Deductive Reasoning using TIL

Martina Číhalová, Nikola Ciprich, Marie Duží, Tomáš Frydrych, and Marek Menšík

VŠB-Technical University Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
m.tina.cihal@gmail.com, nikola.ciprich@linuxbox.cz, marie.duzi@vsb.cz, frydrych.t@gmail.com, mensikm@gmail.com

Abstract. Transparent Intensional Logic (TIL) is a highly expressive logical system apt for the logical analysis of natural language. It operates with a single *procedural semantics* for all kinds of logical-semantic context, whether extensional, intensional or hyper-intensional, while adhering to the compositionality principle throughout. The reason why we vote for a rich procedural semantics is this. A coarse-grained analysis of assumptions yields paradoxes and puzzles, while an expressive formal system such as TIL makes it possible to build an inference machine that neither over-infers (which yields paradoxes) nor under-infers (which leads to the lack of knowledge). From the formal point of view, TIL is a hyperintensional, partial, typed lambda calculus. By way of examples we illustrate how TIL deals with particular 'puzzles' in a smooth way while adhering to Leibniz's law of substitution of identicals and to the principle of compositionality.

Key words: TIL; deductive reasoning; *TIL-Script* language; inference machine

1 Introduction

The way we understand the enterprise of logical analysis of natural language in TIL is selective. The analysis leaves aside pragmatic features of language, but makes all the logically salient features explicit and logically tractable. Yet the very name of our theory, 'Transparent Intensional Logic', is likely to strike one as being an oxymoron, like 'roaring silence'. How can there possibly be a *logic* that is *intensional* and at the same time *transparent*? Isn't any intensional logic such that it fails to heed various laws of extensional logic, such as referential transparency, substitution of identicals, and compositionality? Certainly yes, if 'intensional' is synonymous with 'non-extensional'. But 'intensional' may also mean—and this is the notion of intensionality germane to TIL—that the logic in question comes with a rich ontology of entities and the means to logically manipulate these entities. Due to its rich ontology of entities organized in a bi-dimensional that is ramified hierarchy of types TIL flouts none of the principles of *extensional* logic and is, insofar, an *extensional* logic.[1]

[1] Portions of this paper draw on material presented in [7], in particular Sections 2.6 and 2.7.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2009, pp. 25–38, 2009.

TIL operates with a single *procedural semantics* for all kinds of logical-semantic context, whether extensional, intensional or hyper-intensional. It means that it explicates the meaning of an expression as an abstract procedure encoded by the expression. Such procedures are rigorously defined as TIL constructions and we assign them to expressions as their context-invariant meanings. From the formal point of view, TIL is a hyper-intensional, partial, typed λ-calculus. Hyper-intensional, because the terms of the TIL formal language in which constructions are encoded are interpreted as procedures (generalized algorithms) rather than their products; partial, because the primitive notion of TIL is a function understood as a partial mapping that assigns to each element of its domain *at most* one element of its range; and typed, because all the entities of TIL ontology, including constructions, receive a type.

Yet such an expressive system is often characterized as being logically or computationally intractable. In our opinion, it is not the language in which a problem is encoded what can be intractable, but the *problem* itself. Problems are easy or difficult to solve. For instance, it is a well known fact that though the problem of logical validity in propositional logic is decidable, it is not computationally tractable. In Stephen Cook's famous [3] the theorem is proved that the satisfiability problem is NP-complete, and the tautology problem is co-NP-complete. This means that by a commonly accepted conjecture, these problems are regarded as computationally intractable. And it is also a well-known fact that as a consequence of Gödel's incompleteness theorem, the problem of logical validity in the first-order predicate logic is not even algorithmically decidable. Does it mean that we should reduce our reasoning to computationally tractable sub-systems of propositional logic? Certainly not. Though a great expressive power is inversely proportional to an easy implementation of a suitable deductive system, we need to know *what* should be solved prior to seeking plausible methods of the problem in question solving.

Moreover, there is another strong reason to vote for an expressive, fine-grained semantics. A coarse-grained analysis of assumptions yields paradoxes and puzzles. On the other hand, a rich formal system such as TIL makes it possible to build an inference machine that in principle neither over-infers (which yields paradoxes) nor under-infers (which leads to the lack of knowledge). However, there are two problems connected with TIL deduction system.

First, since the system is hyper-intensional, which means that the meaning of an expression is a *construction* specified by the analyzed expression, rather than the product of the construction denoted by the expression, we must strictly distinguish between constructions and their products. This amounts for distinguishing a context in which a construction is used to produce an entity (if any), and the *hyper-intensional* context in which the construction itself is only *mentioned* as an object of predication. And if the construction is used, we must distinguish the context in which it is *used intensionally* and the context in which it is *used extensionally*. If the former, then the so constructed *function* is an object of predication; and if the latter, the *value* of the constructed function is an object of predication. Only then can we specify rules of deduction for extensional, intensional as well as hyper-intensional context.

The second problem is *partiality*. The primitive notion of TIL is function rather than relation. And it is a brute fact that we need to work with *properly* partial functions, i.e. functions that are undefined at some arguments. The problem crops up when a properly partial function is applied to an argument at which the function does not return any value. Traditional formal lambda-calculi avoid this problem by simply excluding non-denoting terms as meaningless. Yet, this is not a plausible solution for the semantics of natural language. Though, for instance, the present King of France does not exist (the office is vacant) this does not mean that the term 'King of France' is meaningless. If it were, we could not reasonably and truly assert that the King of France does not exist. Similarly mathematicians had to know the meaning of 'the greatest prime' prior to proving that there is no greatest prime. Thus we cannot avoid application of a properly partial function to an argument. Only we have to take into account that the operation can fail to produce a product.[2]

The goal of the paper is not to introduce the TIL inference machine in full. Instead, by way of examples, we illustrate how TIL deals with particular 'puzzles' in a smooth way while adhering to Leibniz's law of substitution of identicals and the principle of compositionality in all kinds of context. TIL constructions are assigned to semantically unambiguous expressions as their *context-invariant* meanings. What differs dependently on the context in which the analyzed expression is used are *logical operations* applied to a meaning constituent rather than the assigned meaning itself.

The paper is organized as follows. The next Section 2 presents basic principles and definitions of TIL. In Section 3 we first characterize the three kinds of context in which a construction can occur, and then we illustrate by examples the unique way how TIL operates in these contexts. Concluding Section 4 sums up the results and outlines further research.

2 TIL in brief

Since Frege's pioneering paper [8] logicians and semanticists have striven to define so-called *structured meanings* that would comply with the principles of compositionality and universal referential transparency. Various adjustments of Frege's semantic schema have been proposed, shifting the entity named by an expression from the extensional level of atomic (physical/abstract) objects to the intensional level of abstract objects such as sets or functions/mappings. Yet natural language is rich enough to generate expressions that talk neither about extensional nor intensional objects. Propositional attitudes are notoriously known as the hard cases that are neither extensional nor intensional, as Carnap

[2] To this end we introduce a generally valid rule of β-reduction 'by value'. The rule has been precisely defined in [7], Section 2.6. Roughly, the idea resembles lazy evaluation of functional programming languages. We first check whether the construction is proper and only then substitute the Trivialization of the constructed entity for the formal parameter x. See also [4] where this substitution method is applied to anaphora pre-processing.

in [1] characterized them. It has become increasingly clear since the 1970s that we need to individuate meanings more finely than by possible-world intensions, and the need for *hyperintensional* semantics is now broadly recognised. Our position is a plea for such a semantics, which takes expressions as encoding *algorithmically structured procedures* producing extensional/intensional entities (or lower-order procedures) as their products. This approach—which could be characterized as being informed by an *algorithmic* or *computational turn*—has been advocated by, for instance, Moschovakis in [10]. Yet much earlier, in the early 1970s, Tichý introduced his notion of *construction* and developed the system of Transparent Intensional Logic, as presented in [11] and [12].

Constructions, as well as the entities they construct, all receive a type. The ontology of TIL is organized in an infinite, bi-dimensional hierarchy of types. One dimension is made up of non-constructions, i.e., entities unstructured from the algorithmic point of view. The other dimension of the type hierarchy is made up of structured, higher-order constructions which construct lower-order entities. Thus our definitions are inductive, and they proceed in three stages. First, we define the simple types of order 1 comprising non-constructions. Then we define constructions and, finally, the ramified hierarchy of types.

Definition 1 *(Types of order 1)* Let B be base, i.e., a collection of non-empty sets.

1. Every member of B is a *type of order 1 over B*.
2. Let $\alpha, \beta_1, \ldots, \beta_m$ be *types of order 1*. Then the set $(\alpha\beta_1 \ldots \beta_m)$ of partial functions with values in α and arguments in $\beta_1, \ldots, \beta_m$, respectively, is a *type of order 1 over B*.
3. Nothing is a *type of order 1 over B* unless it so follows from (1) and (2). □

The choice of the base depends on the area and language we happen to be investigating. When investigating purely mathematical language, the base can consist of, e.g., two atomic types; o, the type of truth-values, and v, the type of natural numbers. When analyzing an ordinary natural language, we use the *epistemic base* which is a collection of four atomic types, o, ι, τ, ω, where o= $\{\mathbf{T}, \mathbf{F}\}$ is the set of truth-values, ι is the universe of discourse (members: individuals), τ is the set of real numbers (or of time moments) and ω is the logical space, the set of possible worlds.

Definition 2 *(intensions and extensions)*; (PWS) *intensions* are entities of type $(\beta\omega)$: mappings from possible worlds to some type β. The type β is frequently the type of the *chronology* of α-objects, i.e., mapping of type $(\alpha\tau)$. Thus α-intensions are frequently functions of type $((\alpha\tau)\omega)$, abbreviated as '$\alpha_{\tau\omega}$'. *Extensions* are entities of a type α where $\alpha \neq (\beta\omega)$ for any type β. □

Examples of frequently used intensions are: *Propositions* (denoted by declarative sentences) are of type $o_{\tau\omega}$; *properties of individuals* (usually denoted by nouns or intransitive verbs like 'is a student', 'walks') are of type $(o\iota)_{\tau\omega}$; binary *relations-in-intension* between individuals are of type $(o\iota\iota)_{\tau\omega}$;[3] *individual offices/roles* (cf.

[3] Since *function* rather than relation is a primitive notion of TIL, we model *sets* and relations by their characteristic functions. Thus, for example, the set of prime numbers is a function of type $(o\tau)$ that associates any number with **T** or **F** according as the given number is a prime.

Church's individual concepts, usually denoted either by superlatives like 'the highest mountain' or terms with built-in uniqueness, like 'The President of the USA') are of type $\iota_{\tau\omega}$. *Expressions* which denote non-constant intensions (i.e. functions that take different values in at least two world-time pairs) are *empirical*.

Quantifiers $\forall^\alpha, \exists^\alpha$, are extensions, viz. type-theoretically polymorphous functions of type(s) $(o(o\alpha))$ defined as follows: The *universal quantifier* $\forall^\alpha$ is a function that associates a class C of α-elements with **T** if C contains all elements of the type α, otherwise with **F**. The *existential quantifier* $\exists^\alpha$ is a function that associates a class C of α-elements with **T** if C is a non-empty class, otherwise with **F**. The *singulariser* $Sing^\alpha$ is a partial type-theoretically polymorphic function of type(s) $(\alpha(o\alpha))$ that associates a class C with the only α-element of C if C is a singleton, otherwise the function $Sing^\alpha$ is undefined. We will often use the abbreviated notation '$\forall x\, A$', '$\exists x\, A$' and '$\iota x\, A$' instead of '$[^0\forall^\alpha \lambda x\, A]$', '$[^0\exists^\alpha \lambda x\, A]$', '$[^0 Sing^\alpha \lambda x\, A]$', respectively, when no confusion can arise.

When claiming that constructions are algorithmically structured, we mean the following. A construction C consists of one or more particular steps, or *constituents*, that are to be individually executed in order to execute C. The objects a construction operates on are not constituents of the construction. Just like the constituents of a computer program are its sub-programs, so the constituents of a construction are its sub-constructions. Thus on the lowest level of non-constructions, the objects that constructions work on have to be supplied by other (albeit trivial) constructions. The constructions themselves may occur not only as constituents to be executed, but also as objects that still other constructions operate on. Therefore, one should not conflate *using* constructions as constituents of compound constructions and *mentioning* constructions that enter as input/output objects into compound constructions. Mentioning is, in principle, achieved by using atomic constructions. A construction C is atomic if it does not contain any other construction as a used sub-construction (a 'constituent' of C) but C. There are two atomic constructions that supply entities (of any type) on which compound constructions operate: *Variables* and *Trivializations*. *Compound* constructions, which consist of other constituents than just themselves, are *Composition* and *Closure*. *Composition* is the instruction to apply a function to an argument in order to obtain its value (if any) at the argument. It is *improper*, i.e., does not construct anything, if the function is not defined at the argument. *Closure* is the instruction to construct a function by abstracting over variables in the ordinary manner of the λ-calculi. Finally, higher-order constructions can be used once or twice over as constituents of constructions. This is achieved by a fifth and sixth construction called *Execution* and *Double Execution*, respectively.

Definition 3 *(construction)*

1. The *Variable* x is a **construction** that constructs an object O of the respective type dependently on a valuation v; it v-constructs O.
2. *Trivialization*: Where X is an object whatsoever (an extension, an intension or a construction), ^{0}X is the **construction** of *Trivialization*. It constructs X without any change.

3. The *Composition* $[XY_1 \dots Y_m]$ is the following **construction**. If X v-constructs a function f of a type $(\alpha\beta_1 \dots \beta_m)$, and $Y_1, \dots, Y_m$ v-construct entities $B_1, \dots, B_m$ of types $\beta_1, \dots, \beta_m$, respectively, then the Composition $[XY_1 \dots Y_m]$ v-constructs the value (an entity, if any, of type α) of f on the tuple-argument $\langle B_1, \dots, B_m \rangle$. Otherwise the *Composition* $[XY_1 \dots Y_m]$ does not v-construct anything and so is v-improper.
4. The *Closure* $[\lambda x_1 \dots \lambda x_m Y]$ is the following construction. Let $x_1, x_2, \dots, x_m$ be pairwise distinct variables v-constructing entities of types $\beta_1, \dots, \beta_m$ and Y a construction v-constructing an α-entity. Then $[\lambda x_1 \dots \lambda x_m Y]$ is the **construction** λ-Closure (or *Closure*). It v-constructs the following function $f/(\alpha\beta_1 \dots \beta_m)$. 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 3), then f is undefined on $\langle B_1, \dots, B_m \rangle$. Otherwise the value of f on $\langle B_1, \dots, B_m \rangle$ is the α-entity $v(B_1/x_1, \dots, B_m/x_m)$-constructed by Y.
5. The *Execution* ^1X is the **construction** that either v-constructs the entity v-constructed by X or, if X v-constructs nothing, is v-improper.
6. The *Double Execution* ^2X is the following **construction**. Let X be 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 X' and X' v-construct an entity Y. Then ^2X v-constructs Y.
7. Nothing is a **construction**, unless it so follows from (1) through (6). □

Notation and abbreviations:

- 'X/α' means that the 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 use '$X \rightarrow \alpha$' if what is v-constructed does not depend on a valuation v.
- We will standardly use the variables $w \rightarrow_v \omega$ and $t \rightarrow_v \tau$;
- If $C \rightarrow_v \alpha_{\tau\omega}$, the frequently used Composition $[[C\,w]\,t]$, the intensional descent of the α-intension v-constructed by C, will be written as 'C_{wt}'.
- When using constructions of truth-value functions, namely $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication) of type (ooo), and $\neg$ (negation) of type (oo), we often omit Trivialisation and use infix notion.
- When using identity relations $=^{\alpha}$ $/(o\alpha\alpha)$, we often omit the superscript α and use infix notation, whenever no confusion arises.

As mentioned above, constructions themselves are objects and thus also receive a type. Only it cannot be a type of order 1, because a construction cannot be of the same type as the object it constructs. Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted by '$\star_1$'. This type $\star_1$, together with atomic types of order 1, serves as the base for the following induction rule: any collection of partial mappings, type $(\alpha\beta_1 \dots \beta_n)$, involving $\star_1$ in their domain or range is a *type of order 2*. Constructions belonging to the type $\star_2$, which 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*.

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

Definition 4 *(Ramified hierarchy of types)* Let B be base.
$\boldsymbol{T_1}$ (types of order 1)–defined by Definition 1.
$\boldsymbol{C_n}$ (constructions of order n)

1. Let x be a variable ranging over a type of order n. Then x is a *construction of order n over B*.
2. Let X be a member of a type of order n. Then ^{0}X, ^{1}X, ^{2}X are *constructions of order n over B*.
3. Let X, $X_1 \ldots X_m (m > 0)$ be constructions of order n over B. Then $[XX_1 \ldots X_m]$ is a *construction of order n over B*.
4. Let $x_1 \ldots x_m, X (m > 0)$ be constructions of order n over B. Then $[\lambda x_1 \ldots \lambda x_m\, X]$ is a *construction of order n over B*.
5. Nothing is a construction of order n over B unless it so follows from $\boldsymbol{C_n}$ (1)-(4).

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

1. $*_n$ and every type of order n are types of order $n+1$.
2. 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 $\boldsymbol{T_1}$ 2) is a type of order $n+1$ over B.
3. Nothing is a type of order $n+1$ over B unless it so follows from T_{n+1} (1) and (2). □

So much for the philosophy and basic definitions of TIL.

3 The outline of TIL calculus

In this section we deal with the deduction system based on TIL. We are not going to define the calculus entirely, since it would be beyond the scope of this paper. Instead we informally explain particular rules as they are valid in the three kinds of context and illustrate them by examples.

3.1 Three kinds of context

As mentioned above, 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. This is so, because expressions of natural language, when used in a communicative act, can be used in three different ways. True, expressions are always used to express their *meaning* explicated in TIL as a *construction*. But when using an expression E, its meaning C can occur with three different suppositions:

1. The meaning C is not used to identify an object about which something is predicated; rather, C itself is an object of predication within another expression E' of which E is a sub-expression. We will say that the meaning C (and thus also the expression E) occurs *hyper-intensionally*.
2. The meaning C is used to identify an object that is a function F (possibly a 0-ary one, which is a function without arguments). Now again there are two possible suppositions:
   (a) The function F itself is an object of predication; in this case we say that the meaning C (and thus also the expression E) is used *intensionally*.
   (b) The value of F is an object of predication; in this case we say that the meaning C (and thus also the expression E) is used *extensionally*.

Note that the notions 'intensionally' and 'extensionally' are used here in a broader sense than in possible-world semantics. Whenever confusion might arise, we will explicitly say PWS-intension. Using medieval terminology, we will also talk about *de dicto* and *de re* supposition, in case that a construction of a *PWS-intension* occurs *intensionally* and *extensionally*, respectively.

Thus we must distinguish between the context of *mentioning* a construction hyper-intensionally as an input/output object on which another construction operates and *using* a construction as a constituent of another construction in two different ways, either intensionally or extensionally. However, there is another complication here. A higher context is *dominant* over a lower one. It means that if a meaning C occurs extensionally as a constituent of another construction C' which in turn occurs intensionally (as a constituent of some D), then C occurs in C' (as well as in D) intensionally. And if a meaning C is used extensionally or intensionally as a constituent of another construction C' which in turn occurs hyper-intensionally (as mentioned in some D), then C occurs in C' (as well as in D) hyper-intensionally.

The three kinds of context are specified as follows[4]:

1. *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.
   Examples: Consider the sentence "Charles is solving the equation $2 + x = 7$". When Charles is looking for the solution of '$2 + x = 7$', he is not looking for the number 5. And though the solution of '$2 + x = 7$' is the same as, e.g., of '$13 - x = 8$', it does not follow that Charles is solving the latter equation. Thus the meaning of the solution of '$2 + x = 7$' is only mentioned here. It is predicated of this very meaning that Charles is striving to find the object constructed by this meaning. When evaluating the truth-conditions of this sentence, we do not solve the equation, it is Charles' matter.
   For another example, consider the sentence "Charles believes that the President of Finland is elected directly by public but does not believe that the Head of state of Finland is elected by public". Suppose that 'the President of

[4] An exact definition is out of the scope of this paper. See, however, [7], in particular Chapters 2.6 and 2.7.

Finland' and 'the head of state of Finland' denote one and the same office.[5] Then if 'the President of Finland' and 'the head of state of Finland' were used extensionally or intensionally, the sentence would be a contradiction. Yet it is not. Thus both the expressions, or rather their meanings, occur here hyper-intensionally.

2. *Intensional context*: the sort of context in which a construction C is used to v-construct a function but not a particular value of the function, and C does not occur within another hyperintensional context.
   Example: "Sinus is a periodical function". The object of predication is here the entire function *sinus* rather than its particular value. Thus the meaning of 'sinus' is used intensionally here, and the analysis comes down to $[{}^0Periodical\ {}^0Sinus]$; $Periodical/(o(\tau\tau))$; $Sinus/(\tau\tau)$.
   For a non-mathematical example, consider the sentence "The President of Finland is elected for the period of six years". The object of predication is the office of the president, that is the function of type $\iota_{\tau\omega}$. The sentence predicates of this office (rather than of its contingent holder, if any) that it has the property of being eligible for the period of six years. The sentence is true even if the office is vacant; its truth is established by the Finnish constitution. Hence the meaning of 'the President of Finland' occurs here with *de dicto* supposition.
3. *Extensional context*: the sort of context in which a construction C of a function is used to construct a particular value of the function at a given argument, and C does not occur within another intensional or hyperintensional context.
   Example: "$\sin(\pi) = 0$" expresses the Composition $[[{}^0Sinus\ {}^0\pi] = {}^00]$, where ${}^0Sinus$ occurs extensionally; the Composition is used to construct the value of the sinus function at the argument π of type τ.
   For a non-mathematical example, consider the sentence "The President of Finland is the first female holder of the office". Now the property of being the first female holder of the office is not predicated of the entire office, but of its present holder, i.e. the value of type ι of the function of type $\iota_{\tau\omega}$. Hence the meaning of 'the President of Finland' occurs here extensionally, with *de re* supposition.

To avoid a misconception, we want to stress that the specification of particular contexts in which a construction occurs, does not involve a reference shift or even a meaning shift, as Frege proposed. Our analysis is *anti-contextualistic*. This is to say that the meaning of an unambiguous expression is the same in all the contexts. Indeed, why should the meaning of 'the President of Finland' as used in the sentence "The President of Finland is a female" be different from the meaning of this very same expression as used in "Charles believes that the President of Finland is a female"? What is dependent on the context in which one and the same meaning occurs is the way we logically manipulate the respective construction. This we are going to demonstrate in the next section.

[5] This is a simplification, because it is true that the President of Finland is the Head of state of Finland, but if Finland were for instance a kingdom, then the head of state would be a king or a queen. Thus the Head of state of Finland is a requisite of the presidential office. Yet this minor simplification is irrelevant here.

3.2 Substitution and Leibniz's Law

The extensional, intensional and hyperintensional occurrences of constructions were introduced in order to define valid inference rules for TIL in its capacity as a hyperintensional logic of partial functions. Once the difference between mentioning and using a construction, and the difference between using a construction either intensionally or extensionally, have both been defined, the specification of the rules is smooth sailing. They can be formulated as follows.

Improperness (non-existence). A construction C v-constructing an entity of a type α can be v-improper only due to a constituent D occurring *extensionally* in C. This is to say that improperness stems from using Composition, which is the procedure of applying a function f to an argument; either f has a *value gap*, or Composition C does not obtain an argument to operate on because some of the constituents of C are v-improper. In this way *partiality is strictly propagated upwards*.

Existence. If a construction C is v-proper then all its constituents D_i occurring extensionally are v-proper as well. In other words, the respective values of the functions constructed by these constituents *exist*.

Leibniz's law of substitution.

Extensional context. A collision-less replacement of v-congruent constructions D, D' in C is valid for extensionally occurring constituents; constructions D, D' are v-congruent if they v-construct one and the same entity.

Intensional context. A collision-less replacement of equivalent constructions D, D' in C is valid for all constituents of C; constructions D, D' are equivalent if they v-construct one and the same entity for all valuations v.

Hyper-intensional context. A collision-less replacement of procedurally isomorphic constructions D, D' in C is valid for all sub-constructions of C; constructions D, D' are procedurally isomorphic if they v-construct one and the same entity for all valuations v in the same procedural way. More precisely, procedural isomorphism is defined as the transitive closure of α- and η-equivalence. For instance, the constructions ${}^0Prime$, $\lambda x[{}^0Prime\,x]$, $\lambda y[{}^0Prime\,y]$, $\lambda z[\lambda x[{}^0Prime\,x]\,z]$, are procedurally isomorphic, while $\lambda x[[{}^0Card\lambda y[{}^0Divide\,y\,x]] = {}^02]$ is only equivalent to them; it does construct the set of primes, but does so in a non-isomorphic manner.

Moreover, for v-proper constituents occurring extensionally, the classical extensional rules of inference (as for instance those of a sequent calculus) are valid.

To illustrate the rules, we are now going to analyze some of the examples adduced in the previous section.

Example 1 (hyper-intensional context) "Charles is solving the equation $2 + x = 7$". As always, we begin with assigning types to the objects that receive mention in the analyzed sentence: $Charles/\iota$; $Solve/(o\iota\star_n)_{\tau\omega}$; $2, 7/\tau$; $x \rightarrow \tau$. When solving the equation, Charles wants to find out which set (here a singleton) is constructed by the Closure $\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]$. Thus he is related to the Closure itself rather than its product, a particular set. Otherwise the seeker would be immediately a finder and Charles' solving would be a pointless activity. The analysis comes down to

$$\lambda w \lambda t[^0 Solve_{wt}\, {}^0 Charles\, {}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]]. \tag{1}$$

Thus the following argument is *invalid*:

"Charles is solving the equation 2 + x = 7"
"The solution of 2 + x = 7 is equal to the solution of 13 - x = 8"
———————————————
"Charles is solving the equation 13 - x = 8"

This is revealed by the analysis:

$$\frac{\begin{array}{c}\lambda w \lambda t[^0 Solve_{wt}\, {}^0 Charles\, {}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]] \\ \lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7] = \lambda y[^0\!= [^0\!- {}^0\!13\, y]\, {}^0\!8]\end{array}}{\lambda w \lambda t[^0 Solve_{wt}\, {}^0 Charles\, {}^0[\lambda y[^0\!= [^0\!- {}^0\!13\, y]\, {}^0\!8]]]}$$

The construction $[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]$ occurs in the first premise *hyper-intensionally*. Thus a substitution *salva veritate* is valid here only for *procedurally isomorphic* constructions. Yet the second premise guarantees only the *equivalence* of the two Closures; they construct the same set of numbers, but in a non-isomorphic way.

On the other hand, the Trivialization ${}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]$ is a constituent used in (1) intensionally. It can never be v-improper, and the following argument is valid:

$$\frac{\begin{array}{c}\text{“Charles is solving the equation 2 + x = 7”} \\ \lambda w \lambda t[^0 Solve_{wt}\, {}^0 Charles\, {}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]]\end{array}}{\begin{array}{c}\text{“There is something Charles is solving”} \\ \lambda w \lambda t \exists c[^0 Solve_{wt}\, {}^0 Charles\; c]\end{array}}$$

The variable c is ranging over $\star_1$.

Proof. Let $Proper/(o\star_n)$ be the class of constructions that are not v-improper for any valuation v. Then in any world w at any time t the following steps are truth-preserving:

$[^0 Solve_{wt}\, {}^0 Charles\, {}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]]$	assumption
$[^0 Proper_{wt}\, {}^0[\lambda x[^0\!= [^0\!+ {}^0\!2\, x]\, {}^0\!7]]]$	the rule of improperness
$\exists c[^0 Solve_{wt}\, {}^0 Charles\; c]$	existential generalisation

Example 2 (intensional context) "Charles wants to be The President of Finland". Types. $Charles/\iota$; $Want\_to\_be/(o\iota_{\tau\omega})$; $President\_of/(\iota\iota)_{\tau\omega}$; $Finland/\iota$

$$\lambda w \lambda t[^0 Want\_to\_be_{wt}\, {}^0 Charles\; \lambda w \lambda t[^0 President\_of_{wt}\, {}^0 Finland]]. \tag{2}$$

The Closure $\lambda w \lambda t[^0 President\_of_{wt}\, {}^0 Finland]$ occurs intensionally, i.e. with *de dicto* supposition, because it is not used in (2) to v-construct the holder of the office (particular individual, if any). Thus the following argument is *invalid*:

"Charles wants to be the President of Finland"
"The President of Finland is the first female holder of the office"
———————————————
"Charles wants to be the first female holder of the office"

The analysis reveals the invalidity of the argument:

$$\frac{\begin{array}{c}\lambda w\lambda t[{}^0Want\_to\_be_{wt}\ {}^0Charles\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]] \\ \lambda w\lambda t[{}^0{=}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt} \\ [{}^0First_{wt}\ \lambda x\ [[{}^0Female_{wt}\ x] \wedge [{}^0{=}\ x\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}]]]]\end{array}}{\begin{array}{c}\lambda w\lambda t[{}^0Want\_to\_be_{wt}\ {}^0Charles \\ \lambda w\lambda t[{}^0First_{wt}\lambda x\ [[{}^0Female_{wt}\ x] \wedge [{}^0{=}\ x\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}]]]\end{array}}$$

Additional types: $x \to \iota$; $Female/(o\iota)_{\tau\omega}$; $First/(\iota(o\iota))_{\tau\omega}$: the function that selects from the set of individuals the only individual that is the first one at a given $\langle w, t\rangle$ of evaluation.

The argument is obviously invalid, because $\lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]$ occurs in the first premise with *supposition de dicto*, i.e. intensionally, while the second premise guarantees only v-congruence of this Closure with $\lambda w\lambda t[{}^0First_{wt}\lambda x\ [[{}^0Female_{wt}\ x] \wedge [{}^0{=}\ x\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}]]]$, i.e. a contingent co-occupancy of the two offices, rather than equivalence needed for a valid substitution.

Example 3 (extensional context) "The President of Finland is watching TV". The analysis of this sentence comes down to the Closure

$$\lambda w\lambda t[{}^0Watch_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0TV] \quad (3)$$

Additional types: $Watch/(o\iota\iota)_{\tau\omega}$; TV/ι. The meaning of 'the President of Finland' occurs with *de re* supposition in (3), i.e. extensionally. Thus we can apply the extensional rules that are also known as *two principles de re*. They are the *Principle of existential presupposition* and the *Substitutivity of co-referential expressions*. The following arguments are valid:

Argument 1:

"The President of Finland is watching TV"
———
"The President of Finland exists"

$$\frac{\lambda w\lambda t[{}^0Watch_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0TV]}{\lambda w\lambda t[{}^0Exist_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]]}$$

Argument 2:

"The President of Finland is watching TV"
"The President of Finland is Tarja Halonen"
———
"Tarja Halonen is watching TV"

$$\frac{\begin{array}{c}\lambda w\lambda t[{}^0Watch_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0TV] \\ \lambda w\lambda t[{}^0{=}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0Halonen]\end{array}}{\lambda w\lambda t[{}^0Watch_{wt}\ {}^0Halonen\ {}^0TV]}$$

Here are the proofs:

(*Ad Argument 1*) First, existence is here a property of an individual office rather than of some non-existing individual, whatever that would mean. Thus we have $Exist/(o\iota_{\tau\omega})_{\tau\omega}$. To prove the validity of the argument, we define $Exist/(o\iota_{\tau\omega})_{\tau\omega}$ as the property of an office of being occupied at a given world/time pair:

$${}^0Exist =_{of} \lambda w\lambda t\lambda c[{}^0\exists\lambda x[x = c_{wt}]], \text{ i.e., } [{}^0Exist_{wt}\ c] =_o [{}^0\exists\lambda x[x = c_{wt}]]$$

Types: $\exists/(o(o\iota))$: the class of non-empty classes of individuals; $c \rightarrow_v \iota_{\tau\omega}$; $x \rightarrow_v \iota$; $=_o /(ooo)$: the identity of truth-values; $=_{of} /(o(o\iota_{\tau\omega})_{\tau\omega}(o\iota_{\tau\omega})_{\tau\omega})$: the identity of individual-office properties.

Let $=_i /(o\iota\iota)$ be the identity of individuals, $Empty/(o(o\iota))$ the singleton containing the empty set of individuals and $Improper/(o\star_1)_{\tau\omega}$ the property of constructions of being v-improper in a given $\langle w, t\rangle$-pair, the other types as above. Then in any $\langle w, t\rangle$ the following proof steps are truth-preserving:

$[{}^0Watch_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0TV]$	assumption
$\neg[{}^0Improper_{wt}\ {}^0[[\lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]]_{wt}]]$	def. of Composition
$\neg[{}^0Empty\ \lambda x[x =_i [\lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]]_{wt}]]$	obvious from the prev. step
$[{}^0\exists\lambda x[x =_i [\lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]]_{wt}]]$	existential generalisation
$[{}^0Exist_{wt}\ [\lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]]]$	by def of Exist

(*Ad Argument 2*) substitution:

$[{}^0Watch_{wt}\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0TV]$	assumption
$[{}^0=\ \lambda w\lambda t[{}^0President\_of_{wt}\ {}^0Finland]_{wt}\ {}^0Halonen]$	assumption
$[{}^0Watch_{wt}\ {}^0Halonen\ {}^0TV]$	substitution of identicals

Note that if the President of Finland does not exist, then neither "The President of Finland is watching TV" nor the negated "The President of Finland is *not* watching TV" have any truth-value. This is due to compositionality and the extensional rule for existence. In both sentences 'the President of Finland' occurs extensionally. Thus if one of these sentences (either positive or negative one) is True, the President of Finland exists. As a consequence, if the president does not exist, then *neither* of these sentences is true. Hence, both sentences have *no* truth value, which is just the *Principle of exitential presupposition*: the existence of the president is not only entailed but also presupposed.[6]

4 Conclusion

Partiality, as we know all too well, is a complicating factor. Yet we are convinced that logic should assist in unearthing the objective structures underlying the expressions of a given language. In order to reflect 'gaps in reality' faithfully (i.e., to obtain a counterpart of Bolzano's *Gegenstandslosigkeit*), TIL adopts *properly partial functions* and *improper constructions*. In short, part of the task of a logician must be to adequately model the semantic features of (fragments of) a given language even at the cost of incurring technical complications. This explains why we are not going to join the game of playing fast and loose with existing logical symbols in order to define new *ad hoc* connectives and 'entailment relations' so as to either preserve or invalidate this or that commonly accepted law. Instead, we deploy methods that overcome these technical complications and are at the same time in full accordance with the principles of TIL as outlined in this paper.

---

[6] This is valid in case 'the President of Finland' is the topic of the sentence about which the property of watching TV (focus) is predicated. Yet there is another reading with 'TV' occurring as the topic and 'president of Finland' occurring in the focus clause. Then the existence of the president is only entailed. For details on Topic-Focus ambiguities see [6] and also [9].

Yet the theoretical specification of particular rules is only the first step. When making these features explicit we keep in mind an *automatic* deduction that will make use of these rules. To this end we currently develop a computational variant of TIL, the functional programming language $\mathcal{T}$*IL-Script* (see [2]). The direction of further research is clear. We are going to continue the development the $\mathcal{T}$*IL-Script* language in its full-fledged version equivalent to TIL calculus.

Acknowledgements. This research has been supported by the Grant Agency of the Czech Republic, projects GACR 401/07/0451, 'Semantization of Pragmatics' and 401/09/H007 'Logical Foundations of Semantics', and also by the internal grant agency of FEECS VSB-TU Ostrava - IGA 22/2009, 'Modeling, simulation and verification of software processes'.

References

1. Carnap, R.: *Meaning and Necessity*. Chicago: Chicago University Press, 1974.
2. Ciprich, N., Duží, M. and Košinár, M.: The $\mathcal{T}$*IL-Script* language. In *Information modelling and Knowledge Bases XX*, Y. Kiyoki, T. Tokuda, H. Jaakola, X. Chen, N. Yoshida (eds.), Amsterdam: IOS Press, pp. 166–179, 2009.
3. Cook, S. A.: The Complexity of Theorem-Proving Procedures. In: *STOC '71*: Proceedings of the 3rd annual ACM symposium on Theory of computing. New York, NY, USA, pp. 151–158, ACM Press, 1971.
4. Duží, M.: Semantic pre-processing of anaphoric references. In Proceedings *RASLAN 2007*, P. Sojka and A. Horák (eds.), Masaryk University Brno, pp. 43–56, 2007.
5. Duží, M.: Intensional logic and the irreducible contrast between de dicto and de re, *ProFil 5*, pp. 1–34. 2004, Retrievable at `http://profil.muni.cz/01_2004/duzi_de_dicto_de_re.pdf`
6. Duží, M.: Topic-Focus articulation from the semantic point of view. In *Computational Linguistics and Intelligent Text Processing*, ed. A. Gelbukh, pp. 220–232, Springer, LNCS 5449, 2009.
7. Duží, M., Jespersen B., Materna P.: *Procedural Semantics for Hyperintensional Logic; Foundations and Applications of Transparent Intensional Logic.* Series Logic, Epistemology and the Unity of Science. Springer, Berlin (forthcoming).
8. Frege G., Über Sinn und Bedeutung, *Zeitschrift für Philosophie und philosophische Kritik*, vol. 100, pp. 25–50, 1892.
9. Hajičová, E.: What we are talking about and what we are saying about it. In *Computational Linguistics and Intelligent Text Processing*, LNCS Springer Berlin/Heidelberg, vol. 4919, pp. 241–262, 2008.
10. Moschovakis, Y. N.: Sense and denotation as algorithm and value. In: J. Väänänen, J. Oikkonen (eds.), *Lecture Notes in Logic*, vol. 2, Berlin: Springer, pp. 210–249, 1994.
11. Tichý, P. *The Foundations of Frege's Logic*. Walter de Gruyter, Berlin-New York, 1988.
12. Tichý, P. (2004): *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press.