

Deduction in TIL: From Simple to Ramified Hierarchy of Types

MARIE DUŽÍ

Department of Computer Science, VSB-Technical University Ostrava,
17. listopadu 15, 708 33 Ostrava, Czech Republic.
marie.duzi@gmail.com

RECEIVED: 13-02-2013 • ACCEPTED: 05-03-2013

Abstract: Tichý's Transparent Intensional Logic (TIL) is an overarching logical framework apt for the analysis of all sorts of discourse, whether colloquial, scientific, mathematical or logical. The theory is a *procedural* (as opposed to denotational) one, according to which the meaning of an expression is an abstract, extra-linguistic procedure detailing what operations to apply to what procedural constituents to arrive at the product (if any) of the procedure that is the object denoted by the expression. Such procedures are rigorously defined as TIL *constructions*. Though TIL analytical potential is very large, deduction in TIL has been rather neglected. Tichý defined a sequent calculus for pre-1988 TIL, that is TIL based on the simple theory of types. Since then no other attempt to define a proof calculus for TIL has been presented. The goal of this paper is to propose a generalization and adjustment of Tichý's calculus to TIL 2010. First I briefly recapitulate the rules of simple-typed calculus as presented by Tichý. Then I propose the adjustments of the calculus so that it be applicable to hyperintensions within the ramified hierarchy of types. TIL operates with a single procedural semantics for all kinds of logical-semantic context, be it extensional, intensional or hyperintensional. I show that operating in a hyperintensional context is far from being technically trivial. Yet it is feasible. To this end we introduce a substitution method that operates on hyperintensions. It makes use of a four-place substitution function (called *Sub*) defined over hyperintensions.

Keywords: Existential generalisation – extensional rules – hyperintensions – sequent calculus – substitution.

1 Foundations of TIL

From the formal point of view, TIL is a hyperintensional, partial typed λ -calculus. Thus the syntax of TIL is Church's (higher-order) typed λ -calculus with the important difference that the syntax has been assigned a *procedural* (as opposed to denotational) semantics. TIL λ -terms do not denote functions; rather they denote procedures (constructions in TIL terminology) that produce functions or functional values as their product. A linguistic sense of an expression is an abstract *procedure* detailing how to arrive at an object of a particular logical type denoted by the expression. TIL *constructions* are such procedures. Thus, abstraction transforms into the molecular procedure of forming a function, application into the molecular procedure of applying a function to an argument, and variables into atomic procedures for arriving at their values assigned by a valuation.

There are two kinds of constructions, atomic and compound (molecular). Atomic constructions (*Variables* and *Trivializations*) do not contain any other constituent but themselves; they specify objects (of any type) on which compound constructions operate. The *variables* x, y, p, q, \dots , construct objects dependently on a valuation; they *v-construct*. The *Trivialisation* of an object X (of any type, even a construction), in symbols 0X , constructs simply X without the mediation of any other construction. *Compound* constructions, which consist of other constituents as well, are *Composition* and *Closure*. *Composition* $[F A_1 \dots A_n]$ is the operation of functional application. It *v-constructs* the value of the function f (*valuation-*, or *v-*, *-constructed* by F) at a tuple-argument A (*v-constructed* by A_1, \dots, A_n) if the function f is defined at A , otherwise the *Composition* is *v-improper*, i.e., it *fails* to *v-construct* anything.¹ *Closure* $[\lambda x_1 \dots x_n X]$ spells out the instruction to *v-construct* a function by abstracting over the values of the variables x_1, \dots, x_n in the ordinary manner of the λ -calculus. Finally, higher-order constructions can be used twice over as constituents of composite constructions. This is achieved by a construction called *Double Execution*, 2X , that behaves as follows: If X *v-constructs* a construction X' , and X' *v-constructs* an entity Y , then 2X *v-constructs* Y ; otherwise 2X is *v-improper*, failing as it does to *v-construct* anything.

¹ We treat functions as partial mappings, i.e., flat set-theoretical objects, unlike the *constructions* of functions which are structured procedures consisting of constituents.

TIL constructions, as well as the entities they construct, all receive a type. The formal ontology of TIL is bi-dimensional; one dimension is made up of constructions, the other dimension encompasses non-constructions. On the ground level of the type hierarchy, there are non-constructional entities unstructured from the algorithmic point of view belonging to a *type of order 1*. Given a *base of atomic types* of order 1, the induction rule for forming functional types is applied: where $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \dots \times \beta_n$ to α , denoted ' $(\alpha \beta_1 \dots \beta_n)$ ', is a type of order 1 as well. Constructions that construct entities of order 1 are *constructions of order 1*. They themselves belong to a *type of order 2*, denoted '*₁'. The type *₁ together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, type $(\alpha \beta_1 \dots \beta_n)$, involving *₁ in their domain or range is a *type of order 2*. Constructions belonging to a type *₂ that identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*. And so on *ad infinitum*.²

The first three definitions below constitute the logical heart of TIL.

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 $\alpha, \beta_1, \dots, \beta_m$ ($m > 0$) be types of order 1 over B . Then the collection $(\alpha \beta_1 \dots \beta_m)$ of all m -ary partial mappings from $\beta_1 \times \dots \times \beta_m$ 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). \square

Remark. For the purposes of natural-language analysis, we are currently assuming the following epistemic base of ground types, each of which is part of the ontological commitments of TIL:³

² For details see, for instance, Duži – Jespersen – Materna (2010, Ch. 1.3, 1.4), or Duži – Materna (2012, Ch. 2).

³ TIL is an open-ended system. The above epistemic base $\{o, t, \tau, \omega\}$ was chosen, because it is apt for natural-language analysis, but the choice of base depends on the area and language to be analysed. For instance, possible worlds and times are out of place in case of mathematics, and the base might consist of, e.g., o and v , where v is the type of natural numbers.

- o: the set of truth-values $\{\mathbf{T}, \mathbf{F}\}$;
 u: the set of individuals (a constant universe of discourse);
 τ : the set of real numbers (doubling as temporal continuum);
 ω : the set of logically possible worlds (the 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 in X .
- (iii) The *Composition* $[X Y_1 \dots Y_m]$ is the following *construction*. If X v -constructs a function f of type $(\alpha\beta_1 \dots \beta_m)$, and Y_1, \dots, Y_m v -construct entities B_1, \dots, B_m of types β_1, \dots, β_m , respectively, then the *Composition* $[X Y_1 \dots Y_m]$ v -constructs the value (an entity, if any, of type α) of f at the tuple argument $\langle B_1, \dots, B_m \rangle$. Otherwise the *Composition* $[X Y_1 \dots Y_m]$ does not v -construct anything and so is *v-improper*.
- (iv) The *Closure* $[\lambda x_1 \dots x_m Y]$ is the following *construction*. Let x_1, x_2, \dots, x_m be pair-wise distinct variables v -constructing entities of types β_1, \dots, β_m and Y a *construction* v -constructing an α -entity. Then $[\lambda x_1 \dots x_m Y]$ is the *construction λ -Closure* (or *Closure*). It v -constructs the following function f of the type $(\alpha\beta_1 \dots \beta_m)$. Let $v(B_1/x_1, \dots, B_m/x_m)$ be a valuation identical with v at least up to assigning objects $B_1/\beta_1, \dots, B_m/\beta_m$ to variables x_1, \dots, x_m . If Y is $v(B_1/x_1, \dots, B_m/x_m)$ -improper (see *iii*), then f is undefined at $\langle B_1, \dots, B_m \rangle$. Otherwise the value of f at $\langle B_1, \dots, B_m \rangle$ is the α -entity $v(B_1/x_1, \dots, B_m/x_m)$ -constructed by Y .
- (v) The *Single Execution* 1X is the *construction* that either v -constructs the entity v -constructed by X or, if X v -constructs nothing, 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). \square

The definition of the ramified hierarchy of types decomposes into three parts. First, simple types of order 1 that were already defined by Definition 1. Second, constructions of order n , and third, types of order $n + 1$.

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, \dots, X_m ($m > 0$) be constructions of order n over B . Then $[X X_1 \dots X_m]$ is a *construction of order n over B* .
- (iv) Let x_1, \dots, x_m, X ($m > 0$) be constructions of order n over B . Then $[\lambda x_1 \dots x_m X]$ 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).

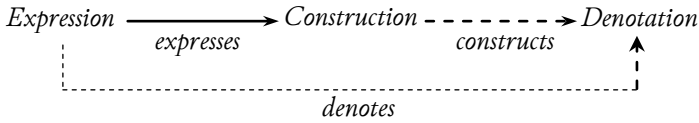
\mathbf{T}_{n+1} (*types of order $n + 1$*) Let $*_n$ be the collection of all constructions of order n over B . Then

- (i) $*_n$ and every type of order n are *types of order $n + 1$* .
- (ii) If $0 < m$ and $\alpha, \beta_1, \dots, \beta_m$ are types of order $n + 1$ over B , then $(\alpha \beta_1 \dots \beta_m)$ (see \mathbf{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). \square

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 a *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 ‘ $\alpha_{\tau\omega}$ ’. We typically say that an index of evaluation is a world/time pair $\langle w, t \rangle$. *Extensional entities* are entities of some type α where $\alpha \neq (\beta\omega)$ for any type β . Where w ranges over ω and t over τ , the following schematic Closure characterizes the logical syntax of any empirical language: $\lambda w \lambda t [\dots w \dots t \dots]$.

Logical objects like *truth-functions* and *quantifiers* are extensional: \wedge (conjunction), \vee (disjunction) and \supset (implication) are of type (ooo), and \neg (negation) of type (oo). *Quantifiers* \forall^α , \exists^α are type-theoretically polymorphous, total functions of type (o(o α)), for an arbitrary type α , defined as follows. The *universal quantifier* \forall^α 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^α is a function that associates a class A of α -elements with **T** if A is a non-empty class, otherwise with **F**. 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 *valuation*-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 intensional descent (a.k.a. extensionalization) of the α -intension v -constructed by C , will be encoded as ' C_{wt} '.

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, i.e., a construction. Once *constructions* have been defined, we can logically examine them; we can investigate *a priori* what (if anything) a construction constructs and what is entailed by it. Thus meanings/constructions are semantically primary, denotations secondary, because an expression denotes an object (if any) *via* its meaning that is a construction *expressed* by the expression. Once a construction is explicitly given as a result of logical analysis, the entity (if any) it constructs is already implicitly given. As a limiting case, the logical analysis may reveal that the construction fails to construct anything by being *improper*.

Any given unambiguous term or expression (even one involving indexicals or anaphoric pronouns) expresses the same construction as its sense whatever sort of context the term or expression is embedded within. And the meaning of an expression determines the respective denoted entity (if any), but not vice versa. The denoted entities are (possibly 0-ary) functions

understood as set-theoretical mappings. Thus we strictly distinguish between a procedure (construction) and its product (here, a constructed function), and between a function and its value. What makes TIL *anti-contextual* and *compositional* is the fact that the theory construes the semantic properties of the sense and denotation relations as remaining invariant across different sorts of linguistic contexts. We do not develop a special extensional logic for extensional contexts, intensional logic for intensional contexts and hyperintensional logic for hyperintensional contexts. Logical operations are universal and context-invariant. What is context dependent are the arguments on which these operations operate. In a *hyperintensional* context they are *constructions* themselves; in an *intensional* context the arguments of logical rules and operations are the *products* of constructions, that is set-theoretical *functions*; finally, in an *extensional* context we operate on functional *values*.

Technically speaking, some constructions are modes of presentation of functions, including 0-place functions such as individuals and truth-values, and the rest are modes of presentation of other constructions. Thus, with constructions of constructions, constructions of functions, functions, and functional values in our stratified ontology, we need to keep track of the traffic between multiple logical strata. The ramified type hierarchy does just that. What is important about this traffic is, first of all, that constructions may themselves figure as functional arguments or values. Thus we consequently need constructions of one order higher in order to present those being arguments or values of functions. Typically, constructions that serve as arguments to operate on are supplied by the two kinds of atomic constructions, viz. Trivialization and variables. For instance, if $x/*_1 \rightarrow \tau$ is a variable belonging to $*_1$, the type of order 2, then ${}^0x/*_2 \rightarrow *_1$ is a construction belonging to $*_2$, the type of order 3, which constructs just the variable x .

It should be clear now that we need to distinguish three kinds of context. Here I only recapitulate the characterizations of these contexts. Rigorous definitions are rather complicated and thus out of the scope of this paper.⁴ When defining the three kinds of context we proceed in a top-down way. First we distinguish two kinds of occurrence of a subconstruction D in a construction C , which means that we define the *use-mention*

⁴ For these rigorous definitions see Duží et al. (2010, §2.6) or Duží – Materna (2012, Chapter 11).

distinction. Then we define two kinds of *using* a construction D as a constituent of C . The constituent D can be used either intensionally or extensionally in C .

The *use-mention* distinction is traditionally understood as the distinction between using an *expression* (or any piece of language) and mentioning it using a quotation in a meta-language. However, we do not analyse the semantics of quotation, which is not to say that it is not an interesting topic in the philosophy of language. Instead, we analyse the semantics of using expressions in a communicative act. An expression E is used to communicate its meaning that we explicate as a TIL construction C . In principle, there are three ways of using an expression within a linguistic discourse.

First, the *meaning* of E can be just *mentioned* as an object of predication rather than used to point at the object denoted by it (if any). This is in particular the case of sentences expressing attitudes. For instance, in the sentence “ a believes explicitly that Cracow is greater than Warsaw” the meaning of the embedded clause ‘Cracow is greater than Warsaw’ rather than the proposition denoted by it is the object of predication, because it is possible that a believes that Cracow is greater than Warsaw without believing that Warsaw is smaller than Cracow. The believer a is related explicitly to this mode of presentation of the proposition that Cracow is greater than Warsaw.

Second, if the meaning of E is *used* to refer to the denoted object (that we conceive as a function) it can be used either *intensionally* or *extensionally*. If the former, then the *entire function* is an object of predication; and if the latter, then the *functional value* is an object of predication.

Hence we must distinguish between three kinds of an occurrence of a subconstruction D in a construction C .

Hyperintensional context within C : the kind of context in which a construction D occurs in such a way that it is not used to v -construct a function (or its value). Instead the *construction D* itself is the argument of another function; the construction is merely *mentioned*. Only in a hyperintensional context can a construction figure as the subject of predication.

Example. “ a is solving the equation $\sin(x) = 0$ ”. Here a cannot be related to the solution, that is to the class of real numbers $\{\dots -2\pi, -\pi, 0, \pi, 2\pi, \dots\}$, because in such a case there would be nothing to solve for a . Rather, a is related to the *meaning* of ‘ $\sin(x) = 0$ ’, which is the con-

struction $\lambda x [[^0\text{Sin } x] = ^00]$. He wants to find out which class of real numbers is so constructed. Thus *Solving* is a relation-in-intension of an individual to a construction, and the analysis of the sentence comes down to this construction:

$$\lambda w \lambda t [^0\text{Solving}_{wt} a [^0[\lambda x [[^0\text{Sin } x] = ^00]]]]$$

Types: *Solving*/(oi^*n) $_{\tau\text{o}}$; $a \rightarrow \iota$; $x \rightarrow \tau$; *Sin*/($\tau\tau$); $0/\tau$; $=/(\text{o}\tau\tau)$.

The Closure $\lambda x [[^0\text{Sin } x] = ^00]$ occurs here hyperintensionally. When evaluating the truth-value of the sentence, we do not evaluate this Closure; it is a matter of a . Hence the whole Closure is the (second) argument of the relation *Solving*, the first being a .

Intensional context of C: the kind of context in which a construction D is used intensionally as a constituent of C to v -construct a function rather than a particular value of the function.

Example. “Sine is a trigonometric function.” This sentence expresses the fact that sine belongs to the class of trigonometric functions. Hence the entire function sine rather than its particular value is an object of predication here.

When *Trigonometric*/($\text{o}(\tau\tau)$) is the class of trigonometric functions of type ($\tau\tau$), the analysis of the sentence comes down to this Composition:

$$[^0\text{Trigonometric } ^0\text{Sin}]$$

^0Sin is used *intensionally* within this Composition. It is not composed with a τ -argument in order to construct a value of the sine function. The subject of predication is not a value but this very function.

Extensional context of C: the kind of context in which a construction D of a function is used extensionally as an instruction to apply the function in order to v -construct a particular value of the function.

Example. “Sine of π equals to zero” expresses the Composition $[[^0\text{Sin } \pi] = ^00]$ where ^0Sin occurs extensionally; the Composition is used to construct the value of the sine function at π . Also, in the previous example, $^0\text{Trigonometric}$ is used extensionally.

The details, however, are somewhat more involved. The basic idea is that a ‘higher’ context is dominant over a ‘lower’ one. Thus, as we have just showed, in the meaning of the sentence “ a is solving the equation $\sin(x) =$

0”, the Closure $\lambda x [[{}^0\text{Sin } x] = {}^00]$ occurs hyperintensionally and it thus produces a hyperintensional context. And since a higher-order context is dominant over a lower-order one, all the subconstructions of this Closure including the construction of the function sine, that is ${}^0\text{Sin}$, occur hyperintensionally as well. Hence the subconstructions $[\lambda x [[{}^0\text{Sin } x] = {}^00]]$, $[[{}^0\text{Sin } x] = {}^00]$, $[{}^0\text{Sin } x]$, ${}^0\text{Sin}$, x , 00 are not constituents of the entire Closure $\lambda w \lambda t [{}^0\text{Solving}_{wt} a [{}^0[\lambda x [[{}^0\text{Sin } x] = {}^00]]]]$. Rather they are only *mentioned* within the second argument of the function v -constructed by ${}^0\text{Solving}_{wt}$. The constituents of this Closure are those subconstructions that are *used* (intensionally or extensionally) to construct a function or a functional value. They are: $\lambda w \lambda t [{}^0\text{Solving}_{wt} a [{}^0[\lambda x [[{}^0\text{Sin } x] = {}^00]]]]$, $[{}^0\text{Solving}_{wt} a [{}^0[\lambda x [[{}^0\text{Sin } x] = {}^00]]]]$, ${}^0\text{Solving}_{wt}$, ${}^0\text{Solving}_{w,t}$, ${}^0\text{Solving}$, w , t , a and ${}^0[\lambda x [[{}^0\text{Sin } x] = {}^00]]$.

Similarly in the meaning of the sentence “Sine a trigonometric function” the Trivialization ${}^0\text{Sin}$ occurs intensionally whereas in the meaning of “The sine of π equals 0” it occurs extensionally. However, if the latter is included in an intensional context, ${}^0\text{Sin}$ occurs intensionally. And if included into a hyperintensional context, it occurs hyperintensionally.

The main reason for defining and taking into account these three kinds of context is this. Not respecting particular levels of abstraction yields many paradoxes and fallacies that arise from an incorrect application of extensional rules like Leibniz’s law of substitution of identicals and the rule of existential generalisation.

Traditionally, the validity of extensional rules has been fielded as a logical criterion for distinguishing (i) extensional/ transparent/’relational’ contexts from (ii) non-extensional/opaque/’notional’ contexts. The idea is that extensional (etc.) contexts are those that validate rules of substitution and quantifying-in. What we are saying is that these rules are valid also in intensional and hyperintensional contexts, but that the feasibility of applying them presupposes that it be done within an extensional logic of hyperintensionals. Deploying a non-extensional logic of hyperintensions generates opacity and thus makes (hyper)intensional contexts logically intractable.⁵ Tichý issues in (1986, 256; 2004, 654) a warning against inter-defining the notion of extensional context and the validity of the rules of substitution of co-referring terms and existential generalization on pain of circularity (where TIL and Quine agree on the use of ‘co-referential’):

⁵ Here I draw on material from Duží – Jespersen (2012) and (2013) where more details can be found.

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 (i) *hyperintensions* presenting functions, for (ii) *functions* (including possible-world intensions), and for (iii) *functional values*. These three levels are squared off with the three kinds of context as introduced above.

The last notion we will need is that of *procedural isomorphism*. Laying out the required semantics requires a fair amount of footwork. The main problem we encounter here is the problem how to define synonymy. The solution might seem to be simple; two expressions are synonymous if they have the same meaning. Since we explicate meanings as constructions, the related problem to solve is the individuation of constructions. When operating on hyperintensional level the individuation up to equivalence is too coarse-grained, since different constructions may be equivalent by constructing one and the same object. On the other hand, constructions are too fine-grained from the procedural point of view. Our working hypothesis is that hyperintensional individuation is procedural individuation and that the relevant procedures are isomorphic modulo α - or η - or restricted β -convertibility. Any two terms or expressions whose respective meanings are procedurally isomorphic are deemed semantically indistinguishable, hence synonymous.

The degree to which ‘functions-in-intension’ should be fine-grained was of the outmost importance for Church.⁶ When summarising Church’s Alternatives, Anderson (1998, 162) presents these options considered by Church. Senses are identical if the respective expressions are (A0) ‘synonymously isomorphic’, (A1) mutually λ -convertible, (A2) logically equivalent. (A0) is α -conversion and synonymies resting on meaning postulates; (A1) is α - and β -conversion; Church also considered Alternative (A1’) that is α -, β - and η -conversion; (A2) is logical equivalence (see Church 1993). (A2), the weakest criterion, was refuted already by Carnap in his (1947), and would not be acceptable to Church, anyway. (A1) is surely more fine-

⁶ For Church, functions-in-intension are modes of presentation of functions-in-extensions that is set-theoretical mappings. Hence functions-in-intension roughly correspond to our constructions.

grained. However, we are not willing to include unrestricted β -conversion. The reasons are these. First, β -conversion is not guaranteed to be an equivalent transformation as soon as partial functions are involved. Second, even in those cases when β -reduction is an equivalent transformation, it can yield a loss of analytic information.⁷

Church's Alternatives (0) and (1) leave room for additional Alternatives. One such would be Alternative ($\frac{1}{2}$), another Alternative ($\frac{3}{4}$).⁸ The former includes α - and η -conversion while the latter adds a *restricted* β -conversion by name. If we must choose, we would prefer Alternative ($\frac{3}{4}$) to soak up those differences between β -transformations that concern only λ -bound variables and thus (at least appear to) lack natural-language counterparts.

Another reason for excluding unrestricted β -conversion is that occasionally even β -equivalent constructions have different natural-language counterparts; witness the difference between attitude reports *de dicto* vs. *de re*. Thus the difference between “*a* believes that Warsaw is smaller than Cracow” and “Warsaw is believed by *a* to be smaller than Cracow” is just the difference between β -equivalent meanings provided the meaning of the ‘Warsaw’ is a proper construction of an individual. Where attitudes are construed as in possible-world semantics, that is as relations to intensions (rather than hyperintensions), the attitude *de dicto* receives the analysis

$$\lambda w \lambda t [{}^0 \text{Believe}_{wt} a \lambda w \lambda t [{}^0 \text{Smaller}_{wt} {}^0 \text{Warsaw} {}^0 \text{Cracow}]]$$

while the attitude *de re* receives the analysis

$$\lambda w \lambda t [\lambda x [{}^0 \text{Believe}_{wt} a \lambda w \lambda t [{}^0 \text{Smaller}_{wt} x {}^0 \text{Cracow}]] {}^0 \text{Warsaw}]$$

Types: *Smaller*/($\text{O}\iota$) $_{\tau\omega}$; $x \rightarrow_v \iota$; *Warsaw, Cracow*/ ι ; *Believe*/($\text{O}\iota\text{O}_{\tau\omega}$) $_{\tau\omega}$.

The *de dicto* variant is the β -equivalent contractum of the *de re* variant. The variants are equivalent because they construct one and the same proposition, the two sentences denoting the same truth-condition. Yet they denote this proposition in *different ways*, hence 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 *construc-*

⁷ For the notion of analytic information, see Duží (2010) and Duží et al. (2010, §5.4). The solution of the problem of the loss of analytic information is proposed in Duží – Jespersen (2013).

⁸ For (A $\frac{1}{2}$) see Jespersen (2010).

tions, has been used to construct this proposition. In this particular case the loss seems to be harmless, though, because there is only one, hence 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. Here is an example. The Compositions

$$\begin{aligned} (C_1) & \quad [\lambda x [{}^0_+ x {}^0_1] {}^0_3] \\ (C_2) & \quad [\lambda y [{}^0_+ {}^0_3 y] {}^0_1] \end{aligned}$$

both β -reduce to the contractum $[{}^0_+ {}^0_3 {}^0_1]$. It is uncontroversial that the contractum can be equivalently expanded back both to (C_1) and (C_2) . However, the problem is that there is no way to reconstruct *which* of (C_1) , (C_2) would be the correct redex. We do not know which function has been applied to which argument.

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*. Restricted β -conversion is just a formal manipulation with λ -bound variables that has much in common with η -conversion and less with β -reduction. The latter is the operation of applying a function $f/(\beta\alpha)$ to its argument value a/α in order to obtain the value of f at a (leaving it open whether a value emerges). It is the fundamental computational rule of functional programming languages. Thus if f is constructed by the Closure $C = \lambda x [\dots x \dots]$ then β -reduction is here the operation of calling the procedure C with a formal parameter x at the actual parameter value a : $[\lambda x [\dots x \dots] {}^0_a]$. Now a construction of the value a is substituted for x and the ‘body’ of the procedure C is computed, which means that the Composition $[\dots {}^0_a \dots]$ is evaluated in order to obtain the value of the function f at a .

No such features can be found in β_r -reduction. If a variable $y \rightarrow_v \alpha$ is not free in C then the β_r -contractum of $[\lambda x [\dots x \dots] y]$ is $[\dots y \dots]$. Now

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 role *Pope*/ ι_{to} .

the evaluation of the Composition [... y ...] does not yield a value of f . As a result we just obtain a formal simplification of $[\lambda x [... x ...] y]$.

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”.¹⁰ The latter sentence expresses

$$\lambda w \lambda t [[\lambda w \lambda t' [\lambda x [{}^0\text{Believe}_{wt'} a \lambda w \lambda t [{}^0\text{Happy}_{wt} x]]]]]_{wt} {}^0 b]$$

This is merely a β_r -expanded form of

$$\lambda w \lambda t [\lambda x [{}^0\text{Believe}_{wt} a \lambda w \lambda t [{}^0\text{Happy}_{wt} x]]] {}^0 b]$$

Thus we define:

Definition 4 (*procedurally isomorphic constructions: Alternative (3/4)*)

Let C, D be constructions. Then C, D are α -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 ${}^0 C \approx_* {}^0 D$, $\approx_*/(o*_n*_n)$, iff there are closed constructions C_1, \dots, C_m , $m \geq 1$, such that ${}^0 C = {}^0 C_1$, ${}^0 D = {}^0 C_m$, and all C_i, C_{i+1} ($1 \leq i < m$) are either α -, η - or β_r -equivalent. \square

There are two weaker relations between constructions that we I will need as well. They are equivalency and v -congruency:

Definition 5 (*congruency and equivalence of 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 . Then

C, D are *v -congruent*, ${}^0 C \approx_v {}^0 D$, iff either C and D v -construct the same α -entity, or both C and D are v -improper;

C, D are *equivalent*, ${}^0 C \approx {}^0 D$, iff C, D are v -congruent for all valuations v . \square

¹⁰ 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*. Or if the believer is a self-assured nominalist then he may protest that while he does believe that a is happy he does not believe that a has any properties. Further research is required to decide one way or the other.

Corollaries

If C, D are *procedurally isomorphic*, then C, D are equivalent, but not *vice versa*: ${}^0C \approx_* {}^0D \Rightarrow {}^0C \approx {}^0D$.

If C, D are equivalent, then C, D are *v-congruent*, but not *vice versa*: ${}^0C \approx {}^0D \Rightarrow {}^0C \approx_v {}^0D$.

Examples

- a) Expressions ‘President of Czech Republic’ and ‘the husband of Livia Klaus’ are co-referential. They just contingently happen to refer to the same individual, Václav Klaus, in the given world and time. After March 8th 2013 they will be no more co-referential. Hence their meanings are *v-congruent* in the given $\langle w, t \rangle$: $[\lambda w \lambda t [{}^0President_of_{wt} {}^0CR]]_{wt} \approx_v [\lambda w \lambda t [{}^0Husband_of_{wt} {}^0Livia]]_{wt}$
- b) Assume that ‘Pope’ and ‘the head of Catholic church’ are co-denotational terms by denoting one and the same office. Then their meanings are *equivalent*: ${}^0Pope \approx \lambda w \lambda t [{}^0Head_of_{wt} {}^0Catholics]$
- c) Assume that the expressions ‘azure’ and ‘sky-blue’ are synonymous. Then their meanings are procedurally isomorphic: ${}^0Azure \approx_* {}^0Sky_Blue$

So much for the logical and philosophical foundations of TIL as it is in 2010. As mentioned above, Tichý defined a sequent calculus only for pre-1988 TIL that differed from the current version of TIL in these three main issues. First, it was based on simple hierarchy of types. Second, as a consequence of the first, pre-1988 TIL constructions did not involve Trivialisation and Double Execution. An object was a construction of itself. Hence this version did not, strictly speaking, distinguish between an object and a mode of presentation of the object. Finally, as a result, constructions were not objects *sui generis*. They could be only used to construct objects but could not figure as arguments of functions. In other words, pre-1988 TIL did not take into account hyperintensional contexts in which we operate on constructions.¹¹

¹¹ In his (1988) Tichý deals with inference in Chapter 13 where he discusses the distinction between a one-dimensional and two-dimensional conception of proofs and argues for the latter. He provides logical and philosophical reasons for his conception of a two-dimensional inference. The two-dimensional view is no doubt a rigorous explication of the way to execute proofs correctly. However, Tichý does not develop a new proof calculus for TIL-1988, nor does he compare the two-dimensional inference and the sequent calculus for TIL as introduced earlier in his (1982) and (1986). Actually,

2. Tichý's sequent calculus

When defining extensional rules for operating in (hyper-)intensional contexts we encounter two main problems, namely the problem of *substitution* of identicals (Leibniz) and *existential generalization*. Tichý proposed a solution of the substitution and existential generalization problem in his (1982, 1986) and defined a sequent calculus for the pre-1988 TiL, that is for extensional and intensional contexts. Moreover, the solution is restricted to the so-called linguistic constructions of the form $\lambda w \lambda t [C_1 C_2 \dots C_m]$ or $\lambda w \lambda t [\lambda x_1 \dots x_m C]$. In order to explain and recapitulate Tichý's calculus and rules I will now use terminology as introduced above. In particular I will speak about a (hyper-)intensional and extensional context though Tichý does not use these terms.

2.1. Substitution and existential generalization

a) *Substitution*. $a = b; C(a/x) \vdash C(b/x)$

This rule seems to be invalid in intensional contexts. For instance, the following argument is obviously invalid:

The President of ČR is the husband of Livie.
Miloš Zeman has been elected for the President of ČR.
Miloš Zeman has been elected for the husband of Livie.

b) *Existential generalization*. $C(a/x) \vdash \exists x C(x)$

Again, in intensional contexts this rule seems to be invalid. For instance, the following argument is obviously invalid:

Miloš Zeman wants to be the President of ČR.
The President of ČR exists.

Now we must take into account that Tichý solves these problems only for a particular case of linguistic constructions. Thus he does not define e.g. substitution of identicals in general. He specifies the intensional de-

the adoption of the sequent calculus is not necessarily connected with the two-dimensional inference, as was evident in Gentzen's work.

scent of a construction C with respect to w , t , or both, and proves under which conditions is the so-descended construction substitutable for another construction.

Ad a) In order to solve the problem of substitution, Tichý introduces in (1986) the notion of *hospitality* of a construction for a variable z occurring in a construction $C(z)$. In principle, there are four cases. If a variable z is (1,1) hospitable, then the construction of the form $[X_{wt}]$ is substitutable for z . That is, z occurs in an extensional (*de re*) context. If a variable z is (1,0) hospitable, then the construction of the form $[X w]$ is substitutable for z . That is, z occurs in an intensional (*de dicto*) context with respect to time t . If a variable z is (0,1) hospitable, then the construction of the form $[X t]$ is substitutable for z . That is, z occurs in an intensional (*de dicto*) context with respect to a world w . Finally, if a variable z is (0,0) hospitable, then the construction of the form X is substitutable for z . That is, z occurs in an intensional (*de dicto*) context with respect to both t and w .

Ad b) Existential generalization. Tichý first defines an *exposure* of a variable. In brief, a variable $z \rightarrow \alpha$ is exposed in a construction C if it is free and occurs extensionally in C , that is it does not occur in an intensional context like $\lambda t [\dots z \dots]$. Second, he takes into account the *hospitality* of a variable z . Then he defines the rule of existential generalisation for extensional contexts like this.

Let $x \rightarrow \alpha$ be (1,1)-hospitable and let $D(k,l)$, $0 \leq k \leq 1$, $0 \leq l \leq 1$, be a construction substitutable for x in a construction $C(x)$. Then the following rule is valid:

$$\lambda w \lambda t C(D(k,l)/x) \vdash \lambda w \lambda t \exists x C(x)$$

This rule needs to be explained. First, $D(k,l) \rightarrow \alpha$, $0 \leq k \leq 1$, $0 \leq l \leq 1$, is an abbreviation for a construction of one of these forms D_{wt} , D_w , D_t , D . Second, if x is exposed and (1,1)-hospitable then the existential generalisation is valid.

Example. “The president of the Czech Republic is an economist.”

$$\lambda w \lambda t [{}^0 \text{Economist}_{wt} {}^0 \text{PCR}_{wt}] \vdash \lambda w \lambda t \exists x [{}^0 \text{Economist}_{wt} x];$$

Types. $\text{Economist}/(\text{o})_{\tau\text{o}}$; $\text{PCR}/\iota_{\tau\text{o}}$: the office of the president of CR; $x \rightarrow_v \iota$.

2.2. Sequent calculus

The basic notions we need are these.

Match is a pair $a:C$, where $a, C \rightarrow \alpha$ and a is an atomic construction. A match $a:C$ is *satisfied* by a valuation v , if a and C v -construct the same object; match $:C$ is satisfied by v , if C is v -improper; matches $a:C$ and $b:C$ are *incompatible*, if a, b construct different objects; matches $a:C, :C$ are *incompatible*.

Sequent is a tuple of the form $a_1:C_1, \dots, a_m:C_m \rightarrow b:D$, for which we use a generic notation $\Phi \rightarrow \Psi$; A sequent $\Phi \rightarrow \Psi$ is *valid* if each valuation satisfying Φ satisfies also Ψ ;

Remark. Note that due to this definition a valid sequent can be viewed as a valid argument. Thus Tichý actually applies here his two-dimensional conception of inference introduced later in 1988 book though he does not explicitly speak about the two-dimensional inference in his 1982 paper.

The rules preserving validity of sequents are specified like this.

Structural rules.

1. $\| \Phi \rightarrow \Psi$, if $\Psi \in \Phi$ (trivial sequent)
2. $\Phi \rightarrow \Psi \| \Phi_s \rightarrow \Psi$, if $\Phi \subseteq \Phi_s$ (redundant match due to monotonicity)
3. $\Phi, \mathcal{G} \rightarrow \Psi$; $\Phi \rightarrow \mathcal{G} \| \Phi \rightarrow \Psi$ (simplification)
4. $\| \Phi \rightarrow y$;y (trivial match)
5. $\Phi \rightarrow \mathcal{G}_1$; $\Phi \rightarrow \mathcal{G}_2 \| \Phi \rightarrow \Psi$, if \mathcal{G}_1 and \mathcal{G}_2 are incompatible
6. $\Phi, : \mathcal{G} \rightarrow \Psi$; $\Phi, y: \mathcal{G} \rightarrow \Psi \| \Phi \rightarrow \Psi$ (y is not free in Φ, Ψ)

Application rules.

7. *a-instance* (modus ponens):
 $\Phi \rightarrow y:[FX_1 \dots X_m]$, $\Phi, f:F, x_1:X_1, \dots, x_m:X_m \rightarrow \Psi \| \Phi \rightarrow \Psi$
 $(f, x_i, \text{ different variables, free in } \Phi, \Psi, F, X_i)$
8. *a-substitution*:
 (i) $\Phi \rightarrow y:[FX_1 \dots X_m]$, $\Phi \rightarrow x_1:X_1, \dots, \Phi \rightarrow x_m:X_m \| \Phi \rightarrow y:[Fx_1 \dots x_m]$
 (ii) $\Phi \rightarrow y:[Fx_1 \dots x_m]$; $\Phi \rightarrow x_1:X_1, \dots, \Phi \rightarrow x_m:X_m \| \Phi \rightarrow y:[FX_1 \dots X_m]$
9. *extensionality*:
 $\Phi, y:[f x_1 \dots x_m] \rightarrow y:[g x_1 \dots x_m]$; $\Phi, y:[g x_1 \dots x_m] \rightarrow y:[f x_1 \dots x_m] \|$
 $\Phi \rightarrow f;g$
 $(y, x_1, \dots, x_m \text{ are different variables that are not free in } \Phi, f, g)$

λ -rules.

10. $\Phi, f:\lambda x_1 \dots x_m Y \rightarrow \Psi \parallel \Phi \rightarrow \Psi$ (f is not free in Φ, Y, Ψ)
11. β -reduction:
 $\Phi \rightarrow y: [[\lambda x_1 \dots x_m Y] X_1 \dots X_m] \parallel \Phi \rightarrow y: Y(X_1 \dots X_m / x_1 \dots x_m)$
 (X_i is substitutable for x_i)
12. β -expansion:
 $\Phi \rightarrow x_1: X_1; \dots; \Phi \rightarrow x_m: X_m; \Phi \rightarrow y: Y(X_1 \dots X_m / x_1 \dots x_m) \parallel$
 $\Phi \rightarrow y: [[\lambda x_1 \dots x_m Y] X_1 \dots X_m]$

So much for Tichý's proof calculus as he introduced it in the two papers (1982) and (1986). In the next Section 3 I am going to generalize the calculus for TIL 2010 as presented in Duží – Jespersen – Materna (2010).

3. Generalization for TIL 2010

Our goal is to generalize the calculus so that it involves *ramified theory of types*, *all kinds* of constructions that is not only Tichý's linguistic constructions, existential generalization to *any context* and substitution of identicals in any kind of context be it extensional, intensional or hyperintensional. For the sake of convenience I first briefly recapitulate the free kinds of context as introduced in Section 1.

3.1. Three kinds of context

Constructions are full-fledged objects that can be not only used to construct an object (if any) but also serve themselves as input/output objects on which other constructions (of a higher-order) operate. Thus we have:

Hyperintensional context: the sort of context in which a construction is not used to v -construct an object. Instead, the construction itself is an argument of another function; the construction is just mentioned.

Example. "Charles is solving the equation $1 + x = 3$." When solving the equation, Charles wants to find out which set (here a singleton) is constructed by the Closure $\lambda x [^0 = [^0 + ^0 1 x] ^0 3]$. Hence this Closure must occur hyperintensionally, because Charles is related to the Closure itself rather than its product, a particular set. Otherwise the seeker would be immediately afinder and Charles's solving would be a pointless activity. The analysis comes down to:

$$\lambda\omega\lambda t \ [^0\text{Solve}_{wt} \ ^0\text{Charles} \ ^0[\lambda x \ [^0 = \ [^0 + \ ^0 1 \ x \] \ ^0 3]]].$$

Intensional context: the sort of context in which a construction C is used to v -construct a function f but not a particular value of f ; moreover, C does not occur within another hyperintensional context.

Example. “Charles wants to be The President of Finland.” Charles is related to the office itself rather than to its occupier, if any. Thus the Closure $\lambda\omega\lambda t \ [^0\text{President}_{of_{wt}} \ ^0\text{Finland}]$ must occur intensionally, because it is not used to v -construct the holder of the office (particular individual, if any). The sentence is assigned as its analysis the construction

$$\lambda\omega\lambda t \ [^0\text{Want}_{to_be_{wt}} \ ^0\text{Charles} \ \lambda\omega\lambda t \ [^0\text{President}_{of_{wt}} \ ^0\text{Finland}]].$$

Extensional context: the sort of context in which a construction C of a function f is used to construct a particular value of f at a given argument, and C does not occur within another intensional or hyperintensional context.

Example. “The President of Finland is watching TV.” The analysis of this sentence comes down to the Closure

$$\lambda\omega\lambda t \ [^0\text{Watch}_{wt} \ \lambda\omega\lambda t \ [^0\text{President}_{of_{wt}} \ ^0\text{Finland}]_{wt} \ ^0\text{TV}].$$

The meaning of ‘the President of Finland’ occurs here with *de re* supposition, i.e. extensionally.

3.2. Extensional calculus of hyperintensions

In this section I generalise Tichý’s extensional rules for substitution and existential generalization so that they be applicable to constructions of any kind (not only linguistic) and in any context. Moreover, I am going to explain the way in which partiality is being ‘propagated up’, and finally I will deal with the problem of β -conversion.

3.2.1. Rules of existential generalisation

Now I specify the *rules for existential generalisation* into the three kinds of context.¹² These rules will be specified in a schematic way. Let $F \rightarrow (\alpha\beta)$; $a \rightarrow \alpha$. We will examine an extensional, intensional and hyperinten-

¹² For details see Duží (2012) and Duží – Jespersen (2012).

sional occurrence of a construction of the schematic form [... $[F a]$...] within a construction D .

a) *Extensional* context

Let an occurrence of a construction [... $[F a]$...] be extensional and let it v -construct the truth-value \mathbf{T} . Then the following rule is truth-preserving:

$$[\dots [F a] \dots] \vdash \exists x [\dots [F x] \dots]; \quad x \rightarrow_v \alpha$$

Example. Pope is wise. \models Somebody is wise.

$$\lambda w \lambda t [{}^0 \text{Wise}_{wt} {}^0 \text{Pope}_{wt}] \models \lambda w \lambda t \exists x [{}^0 \text{Wise}_{wt} x].$$

Types: $\text{Wise}/(\text{oi})_{\tau\omega}$; $\text{Pope}/\iota_{\tau\omega}$; $x \rightarrow \iota$.

Hence we can quantify into an extensional context by an abstraction over the *value* of the function constructed by F . This is quite comprehensible. In an extensional context the value of the function is an object of predication.

b) *Intensional* context

Let $[F a]$ occur intensionally in a construction [... $[F a]$...] that v -constructs \mathbf{T} . Then the following rule is truth-preserving:

$$[\dots [F a] \dots] \vdash \exists f [\dots [f a] \dots]; \quad f \rightarrow_v (\alpha\beta)$$

Example. b believes that Pope is wise. \models There is an office such that b believes that its holder is wise.

$$\lambda w \lambda t [{}^0 \text{Believe}_{wt} {}^0 b \lambda w \lambda t [{}^0 \text{Wise}_{wt} {}^0 \text{Pope}_{wt}]] \models \\ \lambda w \lambda t \exists f [{}^0 \text{Believe}_{wt} {}^0 b \lambda w \lambda t [{}^0 \text{Wise}_{wt} f_{wt}]].$$

Types: $\text{Believe}/(\text{oi}\text{O}_{\tau\omega})_{\tau\omega}$: an intensional belief; b/ι ; $\text{Wise}/(\text{oi})_{\tau\omega}$; $\text{Pope}/\iota_{\tau\omega}$; $f \rightarrow \iota_{\tau\omega}$.

In an intensional context we cannot quantify over the value of a function, because the entire function is an object of predication. Hence we must quantify over the entire function.

c) *Hyperintensional* context

Let $[F a]$ occur hyperintensionally in a construction [... $[F a]$...] that v -constructs \mathbf{T} . Now if we want to validly quantify into hyperproposi-

tional context, we come up against a major technical complication. An attempt analogous to the intensional one above yields:

$$\frac{[\dots \overset{0}{[}\dots [F a] \dots \dots]}{\exists f [\dots \overset{0}{[}\dots [f a] \dots \dots]; f \rightarrow_v (\alpha\beta)}$$

Why is the conclusion no good? The occurrence of f in $\overset{0}{[}\dots [f a] \dots$ – notice the leftmost Trivialization – is $\overset{0}{[}$ bound that is bound by Trivialisation, because the variable f occurs within the hyperintensional context of the construction $[\dots [f a] \dots]$. So f is *mentioned*, hence not available for a direct logical manipulation. It is shielded from \exists by Trivialization in $\overset{0}{[}\dots [f a] \dots$.

Yet it would be a serious flaw of an extensional logic of hyperintensions if one of the fundamental extensional rules were not applicable in a hyperintensional context. For instance, from the premise that a believes* (hyperintensionally) that the Evening Star is a planet it does follow that there is a concept ES of an individual role $Evening_Star/\iota_{\tau_{\text{to}}}$ such that a believes* that its occupant is a planet. But how?

First, we must realize that in a hyperintensional context the object of predication is a *construction* of a function. Hence we must quantify over construction. Second, the solution consists in the application of the following *substitution technique*. A valid argument is obtained by applying the function Sub of the polymorphous type $(*_n*_n*_n*_n)$ that operates on constructions in this way. Let X, Y, Z be constructions of order n . Then Sub is a mapping which, when applied to $\langle X, Y, Z \rangle$, returns the construction that is the result of correctly substituting X for Y in Z . A correct substitution is one that does not make any variable occurring free in X become bound in the resulting construction (no ‘collision’). For illustration, the Composition $[\overset{0}{Sub} \overset{00}{2} \overset{0}{x} \overset{0}{[}\overset{0}{+} \overset{0}{x} \overset{0}{1}]]$ constructs the result of substituting $\overset{0}{2}$ for x into $[\overset{0}{+} \overset{0}{x} \overset{0}{1}]$, which is the Composition $[\overset{0}{+} \overset{0}{2} \overset{0}{1}]$. Therefore, the Composition $[\overset{0}{Sub} \overset{00}{2} \overset{0}{x} \overset{0}{[}\overset{0}{+} \overset{0}{x} \overset{0}{1}]]$ is equivalent to $[\overset{0}{+} \overset{0}{2} \overset{0}{1}]$, both constructing the Composition $[\overset{0}{+} \overset{0}{2} \overset{0}{1}]$.

In Duží – Jespersen (submitted) we analyze four different arguments that share the same premise, but have different conclusions. Two of them are invalid, while the first of the valid ones leads up to the solution of the problem of quantifying into hyperpropositional contexts. Here I reproduce the two valid rules and introduce another one, which I argue for as the most general one.

$$(1) \quad \frac{[\dots {}^0[\dots [F a] \dots] \dots]}{[{}^0\exists^*\lambda c [\dots [{}^0Sub c {}^0c {}^0[\dots [c a] \dots]] \dots]]}$$

Types: $\exists^*/(o(o*_n))$; $c \rightarrow_v *_n$; ${}^2c \rightarrow_v (\alpha\beta)$

Proof.

- (i) ${}^0[\dots [F a] \dots]$ constructs $[\dots [F a] \dots] \quad \emptyset$
- (ii) $[{}^0Sub c {}^0c {}^0[\dots [c a] \dots]]$ $v(F/c)$ -constructs the construction $[\dots [F a] \dots]$; (the first occurrence of c is free)
- (iii) $\lambda c [\dots [{}^0Sub c {}^0c {}^0[\dots [c a] \dots]] \dots]$ constructs a non-empty class of constructions
- (iv) $[{}^0\exists^*\lambda c [\dots [{}^0Sub c {}^0c {}^0[\dots [c a] \dots]] \dots]]$

Note, however, that $[\dots [c a] \dots]$ in the conclusion of (1) comes with *wrong typing* and so is necessarily an *improper* Composition. The variable c ranges over constructions rather than functions, so c cannot be Composed with an argument of a function (see Definition 2, *iii*). To be sure, the impropriety of a construction matters only if the construction is introduced as *used* in order to produce a product, at which it fails. If a construction is merely *mentioned* as an argument of a function, it occurs as an ordinary object that can be operated on, and its failure to produce a product is logically immaterial.

In the present case $[\dots [c a] \dots]$ is an argument of *Sub*, and the Composition $[{}^0Sub c {}^0c {}^0[\dots [c a] \dots]]$ is a *proper* constituent, because c v -constructs a construction of a *function*. In a word, *lazy* evaluation of c in this Composition is an option because the second and third occurrence of c is Trivialization-bound, i.e. mentioned. But, though technically feasible, it is methodologically and philosophically unsatisfactory that *wrong typing* should be an integral part of the solution. Fortunately, there are attractive alternatives:

$$(2) \quad \frac{[\dots {}^0[\dots [F a] \dots] \dots]}{[{}^0\exists^*\lambda c [\dots [{}^0Sub c F {}^0[\dots [F a] \dots]] \dots]]}$$

Proof.

- (i) the Composition $[{}^0Sub c F {}^0[\dots [F a] \dots]]$ $v(F/c)$ -constructs ${}^0[\dots [F a] \dots]$
- (ii) the Closure $\lambda c [\dots [{}^0Sub c F {}^0[\dots [F a] \dots]] \dots]$ constructs a non-empty class
- (iii) $[{}^0\exists^*\lambda c [\dots [{}^0Sub c F {}^0[\dots [F a] \dots]] \dots]]$

This rule does not have a flaw of wrong typing. Yet it is not general enough. The reason is this. The construction F may have more than one occurrence in the hyperintensional context and we may want to quantify only over some of these occurrences. Hence the *general rule for quantifying into hyperpropositional attitudes* is this:¹³

$$(3) \quad \frac{[\dots {}^0[\dots [F a] \dots] \dots]}{[{}^0\exists^*\lambda c [\dots [{}^0Sub\ c\ {}^0d\ {}^0[\dots [d a] \dots]] \dots]]}$$

Additional type: $d \rightarrow (\alpha\beta)$.

Proof.

- (i) the Composition $[{}^0Sub\ c\ {}^0d\ {}^0[\dots [d a] \dots]]\ v(F/c)$ -constructs ${}^0[\dots [F a] \dots]$
- (ii) the Closure $\lambda c [\dots [{}^0Sub\ c\ {}^0d\ {}^0[\dots [d a] \dots]] \dots]$ constructs a non-empty class
- (iii) $[{}^0\exists^*\lambda c [\dots [{}^0Sub\ c\ {}^0d\ {}^0[\dots [d a] \dots]] \dots]]$

Example.

b believes* that Pope is wise. \models There is a concept of an office such that b believes* that its holder is wise.

$$\lambda\omega\lambda t [{}^0Believe^*_{\omega t} {}^0b\ {}^0[\lambda\omega\lambda t [{}^0Wise_{\omega t} {}^0Pope_{\omega t}]]] \models \lambda\omega\lambda t \exists^*c [{}^0Believe^*_{\omega t} {}^0b\ [{}^0Sub\ c\ {}^0d\ {}^0[\lambda\omega\lambda t [{}^0Wise_{\omega t} d_{\omega t}]]]];$$

Types: $Believe^*/(\omega t)_\tau\omega$: hyperpropositional attitude; $c \rightarrow_v *{}_n$; ${}^2c \rightarrow_v \iota_\tau\omega$; $d \rightarrow_v \iota_\tau\omega$.

If we wanted to infer more, namely that there is an *office* such that b believes* that its holder is wise, we would need another assumption, namely that the construction of this office is proper. In this case this additional assumption is valid, because the Trivialisation 0Pope is not v -improper for any valuation v . Hence a stronger argument is also valid:

¹³ This solution has been suggested by Jakub Macek as a reaction to (1). I am indebted to him for making this suggestion.

b believes* that Pope is wise. \models There is an office such that b believes* that its holder is wise.

$$\lambda w \lambda t \left[\left[{}^0 \text{Believe}^*_{wt} \right] {}^0 b \left[\lambda w \lambda t \left[{}^0 \text{Wise}_{wt} \right] {}^0 \text{Pope}_{wt} \right] \right] \wedge \left[{}^0 \text{Proper} \right] {}^0 \text{Pope} \left] \models \right. \\ \left. \lambda w \lambda t \exists f \exists c^* \left[\left[f = {}^2 c \right] \wedge \left[{}^0 \text{Believe}^*_{wt} \right] {}^0 b \left[{}^0 \text{Sub } c \right] {}^0 d \left[\lambda w \lambda t \left[{}^0 \text{Wise}_{wt} \right] d_{wt} \right] \right] \right];$$

Additional type: $f \rightarrow \iota_{\tau_0}$.

3.2.2. Substitution

In order to specify *substitution of identicals* (Leibniz) in the three kinds of context, recall Definitions 4 and 5 of v -congruent, equivalent and procedurally isomorphic constructions. It should be clear now that

- a) In a *hyperintensional context* (propositional attitudes, mathematical sentences, ...) substitution of *procedurally isomorphic* (but neither equivalent nor v -congruent) constructions is valid.
- b) In an *intensional context* (modalities, some notional attitudes, ...) substitution of *equivalent* or procedurally isomorphic (but not only v -congruent) constructions is valid;
- c) In an *extensional context* substitution of v -congruent (or equivalent or procedurally isomorphic) constructions is valid.

Examples.

- a) extensional context

The temperature in Amsterdam is the same
as the temperature in Prague.

The temperature in Amsterdam is 20 °C.

The temperature in Prague is 20 °C.

Proof of the validity of this argument. In any $\langle w, t \rangle$ -pair the following steps are truth-preserving:

1. $\left[{}^0 \text{Temperature}_{in_{wt}} \right] {}^0 \text{Amsterdam} = \left[{}^0 \text{Temperature}_{in_{wt}} \right] {}^0 \text{Prague}$ assumption
2. $\left[{}^0 \text{Temperature}_{in_{wt}} \right] {}^0 \text{Amsterdam} = {}^0 20$ assumption
3. $\left[{}^0 \text{Temperature}_{in_{wt}} \right] {}^0 \text{Prague} = {}^0 20$ substitution 1, 2

Types: $\text{Temperature}_{in}/(\tau \iota)_{\tau_0}$; $\text{Amsterdam, Prague}/\iota$; $20/\tau$.

b) intensional context

To illustrate invalidity of substitution of v -congruent constructions in an intensional context, consider an example of B. Partee:

The temperature in Amsterdam is rising.
The temperature in Amsterdam is 20 °C.

20 °C is rising.

1. [${}^0\text{Rising}_{wt} \lambda w \lambda t$ [${}^0\text{Temperature_in}_{wt} {}^0\text{Amsterdam}$]] assumption
2. [${}^0\text{Temperature_in}_{wt} {}^0\text{Amsterdam}$] = 020 assumption
3. [${}^0\text{Rising}_{wt} {}^020$] ???

Additional types: $\text{Rising}/(\sigma\tau_{\tau\omega})_{\tau\omega}$: the property of magnitude; $\lambda w \lambda t$ [${}^0\text{Temperature_in}_{wt} {}^0\text{Amsterdam}$] $\rightarrow \tau_{\tau\omega}$.

The last step is invalid. In the first assumption the Composition [${}^0\text{Temperature_in}_{wt} {}^0\text{Amsterdam}$] occurs intensionally. To be rising is the property of magnitude (function) rather than of its value. Hence v -congruent constructions [${}^0\text{Temperature_in}_{wt} {}^0\text{Amsterdam}$], 020 cannot be mutually substituted.

As a valid argument we can adduce the recent one:

Benedict XVI resigns as Pope.

Pope is the same office as the Head of the Catholic church.

Benedict XVI resigns as the Head of Catholic church.

Proof of the validity of this argument. In any $\langle w, t \rangle$ -pair the following steps are truth-preserving:

1. [${}^0\text{Resign}_{wt} {}^0\text{Benedict} {}^0\text{Pope}$] assumption
2. [${}^0\text{Pope} = \lambda w \lambda t$ [${}^0\text{Head_of}_{wt} {}^0\text{Catholics}$]] assumption
3. [${}^0\text{Resign}_{wt} {}^0\text{Benedict} \lambda w \lambda t$ [${}^0\text{Head_of}_{wt} {}^0\text{Catholics}$]] substitution 1, 2

Types: $\text{Resign}/(\sigma\iota_{\tau\omega})_{\tau\omega}$; $\text{Pope}/\iota_{\tau\omega}$; $\text{Benedict}/\iota$.

c) hyperintensional context

To illustrate invalidity of substitution of *equivalent* constructions in a hyperintensional context, consider this example:

Marie knows* that Benedict XVI resigns as Pope.
 Pope is the same office as the Head of the Catholic Church.

Marie knows* that Benedict XVI resigns as the Head
 of the Catholic Church.

1. $[{}^0\textit{Know}^*_{wt} {}^0\textit{Marie} [{}^0\lambda w\lambda t [{}^0\textit{Resign}_{wt} {}^0\textit{Benedict} {}^0\textit{Pope}]]]$ assumption
2. $[{}^0\textit{Pope} = \lambda w\lambda t [{}^0\textit{Head_of}_{wt} {}^0\textit{Catholics}]]$ assumption
3. $[{}^0\textit{Know}^*_{wt} {}^0\textit{Marie} [{}^0\lambda w\lambda t [{}^0\textit{Resign}_{wt} {}^0\textit{Benedict}$
 $\lambda w\lambda t [{}^0\textit{Head_of}_{wt} {}^0\textit{Catholics}]]]]$???

Additional type: $\textit{Know}^*/(\text{oι}^*n)_{\tau\text{o}}$: hyperpropositional attitude.

Marie may know (hyperintensionally) that Pope Benedict XVI resigns but does not have to know that Pope is the Head of the Catholic Church. In hyperintensional attitudes we must fully respect the perspective of the attributee.

3.2.3. Sequent calculus and β -conversion

If we apply extensional rules of substitution of identicals and existential generalisation as specified above, Tichý's sequent calculus (or any other extensional calculus) can be applied. Yet we must be aware of the problematic nature of β -conversion. Though it is the fundamental computational rule of λ -calculi, it is underspecified by β -conversion. This rule can be executed in two different ways; 'by value' and 'by name'. There are two problems with the latter.

First, in logic of *partial functions* such as TIL the rule of transformation 'by name'

$$[[\lambda x_1 \dots x_m Y] X_1 \dots X_m] \vdash Y(X_1 \dots X_m / x_1 \dots x_m)$$

is not equivalent, because the left-hand side can be v -improper whereas the right-hand side v -proper by constructing a degenerated function that is undefined for all its arguments. To illustrate it, consider two constructions C_1 and C_2 that are not equivalent, because they construct *different* functions:

$$\begin{array}{ll} C_1 & [[\lambda x [\lambda y [{}^0\textit{Divide} y x]]] [{}^0\textit{Cot} {}^0\pi]] \\ C_2 & [\lambda y [{}^0\textit{Divide} y [{}^0\textit{Cot} {}^0\pi]]] \end{array}$$

Types: $x, y \rightarrow \tau$; $\textit{Divide}/(\tau\tau\tau)$: the function of dividing y by x ; $\textit{Cot}/(\tau\tau)$: the cotangent function; π/τ .

The construction C_1 is the Composition of the Closure $[\lambda x [\lambda y [{}^0\text{Divide } y \ x]]]$ with the Composition $[{}^0\text{Cot } {}^0\pi]$. Since the contagant function is not defined at the argument π , $[{}^0\text{Cot } {}^0\pi]$ is improper by failing to construct anything. Due to compositionality principle the entire Composition C_1 is *improper*, because the function constructed by the Closure $[\lambda x [\lambda y [{}^0\text{Divide } y \ x]]]$ does not receive an argument to be applied at. However, the Closure is never improper, it always constructs a function. Hence C_2 is a proper construction. It constructs a ‘degenerated function’ of type $(\tau\tau)$ undefined on all its arguments. But C_2 is a β -contractum of C_1 , that is the entire Composition $[{}^0\text{Cot } {}^0\pi]$ has been substituted for the variable (formal parameter) x . Thus whereas C_1 does not construct anything, C_2 constructs a degenerated function that is an object (though a peculiar one).

Partiality, as we know all too well, is a complicating factor. Though lambda logic can be modified so as to allow ‘undefined terms’, application of a function in λ -calculi had always been total. E. Moggi (1988) would appear to have been the first to advance a definition of a partial λ -calculus, and S. Feferman (1995) introduced axioms (λ_p) for *Partial Lambda Calculus*. However, they both consider only a ‘total application’ to a term that is denoting.

The second problem is this. Even if we apply an equivalent ‘total β -reduction’, it can yield a loss of analytic information as illustrated in Section 1. This is due to the fact that we do not keep track of the function and arguments that have been used in the transformation.

An analogy from programming languages can be helpful to explain the problem. Imagine you have a procedure (program) $\lambda x \ C(x)$ with a “hole” x (unsaturated procedure with a formal parameter x). It does not make sense to compute $C(x)$ in this stage. Before calling the program $\lambda x \ C(x)$ one must ‘fill in the hole’ x that is to supply an argument (value) for which C should be computed. To this end there is a subprogram D that specifies the material (argument value) to be filled into the hole x .

There are two possibilities how to do it.

1. Insert into the hole x the whole subprocedure D and then compute $C(D)$. This corresponds to calling D ‘by name’.
2. Compute D first in order to obtain “the material” (argument value) a . Then insert a into the hole x and compute $C(a)$. This corresponds to calling D ‘by value’.

In case 1 there may be an undesirable side-effect. Imagine that the subprogram D is somehow garbled and as a result the whole procedure C gets

garbled after the insertion ('damage being propagated up'). Moreover, instead of the hole x you have now got D and D may conflict with C . Again, no good, damage. This corresponds to the case of an invalid beta reduction that does not preserve equivalence. And still moreover, even if D does not damage C when computing $C(D)$, after the execution of $C(D)$ you lost the track of D and of the result D produces. The two procedures have been merged together. Now you want to compute another procedure $E(x)$ and to supply the same material for the hole x . Even if the execution of $C(D)$ were successful, D might have been changed by the execution. There is no guarantee that the same material would be supplied to the hole x in $E(x)$. This case corresponds to a valid β -reduction preserving equivalence but yielding the loss of information.

Fortunately, there is a remedy. Call the subprocedure D 'by value'. The idea is simple. Compute D first to obtain its result a (if any), and then substitute this result for x . This solution is preserving equivalence, avoids the problem of the loss of analytic information, and moreover, it is in practice more effective. If the execution of the subprocedure D fails to produce a product, it is pointless to call $C(x)$ or $E(x)$ or any other procedure that should operate on this product. Thus we know it in advance, rather than only when executing $C(x)$. We keep all the procedures $C(x)$, $E(x)$, D separated and evaluate them only when needed. Everything is all right, as it should be.

Tichý's λ -rules involve β -reduction 'by name'. Due to the version of sequent calculus that operates on *matches* the rule is validity preserving; β -reduction 'by name' in the sequent calculus is this rule:

$$\Phi \rightarrow y: [[\lambda x_1 \dots x_m Y] X_1 \dots X_m] \parallel \Phi \rightarrow y: Y(X_1 \dots X_m / x_1 \dots x_m);$$

(X_i is substitutable for x_i)

The left-hand match is satisfied by a valuation v if y and $[[\lambda x_1 \dots x_m Y] X_1 \dots X_m]$ v -construct the same object. Hence the Composition $[[\lambda x_1 \dots x_m Y] X_1 \dots X_m]$ is v -proper and the rule is validity preserving.

In this way the problem of partiality is avoided rather than solved. It is the standard way to deal with application as presented, for instance, by Moggi and Feferman. The rule does not give us any hint what to do in case that the Composition $[[\lambda x_1 \dots x_m Y] X_1 \dots X_m]$ is v -improper, because in this case the left-hand side match is not satisfied by the valuation v due to the fact that the atomic construction y is always satisfied. Thus the rule is not applicable.

For these reasons we developed a *substitution method* that makes it possible to define a generally valid β -transformation ‘by value’ that does not exhibit the above specified defects. To this end we make use of the function *Sub* defined in Section 3.2.1. Moreover we occasionally need another function $Tr_{\alpha}/(*_n\alpha)$ that takes an object of type α to its Trivialisation. Note that there is a substantial difference between the application of this function and the construction Trivialisation. For instance, if $x \rightarrow_v \iota$, 0x *v*-constructs just the variable x . The variable is 0-bound in 0x and thus it occurs hyperintensionally. On the other hand $[Tr_{\iota} x]$ *v*(*John/x*)-constructs 0John . The variable x is free in $[Tr_{\iota} x]$ and thus occurs intensionally.

Let $x_i \rightarrow_v \alpha_i$ be mutually distinct variables and $D_i \rightarrow_v \alpha_i$ ($1 \leq i \leq m$) constructions. Then the following rule of β -reduction ‘by value’ is valid:

$$[[[\lambda x_1 \dots x_m Y] D_1 \dots D_m] \vdash^2 [{}^0Sub [{}^0Tr_{\alpha_1} D_1] {}^0x_1 \dots \\ [{}^0Sub [{}^0Tr_{\alpha_m} D_m] {}^0x_m {}^0Y]]]$$

Note the Double Execution on the right-hand side. The result of applying *Sub* is a *construction* that must be afterwards executed; hence Double Execution.

Example. “John loves his own wife. So does the Mayor of Ostrava.”

$$\begin{aligned} \lambda w \lambda t [\lambda x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]] {}^0John &=_{\beta v} \\ \lambda w \lambda t [{}^2[{}^0Sub {}^0John {}^0x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]] \\ \lambda w \lambda t [so_does_{wt} {}^0MO_{wt}] &\rightarrow \\ \lambda w \lambda t [{}^2[{}^0Sub [{}^0\lambda w \lambda t \lambda x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]] {}^0so_does [{}^0so_does_{wt} \\ {}^0MO_{wt}]]] &=_{\beta v} \lambda w \lambda t [\lambda x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]] {}^0MO_{wt} =_{\beta v} \\ \lambda w \lambda t [{}^2[{}^0Sub [{}^0Tr {}^0MO_{wt}] {}^0x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]] &]. \end{aligned}$$

Types. $Lovel/(oi)_{\tau_{00}}$; $Wife_{of}/(\iota)_{\tau_{00}}$; $John/\iota$; $MO/\iota_{\tau_{00}}$: the office of the Mayor of Ostrava; $x \rightarrow_v \iota$.

One can easily check that in all these constructions, whether reduced or non-reduced, the track of the property of loving *one’s own* wife is being kept. This property is constructed by the Closure $\lambda w \lambda t \lambda x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}} x]]$. When applied to John it does not turn into the property of loving John’s wife. And the same property is substituted for the variable *so_does* into the second sentence. Thus we can easily infer that John and the Mayor of Ostrava share the property of loving their own wives. If we used β -reduction ‘by name’ the Closure would be reduced to $\lambda w \lambda t [{}^0Love_{wt} {}^0John [{}^0Wife_{of_{wt}} {}^0John]]$. No doubt that it can be β -expanded to the original Closure. The problem is that it can be also expanded to another Closure $\lambda w \lambda t [\lambda x [{}^0Love_{wt} x [{}^0Wife_{of_{wt}}$

${}^0John]] {}^0John]$, which means that the property of loving *John's* wife has been applied to John. In this way we lost the track of the property that has been applied to John and that we want to apply to Mayor of Ostrava. And it also shows exactly how the β -reduction by value works to our advantage.¹⁴

4. Conclusion

We described generalization of Tichý's sequent calculus to the calculus for TIL 2010. The generalization concerns these issues. First, the extensional rules of quantifying in and substitution of identicals were generalized so that to be valid in any context, including intensional and hyperintensional ones. Second, we showed that the sequent calculus remains to be the calculus for TIL based on the ramified hierarchy of types with one important exception, which is the rule of β -reduction. We specified a generally valid rule of β -reduction 'by value' that does not yield a loss of analytic information about which function has been applied to which argument. No doubt that these are valuable results.

Yet some open problems remain. Among them there are in particular the question on the properties of the calculus like completeness and the problem of its implementation. There is also a question whether another similarly extensional calculus of hyperintensions would not be more convenient for implementation. To this end we develop a computational variant of TIL, the functional programming language *TIL-Script*. Till now we managed to develop and test the modules for recognizing particular types of context and we implemented β -reduction by value that we use universally. Moreover, we specified and implemented the algorithm that makes it possible to exploit Prolog inference machine (see Duží et. al. 2009). However, the full-fledged TIL inference machine is still a future work.

Acknowledgements

The research reported herein was funded by Grant Agency of the Czech Republic, Projects No. 401/10/0792, "Temporal Aspects of Knowledge and Information" and also by the internal grant agency of VSB-Technical University Ostrava, Project No. SP2013/207 "Application of artificial intelligence in process-knowledge mining, modeling and management".

¹⁴ For details see Duží – Jespersen (2013).

References

- ANDERSON, C. A. (1998): Alonzo Church's contributions to philosophy and intensional logic. *The Bulletin of Symbolic Logic* 4, 129-171.
- CARNAP, R. (1947): *Meaning and Necessity*. Chicago: Chicago University Press.
- CHURCH, A. (1993): A revised formulation of the logic of sense and denotation. *Alternative* (1). *Noûs* 27, 141-157
- DUŽÍ, M. (2010): The paradox of inference and the non-triviality of analytic information. *Journal of Philosophical Logic* 39, No. 5, 473-510.
- DUŽÍ, M. (2012): Towards an extensional calculus of hyperintensions. *Organon F* 19, supplementary issue 1, 20-45.
- DUŽÍ, M. – ČÍHALOVÁ, M. – CIPRICH, N. – MENŠÍK, M. (2009): Agents' reasoning using TIL-Script and Prolog. In: Weltzer Družovec T. - Jaakkola, H. – Kiyoki, Y. – Tokuda, T. – Yoshida, N. (eds.): *Information Modelling and Knowledge Bases XXI*. Amsterdam: IOS Press, 2010, 135-154.
- DUŽÍ, M. – JESPERSEN, B. – MATERNA, P. (2010): *Procedural Semantics for Hyperintensional Logic: Foundations and Applications of Transparent Intensional Logic*. Series Logic, Epistemology and the Unity of Science. Berlin, Heidelberg: Springer.
- DUŽÍ, M. – JESPERSEN, B. (2012): Transparent quantification into hyperintensional contexts *de re*. *Logique & Analyse* 220, 513-554.
- DUŽÍ, M. – JESPERSEN, B. (submitted): Transparent quantification into hyperpropositional contexts *de dicto*. Paper in revision.
- DUŽÍ, M. – JESPERSEN, B. (2013): Procedural isomorphism, analytic information, and β -conversion by value. *Logic Journal of the IGPL* 21, 291-308. doi: 10.1093/jigpal/jzs044.
- DUŽÍ, M. – MATERNA, P. (2012): *TIL jako procedurální logika. Průvodce zřídavého čtenáře Transparentní intensionální logikou*. Bratislava: aleph.
- FEFERMAN, S. (1995): Definedness. *Erkenntnis* 43, 295-320.
- JESPERSEN, B. (2010): Hyperintensions and procedural isomorphism: Alternative (1/2). In *The Analytical Way*. Proceedings of the 6th European Congress of Analytic Philosophy, ECAP VI. London: College Publications, 299-320.
- MOGGI, E. (1988): *The Partial Lambda-Calculus*. PhD thesis. University of Edinburg, available as *LFCS report* at <http://www.lfcs.inf.ed.ac.uk/reports/88/ECS-LFCS-88-63/>.
- TICHÝ, P. (1982): Foundations of partial type theory. *Reports on Mathematical Logic* 14, 52-72. Reprinted in Tichý (2004, 467-480).
- TICHÝ, P. (1986): Indiscernibility of identicals. *Studia Logica* 45, 251-273. Reprinted in Tichý (2004, 649-671).
- TICHÝ, P. (1988): *The Foundations of Frege's Logic*. Berlin, New York: De Gruyter.
- TICHÝ, P. (2004): *Collected Papers in Logic and Philosophy*. V. Svoboda, B. Jespersen, C. Cheyne (eds.). Prague: Filosofia, Czech Academy of Sciences; Dunedin: University of Otago Press.