0Tr\ C]\ {}^0x\ {}^0Y]$ is v-improper and so is $^2[{}^0Sub\ [{}^0Tr\ C]\ {}^0x\ {}^0Y]$ (see Definition 2, iii) and vi)). In this manner compositionality is preserved and the above rule of β-conversion by value is always valid even when C is v-improper.

Remark. In the project of a multi-agent system that our Laboratory of Intelligent Systems dealt with in 2004-2008 we use the computational variant of TIL, the *TIL-Script* functional programming language as the language of communication between agents.[17] In the *TIL-Script* language we apply *only* this computational rule of *conversion by value*. The reason is this. Since the construction C can be v-improper, we need to implement a lazy evaluation mechanism in order to evaluate C only when needed. However, the properness of C can be checked only in the run time, because valuation v would supply values dependently on states-of-affairs.

---

[16] Since TIL is a λ-calculus, all variable binding is λ-binding, except for Trivialization-binding. One area where Trivialization-binding plays a key role is existential quantification into hyperintensional contexts, where a quantifier is introduced with a view to binding a variable that occurs bound by Trivialization, because the variable occurs inside a Trivialized context. For discussion, see Duží et al. (2010, §5.3). For improved solutions, see Duží – Jespersen (to appear).

[17] Project *No. 1ET101940420: "Logic and Artificial Intelligence for multi-agent systems"; supported by the program "Information Society" of the Czech Academy of Sciences. For details see http://labis.vsb.cz/.*

4 Rules for the Three Kinds of Context

At this point we have lined up everything we need in order to introduce the extensional logic of hyperintensions. Yet some may protest that extensional logic of hyperintensions sounds as an oxymoron like a roaring silence. For at least since the milestones Quine (1956) and Kaplan (1968) the validity of extensional principles, in particular of quantifying-in and existential generalization, has been fielded as a logical criterion for distinguishing

a) extensional/transparent/'relational' contexts from
b) non-extensional/opaque/'notional' contexts,

the idea being that extensional (etc.) contexts are those that validate quantifying-in.[18] And conversely, if a context resists quantifying-in, it is caught in violation of one or more of the laws of extensional logic and as such eludes full logical analysis. What we are saying is that also intensional and hyperintensional contexts may be quantified into, but that the feasibility of doing so presupposes that it be done within an *extensional logic of hyperintensional contexts*. Deploying a non-extensional logic of hyperintensions to quantify into hyperintensional contexts would, indeed, be a non-starter, generating opacity and thereby making hyperintensional attitude contexts logically intractable. However, whether one accepts quantifying into (hyper-) intensional contexts or wants to restrict quantification to extensional contexts, like "Mary is happy", the logical question remains which contexts validate quantifying-in.

Tichý issues in (1986, 256; 2004, 654) a warning against a circular definition:

Q: When is a context extensional?
A: A context is extensional if it validates
(i) *the rule of substitution of co-referential terms* and
(ii) *the rule of existential generalization*.
Q: And when are (i), (ii) valid?
A: Those two rules are valid when applied to extensional contexts.

We steer clear of the circle by defining extensionality for

a) *hyperintensions* presenting functions,

[18] See Forbes (1996).

b) *functions* (including possible-world intensions), and
c) *functional values*.

These three levels are squared off with three kinds of context:[19]

1. *hyperintensional contexts*, in which a *construction* is not *used* to present an object, but is itself *mentioned* as functional argument (though a construction of one order higher needs to be used to mention this lower-order construction);
2. *intensional context*s, in which a construction is *used* to present a *function* without presenting a particular value of the function (moreover, the construction does not occur within another hyperintensional context);
3. *extensional context*s, in which a construction is *used* to produce a particular *value* of the function at a given argument (moreover, the construction does not occur within another intensional or hyperintensional context).

Example. Let the types of entities be: *Periodic*/(o(ττ)); *Sin*/(ττ); *Solve*/$(o(\iota*_1))_{\tau\omega}$: the relation-in-intension between an individual and a construction the product of which the individual is seeking; π/τ ; *Tom*/ι.

- *Extensional context*: functional *value* is an object of predication (functional argument):

  "*sin π = 0*"
  [[^{0}*Sin* $^0\pi$] = ^{0}0]

- *Intensional context*: function-in-extension is an object of predication:

  "*Sine is a periodic function*"
  [^{0}*Periodic* ^{0}*Sin*]

- *Hyperintensional context*: construction ("function-in-intension") is an object of predication (a functional argument):

  "*Tom is solving the equation sin x = 0*"
  $\lambda w \lambda t$ [^{0}*Solve*$_{wt}$ ^{0}*Tom* ^{0}[λx [^{0}*Sin x*] = ^{0}0]]

A dual constraint of TIL has impact on the rules of inference. It is the constraint dictated by *properly partial functions*, which are undefined for some or all of their arguments, and *improper constructions*, which

[19] For the definition see Duží et al. (2010, § 2.6 and 2.7).

fail to produce a product. Improperness stems from the procedure of applying a properly partial function *f* to an argument *a*, such that *f* returns no value at *a*. The procedure of functional application induces an extensional context. Thus when specifying the rules of quantifying-in, we must check whether particular constituent constructions occurring extensionally are improper. If none is, the particular rule of quantifying-in is valid.

The rules of *improperness* can be schematically summarized as follows. If a Composition is used in an extensional context as a procedure of application a properly partial function $F/(\beta\alpha)$ to an argument a/α and if *F* has no-value at *a* (value gap) then

$[^0F\ ^0a]$ is *v-improper*

and so is any construction *C* occurring extensionally and containing $[^0F\ ^0a]$ as a constituent; *partiality is strictly propagated up*:

$[\ldots\ [\ \ldots\ [^0F\ ^0a]\ \ldots]\ \ldots]$ is *improper*

until the context is raised up to *hyper/intensional*:

intensional: $\lambda x \ldots\ [\ldots\ [\ \ldots\ [^0F\ ^0a]\ \ldots]\ \ldots]$ is *proper*
hyperintensional: $^0[\ldots\ [\ \ldots\ [^0\mathrm{F}\ ^0\mathrm{a}]\ \ldots]\ \ldots]$ is *proper*

4.1 The Rules of Existential Generalization

Let $F/(\beta\alpha)$; a/α. Then the rules of existential generalization for particular contexts are as follows:

a) ***extensional*** context. Let $[\ldots[^0F\ ^0a]\ldots]$ *v*-construct the truth-value **T**. Then the following rule is truth-preserving:

$[\ldots[^0F\ ^0a]\ldots] \vdash \exists x\ [\ldots[^0F\ x]\ \ldots]; x \rightarrow_v \alpha$

Proof:

1.	$[\ldots[^0F\ ^0a]\ldots]$	assumption
2.	$[\ldots[^0F\ x]\ \ldots]$	*v(a/x)*-constructs **T**
3.	$\lambda x\ [\ldots[^0F\ x]\ldots]$	*v*-constructs a non-empty class
4.	$[^0\exists\lambda x\ [\ldots[^0F\ x]\ldots]]$	EG, 3

Example: $\lambda w\lambda t\ [^0Wise_{wt}\ ^0Pope_{wt}] \models \lambda w\lambda t\ \exists x\ [^0Wise_{wt}\ x]$;
Types: $Wise/(o\iota)_{\tau\omega}$; $Pope/\iota_{\tau\omega}$; $x \rightarrow_v \iota$.

b) ***intensional*** context. Let $[\ldots\lambda y\ [\ \ldots\ [^0F\ ^0a]\ \ldots]]$ *v*-construct **T**. Then the following rule is truth-preserving:

$[\ldots\lambda y\ [\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]] \vdash \exists f\,[\ldots\lambda y\ [\ \ldots\ [f\,{}^{0}a]\ \ldots]];\ f \rightarrow_{v} (\beta\alpha)$

Proof:

1.	$[\ldots\lambda y\ [\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]$	assumption
2.	$[\ldots\lambda y\ [\ \ldots\ [f\,{}^{0}a]\ \ldots]]$	*v(F/f)*-constructs **T**
3.	$\lambda f\,[\ldots\lambda y\ [\ \ldots\ [f\,{}^{0}a]\ \ldots]]$	*v*-constructs a non-empty class
4.	$[{}^{0}\exists\lambda f\,[\ldots\lambda y\ [\ \ldots\ [f\,{}^{0}a]\ \ldots]]]$	EG, 3

Example: $\lambda w\lambda t\ [{}^{0}Believe_{wt}\ {}^{0}b\ \lambda w\lambda t\ [{}^{0}Wise_{wt}\ {}^{0}Pope_{wt}]] \models$
$\lambda w\lambda t\ \exists f\,[{}^{0}Believe_{wt}\ {}^{0}b\ \lambda w\lambda t\ [{}^{0}Wise_{wt}\ f_{wt}]]$;

Additional types:

Believe/$(o\iota o_{\tau\omega})_{\tau\omega}$: an intensional attitude to a proposition; $f \rightarrow_{v} \iota_{\tau\omega}$

Gloss: If *b* believes that the Pope is wise then there is an office such that *b* believes that its holder is wise.

c) ***hyperintensional*** context. Let $[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]$ *v*-construct **T**. Then the following rule is truth-preserving:[20]

$[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]] \vdash \exists c\ {}^{2}[{}^{0}Sub\ c\ {}^{00}F\ {}^{0}[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]];$
$c \rightarrow_{v} *_{n};\ {}^{2}c \rightarrow_{v} (\beta\alpha)$

Proof.

1.	$[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]$	assumption
2.	${}^{2}[{}^{0}Sub\ c\ {}^{00}F\ {}^{0}[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]]$	*v*(^{0}*F/c*)-constructs **T**
3.	$\lambda c\ {}^{2}[{}^{0}Sub\ c\ {}^{00}F\ {}^{0}[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]$	*v*-constructs a non-empty class
4.	$[{}^{0}\exists\lambda c\ {}^{2}[{}^{0}Sub\ c\ {}^{00}F\ {}^{0}[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]]$	EG, 3

The step 2 may be difficult to understand. Here is an additional explanation. The Composition $[{}^{0}Sub\ c\ {}^{00}F\ {}^{0}[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]]$ *v*(^{0}*F/c*)-constructs the Composition $[\ldots\ {}^{0}[\ \ldots\ [{}^{0}F\ {}^{0}a]\ \ldots]]$. In order to *v*-construct **T**, this Composition must be executed again. Therefore, Double Execution.

Example: $\lambda w\lambda t\ [{}^{0}Believe^{*}_{wt}\ {}^{0}b\ {}^{0}[\lambda w\lambda t\ [{}^{0}Wise_{wt}\ {}^{0}Pope_{wt}]] \models$
$\lambda w\lambda t\ \exists c\ [{}^{0}Believe^{*}_{wt}\ {}^{0}b\ [{}^{0}Sub\ c\ {}^{00}Pope\ {}^{0}[\lambda w\lambda t\ [{}^{0}Wise_{wt}\ {}^{0}Pope_{wt}]]]]$;

Additional types:

*Believe**/$(o\iota *_{n})_{\tau\omega}$: a hyperpropositional belief; $c \rightarrow_{v} *_{n};\ {}^{2}c \rightarrow_{v} \iota_{\tau\omega}$.

Gloss: If *b* (explicitly) believes* that the Pope is wise, then there is a concept of an office such that *b* believes* that the holder of the office is wise.

---

[20] I distinguish between a derivation rule, denoted '$\vdash$', and analytic entailment, denoted '$\models$'. Of course, if a particular rule is truth-preserving then the corresponding entailment is valid.

Note: In this example the $[^0Sub\ c\ ^{00}Pope\ ^0[\lambda w\lambda t\ [^0Wise_{wt}\ ^0Pope_{wt}]]]$ is not Double executed, because *b* is related just to the Composition itself constructed by this substitution.

Hyperpropositional attitudes must be used if the attributer is reproducing faithfully *b*'s perspective. For instance, suppose that the office of Pope is identical with the office of the Bishop of Rome. Then it may be the case that though *b* believes that the Pope is wise, he may disbelieve that the Bishop of Rome is wise.

4.2 Leibniz's Rule of Substitution in the Three Kinds of Context

a) In an ***extensional context*** substitution *salva veritate* of *v-congruent* constructions is valid.

Example.

"The president of CR is the husband of Livia Klausova"
"The president of CR is an economist"

"The husband of Livia Klausova is an economist"

Proof:

1. $\lambda w\lambda t\ [^0President\_of_{wt}\ ^0CR]_{wt} \approx_v \lambda w\lambda t\ [^0Husband\_of_{wt}\ ^0Livia]_{wt}$ assumpt.
2. $[^0Economist_{wt}\ \lambda w\lambda t\ [^0President\_of_{wt}\ ^0CR]_{wt}]$ assumpt.

---

3. $[^0Economist_{wt}\ \lambda w\lambda t\ [^0Husband\_of_{wt}\ ^0Livia]_{wt}]$ Leibniz 2)

Types*: President_of*/$(\iota\iota)_{\tau\omega}$; *CR*/$\iota$; *Husband_of*/$(\iota\iota)_{\tau\omega}$; *Livia*/$\iota$; *Economist*/$(o\iota)_{\tau\omega}$;

b) In an ***intensional context*** substitution *salva veritate* of *equivalent* constructions is valid.

Example.

"The president of CR is the highest representative of CR"
"Tom wants to become the president of CR"

"Tom wants to become the highest representative of CR''

Proof:

1. $\lambda w\lambda t\ [^0President\_of_{wt}\ ^0CR] \approx \lambda w\lambda t\ [^0Highest\_Rep\_of_{wt}\ ^0CR]$ assumpt.
2. $[^0Want\_be_{wt}\ ^0Tom\ \lambda w\lambda t\ [^0President\_of_{wt}\ ^0CR]]$ assumpt.

---

3. $[^0Want\_be_{wt}\ ^0Tom\ \lambda w\lambda t\ [^0Highest\_Rep\_of_{wt}\ ^0CR]]$ Leibniz 2)

Additional types. *Highest_Rep_of*/$(\iota\iota)_{\tau\omega}$; *Want_be*/$(o\iota\iota_{\tau\omega})_{\tau\omega}$: the relation-in-intension of an individual to an individual office; *Tom*/ι.

c) In a ***hyperintensional context*** substitution *salva veritate* of *procedurally isomorphic* constructions is valid.

Example. Suppose that 'azure' and 'sky-blue' are synonymous.

"Tom believes* that Marie's blouse is azure"

"Tom believes* that Marie's blouse is sky-blue"

Proof.

1. $[^0Believe^*_{wt}\ ^0Tom\ ^0[\lambda w\lambda t\ [^0Azure_{wt}\ [^0Blouse\_of_{wt}\ ^0Marie]]]]$ assumpt.
2. $^{00}Azure \approx {}^{00}Sky\_Blue$ assumpt.

3. $[^0Believe^*_{wt}\ ^0Tom\ ^0[\lambda w\lambda t\ [^0Sky\_Blue_{wt}\ [^0Blouse\_of_{wt}\ ^0Marie]]]]$ Leibniz

Some might object that this argument is invalid, because it is possible that Tom believes that Marie's blouse is azure without believing that Marie's blouse is sky-blue. We disagree and on this point we refer to Richard who says:

> … It is impossible for a (normal, rational) person to understand expressions which have identical senses but not be aware that they have identical senses. (Richard 2001, 546-7)

Since the meanings of 'azure' and 'sky-blue' are identical by assumption, the two Trivializations ^0Azure and $^0Sky\_Blue$ are not only equivalent but also *identical*, hence *procedurally isomorphic.* Hence the paradox of analysis is *not* a problem of hyperintensionality. Rather, it is a matter of linguistic incompetence (failure to recognize pairs of synonyms) and *not of logical incompetence* (failure to recognize pairs of procedurally isomorphic hyperintensions).

5 Conclusion

Once the three kinds of context, namely extensional, intensional and hyperintensional are defined, the extensional rules like the substitution of identicals and quantification into non-extensional contexts are trivially valid. There is no cogent reason for invalidity of Leibniz's law. Only that we must substitute that object which is the object of predication in a given context.

Quantifying into hyperintensional contexts requires an *extensional logic of hyperintensions*. Much non-trivial footwork is required to lay out such a large-scale logical semantics. Once this is done, though, quanti-

fying into hyperintensional and intensional contexts turns out to be as trivially valid as quantifying into extensional contexts. However quantifying into hyperintensional contexts introduces a *technical complication* absent in quantifying into intensional and extensional contexts. In a hyperintensional context a construction occurs *mentioned* (as an argument of another function) rather than used (to construct a function). The complication is that, since every constituent of a *mentioned* construction itself occurs mentioned, the quantifier cannot bind any variables inside the hyperintensional context, thus rendering quantifying-in impossible. The solution consists in applying a *substitution technique* that makes the variables amenable to binding. Moreover, the substitution method made it possible to introduce the generally valid computational rule of a partial lambda calculus, *viz.* reduction 'by value'.

Department of Computer Science
VSB – Technical University
17. listopadu 15
708 33 Ostrava
Czech Republic
marie.duzi@vsb.cz

Acknowledgements. This research was funded by the Grant Agency of the Czech Republic, project No. 401/10/0792, "Temporal aspects of knowledge and information" and also by the internal grant agency of VSB-TU Ostrava, project No. SP2012/26 „An utilization of artificial intelligence in knowledge mining from software processes". Versions of this study were read by the author as an invited talk at the University of Western Australia, Perth, Australia, February 25th, 2011.

References

Anderson, C. A. (1998): Alonzo Church's contributions to philosophy and intensional logic. *The Bulletin of Symbolic Logic* 4, 129-171.
Bealer, G. (1982): *Quality and Concept*. Oxford: Clarendon Press.
Bealer, G. (2004): An inconsistency in direct reference theory. *The Journal of Philosophy* 111, 574-93.
Carnap, R. (1947): *Meaning and Necessity*. Chicago: Chicago University Press.
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). *Noûs* 27, 141-157.

Cleland, C. E. (2002): On effective procedures. *Minds and Machines* 12, 159-179.
Cresswell, M. J. (1975): Hyperintensional logic. *Studia Logica*, No. 34, 25-38.
Cresswell, M. J. (1985): *Structured Meanings*. Cambridge: MIT Press.
Duží, M. (2010): The paradox of inference and the non-triviality of analytic information. *Journal of Philosophical Logic*, 39, 5, 473-510.
Duží, M. – Jespersen, B. – Materna, P. (2010): *Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic*. First edition. Berlin: Springer.
Duží, M. – Jespersen, B. (to appear): Procedural isomorphism and restricted β-reduction. *Logic Journal of the IGPL*.
Duží, M. – Materna, P. (2010): Can concepts be defined in terms of sets? *Logic and Logical Philosophy*, 19, 195-242.
Eijck, J. van – Francez, N. (1995): Verb-phrase ellipsis in dynamic semantics. In: Masuch, M. – Polos, L. (eds.): *Applied Logic: How, What and Why?* Kluwer, 29-60.
Forbes, G. (1996): Substitutivity and the coherence of quantifying in. *The Philosophical Review*, 105, 337-371.
Fox, C. – Lappin, S. (2001): A framework for the hyperintensional semantics of natural language with two implementations. *Lecture Notes in Computational Linguistics*, vol. 2009, 175-92.
Frege, G. (1892): Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik* 100, 25-50.
Hyde, R. (1996): *The Art of Assembly Language Programming*. Available: http://www.arl.wustl.edu/~lockwood/class/cs306/books/artofasm/toc.html.
Jespersen, B. (2003): Why the tuple theory of structured propositions isn't a theory of structured propositions. *Philosophia* 31, 171-183.
Jespersen, B. (2010): How hyper are hyperpropositions? *Language and Linguistics Compass* 4, 96-106.
Kaplan, D. (1968): Quantifying in. *Synthese* 19, 178-214.
Kaplan, D. (1986): Opacity. In: Hahn, L. (ed.): *The Philosophy of W.V. Quine*. La Salle: Open Court, 229-289.
Kaplan, D. (1990): Dthat. In: Cole, P. (ed.): *Syntax and Semantics*. Vol. 9. New York: Academic Press. Reprinted in: Yourgrau, P. (ed.): *Demonstratives.* Oxford: Oxford University Press.
Klement, K. C. (2002): *Frege and the Logic of Sense and Reference*. New York: Routledge.
Kripke, S. (1963): Semantical considerations on modal logic. *Acta Pilosophica Fennica*, 16, 83-94.
Lewis, D. (1972): General semantics. In: Davidson, D. – Harman, G. (eds.): *Semantics of Natural Language*. Dordrecht: Reidel.
Moschovakis, Y. N. (1994): Sense and denotation as algorithm and value. In: Väänänen, J. – Oikkonen, J. (eds.): *Lecture Notes in Logic*. Vol. 2. Berlin: Springer, 210-249.
Moschovakis, Y. N. (2006): A logical calculus of meaning and synonymy. *Linguistics and Philosophy* 29, 27-89.
Pierce, C. B. (2002): *Types and Programming Languages*. London: MIT Press.

Plotkin, G. D. (1975): Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1, 125-159.
Quine, W. V. O. (1956): Quantifiers and propositional attitudes. *Journal of Philosophy* 53, 177-186.
Richard, M. (2001): Analysis, synonymy and sense. In: Anderson, C. A. – Zeleny, M. (eds.): *Logic, Meaning and Computation*: *Essays in Memory of Alonzo Church*. Synthese Library, vol. 305, Dordrecht: Kluwer, 545-71.
Tichý, P. (1968): Smysl a procedura. *Filosofický časopis* 16, 222-232. Translated as 'Sense and procedure' in Tichý (2004, 77-92).
Tichý, P. (1969): Intensions in terms of Turing machines. *Studia Logica* 26, 7-25. Reprinted in Tichý (2004, 93-109).
Tichý, P. (1988): *The Foundations of Frege's Logic*. Berlin – New York: De Gruyter.
Tichý, P. (1994): The analysis of natural language. *From the Logical Point of View* 3, 42-80. Reprinted in Tichý (2004, 801-841).
Tichý, P. (2004): *Collected Papers in Logic and Philosophy*. Edited by V. Svoboda, B. Jespersen, C. Cheyne. Prague: Filosofia, Dunedin: University of Otago Press.