ISSN 2007-9737

Inferring Knowledge from Textual Data by Natural Deduction

Marie Duží, Marek Menšík

VSB-Technical University of Ostrava, Czech Republic

{marie.duzi, marek.mensik}@vsb.cz

Abstract. In this paper, we introduce the system for inferring implicit computable knowledge from textual data by natural deduction. Our background system is Transparent Intensional Logic (TIL) with its *procedural* semantics that assigns abstract procedures known as TIL *constructions* to terms of natural language as their context-invariant meanings. The input data for our method are produced by the so-called Normal Translation Algorithm (NTA). The algorithm processes natural-language texts and produces TIL constructions. In this way we have obtained a large corpus of TIL meaning procedures. These procedures are furthermore processed by our algorithms for type checking and context recognition, so that the rules of natural deduction for inferring computable knowledge can be afterwards applied.

Keywords. Natural deduction, inference rules, Transparent Intensional Logic, TIL, β-conversion.

1 Introduction

There are large amounts of knowledge in textual data. Yet in the era of information surfeit, it is difficult to obtain just those pieces of information that one needs. To this end it is necessary to build up systems of natural-language processing that derive not only explicit knowledge but also implicit, or rather inferable or computable knowledge from these text corpuses. In order to achieve such a goal, we have to combine linguistic, semantic and logical methods.

As Nevěřilová in [23] says "[…] in computational linguistics, making implicit information explicit forces syntactic, semantic and pragmatic modules to interact. Firstly, it is necessary to discover 'gaps' in the text, secondly, the correct missing entities have to be found, and finally, those entities can be filled in. For example, missing entities at the syntactic level are unexpressed (but obligatory), and such sentence constituents and the gaps are called ellipses. At the semantic level, such missing entities are the unfilled semantic roles [24]." Not only that, we also need to combine linguistic and *logical* methods. For instance, a logical method for computing the complete meaning of sentences with anaphoric references has been presented in [8]. The method is similar to the one applied in general by Hans Kamp's Discourse Representation Theory (DRT).[1] 'DRT' is an umbrella term for a collection of logical and computational linguistic methods developed for a dynamic interpretation of natural language, where each sentence is interpreted within a certain discourse, which is a sequence of sentences uttered by a group of speakers.

These methods are mostly based on first-order logics, and thus only terms referring to individuals (indefinite or definite noun phrases) can introduce so-called discourse referents, which are free variables that are updated when interpreting the discourse. However, Pavel Tichý's Transparent Intensional Logic (TIL, see [26]) makes it possible to substitute not only individuals, but entities of any type, like properties of individuals, propositions and hyperpropositions, relations-in-intension, and even constructions (i.e., meanings of antecedent expressions) for anaphoric variables. Moreover, the thoroughgoing typing of the universe of TIL makes it possible to determine the respective type-theoretically appropriate antecedent.

In this paper, we introduce a method of deriving inferable, or computational, knowledge from the explicit textual data by means of the

[1] See, for instance, [17, 18, 22].

system of natural deduction adjusted to our background TIL system.

In TIL we assign abstract *procedures* to terms of natural language as their context-invariant meanings. These procedures are rigorously defined as TIL constructions that produce lower-order objects as their products or in well-defined cases fail to produce an object by being improper. The input data for our method are produced by the so-called Normal Translation Algorithm (NTA) that processes text data and produces TIL constructions as their meanings. In this way we have obtained a large corpus of TIL meaning procedures.[2]

The rest of the paper is organised as follows. Section 2 introduces the fundamentals of TIL. In Section 3 we describe the three kinds of context in which a given natural-language term or rather its meaning can occur. Section 4 introduces the rules of natural deduction adjusted to TIL together with the principles of their correct application with respect to a given context and type of an entity to operate on. Section 5 deals with the rules of β-conversion validly applicable in the logic of partial functions such as TIL. Concluding remarks can be found in Section 6.

2 Basic Notion of TIL

The TIL system will be familiar to those who are acquainted with Montague system of IL.[3] The most important distinction between TIL and IL is that TIL comes with *procedural* rather than model set-theoretic semantics.[4] It means that we assign to terms of natural language *procedures* encoded by these terms as their meanings. These procedures are defined as TIL *constructions*. For instance, the sentence "the Pope is wise" encodes the procedure the evaluation of which in *any* possible world *w* and time *t* consists of these steps:

- Take the Papal office ('*Pope*).
- Extensionalise this office with respect to a world *w* and time *t* of evaluation to obtain the holder of this office, if any ('$Pope_{wt}$).
- If there is no holder (the office goes vacant), finish with a truth-value gap.
- Take the property of being wise ('*Wise*).
- Produce a truth-value **T** or **F** according as the holder of the papal office has the property of being wise ('$Wise_{wt}$) in the world *w* and time *t* of evaluation.

Definition 1 (*constructions*)

(i) *Variables* x, y, … are *constructions* that *construct* objects (elements of their respective ranges) dependently on a valuation v; they *v-construct*.

(ii) Where X is an object whatsoever (even a *construction*), 'X is the *construction Trivialization* that *constructs* X without any change of X.

(iii) Let X, $Y_1,\dots,Y_n$ be arbitrary *constructions*. Then *Composition* $[X\ Y_1\dots Y_n]$ is the following *construction*. For any v, the *Composition* $[X\ Y_1\dots Y_n]$ is *v-improper* if at least one of the *constructions* X, $Y_1,\dots,Y_n$ is v-improper, or if X does not *v-construct* a function that is defined at the n-tuple of objects *v-constructed* by $Y_1,\dots,Y_n$. If X does *v-construct* such a function, then $[X\ Y_1\dots Y_n]$ *v-constructs* the value of this function at the n-tuple.

(iv) (λ-) *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 and Y a *construction*. Then $[\lambda x_1\dots x_m\ Y]$ *v-constructs* the function f that takes any members $B_1, \dots, B_m$ of the respective ranges of the variables $x_1, \dots, x_m$ into the object (if any) that is $v(B_1/x_1,\dots,B_m/x_m)$-*constructed* by Y, where $v(B_1/x_1,\dots,B_m/x_m)$ is like v except for assigning B_1 to $x_1, \dots, B_m$ to x_m.

(v) Where X is an object whatsoever, ^1X is the *construction Single Execution* that *v*-constructs what X *v-constructs*. Thus, if X is a v-improper *construction* or not a *construction* as all, ^1X is v-improper.

[2] For details, see [19] or [20].

[3] For details on Montague's system see, for instance, [21].

[4] A critical survey and comparison of IL and TIL can be found in [11, §2.4].

ISSN 2007-9737

(vi) Where X is an object whatsoever, ${}^2X$ is the *construction Double Execution*. 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*, then ${}^2X$ is *v*-improper. Otherwise ${}^2X$ *v-constructs* what is *v-constructed* by the *construction v-constructed* by X.

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

Comments. Constituents of constructions are their sub-constructions, rather than the objects on which constructions operate. Thus, we need some simple constructions as 'suppliers' of or referents to the objects. Trivialization and variables are such simple suppliers. TIL standard notation for Trivialization of an object X is '${}^0X$'. Yet, due to easier typing, here we use the notation 'X.

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. The type of first-order objects includes all non-procedural objects. Therefore, it includes not only the standard objects of individuals, truth-values, sets, mappings, etc., but also functions defined on possible worlds (i.e., the intensions typical of possible-world semantics). The type of second-order objects includes constructions of first-order objects and functions with such constructions in their domain or range. The type of third-order objects includes constructions of first- and/or second-order objects and functions with such constructions in their domain or range. And so on, ad infinitum.

Definition 2 (*Ramified Hierarchy of Types*). Let B be a *base*, where a base is a collection of pair-wise disjoint, non-empty sets. Then:

$\mathbf{T_1}$ *(types of order 1)*:

i) Every member of B is an elementary *type of order 1 over B.*
ii) Let α, β_1, ..., β_m ($m > 0$) be types of order 1 over B. Then the collection (α β_1 ... β_m) of all m-ary partial mappings from $\beta_1 \times ... \times \beta_m$ 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).

$\mathbf{C}_n$ *(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 'X, ${}^1X$, ${}^2X$ are *constructions of order n over B.*

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

iv) Let x_1, ..., x_m, X ($m > 0$) be constructions of order n over B. Then [$\lambda x_1...x_m$ 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 $m > 0$ and α, β_1, ..., β_m are types of order $n+1$ over B, then (α β_1 ... β_m) (see T_1 ii)) is a *type of order n+1 over B*.

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

For the purposes of natural-language analysis, we are assuming the following base of ground types:

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

We model sets and relations by their characteristic functions. Thus, for instance, (οι) is the type of a set of individuals, while (οιι) is the type of a relation-in-extension between individuals. Empirical expressions denote *empirical conditions* that may or may not be satisfied at the particular world/time pair of evaluation.

We model these empirical conditions as possible-world-semantic *(PWS-) intensions.* PWS-intensions are entities of type $(\beta\omega)$: mappings from possible worlds to an arbitrary type β. The type β is frequently the type of the *chronology* of α-objects, i.e., a mapping of type $(\alpha\tau)$. Thus α-intensions are frequently functions of type $((\alpha\tau)\omega)$, abbreviated as '$\alpha_{\tau\omega}$'. *Extensional entities* are entities of a type α where $\alpha \neq (\beta\omega)$ for any type β. Where w ranges over ω and t over τ, the following logical form essentially characterizes the logical syntax of empirical language:

$\lambda w\lambda t$ […w….t…].

Examples of frequently used PWS intensions are: *propositions* of type $o_{\tau\omega}$, *properties of individuals* of type $(o\iota)_{\tau\omega}$, binary *relations-in-intension* between individuals of type $(o\iota\iota)_{\tau\omega}$, *individual offices* (or *roles*) of type $\iota_{\tau\omega}$, *intensional attitudes*/$(o\iota\alpha)_{\tau\omega}$; *hyperintensional attitudes*/$(o\iota{*}_n)_{\tau\omega}$.

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* $\forall^\alpha$, $\exists^\alpha$ are type-theoretically polymorphic functions of type $(o(o\alpha))$, for an arbitrary type α, defined as follows.

Definition 3 (*quantifiers*). The *universal quantifier* $\forall^\alpha$ is a polymorphic total function that associates a class A of α-elements with **T** if A contains all elements of the type α, otherwise with **F**. The *existential quantifier* $\exists^\alpha$ is a polymorphic total 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. The outermost brackets of the Closure will be omitted whenever no confusion arises. Furthermore, 'X/α' means that an object X is (a member) of type α. '$X \rightarrow_v \alpha$' means that X is typed to v-construct an object of type α, if any. We write '$X \rightarrow \alpha$' if a valuation v does not matter. Throughout, it holds that the variables $w \rightarrow \omega$ and $t \rightarrow \tau$. If $C \rightarrow \alpha_{\tau\omega}$ 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}'. For instance, if *Student*/$(o\iota)_{\tau\omega}$ is the property of being a student, the procedure of extensionalizing this property to obtain its population in a given world w and time t is the Composition [['*Student* w] t], or '$Student_{wt}$, for short.

Whenever no confusion arises, we use traditional infix notation without Trivialisation for truth-functions and the identity relation, to make the terms denoting constructions easier to read. Thus, for instance, instead of $\lambda w\lambda t$ ['$\wedge$ ['= ['+ '2 '5] '7] [[['*Know* w] t] '*Tilman it*]] we usually write $\lambda w\lambda t$ [[['+ '2 '5] = '7] $\wedge$ ['$Know_{wt}$ '*Tilman it*]].

3 Three Kinds of Context

TIL operates with a fundamental dichotomy between procedures, i.e. constructions, and their products, i.e. functions.[5] This dichotomy corresponds to two basic ways in which a construction can occur within another construction, namely *displayed,* or *executed*. If the construction is displayed then the construction itself is an object of predication; we say that it occurs *hyperintensionally*. If the construction is executed, then it is a *constituent* of another construction, and an additional distinction can be found at this level.

The constituent presenting a function may occur either *intensionally* (de dicto) or *extensionally* (de re). If intensionally, then the whole function is an object of predication; if extensionally, then a functional value is an object of predication. Both distinctions are instrumental in selecting a construction or else what the meaning construction produces, which is either a function or a functional value, as the functional argument of a function v-constructed within a superconstruction.

For an example of the contrast between displayed and executed procedures, consider the mathematical equation $sin(x) = 0$.

If Tilman is solving this equation then Tilman is related to the very meaning of "$sin(x) = 0$" rather

[5] Formally speaking, extensional entities like individuals, numbers and truth-values are extreme forms of 0-ary functions, whereas sets are identified with their characteristic functions.

than the set of multiples of the number π. Tilman wants to execute the procedure expressed by "*sin*(*x*) = 0" in order to find out which set of real numbers matches the equation. Hence in "Tilman is solving the equation *sin*(*x*) = 0" the meaning of "*sin*(*x*) = 0", i.e. the Closure λx [['*Sin* *x*] = '0] is *displayed*. This very Closure is an object of predication here.

On the other hand, if we claim that the solution of the equation *sin*(*x*) = 0 is the set {…, –2π, –π, 0, π, 2π, …} the meaning of "*sin*(*x*) = 0" is executed to produce this set. Yet the constituent meaning of "*sin*(*x*) = 0" occurs intensionally in the meaning of "The solution of the equation *sin*(*x*) = 0 is the set {…, –2π, –π, 0, π, 2π, …}". The whole set (a characteristic function) is the object of predication. An example of an extensional occurrence of the meaning of '*sin*' would be provided by the simple sentence "*sin*(π) = 0". Here the value of the function *sine* at the argument π is the object of which it is predicated that it is equal to zero.

The same differentiation applies also to the meanings of empirical terms. For an example of the contrast between intensional and extensional occurrence, consider predication. Predication, in TIL, is an instance of functional application: a characteristic function is applied to a suitable argument in order to obtain a truth-value, according as the argument is an element of the set. In the case of predication of empirical properties, the relevant set is obtained by extensionalizing the property.

In the context "The site of Troy is located in Asia Minor" we want the functional value of the office *the site of Troy* to occur either as an argument for the set of entities located in Asia Minor or as an argument for the binary relation (-in-intension) *located in* whose second argument is Asia Minor. Hence the meaning of 'the site of Troy' occurs extensionally here. On the other hand, when Schliemann sought the site of Troy, he was not related to any value of the denoted function. Rather he was related to the whole office aiming to determine its value, if any. As a result, the meaning of 'the site of Troy' occurs intensionally in "Schliemann sought the site of Troy".

Similarly, the meaning of the term 'the temperature in Prague' occurs extensionally in "The temperature in Prague is 13^0C", while in "The temperature in Prague is rising" *the same meaning* of this definite description occurs intensionally. To be rising is a property of the whole function rather than of any value. Finally, in "*a* knows (hyperintensionally) that the temperature in Prague is 13^0C" the same meaning occurs hyperintensionally. When knowing something hyperintensionally, we are related to the very meaning of the embedded clause rather than the produced function (a possible-world proposition in this case).

The two distinctions, between displayed and executed and intensional/extensional, allow us to distinguish between three sorts of *context*. Though the basic ideas of distinguishing these contexts are simple, rigorous definition is rather complicated. Hence, here is just a brief summary of them:[6]

– *hyperintensional context*: one or more constructions occur *displayed* (though a construction at least one order higher need to be executed in order to produce the displayed constructions)
– *intensional context*: one or more constructions are *executed* in order to produce one or more functions (moreover, the executed constructions do not occur within another hyperintensional context)
– *extensional context*: one or more constructions are *executed* in order to produce one or more particular values of one or more functions at one or more given arguments (moreover, the executed constructions do not occur within another intensional or hyperintensional context).

The basic idea underlying the above trifurcation is that the same set of logical rules apply to all three kinds of context, but they operate on different complements: *constructions*, *functions*, and functional *values*, respectively. Thus, in TIL we have no oblique contexts in which the fundamental logical rules were not valid. The rules are all valid for constituent constructions; only that to be validly applied, the rules must respect the type of an entity to operate on. Furthermore, whenever we operate *inside* a non-

[6] The rigorous definition can be found in [11, §2.6].

ISSN 2007-9737

extensional context, we apply our *substitution* method in order not to draw a construction occurring in a lower context into a higher one, which would be incorrect.

4 Natural Deduction in TIL

The rules we introduce here follow the general pattern of the rules of natural deduction that are introduced in the sequent form. We start with the rules dealing with truth-functions, because these rules are applicable in extensional contexts. When applying the rules for quantifiers, we have to take into account a context in which a given construction occurs and the type of an entity that is quantified over. Furthermore, when dealing with empirical propositions, the first steps of each proof are λ-elimination (λ-E) and the last ones λ-introduction (λ-I) of the left-most λ*w*λ*t*, because the whole proof sequence must be truth-preserving in any world *w* and time *t*.

Here is a simple example:

John is sick or went to the theatre.
If he is sick then he calls a doctor.
But he doesn't call a doctor.

John went to the theatre.

To analyse the premises and the conclusion, we apply our method of analysis.[7] As always, we start with type-theoretical analysis of the objects that receive mention here:

Types. John/ι; *Sick*/$(\omicron\iota)_{\tau\omega}$; *Went*/$(\omicron\iota\iota)_{\tau\omega}$; *Theatre*/ι; *Call*$(\omicron\iota\iota)_{\tau\omega}$; *Doctor*/ι.

Synthesis.[8]

*w*λ*t* [[‘$Sick_{wt}$ ‘*John*] ∨ [‘$Went_{wt}$ ‘John ‘*Theatre*]]
λ*w*λ*t* [[‘$Sick_{wt}$ ‘*John*] ⊃ [‘$Call_{wt}$ ‘John ‘*Doctor*]]
λ*w*λ*t* [¬[‘$Call_{wt}$ ‘John ‘*Doctor*]]

λ*w*λ*t* [‘$Went_{wt}$ ‘John ‘*Theatre*]]

The last step of our method is checking whether a given construction is composed in a type-theoretically coherent way. For the sake of simplicity, here we demonstrate the type-checking only for the Closure λ*w*λ*t* [‘$Sick_{wt}$ ‘*John*]:

– [‘*Sick w*] → ((οι)τ)
– [[‘Sick w] t] □ (οι)
– ‘John □ ι
– [[[‘Sick w] t] ‘John] □ ο
– □t [[[‘Sick w] t] ‘John] □ (οτ)
– □w□*t* [[[‘*Sick w*] *t*] ‘*John*] → ((οτ)ω)

The resulting type is the type of a proposition, ((οτ)ω), or $\omicron_{\tau\omega}$ for short, as it should be.
The proof of our argument is as follows:

1)	λ*w*λ*t* [[‘$Sick_{wt}$ ‘*John*] ∨ [‘$Went_{wt}$ ‘John ‘*Theatre*]]	∅
2)	λ*w*λ*t* [[‘$Sick_{wt}$ ‘*John*] ⊃ [‘$Call_{wt}$ ‘John ‘*Doctor*]]	∅
3)	λ*w*λ*t* [¬[‘$Call_{wt}$ ‘John ‘*Doctor*]]	∅
4)	[[‘$Sick_{wt}$ ‘*John*] ∨ [‘$Went_{wt}$ ‘John ‘*Theatre*]]	1, λ-E
5)	[[‘$Sick_{wt}$ ‘*John*] ⊃ [‘$Call_{wt}$ ‘John ‘*Doctor*]]	2, λ-E
6)	¬[‘$Call_{wt}$ ‘John ‘*Doctor*]	3, λ-E
7)	¬[‘$Sick_{wt}$ ‘*John*]	5,6 MTT
8)	[‘$Went_{wt}$ ‘John ‘*Theatre*]	4,7 DS
9)	λ*w*λ*t* [‘$Went_{wt}$ ‘John ‘*Theatre*]	8, λ-I

In what follows, we usually omit the initial and final rules for elimination and introduction of λ*w*λ*t.*

Firstly, we introduce the rules of propositional logic dealing with truth-functions, adjusted to TIL. Though in our example we apply the rules in their linear form, to demonstrate the proofs from assumptions, we present the rules in the sequent form.

4.1 The Rules for Truth-Functions

Let *A*, *B*, *C* → ο. *X* and *Y* represent lists of constructions (assumptions):

1. Rule of Assumption:

$$A \vdash A$$

2. Conjunction Introduction (∧-I):

$$\frac{X \vdash A \qquad Y \vdash B}{X, Y \vdash A \wedge B}$$

3. Conjunction Elimination (∧-E):

$X \vdash A \wedge B$ $X \vdash A \wedge B$

---

[7] For details see [11, pp. 77-79].

[8] For the sake of simplicity, we ignore the past tense here and analyse ‘the theatre’ as denoting an individual, which is a simplification, yet irrelevant for our exposition.

$X \vdash A$ $\quad\quad$ $X \vdash B$

4. Modus Ponendo Ponens (MPP):

$$\frac{X \vdash A \supset B \qquad Y \vdash A}{X, Y \vdash B}$$

5. Conditional Proof (CP):

$$\frac{X, A \vdash B}{X \vdash A \supset B}$$

6. Disjunction Introduction ($\vee$-I):

$$\frac{X \vdash A}{X \vdash A \vee B} \qquad \frac{X \vdash A}{X \vdash B \vee A}$$

7. Disjunction Elimination ($\vee$-E):

$$\frac{X \vdash A \vee B \qquad Y, A \vdash C \qquad Z, B \vdash C}{X, Y, Z \vdash C}$$

8. Double negation Introduction (DNI):

$$\frac{X \vdash A}{X \vdash \neg\neg A}$$

9. Double negation Elimination (DNE):

$$\frac{X \vdash \neg\neg A}{X \vdash A}$$

10. Modus Tollendo Tollens (MTT):

$$\frac{X \vdash A \supset B \qquad Y \vdash \neg B}{X, Y \vdash \neg A}$$

11. Disjunctive Syllogism (DS):

$$\frac{X \vdash A \vee B \qquad Y \vdash \neg A}{X, Y \vdash B} \qquad \frac{X \vdash A \vee B \qquad Y \vdash \neg B}{X, Y \vdash A}$$

12. Reductio Ad Absurdum (RAA):

$$\frac{X, A \vdash B \wedge \neg B}{X \vdash \neg A}$$

Similar to propositional logic, predicate logic has its natural deduction proof system. Needless to say, the rules dealing with truth-functions are the same as those in propositional logic. Additionally, there are rules for quantifiers (general $\forall$ and existential $\exists$). Again, these additional rules are of two kinds, namely introduction and elimination rules.

However, we are building the deduction system for TIL, and since TIL is a hyperintensional λ-calculus of partial functions, there are additional complications. First, quantifiers in TIL (see Def. 3) are not special symbols; rather, they are *functions* applicable to classes of objects. Furthermore, the rules dealing with quantifiers, to be validly applied, must respect the context in which a given construction occurs and the type of an entity to be quantified over. Another serious problem that we have to deal with is the problem of *partiality.* TIL is a logic of *partial functions* and partiality, as we all know too well, brings about technical complication. This concerns in particular the existential quantifiers, as we are going to demonstrate below.

4.2 Rules for General Quantifiers

4.2.1 General Quantifier Elimination ($\forall$-E)

The rule ($\forall$-E) for elimination of a general quantifier in classical predicate logic is non-problematic:

$$\frac{X \vdash \forall x\, \phi}{X \vdash \phi\, [t/x]}$$

ISSN 2007-9737

where ϕ is a formula and the term *t* is substitutable for the variable *x* in ϕ.

In an ordinary vernacular we would say "what holds for everything holds also for something", which is no doubt true. Is it? What about if there is no 'something'? In other words, if the term *t* is not referring to anything? Sure, in classical predicate logic it is not possible, because it is a logic of *total* functions. Yet TIL is a logic of *partial* functions and we have to take this issue into account. We must work with partial functions when processing natural language, because in natural language there are non-referring terms like 'the King of France'. And the method of domain-restriction applied in mathematics or computer science is not applicable here, because we would face the problem of a non-recursive domain explosion. We cannot recursively define in which worlds *w* and times *t* the King of France exists. It is a matter of *empirical* investigation. To adduce a simple example, consider this argument:

All politicians are wise.
The King of Germany is a politician.

The King of Germany is wise.

If both the premises were true, the conclusion would have to be true as well. Hence, the argument is valid[9]. Still the argument is not sound. Even if the first premise were true, the second premise denotes a proposition that is neither true nor false, because there is no King of Germany. Hence, there is no individual at hand to ascribe the property of being or not being a politician.

However, we can prove that the argument is valid. Here is how. As always, first typing[10]:

$\forall/(o(o\iota))$; *Politician*$/(o\iota)_{\tau\omega}$; *Wise*$/(o\iota)_{\tau\omega}$; Germany$/\iota$; King-*of*$/(\iota\iota)_{\tau\omega}$; $[{}^0$*King-of*$_{wt}$ ^{0}*Germany*$] \to \iota$; $x,y \to \iota$.

1) $[{}^0\forall\lambda x\,[[{}^0Politician_{wt}\,x] \supset [{}^0Wise_{wt}\,x]]]$ $\varnothing$
2) $[\lambda x\,[[{}^0Politician_{wt}\,x] \supset [{}^0Wise_{wt}\,x]]\,y]$ 1, $\forall$-E
3) $[[{}^0Politician_{wt}\,y] \supset [{}^0Wise_{wt}\,y]]$ 2, β-r
4) $[{}^0Politician_{wt}\,[{}^0King\text{-}of_{wt}\,{}^0Germany]]$ $\varnothing$
5) $[[{}^0Politician_{wt}\,[{}^0King\text{-}of_{wt}\,{}^0Germany]] \supset$ $[{}^0Wise_{wt}\,[{}^0King\text{-}of_{wt}\,{}^0Germany]]]$ 3, $[{}^0King\text{-}of_{wt}\,{}^0Germany]/y$
6) $[{}^0Wise_{wt}\,[{}^0King\text{-}of_{wt}\,{}^0Germany]]$ 4, 5, MPP

Comment. The substitution of the Composition $[{}^0King\text{-}of_{wt}\,{}^0Germany]$ for *y* in the step (5) is truth-preserving; *provided* the Composition (4) *v*-constructs **T**, which is assumed, it is not *v*-improper (see Def. 1, iii). On this assumption, the Composition $[{}^0King\text{-}of_{wt}\,{}^0Germany]$ is not *v*-improper either.

Now you may ask. Is this new piece of information that we obtained of any value? Of course, it is not. In order it be valuable we must obtain another piece of knowledge, namely whether the King of Germany exists. In other words, we must find out whether such an argument is also sound. To this end we must empirically explore the state of affairs in Germany to find out whether the King of Germany exists.

Another issue we encounter here is this. Though the argument is valid, the corresponding conditional sentence:

"If all politicians are wise and
the King of Germany is a politician,
then the King of Germany is wise"

---

[9] The argument is valid, *provided* the second premise is read extensionally, *de re.* On its *de re* reading, the property of being a politician is ascribed to the holder of the office of King of Germany, if any. If there is no such holder, the sentence denotes a proposition with truth-value gap. However, the sentence "The King of Germany is a politician" is ambiguous. There is another reading, namely intensional (*de dicto*). On this reading the sentence conveys a piece of information that the property of being a politician is a *requisite* of the royal office, where *Requisite*$/(o(o\iota)_{\tau\omega}\iota_{\tau\omega})$. Necessarily, i.e. in all *w* and *t*, if an individual *a* happens to be the King of Germany then *a* is a politician. The requisite relation obtains between intensions (here a property and an office) necessarily and independently of a contingent occupancy of the office. On this reading the argument is *not* valid, because then the second premise is necessarily true, i.e. true even in those $\langle w,t\rangle$-pairs where there is no King of Germany, but the conclusion has a truth-value gap in such $\langle w,t\rangle$-pairs.

[10] For the sake of simplicity, here we apply the 'unrestricted' general quantifier ∀. The literal analysis of the sentence should, however, be composed by applying the restricted quantifier *All*$/((o(o\iota))(o\iota))$ that is the function that associates a given set *S* of individuals with the set of all supersets of *S.* The literal analysis would then be $\square w \square t\,[[{}^0All\,{}^0Politian_{wt}]\,{}^0Wise_{wt}]$.

ISSN 2007-9737

is not analytically true. In other words, the semantic variant of the theorem of deduction does not hold here.

Due to the non-existence of the King of Germany the sentence does not denote a proposition true in all worlds *w* and times *t*. Rather, it denotes the proposition with a *truth-value gap* in the actual world and time of evaluation. Yet this problem does not have to bother us too much, because analytically true sentences convey no empirical information.[11] Our goal is deriving inferable knowledge from textual data, i.e., deriving consequences of assumptions provided by these data. When doing so, we assume that propositions encoded by the assumptions are true.

Back to the rule of general quantifier elimination. As the above example illustrates, the rule must be adjusted for the TIL system. Here is how.

Let $x,y \rightarrow \alpha$, $B(x) \rightarrow o$: the variable x is free in B; $[\lambda x\ B] \rightarrow (o\alpha)$, $\forall/(o(o\alpha))$, $C \rightarrow \alpha$. Then general quantifier elimination in full detail consists of these steps:

$[{}^{\prime}\forall\lambda x\ B]$	$\varnothing$
$[[\lambda x\ B]\ y]$	$\forall$-E
$B(y)$	β–r (see below)
$B(C/y)$	*substitution*

where $B(C/y)$ arises from B by a collision-less, valid substitution of the construction C for all occurrences of the variable y in B.

For the sake of simplicity, we will write this rule in the shortened form:

$$\frac{X \vdash [{}^{\prime}\forall\lambda x\ B]}{X \vdash B(C/x)} \qquad (\forall\text{-E})$$

4.2.2 General Quantifier Introduction (∀-I)

Dual to the general quantifier elimination is the rule for *general quantifier introduction*, ∀-I. This rule is not as simple as the rule ∀-E. Since we can think of a general quantifier as a generalization of conjunction, recall the rule ∧-I:

[11] Yet, such sentences convey *analytical* information. For the difference between analytical and empirical information, see [2].

$$\frac{X \vdash A \qquad Y \vdash B}{X, Y \vdash A \wedge B}$$

This suggests that to introduce a quantifier ∀, i.e., to apply this function to the set produced by $\lambda x\ B$ to obtain the Composition $[{}^{\prime}\forall\lambda x\ B]$, we must prove that the condition specified by the construction B is valid for all possible values of the variable x, i.e. for all the elements of the range of x. This seems impossible. Yet, consider the proofs in mathematics. For instance, suppose we want to prove the theorem:

"Every even natural number is the sum of
two odd natural numbers
whose difference is at most 2."

Phrasing the proof informally, it comes down to this. Let *n* be *any* even natural number. Then *n* is of the form $2k$, for some $k \geq 1$.

If *k* is odd, then we can write $n = k + k$, and the two *k*'s satisfy the theorem.

If *k* is even, we can write $n = (k\text{-}1) + (k\text{+}1)$, and the numbers $k\text{-}1$ and $k\text{+}1$ satisfy the theorem.

What is important here is the fact that by using the variable *n* we consider an *arbitrary* even natural number, and show that this number is the sum of two odd natural numbers whose difference is at most 2. That allows us to conclude that the condition specified by the theorem holds for every natural number *n*, since there is *nothing special* about *n.* It does not appear in the statement of the theorem or anywhere else outside the proof.

Hence, to prove a construction of the form $[{}^{\prime}\forall\lambda x\ B]$, we can prove B with some arbitrary but "fresh" free variable $y \rightarrow \alpha$ substituted for $x \rightarrow \alpha$. That is, we want to prove the construction $B(y/x)$. By "fresh" we mean that the variable has never been used before in the proof. Furthermore, it will not be used once $B(y/x)$ has been proven. It is "local" to this part of the proof. The rule ∀-I thus comes down in this form:

$$\frac{X \vdash B(y/x)}{X \vdash [{}^{\prime}\forall\lambda x\ B]} \qquad (\forall\text{-I})$$

In an ordinary vernacular, we usually do not prove mathematical theorems. Yet, we can demonstrate similar principles of a valid application of the generalization rule by proving an analytically true sentence.

Mathematical sentences are analytical in this sense. When evaluating their truth-values, possible worlds and times do not matter as points of evaluation. Among the sentences involving empirical expressions there are also analytically true sentences. They denote the proposition *TRUE* that takes the truth-value value **T** in all possible worlds and times. Consider sentences like "No bachelor is married", "All whales are mammals" that contain the empirical predicates 'is a bachelor', 'is married', 'is a whale', 'is a mammal'. At no world/time are the properties *being a bachelor* and *being married* co-instantiated by the same individual. And in every world/time is the property of being a mammal a requisite of the property of being a whale. This means that necessarily (in every world/time pair) if an individual *a* happens to be a whale then *a* is a mammal.

Now consider, e.g., the first sentence. Its literal analysis comes down to the Closure

$$\lambda w \lambda t\ [[{}^0No\ {}^0Bachelor_{wt}]\ {}^0Married_{wt}]$$

Types. $No/((o(o\iota))(o\iota))$: the restricted quantifier, i.e. the function that associates a given set *S* of individuals with the set of all those sets of individuals that are disjoint with *S*; *Bachelor*, $Married/(o\iota)_{\tau\omega}$.

This analysis does not reveal that the proposition produced by the Closure takes the value **T** at all $\langle w, t\rangle$-pairs. The analysis itself does not make it possible to *prove* it. We need to refine the analysis. To this end we make use of the fact that the property of being a bachelor is *defined* as the property of being an unmarried man, so the sentence is analytically, *ex definitione*, true. As soon as we replace the simple predicate 'is a bachelor' by this definition, the truth of the sentence is obvious: "No unmarried man is married". Still, to prove it we need a refined analysis that makes use of the definition of the *restricted quantifier No*. It is a function that operates on sets of individuals and returns **T** iff the sets are disjoint. By using the variables *m*, *n* $\rightarrow (o\iota)$, $x \rightarrow \iota$, we obtain the defining equivalences

$${}^0No = \lambda m\ \lambda n\ \neg[{}^0\exists\lambda x\ [[m\ x] \wedge [n\ x]]],$$

$$[[{}^0No\ m]\ n] = \neg[{}^0\exists\lambda x\ [[m\ x] \wedge [n\ x]]].$$

The property of being a bachelor can be defined by composing the constructions of the negation and of the properties *Married* and *Man* as follows:

$${}^0Bachelor = \lambda w\lambda t\ \lambda x\ [\neg[{}^0Married_{wt}\ x] \wedge [{}^0Man_{wt}\ x]].$$

Now by substituting the respective definitions (and applying β-reductions) we obtain:

$[[{}^0No\ {}^0Bachelor_{wt}]\ {}^0Married_{wt}] =$

$\neg[{}^0\exists\lambda x\ [[{}^0Bachelor_{wt}\ x] \wedge [{}^0Married_{wt}\ x]]] =$

$\neg[{}^0\exists\lambda x\ [\neg[{}^0Married_{wt}\ x] \wedge [{}^0Man_{wt}\ x] \wedge [{}^0Married_{wt}\ x]]]$

Since this last construction obviously and provably *v*-constructs **T** for *any* valuation *v* of the variables *w* and *t*, we can generalize to

$$[{}^0\forall\lambda w\ {}^0\forall\lambda t\ \neg[{}^0\exists\lambda x\ [\neg[{}^0Married_{wt}\ x] \wedge [{}^0Man_{wt}\ x] \wedge [{}^0Married_{wt}\ x]]]].$$

We have proven that the sentence "No bachelor is married" denotes the proposition *TRUE.*

When deriving new pieces of information from text data we make use of corpuses like Wordnet, where we can find such definitions of properties and their requisites as above. In our example, the property of being unmarried is a requisite of the property of being a bachelor. Necessarily, if an individual happens to be a bachelor then it is not married. Hence, having a piece of knowledge that

Tom is a bachelor

together with the definition of the property of being a bachelor obtained from, e.g., Wordnet, we can easily infer that

Tom is not married.

It should be obvious now how to do it. We are to prove the argument:

$$\frac{\lambda w\lambda t\ [{}^0Bachelor_{wt}\ {}^0Tom]}{\lambda w\lambda t\ \neg[{}^0Married_{wt}\ {}^0Tom]}$$

Omitting the steps of λ-E and λ-I, we have:

1. $[{}^0Bachelor_{wt}\ {}^0Tom]$ $\varnothing$

ISSN 2007-9737

2. $[\lambda x\, [\neg[{}^{\backprime}Married_{wt}\, x] \wedge [{}^{\backprime}Man_{wt}\, x]]\ {}^{\backprime}Tom]$ 1, subst
3. $[\neg[{}^{\backprime}Married_{wt}\, {}^{\backprime}Tom] \wedge [{}^{\backprime}Man_{wt}\ {}^{\backprime}Tom]]$ 2, β-r
4. $\neg[{}^{\backprime}Married_{wt}\, {}^{\backprime}Tom]$ 3, ∧-E

In step 3, we applied the rule of β-reduction, the definition of which is coming below. Yet to complete this section we are going to introduce the rules dealing with existential quantifiers.

4.3 Rules for Existential Quantifiers

In classical logic the existential quantifier ∃ is dual to the general quantifier ∀. Thus, it might seem that whereas the rule ∃-I for ∃ introduction is unproblematic, the difficulties would arise with the rule ∃-E for elimination of the existential quantifier. This is true in logic of total functions. However, as explained above, TIL is the logic of *partial* functions and we must be careful also with the ∃-I rule not to derive that there is a value of a function at an argument when there is none.

As in classical logic, the rules for existential quantifier function, $\exists/(o(o\alpha))$, are parallel to those for disjunction (∨).

Let $x,y \to \alpha$, $B \to o$, $[\lambda x\ B] \to (o\alpha)$, $\exists/(o(o\alpha))$, $[{}^{\backprime}\exists\lambda x\ B\,] \to o$, $C \to o$.

4.3.1 Existential Quantifier Elimination *(∃-E)*

$$\frac{X \vdash [{}^{\backprime}\exists\lambda x\ B] \qquad Y, B(y) \vdash C}{X, Y \vdash C} \qquad (\exists\text{-E})$$

where the variable *y* does not occur free in *C.*

Comment. Recall the rule for eliminating disjunction; it is rather complicated:

$$\frac{X \vdash A \vee B \qquad Y, A \vdash C \qquad Z, B \vdash C}{X,Y,Z \vdash C}$$

Roughly, it says this; consider both the disjuncts *A* and *B*, and if you manage to prove another construction *C* taking first *A* as an assumption and then *B*, you proved *C* from *A* ∨ *B.*

The rule is well justified. Proving *C* from *A* is equivalent to proving $A \supset C$, and proving *C* from *B* is equivalent to proving $B \supset C$. Hence, we have proved $(A \supset C) \wedge (B \supset C)$, which is equivalent to $(A \vee B) \supset C$. By modus ponendo ponens, we proved *C.*

This suggests that to eliminate an existential quantification $[{}^{\backprime}\exists\lambda x\ B]$ and derive another construction *C*, we should be able to conclude *C* starting from *B* with *any* ‘value’ substituted for *x* in *B*. We do this by substituting a ‘fresh’ free variable *y* that does not occur free in *C* (or anywhere outside the proof sequence).

Example.

There are smart politicians.

There is an individual *x* that is smart.

Proof.

1. $\lambda w\lambda t\ [{}^{\backprime}\exists\lambda x\ [[{}^{\backprime}Smart_{wt}\ x] \wedge [{}^{\backprime}Polician_{wt}\ x]]]$ ∅
2. $[{}^{\backprime}\exists\lambda x\ [[{}^{\backprime}Smart_{wt}\ x] \wedge [{}^{\backprime}Polician_{wt}\ x]]]$ 1, λ-E
3. $[[{}^{\backprime}Smart_{wt}\ y] \wedge [{}^{\backprime}Polician_{wt}\ y]]$ 2, ∃-E
4. $[{}^{\backprime}Smart_{wt}\ y]$ 3, ∧-E
5. $[{}^{\backprime}\exists\lambda x\ [{}^{\backprime}Smart_{wt}\ x]]$ 4, ∃-I
6. $\lambda w\lambda t\ [{}^{\backprime}\exists\lambda x\ [{}^{\backprime}Smart_{wt}\ x]]$ 5, λ-I

Notes. In the analysis (step 1) we make use of the fact that ‘smart’ denotes here an *intersective modifier* of a property, which is a function that takes a property as an argument returning another property as its value, i.e. an entity of type $((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$. The modifier *Smart* is applied to the property of being a politician here. For intersective modifiers the rules of left and right subsectivity hold. In other words, if somebody is a smart politician then he/she is smart and a politician. For details, see for instance [5] and [15].

In the step 5 we applied the ∃-I rule coming below. In the logic of *partial* functions this rule is not as simple as it might seem. To illustrate, consider this argument.

Tilman is seeking an abominable snowman.
Tilman is seeking something.

The argument is valid, for sure. Yet the issue is *what type* of an entity is that *something.* It cannot be an individual, for then we would prove the existence of yeti, which would turn logic to magic. Tilman is related to the *property* of being an abominable snowman the instances of which the seeker wants to find. Hence, the relation of s*eeking* establishes here an intensional context rather than an extensional one. The analysis of the premise and conclusion makes it explicit:

$$\frac{\lambda w\lambda t\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ [{}^0\mathit{Abominable}\ {}^0\mathrm{Snowman}]]}{\lambda w\lambda t\ [{}^0\exists\lambda p\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ p]]}$$

Types. $\mathit{Seek}/(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$: the relation-in-intension of an individual to a property the instances of which the seeker wants to find;[12] Tilman/ι; $\mathit{Abominable}/((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$: a modifier of a property; $\mathit{Snowman}/(o\iota)_{\tau\omega}$; $\exists/(o(o(o\iota)_{\tau\omega}))$: the function that assigns **T** to a non-empty class of properties, otherwise **F**; $p \to (o\iota)_{\tau\omega}$.

Proof.

1. $\lambda w\lambda t\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ [{}^0\mathit{Abominable}\ {}^0\mathrm{Snowman}]]$ $\varnothing$
2. $[{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ [{}^0\mathit{Abominable}\ {}^0\mathrm{Snowman}]]$ 1, λ-E
3. $[\lambda p\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ p]\ [{}^0\mathit{Abominable}\ {}^0\mathrm{Snowman}]]$ 2, β-Ex
4. $\neg[{}^0\mathit{Empty}\ \lambda p\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ p]]$ 3, Def.1 (iii)
5. $[{}^0\exists\lambda p\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ p]$ 4, Def. 3 of $\exists$
6. $\lambda w\lambda t\ [{}^0\exists\lambda p\ [{}^0\mathit{Seek}_{wt}\ {}^0\mathit{Tilman}\ p]$ 5, λ-I

Comments. The proof steps (3) and (4) are necessary, because we work with *partial* function. Hence to make sure that the sequence of proof steps is truth-preserving, before applying the existential quantifier $\exists$, we have to prove that the argument class (here of properties) is not empty ($\mathit{Empty}/(o(o(o\iota)_{\tau\omega}))$).

Yet, we can generalize this proof for any existential quantification *over a constituent construction*. Here is how.

4.3.2 Existential Quantification over a Constituent

First, recall that a *constituent* of *B* is a construction that does not occur displayed in *B*. Now let $t \to \alpha$ be a constituent sub-construction of the construction *B*, the other types as above. Since *B* produces a truth-value and *t* is its constituent, *B* is of the form of a Composition $[\ldots t\ldots]$. Then on the assumption that *B* *v*-constructs **T**, the constituent *t* cannot be *v*-improper and the Composition $[[\lambda x\ B]\ t]$ *v*-constructs **T** as well by Def. 1 of Composition. Thus, the set of α-elements produced by $\lambda x\ B$ is non-empty and the application of $\exists$ quantifier is truth-preserving.

As a result, we obtain the classical $\exists$-I rule.

Existential quantifier Introduction (∃-I)

$$\frac{X \vdash B(t/x)}{X \vdash [{}^0\exists\lambda x\ B]} \qquad (\exists\text{-I})$$

The type α of an entity we abstract over is determined by a proper typing. Here are a few examples:

<u>The Pope is wise</u>
Somebody is wise.

$$\frac{\lambda w\lambda t\ [{}^0\mathit{Wise}_{wt}\ {}^0\mathit{Pope}_{wt}]}{\lambda w\lambda t\ [{}^0\exists\lambda x\ [{}^0\mathit{Wise}_{wt}\ x]]} \qquad \begin{matrix}{}^0\mathit{Pope}_{wt} \to \iota \\ x \to \iota\end{matrix}$$

Comment. Wise is of type $(o\iota)_{\tau\omega}$: the property of individuals. Hence the construction ${}^0\mathit{Pope}$ of the papal office occurs extensionally here. The value of the papal office, i.e. the individual that occupies the office is an object of predication:

<u>Tilman wants to become the Pope</u>
Tilman wants to become something

$$\frac{\lambda w\lambda t\ [{}^0\mathit{Want}_{wt}\ {}^0\mathit{Tilman}\ {}^0\mathit{Pope}]}{\lambda w\lambda t\ [{}^0\exists\lambda y\ [{}^0\mathit{Want}_{wt}\ {}^0\mathit{Tilman}\ y]]} \qquad \begin{matrix}{}^0\mathit{Pope} \to \iota_{\tau\omega} \\ y \to \iota_{\tau\omega}\end{matrix}$$

Comment. Want(-to-become) is of type $(o\iota\iota_{\tau\omega})_{\tau\omega}$: the relation of an individual to an office the individual wants to occupy. Hence the construction ${}^0\mathit{Pope}$ of the papal office occurs intensionally here; the whole office/function is an object of predication:

<u>Tilman calculates *Cotg(π)*</u>
Tilman calculates something

$$\frac{\lambda w\lambda t\ [{}^0\mathit{Calc}_{wt}\ {}^0\mathit{Tilman}\ {}^0[{}^0\mathit{Cotg}\ {}^0\pi]]}{\lambda w\lambda t\ [{}^0\exists\lambda c\ [{}^0\mathit{Calc}_{wt}\ {}^0\mathit{Tilman}\ c]]} \qquad \begin{matrix}{}^0[{}^0\mathit{Cotg}\ {}^0\pi] \to *_1 \\ c \to *_1\end{matrix}$$

[12] Here we consider *intensional* seeking that relates the seeker to an intension. If the seeker's activity were sensitive to the *way* a given intension is conceptualized, we would have to analyze *hyperintensional* seeking of type $(o\iota *_n)_{\tau\omega}$. For details on such objectual attitudes, see, for instance, [10].

Comment. Calc(ulate) is of type $(o\iota*_1)_{\tau\omega}$: the relation of an individual to a *construction* that the individual is executing. Thus, the Composition [‘*Cotg* ‘π] occurs hyperintensionally; the whole construction is displayed by Trivialization and becomes an object of predication.

Additional types. Tilman/ι; *Cotg*/$(\tau\tau)$; π/τ.

As a result, application of this rule is classical, and our type system makes it possible to quantify over an entity of a proper type, even of a higher-order. However, we can deduce more than that. First, if a construction occurs extensionally, or, using medieval terminology, *de re*, two principles *de re* are valid.

4.3.3 Two principles de re

The two principles are *existential pre-supposition* and *substitutivity of v-congruent* constructions.

To illustrate, consider again the premise

The Pope is wise.

Since the meaning of ‘the Pope’, here Trivialization ‘*Pope*, occurs *de re*, the existence of the Pope is a *presupposition* of the sentence. In other words, in order that the sentence have any truth-value, the office must be occupied. If it is not so, there is no individual to whom we might ascribe the property of being wise; the sentence cannot be true. But it cannot be false either, because then the sentence that the Pope is *not* wise would have to be true, which is not the case as well, because likewise there is no individual at hand to ascribe the property of *not being wise* to.[13] Thus, we have:

<u>The Pope is/is not wise</u>
The Pope exists.

$\lambda w\lambda t\ (\neg)[\text{‘}Wise_{wt}\ \text{‘}Pope_{wt}]$ $\qquad$ $\text{‘}Pope_{wt} \rightarrow \iota$

$\lambda w\lambda t\ [\text{‘}Exist_{wt}\ \text{’}Pope]$

[13] Survival under negation is the most important test for a *de re* occurrence. Yet, there are two kinds of negation, to wit, external (wide-scope) and internal (narrow-scope) negation. While the latter is presupposition preserving, the former is presupposition denying. For details, see [6].

Comment. *Exist*/$(o\iota_{\tau\omega})_{\tau\omega}$ is the property of an individual office, namely the property of being occupied at a given world/time pair of evaluation. It is defined as follows. Let $f \rightarrow \iota_{\tau\omega}$, $x \rightarrow \iota$. Then ‘$Exist = \lambda w\lambda t\ \lambda f\ [\text{‘}\exists\lambda x\ [x = f_{wt}]]$; hence $[\text{‘}Exist_{wt}\ f] = [\text{‘}\exists\lambda x\ [x = f_{wt}]]$

Substituting this definition into the conclusion of the above argument, we obtain

$\lambda w\lambda t\ [\text{‘}\exists\lambda x\ [x = \text{‘}Pope_{wt}]]$

The other principle *de re* is illustrated by this argument:

The Pope is wise.
<u>Francisco is the Pope.</u>
Francisco is wise.

If the terms ‘Pope’ and ‘Francisco’ are co-referring, i.e. the constructions ‘$Pope_{wt}$, ‘*Francisco* *v*-congruent, then these constructions are mutually substitutable in an extensional context (*de re*).

The two principles *de re* are not valid in case of an intensional or hyperintensional occurrence of a construction, of course. If Tilman wants to become the Pope, the existence of the Pope cannot be derived; it is neither presupposed nor entailed. Tilman may want to become the Pope just in such a state-of-affairs when the papal office goes vacant. And, if Tilman wants to become the Pope and the Pope is Francisco, we cannot derive that Tilman wants to become Francisco, which would be a nonsense.

Yet, even in case of an intensional or hyperintensional context, we can derive more. We can quantify *into* such a context. Quantifying into an intensional context is driven by the same ∃-I rule as above, because constructions occurring intenisonally are also constituents of a given super-construction. To illustrate, consider this argument:

<u>Tilman is seeking an abominable snowman</u>.
Tilman is seeking something abominable.

Again, we must not derive that there is an *individual* that is abominable and it is sought by Tilman. And we do not derive it, because proper typing blocks such an invalid inference. *Abominable* is an entity of type $((o\iota)_{\tau\omega}\ (o\iota)_{\tau\omega})$: the modifier applicable to a *property* of individuals

ISSN 2007-9737

rather than individuals.[14] Hence, there is a *property* $q \to (o\iota)_{\tau\omega}$ such that Tilman is seeking an abominable *q*, namely the property of being a snowman. Proper analysis and typing make it explicit:

$$\frac{\lambda w \lambda t\, [{}^{\prime}Seek_{wt}\ {}^{\prime}Tilman\ [{}^{\prime}Abominable\ {}^{\prime}\mathrm{Snowman}]]}{\lambda w \lambda t\, [{}^{\prime}\exists \lambda q\ [{}^{\prime}Seek_{wt}\ {}^{\prime}Tilman\ [{}^{\prime}Abominable\ q]]]}$$

This goes smoothly. However, when quantifying *into* a *hyperintensional* context, we contend with technical complications that arise from the fact that all constructions occurring in a hyperintensional context are *displayed* rather than executed. And, as explained above, a displayed construction does not produce an object to operate on. Rather, the construction itself is an object to operate on. Constructions are displayed by Trivialization, which "closes" the construction much closer than λ-abstraction. In particular, variables occurring in a hyperintensional context are bound by Trivialization and thus not amenable to logical operations.

4.3.4 Existential quantification into a hyper-intensional context; substitution method

To illustrate, consider again the assumption that

Tilman calculates cotangent of π.

We *must not* derive that there is a *number x* such that Tilman calculates *x*, because there is no such number. The function cotangent is not defined at π. And even if it were defined, it makes no sense to calculate a number without any mathematical procedure to be executed. But we do not derive it, because the above ∃-I rule is applicable only to *constituents* of a given construction, while the Composition [‘*Cotg* ‘π] is displayed in

$$\lambda w \lambda t\, [{}^{\prime}Calc_{wt}\ {}^{\prime}Tilman\ {}^{\prime}[{}^{\prime}Cotg\ {}^{\prime}\pi]]$$

Yet, it might seem unproblematic to derive that there is a number (to wit the number π) the cotangent of which Tilman calculates, because this argument is obviously valid.

<u>Tilman calculates cotangent of π</u>

Tilman calculates cotangent of something

Still careless application of the ∃-I rule similar to generalization into an intensional or extensional context is *not* valid:

$$\frac{\lambda w \lambda t\, [{}^{\prime}Calc_{wt}\ {}^{\prime}Tilman\ {}^{\prime}[{}^{\prime}Cotg\ {}^{\prime}\pi]] \qquad {}^{\prime}[{}^{\prime}Cotg\ {}^{\prime}\pi] \to *_1}{\lambda w \lambda t\, [{}^{\prime}\exists \lambda x\ [{}^{\prime}Calc_{wt}\ {}^{\prime}Tilman\ {}^{\prime}[{}^{\prime}Cotg\ x]]] \qquad x \to \tau}$$

The reason is this. Trivialisation ‘[‘*Cot x*] constructs the Composition [‘*Cot x*] independently of any valuation *v*. Thus, from the fact that at a ⟨*w*, *t*⟩-pair of evaluation it is true that *Tilman* calculates [‘*Cot* ‘π], we can*not* validly infer that *Tilman* calculates [‘*Cot x*], because *Tilman* calculates the cotangent of π rather than of *x*. Put differently, the class of numbers constructed by

$$\lambda x\, [{}^{\prime}Calc_{wt}\ {}^{\prime}Tilman\ {}^{\prime}[{}^{\prime}Cot\ x]]$$

will be non-empty, according as *Tilman* calculates [‘*Cot x*] and regardless of *Tilman*'s calculating [‘*Cot* ‘π]. The problem just described of λ*x* being unable to catch the occurrence of *x* inside the Trivialized construction is TIL's way of phrasing the standard objection to quantifying-in. Yet in TIL we have a way out (or perhaps rather, a way in). In order to validly infer the conclusion, we need to *preprocess* the Composition [‘*Cot x*] and substitute the Trivialization of π for *x*. Only then can the conclusion be inferred. To this end we developed a *substitution method.* This method deploys the polymorphic functions $Sub^n/(*_n *_n *_n *_n)$ and $Tr^{\alpha}/(*_n \alpha)$ that operate on constructions in the manner stipulated by the following definition.

Definition 4 (*Sub**ⁿ***, *Tr***^α^)** Let $C_1/*_{n+1} \to *_n$, $C_2/*_{n+1} \to *_n$, $C_3/*_{n+1} \to *_n$ *v*-construct constructions D_1, D_2, D_3, respectively*.* Then the Composition

$$[{}^{\prime}Sub^n\ C_1\ C_2\ C_3]$$

v-constructs the construction *D* that results from D_3 by collision-less substitution of D_1 for all occurrences of D_2 in D_3. The function $Tr^{\alpha}/(*_n\ \alpha)$ returns as its value the Trivialization of its α-argument.

Example. Let variable $y \to_v \tau$. Then [‘Tr^{τ} *y*] $v(\pi/y)$-constructs ‘π. The Composition

$$[{}^{\prime}Sub^1\ [{}^{\prime}Tr^{\tau}\ y]\ {}^{\prime}x\ {}^{\prime}[{}^{\prime}Cot\ x]]$$

[14] For details on property modifiers see, for instance, [5] and [16].

$v(\pi/y)$-constructs the Composition $[{'}Cot\ {'}\pi]$.

Note that there is a substantial difference between the *construction* Trivialization and the *function* Tr^{α}. Whereas $'y$ constructs just the variable y regardless of valuation, y being bound by Trivialization in $'y$, $[{'}Tr^{\tau}\ y]$ v-constructs the Trivialization of the object v-constructed by y. Hence y occurs *free* in $[{'}Tr^{\tau}\ y]$.

Below we omit the superscripts n and α and write simply '*Sub*' and '*Tr*' whenever no confusion arises.

It should be clear now how to validly derive that *Tilman* calculates cotangent of something if *Tilman* calculates the cotangent of π. The valid argument, in full TIL notation, is this:

$$\frac{\lambda w\lambda t\ [{'}Calc_{wt}\ {'}Tilman\ {'}[{'}Cot\ {'}\pi]]}{\lambda w\lambda t\ [{'}\exists\lambda x\ [{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ x]\ {'}y\ {'}[{'}Cot\ y]]]]}$$

Proof. Let $Empty/(o(o\tau))$ be the class of empty sets of real numbers. Then for any world-time pair $\langle w, t\rangle$ the following steps are truth-preserving:

1) $[{'}Calc_{wt}\ {'}Tilman\ {'}[{'}Cot\ {'}\pi]]$ $\varnothing$
2) $[{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ {'}\pi]\ {'}y\ {'}[{'}Cot\ y]]]$ 1, def.4
3) $[\lambda x\ [{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ x]\ {'}y\ {'}[{'}Cot\ y]]]\ {'}\pi]$ 2, β-expansion
4) $\neg[{'}Empty$ $\lambda x\ [{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ x]\ {'}y\ {'}[{'}Cot\ y]]]]$ 3, Def. 1 (iii)
5) $[{'}\exists\lambda x\ [{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ x]\ {'}y\ {'}[{'}Cot\ y]]]]$ 4, Def. 3 of $\exists$

Similarly, we can derive that there is a function $f \to (\tau\tau)$ such that Tilman calculates the value of f at π. Here is how.

$$\frac{\lambda w\lambda t\ [{'}Calc_{wt}\ {'}Tilman\ {'}[{'}Cot\ {'}\pi]]}{\lambda w\lambda t\ [{'}\exists\lambda f\ [{'}Calc_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ f]\ {'}g\ {'}[{'}g\ {'}\pi]]]]}$$

Here is another example of valid quantifying into a hyperintensional context:

<u>Tilman believes that Pluto is a planet</u>
Tilman believes that something is a planet

Types. Believe$/(o\iota *_n)_{\tau\omega}$: a hyperintensional attitude, i.e., relation-in-intension of an individual to a *hyperproposition*,[15] i.e., the *construction* of a proposition; $Pluto/\iota$; $Planet/(o\iota)_{\tau\omega}$; $x, y \to \iota$.

$$\frac{\lambda w\lambda t\ [{'}Believe_{wt}\ {'}Tilman\ {'}[\lambda w\lambda t\ [{'}Planet_{wt}\ {'}Pluto]]]}{\lambda w\lambda t\ [{'}\exists\lambda x\ [{'}Believe_{wt}\ {'}Tilman\ [{'}Sub\ [{'}Tr\ x]\ {'}y\ {'}[\lambda w\lambda t\ [{'}Planet_{wt}\ y]]]]]}$$

Note that the above arguments are valid, because we quantified over objects produced by Trivialization, namely $'\pi$, $'Cotg$, $'Pluto$, and these constructions are not v-improper for any valuation v. Trivialization just displays the object that we then quantify over, and the function *Tr* applied to this object (v-produced by a variable) returns as its value just the Trivialization of the object.

In this way, we fully respect an agent's perspective, and our analyses are *literal.* This means that semantically simple terms like 'planet', 'Pluto', 'cotangent' and 'π' are analysed by their Trivializations. Indeed, the sentences do not convey any more information about the meaning of these terms. Strictly respecting agent's perspective is important, because hyperintensional contexts mostly stem from agents' attitudes that are sensitive to the way a given object is conceptualized.

To give a simple example, assume that instead of Trivialization displaying Pluto we conceptualise the dwarf planet Pluto by a definite description 'the first Kuiper belt object that has been discovered'. Then Tilman can believe that Pluto is a planet without believing that the first Kuiper belt object that has been discovered is a planet. Sure, one might object that this definite description does not have to refer to any object, because it might happen that no object was

[15] In general, attitudinal sentences are ambiguous. They come in two variants, intensional and hyperintensional, which roughly correspond to *implicit* and *explicit* knowledge. We usually vote for a hyperintensional analysis, because on this approach the problem of logical/mathematical omniscience does not arise, while it is inevitable in case of an intensional analysis. On the other hand, hyperintensional attitudes are very restrictive as for an agent's inferential capacities. To solve this problem, we developed a method of computing *inferable* knowledge of an agent, provided it is possible to specify agent's inferential capacities, i.e. the set of rules the agent masters. For details, see [13].

ISSN 2007-9737

discovered in Kuiper belt so that we obviously cannot existentially generalize. This is true, but we cannot even derive that there is an individual role of type $\iota_{\tau\omega}$ such that Tilman believes that its occupant is a planet. This would change Tilman's perspective, because we would substitute Trivialization of the role instead of the compose construction which is the meaning of that definite description.

Another objection against the substitution of the definite description 'the first Kuiper belt object that has been discovered' for 'Pluto' in a hyperintensional context is this. Whereas the definite description denotes an individual office that can be occupied by at most one individual, Pluto is the proper name of a definite individual. Hence, the description and the name are not analytically equivalent, and cannot be mutually substituted even in an intensional context. This is also true. Hyperintensional contexts have been characterized just by the fact that the substitution of analytically equivalent terms fails here.

To illustrate, suppose that the Pope denotes exactly the same office as Bishop of Rome. Still, Tilman can (hyperintensionally) believe that the Pope is wise without his believing that Bishop of Rome is wise, because the meanings of 'the Pope' and 'bishop of Rome' are different constructions that are not *procedurally isomorphic*. Thus, 'the Pope' and 'Bishop of Rome' are not synonymous terms and cannot be mutually substituted here, because in a hyper-intensional contexts only *synonymous* terms with procedurally isomorphic meanings can be mutually substituted.[16]

Hence, existential quantifying *into* hyperintensional contexts is valid only if we quantify over objects presented by Trivialization. Our substitution method does precisely this. Generalizing, we formulate the rule for quantifying into a hyperintensional context.

The rule of existential quantifying *into* a hyperintensional context (∃-HI)

Let $C \to o$, and let D be a subconstruction of C that is displayed in C ; furthermore let 'a be a sub-construction of D, a/α, $x,y \to \alpha$. Then the rule ∃-HI is schematically defined as follows:

$$\frac{X \vdash C(\ldots{}'D(y/{}'a)\ldots)}{X \vdash [{}'\exists\lambda x\, C(\ldots [{}'Sub\, [{}'Tr\, x]\, {}'y\, {}'D(y)]\ldots)]} \quad (\exists\text{-HI})$$

Applications of the substitution method introduced in this section are much broader. The method is not applied only for existential quantifying into hyperintensional context. It is used to pre-process a procedural meaning of a sentence with anaphorical references, i.e. to substitute the meaning of an anaphorically referred terms for anaphoric variables (see [8]), and in particular as the correct way of applying a function to an argument, which is specified by β-conversion rules.

5 The Rules for β-Conversion

Since TIL is a partial, typed λ-calculus, besides classical rules of natural deduction introduced above, we also need the so-callled β-conversion rules which specify how to validly apply a function produced by a λ-Closure to an argument

[16] The relation of *procedural isomorphism* has been introduced in TIL to deal with the problem of the structural isomorphism of meanings, hence of co-hyperintensionality, hence of synonymy. It has been demonstrated that the individuation of procedures assigned to expressions as their structured meaning cannot be decided in virtue of a universal criterion applicable to every language. Yet, the positive result is that we have specified a set of rigorously defined criteria of fine-grained procedural individuation, partially ordered according to the degree of their being permissive with respect to synonymy. It turned out that the formalization of procedures in TIL in terms of constructions may become a bit too fine-grained from the point of view of the semantics of natural language. Yet the same problem must be met in any formalization that makes use of □-bound variables, i.e. in any □-calculus, because in an ordinary vernacular we do not use □-bound variables. For this reason, we proposed a criterion that is the most suitable for an ordinary, non-professional language. It is the criterion that declares that procedural isomorphism of TIL constructions obtains whenever the differences between constructions consist just in technical manipulations with □-bound variables. Thus, the rule of co-hyperintensionality (i.e. the rule for substitution of synonymous terms in hyperintensional contexts) has been formulated only conditionally. For details, see [7].

produced by the 'called' subprocedure, i.e., how to compute a functional value. These rules come again in two forms, namely β-reduction and β-expansion, sometimes also called λ-expansion. The problem is this. In the logic of partial functions such as TIL, careless β-conversion 'by name' is not an equivalent transformation. This issue has been delt with in many papers. For recent ones see, for instance, [7, 12, 10]. Thus, here we just briefly summarise.

The rule of β-reduction is a fundamental computational rule of λ-calculi and functional programming languages. In λ-calculi the rule is usually specified thus:

$$[[\lambda x\ M]\ N] \mid\!- M\ (x := N)$$

where *M* is a procedure with a free variable *x* (the 'formal parameter' of the procedure *M*), and this procedure 'calls' another procedure *N* to supply the actual argument value. Hence by '$M\ (x := N)$' is meant the collision-less substitution of *N* for all the occurrences of the variable *x* in the calling procedure *M.*

However, Plotkin in [25] pointed out that this specification is ambiguous. There are two procedurally or operationally non-equivalent ways of executing the rule, namely β-reduction '*by name*' and β-reduction '*by value*'.

From the operational point of view, these two ways differ in the way the argument value is being passed for the formal parameter *x.* If by name, then the *procedure N* is executed *after* its substitution for all the occurrences of the variable *x* in the calling-procedure body *M* (after appropriate renaming of λ-bound variables to prevent collision). If by value, then the procedure *N* is executed *first*, and only if *N* does not fail to produce an argument *value* is this value substituted for all the occurrences of *x* in the body *M.* Plotkin (ibid.) put forward a programming language and a formal calculus for each calling mechanism and then showed how each determines the other.

As a result, he proved that the two mechanisms are *not* operationally equivalent. Furthermore, Duží in [3] and [4] *logically* proved that these two ways of executing the conversion are not only operationally but also *denotationally* *non-equivalent* whenever *partial functions* are involved.[17]

By *validity* of the β-reduction we mean the following. The rule is valid if and only if both the redex (the left-hand side procedure) and the contractum (the right-hand side procedure) are *strictly equivalent* in the sense that under any valuation *v* the two procedures produce the same function/mapping or are both *v*-improper, that is, fail to produce anything.[18]

There are two β-conversions that are strictly equivalent, namely *β-conversion by value* and *restricted β-conversion by name*, which we use in our algorithm for TIL deduction system.

Definition 5 (***β-conversion by value***) Let $Y \rightarrow \alpha$; x_1, $D_1 \rightarrow \beta_1$, …, x_n, $D_n \rightarrow \beta_n$, $[\lambda x_1 \dots x_n\ Y] \rightarrow (\alpha\beta_1 \dots \beta_n)$. Then the conversion

$$[[\lambda x_1 \dots x_n\ Y]\ D_1 \dots D_n] \Rightarrow_\beta$$
$$^2[{}^0 Sub\ [{}^0 Tr\ D_1]\ {}^0 x_1 \dots [{}^0 Sub\ [{}^0 Tr\ D_n]\ {}^0 x_n\ {}^0 Y]]$$

is *β-reduction by value.* The reverse conversion is *β-expansion by value.*

***Claim* 1**. β-reduction and β-expansion by value are valid conversions. In other words, the redex and contractum constructions are strictly equivalent.[19]

This rule is applied not only in hyperintensional contexts, but also in intensional ones. Consider the *de re* reading of the sentence expressing Tom's intensional attitude

"Tom believes *of* the Pope that he is wise".

We can analyse this sentence by applying the property of being believed by Tom to be wise to the holder of papal office, if any. This analysis comes down to the construction

$$\lambda w \lambda t\ [\lambda he\ [{}^0 Believe_{wt}\ {}^0 Tom$$
$$\lambda w^* \lambda t^*\ [{}^0 Wise_{w^*t^*}\ he]]\ {}^0 Pope_{wt}]$$

[17] There are two other flaws of β-conversion by name that are not shared by the conversion by value, to wit 'loss of analytic information' and ineffectiveness. For details, see [9].

[18] As an extreme case the produced function/mapping can be nullary, i.e. an atomic object. The produced object can be also a lower-order procedure.

[19] For the proof see, for instance, [12].

Types. $he \to \iota$; $Believe/(o\iota o_{\tau\omega})_{\tau\omega}$: intensional attitude of an individual to a proposition; Tom/ι; $Wise/(o\iota)_{\tau\omega}$; $Pope/\iota_{\tau\omega}$.

This analysis can be validly reduced in this way:

$$\lambda w\lambda t\ [\lambda he\ [{}^0Believe_{wt}\ {}^0Tom\ \lambda w^*\lambda t^*\ [{}^0Wise_{w^*t^*}\ he]]\ {}^0Pope_{wt}] \Rightarrow_\beta$$

$$\lambda w\lambda t\ {}^2[{}^0Sub\ [{}^0Tr\ {}^0Pope_{wt}]\ {}^0he\ {}^0[{}^0Believe_{wt}\ {}^0Tom\ \lambda w^*\lambda t^*\ [{}^0Wise_{w^*t^*}\ he]]]$$

This reduced construction is the *literal* analysis of the sentence "Tom believes *of* the Pope that *he* is wise". The anaphoric reference 'he' referring to the holder of the papal office is resolved by the substitution of the Trivialization of this holder (if any) for the variable *he*.

Note that Double Execution is necessary here. According to the rule, we first substitute the argument *value* (here ${}^0Pope_{wt}$) for the "formal parameter" (here *he*). As a result, if the Pope is Francsico we obtain the *construction*

$$\lambda w\lambda t\ [{}^0Believe_{wt}\ {}^0Tom\ \lambda w^*\lambda t^*\ [{}^0Wise_{wt}\ {}^0Francisco]]$$

which must be afterwards executed to obtain the proposition that Tom believes. If the Pope does not exist, then both the substitution and the Double execution are *v*-improper in the sense of failing to produce any truth-value. This is as it should be, because ${}^0Pope$ occurs with the supposition *de re.* Hence, existence of the Pope is presupposed here.

In case the argument of a function is produced by Trivialization or by a variable, which are constructions that are not *v*-improper for any valuation *v*, conversion by name is also strictly equivalent, and can thus be applied.

Definition 6 (restricted *β-conversion by name*) Let $Y \to \alpha$; x_1, $D_1 \to \beta_1,\ldots, x_n, D_n \to \beta_n$, $[\lambda x_1\ldots x_n\ Y] \to (\alpha\beta_1\ldots\beta_n)$. Furthermore, let $D_1,\ldots,D_n$ be atomic constructions, i.e. variables distinct from $x_1,\ldots x_n$, respectively, or Trivializations of β_i-objects. Then the conversion

$$[[\lambda x_1\ldots x_n\ Y]\ D_1\ldots D_n] \Rightarrow_{\beta r} Y(D_1/x_1\ldots D_n/x_n)$$

where $Y(D_1/x_1\ldots D_n/x_n)$ arises from *D* by a collision-less substitution of D_1 for x_1, …, D_n for x_n, is the *restricted β-reduction by name.* The reverse conversion is the *restricted β-expansion by name.*

***Claim* 2**. Restricted β-reduction and β-expansion by name are valid conversions. In other words, the redex and contractum constructions are strictly equivalent.

Proof is obvious.

Such a restricted β-reduction is often applied in case we just technically manipulate with λ-bound variables. For instance, the above sentence "The Pope has the property of being believed by Tom to be wise" should obtain the literal analysis as follows:

$$\lambda w\lambda t\ [\lambda w_1\lambda t_1\ [\lambda he\ [{}^0Believe_{w1t1}\ {}^0Tom\ \lambda w^*\lambda t^*\ [{}^0Wise_{w^*t^*}\ he]]_{wt}\ {}^0Pope_{wt}]$$

Which is reducible to

$$\lambda w\lambda t\ [\lambda he\ [{}^0Believe_{wt}\ {}^0Tom\ \lambda w^*\lambda t^*\ [{}^0Wise_{w^*t^*}\ he]]\ {}^0Pope_{wt}]$$

Yet, we do not see any reason to differentiate between the two analyses, and thus mostly use the reduced one.

6 Conclusion

In this paper we introduced the system of natural deduction adjusted for TIL. We first specified the deduction rules applicable in an extensional context that deal with truth-functions. Then the rules for general and existential quantifiers have been introduced. We described a correct application of elimination and introduction rules for quantifiers which are applicable both in an extensional and intensional context. In other words, the rules that quantify *over* a constituent of a given meaning procedure. Furthermore, we specified the rules for quantifying *into* a *hyperintensional* context that make use of the *substitution method*. Finally yet importantly, we dealt with valid rules of β-conversion by value and restricted β-conversion by name.

Though there are systems of automatic theorem provers, known today as HOL (see, for instance, [1] and [14]), we need a system of deduction rules for TIL. The reason is this. HOL provers are broadly used in automatic theorem checking and applied as interactive proof assistants in mathematics. As 'HOL' is an acronym for higher-order logic, the underlying logic is usually a version of a simply typed λ-calculus. This makes it possible to operate both in extensional and intensional contexts, where a value of the denoted function or the function itself, respectively, is an object of predication.

Yet there is another application that is gaining interest, and where HOL systems are not so apt as in mathematics, namely natural-language processing. There are large amounts of text data that we need to analyse and formalize.

Not only that, we also want to have question-answer systems, which would infer implicit computable knowledge from these large explicit knowledge bases. To this end not only intensional but rather hyperintensional logic is needed, because we need to formally analyse natural language in a fine-grained way so that the underlying inference machine is neither over-inferring (that yields inconsistencies) nor under-inferring (that causes lack of knowledge). We need to properly analyse agents' attitudes like knowing, believing, seeking, solving, designing, etc., because attitudinal sentences are part and parcel of our everyday vernacular. And attitudinal sentences, *inter alia*, call for a hyperintensional analysis, because substitution of a logically equivalent clause for what is believed, known, etc. may fail. TIL is a system apt for natural-language processing where these goals can be met.

Acknowledgements

The research reported here in was supported by the Grant Agency of the Czech Republic, project No. GA18-23891S "Hyperintensional Reasoning over Natural Language Texts", and by the internal grant agency of VSB-TU Ostrava, project SGS No. SP2018/172, "Application of Formal Methods in Knowledge Modelling and Software Engineering". Versions of this paper were presented at the 19th International Conference on Computational Linguistics and Intelligent Text Processing, *CICLing 2018*, Vietnam.

References

1. **Benzmüller, Ch. (2015).** Higher-Order Automated Theorem Provers. In *All about Proofs, Proof for All*, **Delahaye, D. & Woltzenlogel Paleo, B. (eds.),** *Mathematical Logic and Foundations*, College Publications, pp. 171–214. DOI: 10.2143/LEA.239.0.3237153.
2. **Duží, M. (2010).** The paradox of inference and the non-triviality of analytic information. *Journal of Philosophical Logic*, Vol. 39, No. 5, pp. 473–510.
3. **Duží, M. (2012).** Extensional logic of hyperintensions. *Lecture Notes in Computer Science*, Vol. 7260, pp. 268–290. DOI : 10.1007/978-3-642-28279-9-19.
4. **Duží, M. (2014).** Structural isomorphism of meaning and synonymy. *Computación y Sistemas*, Vol. 18, No. 3, pp. 439–453. DOI: 10.13053/CyS-18-3-2018.
5. **Duží, M. (2017).** Property modifiers and intensional essentialism. *Computación y Sistemas.* Vol. 21, No. 4, 2018, pp. 601–613., DOI: 10.13053/CyS-21-4-2811.
6. **Duží, M. (2017).** Presuppositions and two kinds of negation. *Logique & Analyse*, Vol. 239, pp. 245–263.
7. **Duží, M. (2017).** If structured propositions are logical procedures then how are procedures individuated? *Synthese,* special issue on the Unity of propositions. DOI: 10.1007/s11229-017-1595-5.
8. **Duží, M. (2018)**. Logic of Dynamic Discourse; Anaphora Resolution. *Frontiers in Artificial Intelligence and Applications*, Information Modelling and Knowledge Bases XXIX, Vol. 301, pp. 263–279. DOI: 10.3233/978-1-61499-834-1-263.
9. **Duží, M. & Jespersen, B. (2013)**. Procedural isomorphism, analytic information, and β-conversion by value. *Logic Journal of the IGPL*, Oxford, Vol. 21, No. 2, pp. 291–308, DOI: 10.1093/jigpal/jzs044.
10. **Duží, M. & Jespersen, B. (2015).** Transparent Quantification into Hyperintensional objectual attitudes. *Synthese*, Vol. 192, No. 3, pp. 635–677. DOI: 10.1007/s11229-014-0578-z.
11. **Duží, M., Jespersen, B., & Materna, P. (2010)**. *Procedural Semantics for Hyperintensional Logic*; *Foundations and Applications of Transparent Intensional Logic*. Dordrecht: Springer.
12. **Duží, M. & Kosterec, M. (2017)**. A valid rule of β-conversion for the logic of partial functions. *Organon F*, vol. 24, No 1, pp. 10-36.

13. **Duží, M. & Menšík, M. (2017).** Logic of Inferable Knowledge. In **Jaakkola, H., Thalheim, B., Kiyoki, Y. & Yoshida, N. (eds.),** *Frontiers in Artificial Intelligence and Applications*, Amsterdam: IOS Press, Vol. 292, pp. 405–425.
14. **Gordon, M.J.C. & Melhan T.F.** (1993) (eds). *Introduction to HOL: A theorem proving environment for higher order logic.* Cambridge University Press.
15. **Jespersen, B. (2016)**: Left subsectivity: how to infer that a round peg is round. *Dialectica*, Vol. 70, No. 4, pp. 531–547.
16. **Jespersen, B., Carrara, M., & Duží, M. (2017)**: Iterated privation and positive predication. *Journal of Applied Logic*, Vol. 25, pp. S48–S71. DOI: 10.1016/j.jal.2017.12.004.
17. **Kamp, H. (1981)**. A theory of truth and semantic representation. In *Formal Methods in the Study of Language*, Part 1, **Groenendijk, J., Janssen, T., & Stokhof, M. (eds.)**, pp. 277–322.
18. **Kamp, H. & Reyle, U. (1993)**. *From Discourse to Logic. Introduction to Model-Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Dordrecht: Kluwer.
19. **Kovář, V., Baisa, V., & Jakubíček, M. (2016).** Sketch Engine for Bilingual Lexicography. *International Journal of Lexicography*, Vol. 29, No. 3, pp. 339–352.
20. **Medveď, M., Šulganová, T., & Horák, A. (2017).** Multilinguality Adaptations of Natural Language Logical Analyzer. *Proceedings of the 11th* Workshop on *Recent Advances in Slavonic Natural Language Processing, RASLAN´17,* pp. 51–58.
21. **Montague, R. (1974)**: English as a formal language. In **Visentini, B. et al. (eds.)**, *Linguaggi nella societa e nella tecnica*. Milan, pp. 189–224,
22. **Muskens, R. (1996)**. Combining Montague Semantics and Discourse Representation. *Linguistic and Philosophy*, Vol. 19, pp. 143–186.
23. **Nevěřilová, Z. (2014).** Paraphrase and Textual Entailment Generation in Czech. *Computación y Sistemas*, Vol. 18, No. 3, pp. 555–568.
24. **Palmer, M.S., Dahl, D.A., Schiffman, R.J., Hirschman, L., Linebarger, M., & Dowding, J. (1986)**. Recovering implicit information. *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics, ACL '86*, Association for Computational Linguistics, Stroudsburg, pp. 10–19. DOI: 10.3115/981131.981135.
25. **Plotkin, G.D. (1975)**. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, Vol. 1, pp. 125–159.
26. **Tichý, P**. **(1988).** *The Foundations of Frege's Logic.* de Gruyter.

Article received on 20/12/2017; accepted on 06/03/2018. Corresponding author is Marie Duží.