

Two kinds of procedural semantics for privative modification

Giuseppe Primiero^{1*} and Bjørn Jespersen^{2**}

¹ Centre for Logic and Philosophy of Science,
Ghent University (Belgium)

giuseppe.primiero@ugent.be

² Department of Computer Science, Technical University of Ostrava;
Institute of Philosophy, Department of Logic,
Czech Academy of Sciences, Prague (Czech Republic)
jespersen@flu.cas.cz

Abstract. In this paper we present two kinds of procedural semantics for privative modification. We do this for three reasons. The first reason is to launch a tough test case to gauge the degree of substantial agreement between a constructivist and a realist interpretation of procedural semantics; the second is to extend Martin-Löf's Constructive Type Theory to privative modification, which is characteristic of natural language; the third reason is to sketch a positive characterization of privation.

1 Introduction

The verbal agreements between constructivist/idealist and platonist/realist semantics are so numerous and so striking that it is worth exploring the extent to which there is also substantial agreement. This paper explores some of the common ground shared by the Constructive Type Theory of Per Martin-Löf³ and the realist Transparent Intensional Logic of Pavel Tichý.⁴ We focus here on the following common features:

- a notion of construction;
- a functional language;
- a typed universe;
- an interpreted syntax.

These four features are sufficient to underpin a neutral notion of procedural semantics. Phrased in neutral terms, linguistic meaning is construed as an abstract procedure, of one or more steps, delineating what operations to apply to what operands in order to obtain a particular product as its outcome. Since

* Postdoctoral Fellow of the Research Foundation - Flanders (FWO). Affiliated Researcher IEG, Oxford University and GPI, Hertfordshire University.

** Project GACR 401/10/0792.

³ See [11] and [15].

⁴ See [3], [23], [24].

the interpreted syntax is susceptible to type-theoretic restrictions, the range of admissible combinations of operations and operands is accordingly constrained. These procedures are structured constructions, each of whose constituents is an abstract object of a particular type.

In this paper we apply the procedural semantics sketched above to the problem of privative modification. We do this for three reasons. The first reason is to launch a tough test case to gauge the degree of substantial agreement; the second is to extend Martin-Löf's Type Theory to privative modification, which is characteristic of natural language; the third reason is to sketch a positive characterization of privation.

Property modification in the Montagovian tradition is a function from properties to properties.⁵ If M is a modifier and F a property, then (MF) is the property formed by applying the function M to the argument F . Thus, $(MF)a$ is the predication of the property (MF) of the individual a . The sentential schema whose semantics we wish to study is

“(MF) a ”.

The interpretation of this schema in a procedural semantics depends on the appropriate explanation of what M , F and a are, and of what logical procedures are involved in modification and predication.

A full semantic theory of modification must be able to account for the following variants:

- *Subsective*: $(M'F)a \therefore Fa$;
- *Intersective*: $(M''F)a \therefore M^*a \wedge Fa$;
- *Modal/intensional*: $(M'''F)a \therefore Fa \vee \neg Fa$;
- *Privative*: $(M''''F)a \therefore \neg Fa$.

The first variant is easily treated in a type-theoretical procedural semantics by standard subset formation, extending the language with quantifiers and λ -terms, and forming ordered pairs $\langle M, F \rangle$ where F is the functional argument of the function M whose functional value is the modified property (MF) . The path from function and argument to value consists in deploying the operation of functional application. The second variant is less straightforward, as it requires a rule for replacing the modifier M by the property M^* .⁶ Our conjecture, in the absence of obvious counterexamples, is that whenever “ Fa ” is an expression in a mathematical or logical theory, $(MF)a$ is exhausted by subsective modification, whereas for F an empirical property and a a person or an artifact, privative modification is unavoidable. In general, any semantic theory of mathematical and logical language must come with an account of modification, since the premise $(M'F)a$ contains the modifier M' .

Two examples to fix ideas:

“ a is a prime number”

⁵ See [13].

⁶ See [1], §4.4, [6]. The third variant will not be considered here. See [7] for discussion.

where *prime* is a modifier of the property *number*; and

“*b* is a large elephant”

where *large* is a modifier of the property *elephant*. In the first example, we consider the least controversial kind of subsective modification, which goes along procedurally with subset formation: given a set of (natural) numbers, the modification of the property of being a number generates the subset of those numbers that have the additional property of being prime numbers.

In empirical languages, we not only have examples like “*b* is a large elephant”, but also cases of privative modification, of which the following would be typical examples:

“*b* is a *forged* banknote”;

“*b* is *sham* jewellery”;

“*b* is a *false* friend”.

According to its definition, privation merely indicates what something is *not*, namely not an *F*. We do not maintain that privation is the converse operation of subsection, and it would be too strong for the constructivist to hold that privation produces the complement of the property *F* (because there are no such types as *not being an F* or *being a non-F*). Instead our thesis is that for the constructivist privation is an extreme case of subsection. Given a set of *F*'s, privation will generate the null set of *F*'s; yet, while forming the null set of a particular property exhausts the logic of privation, its semantics is richer than that. Though both forged banknotes and railroads, say, are not banknotes, there is an intuitive sense in which forged banknotes are somehow ‘closer to’ banknotes than are railroads (or tea mugs or tax forms, etc.) The challenge is to make explicit what this (incomplete) approximation comes down to, which is to say something positive about what properties do define forged banknotes (etc.). Semantically, the quest is for a definition of what it is that banknotes and forged banknotes have in common. The philosophical idea which in our view ought to inform any definition of (*forged F*), say, is that being a forged *F* is as good a property as any. Hence, a procedural semantics needs to show a way of generating such a property: a constructivist semantics needs to have a way of verifying whether a particular individual has the property of being a forged banknote, and a platonist theory must be able to define the proper subset of the complement of any set of banknotes, such that the elements of that subset are forged banknotes. To do so, we characterize a privatively modified property (*MF*) as having some, but not all, of the properties defining *F*. So there is going to be a range of forged *F*'s, such that those sharing more of those properties are closer approximations to *F*. This idea induces a sequence of properties G_1, \dots, G_n jointly defining *F*; the more G_i are satisfied, the closer the approximation to *F*. Those forged banknotes that satisfy most G_i are virtually indistinguishable from banknotes, whereas those satisfying few are shoddy imitations (paper instead of polymer, or vice versa, wrong format, wrong colors, etc.). Still, a very poor forged banknote

will nonetheless share more defining properties with a banknote than will a railroad or a tea mug.⁷

What is wanted, overall, is a philosophically motivated and technically workable account of privative modification interpreted within a basic neutral formulation of procedural semantics. In particular, it must be shown what the type-theoretically constrained procedure for predicating a modified property of an individual looks like. In order to obtain such a technical result in the procedural semantics germane both to the constructivist and the realist approach to type theory, we have recourse to a procedure for subset formation. We then generate an appropriate procedure for privative modification by, accordingly, characterizing one form of subsective modification. However, Martin-Löf's and Tichý's respective theories will, in the final analysis, provide partially diverging accounts of such a procedure.

To sum up, this paper pursues two strands, one methodological, the other problem-oriented. The semantic problem is to provide a procedural account of privative modification in terms of subset formation. The methodological one concerns two different forms that a procedural semantics may take, namely the constructivism of Martin-Löf's Type Theory and the platonism of Tichý's Transparent Intensional Logic. The paper seeks to advance the research both on an ill-understood topic in semantics and the general debate of realism vs. anti-realism.

2 Procedural Semantics for Privative Modification

Both theories start from a notion of construction, which extends to function formation. While both operate within the confines of a typed interpreted syntax, the respective type theories work in different ways. Martin-Löf's type theory assigns a new type to each new property, laying down how to verify whether an individual has that property, whereas Tichý's type theory assigns the same type to all empirical properties of individuals. Consequently, the respective procedures for constructing a modified property are also going to differ.

⁷ We disregard the forger's *intention* to produce forged banknotes. We realize that by disregarding the intentions of someone designing and manufacturing technological artifacts and confining ourselves to physical properties, we are guilty of a philosophical simplification. Logically, however, a property along the lines of *being intended as a forged 100-euro banknote* can be smoothly added to the list of properties jointly defining *being a forged 100-euro banknote*. Another simplification is the absence of a priority relation over the properties jointly defining the modified one. Clearly, a real-life account of modification will discriminate between the properties that are more or less relevant to the modified property. For instance, that a forged 100-euro banknote has got the watermark right may be more relevant than getting the code number wrong. Note that in a procedural semantics like Constructive Type Theory that comes with dependent types, assumptions for hypothetical judgements are normally prioritized: the present formulation is therefore a simplification where presuppositions and assumptions are all introduced at the same level of relevance.

2.1 Construction

On the constructive interpretation, predication starts by laying down all the necessary and sufficient conditions for a judgement of the form $F \text{ set}$ (or equivalently $F \text{ prop}$, on the *props-as-sets* identity) to be formulated: such a type declaration is justified in terms of a judgement $f : F$ that shows a constructor for that set, and an equality judgement $f = f' : F$, to ensure canonicity for that element. The basic formal expressions of the theory are the standard categorical judgements

$$\begin{array}{l} f : F \\ f = f' : F \end{array}$$

with $F \text{ set}$ being the appropriate type declaration. From categorical judgements of the form $f : F$, one extends the language to hypothetical judgements as formulae of the form $F' \text{ set}[x : F]$ which can be understood as a relation between types, corresponding to *functional abstraction*. The justification of such a form of judgement is given by saying that F' is a type whenever an appropriate substitution is performed by a certain canonical constructor f in the type F . Dependent judgements are generalised to an arbitrary number of assumptions contained in contexts (within brackets):

$$\begin{array}{l} f : F[f_1 : F_1, \dots, f_n : F_n] \\ f = f' : F[f_1 : F_1, \dots, f_n : F_n] \end{array}$$

where again each F_i is declared a type in an appropriate way. The theoretical starting point of Martin-Löf's type theory is, therefore, the justification of a typed formula in terms of its instance and the reduction of truth-conditions to assertion-conditions. Formation – with corresponding equalities – is the first computational rule for types; rules are completed by:⁸

- *introduction rule*, to introduce canonical elements of types with equality;
- *elimination rule*, to prove a property for a previously typed element.

As with Martin-Löf's, Tichý's theory construes procedures in a functional fashion. Its syntax is provided by the λ -calculi, but the semantic interpretation of it is explicitly procedural in nature.⁹ The procedural aspect of Tichý's theory is given by the fact that the λ -terms of application and abstraction do not denote, respectively, the *result* of applying a function to an argument or arranging two sets of entities as functional arguments and their values. Rather, in TIL, they denote, respectively, the very *procedure* of applying a function to an argument and of forming a function. The procedure of application is called *Composition* in TIL and is encoded thus: $[X_0 X_1 \dots X_n]$, where X_0 is a construction of a function,

⁸ Cf. [11], p.24.

⁹ However, especially the rules pertaining to β -conversion are susceptible to various constraints. See [1], §2.7 for the details of TIL as a hyperintensional, partial, typed λ -calculus.

X_1, \dots, X_n constructions of its arguments and $[\]$ the procedure of functional application. The procedure of abstraction is called *Closure* in TIL and is encoded thus: $[\lambda x_1 \dots x_n Y]$, where x_1, \dots, x_n construct arguments, Y constructs values of a function and $[\lambda x_1 \dots x_n Y]$ is the procedure of functional abstraction.¹⁰

2.2 Functional Language

The functional extension of CTT is crucial to expressing implicational and quantified formulae. If F is a type, the construction of a new type is possible by considering F' a family of sets over some $x:F$, such that $F'[x:F]$ is also a type. A function can, therefore, be construed as the judgement regarding a certain object F' type based on the prior judgements for a type F , possibly generalized to more types (we skip here identity on types and objects of types):¹¹

$$\frac{F \text{ set}[x:F']}{(x:F')F : \text{set}} \text{Function Formation}$$

$$\frac{f:F[x:F']}{(x)f : (x:F')F} \text{Function Introduction}$$

$$\frac{f:F[x:F'] \quad f':F'}{f(f') : F[x/f]} \text{Function Elimination}$$

The neutral formulation $(MF)a$ of an individual a instantiating the modified property (MF) is constructively expressed as a function M such that for every element x in the set F taken as argument, it returns a function $M(x)$, formally $M(x)[x:F]$. To preserve the functional aspect of M in the constructive notation, we will refer to $M(F)$ type as the modified type satisfied by some $f:F$; this means that the individual a from the original notation corresponds to a *typed* element in F , expressed by a judgement of the form $f:F$, hence it will suffice to translate the modifier M into a function on f , so that $(MF)a$ will be expressed by $M(f)$. Standard modification of a property $M(F)$ is given, therefore, by functional abstraction and it produces subset formation $\{x:F \mid M(x)\}$. The case of privative modification is no exception to this general interpretation: a privative modifier will still take as arguments elements in a basic type F , hence the operation occurs at the level of extensions. It differs from a standard functional type (and standard subset formation) in that it does not define a *set of individuals* of the basic type, because its arguments no longer instantiate the original property F . Rather, the range of this modifier will consist of *functions* from the basic type

¹⁰ Two other constructions are *Trivialization* and *Variable*. Trivializations can be dispensed with here, since we do not need to *mention* constructions; we only *use* them to obtain the entities they construct. For now, think of variables as one-step procedures for obtaining an entity relative to a sequence of assignments of entities to variables. See [1], §§1.1-1.3.2, §2.6.1.

¹¹ See [17], §1.7.

F to the empty set. This shows that constructive privation represents a special case of standard subsection, specified by requiring extra conditions. That the range of the privative modifier is a set of functions of the appropriate type – rather than individuals – can be seen as introducing a type of higher order. The bottom-up approach characteristic of the constructive philosophy is preserved, so that the Introduction Rule uses a construction $f:F$ as a premise to define a privatively modified F in terms of the empty set of F 's.

The functional language of TIL is cast within a ramified type hierarchy encompassing a simple type theory, relative to which each entity of the ontology of TIL receives a type. The entities are organized into a bi-dimensional typed universe. One dimension is made up of non-constructions, the other of constructions. On the ground level of the type hierarchy there are non-constructional entities unstructured from the procedural point of view belonging to a *type of order 1*. Given a so-called *epistemic* (or, equivalently, *objectual*) *base of atomic types* (σ -truth values, ι -individuals, τ -reals doubling as times, ω -possible worlds), the induction rule for forming *functional types* is applied: where $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \dots \times \beta_n$ to α , denoted ' $(\alpha\beta_1 \dots \beta_n)$ ', is a type of order 1 as well. Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted ' $*_1$ '. The type $*_1$ together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, of type $(\alpha\beta_1 \dots \beta_n)$, involving $*_1$ in their domain or range, is a *type of order 2*. Constructions belonging to a type $*_2$ that construct entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*; and so on *ad infinitum*.¹²

Tichý's theory of modification proceeds, therefore, in a strictly top-down manner. First, a modified property is constructed according to the procedure of functionally applying a modifier M to a property F , and only then is the modified property (MF) predicated of an individual a .¹³ What gets predicated of an individual is, strictly speaking, an *extensionalized* property, which is a function from individuals to truth-values.

An intensional entity is any function (mapping) whose domain is in the logical space of possible worlds. For most purposes, TIL takes an intension to be a function from logical space to a function from times to entities, in the manner well-known from possible-world semantics enriched with temporal parameters. Thus, an empirical property of individuals is a function from logical space to a function from times to sets of individuals, where a set of individuals is a characteristic function from individuals to truth-values. Hence, given a particular world/time pair $\langle w, t \rangle$, it is either true or false that a given individual a is a member of the set that is the extension of the property at $\langle w, t \rangle$. Formally, the type of a property is $((\sigma\iota)\tau\omega)$, abbreviated ' $(\sigma\iota)_{\tau\omega}$ '. The TIL abbreviation of a modified empirical property being predicated of an individual will be of the form $\lambda w \lambda t [[MF]_{wt} a]$.

¹² See [1], §1.3.2.

¹³ Note the contrast with the constructivist approach, where a modified property is obtained via application rather than abstraction.

2.3 Interpreted Syntax

The procedural way of generating privatively modified properties is based on the fact that the type-theoretical syntax is interpreted.

Constructive Type Theory can be seen as one of several foundational systems for predicative constructive mathematics,¹⁴ but its additional value is represented by a meaning theory which extends and refines the Brouwer-Heyting-Kolmogorov interpretation of intuitionistic logic.¹⁵ CTT formalizes a proper theory of reasoning and knowledge, an interpreted system whose objects are equipped with meanings.¹⁶ By implementing the Curry-Howard isomorphism, types are intended as polymorphic categories of predication, carrying an internal meaning that can be made explicit in terms of propositions (for which proofs are the appropriate constructors) or sets (correspondingly constructed by their elements). The fact that types represent meanings can be adapted to the interpretation of natural language semantics,¹⁷ where reference is generally construed as the relationship between nouns or pronouns and the objects that are named by them. In a constructive procedural semantics every object comes embedded within its meaning category, by which a type gains its meaning from its constructor, and the constructor is meaningfully expressed whenever accompanied by its type (“no entity without a type”).¹⁸ As a result, any expression occurring in one of the computational rules comes embedded with types that yield meanings, and each meaning category is reduced to the corresponding syntactical construction procedure.

The syntax of TIL (its formal ‘language of constructions’ in which constructions are encoded) is inherently interpreted because both constructions and the entities they construct cannot be introduced without typing them first.¹⁹ A *semantic analysis* of a piece of language executed in accordance with TIL proceeds along the following three steps.²⁰ First, *type-theoretic and logical analysis*: all and only logical entities (operations and their operands) being denoted by sub-expressions occurring in the overall expression under analysis receive a type, which may be drawn from the simple or ramified type hierarchy. Second, *synthesis*: the constructions of the entities mentioned are executed in accordance with the logical operations made explicit by the logical analysis in order to unveil the entity denoted by the overall expression. Third, *type checking*: by means of an annotated tree it is checked whether the type assignments check out.²¹

¹⁴ Constructive set theory, explicit mathematics and predicative topos are other examples of systems of constructive mathematics.

¹⁵ Cf. [10], [11], [17], ch.1.

¹⁶ Cf. [17], ch.1.

¹⁷ See [21].

¹⁸ See also [16].

¹⁹ See [1], §1.5.1, §2.1.2.

²⁰ See [1], §2.1.1.

²¹ See [9] for details.

3 Constructive Privative Modification

Standard subsets are used in the type-theoretical setting in order to express a type that is defined by comprehension in the range of another type. Constructively, this corresponds to nothing other than a propositional function from type F to another type F' , i.e. function formation from sec. 2.2, requiring the definition of the type in terms of the judgement $F : set[x : F']$ with an equality judgement defined on it. The appropriate introduction rule corresponds to functional abstraction $(x)f : F(x : F')$ and it is equivalent to Church's λ -abstraction. To know that the preceding rule is correct, the judgement $f(f') : F[x/f']$ must be obtained by function elimination, showing an object of the type F which satisfies also the subtype F' , a typed version of β -conversion.²²

Let us generalise and consider our subtype as M for 'modifier'; in this way one obtains the subset of elements in F satisfying M :

$$\frac{F \text{ set} \quad M(x) \text{ set } [x:F]}{\{x:F \mid M(x)\} \text{ set}} \quad \text{Standard Subset Formation Rule}$$

By the side condition on canonical elements, if $f = f'$ and $M(x)$ is true for some $x : F$, one obtains equal canonical constructions of the set $\{x : F \mid M(x)\}$ when f or f' is used as input of M . That is, since every propositional function is extensional in the sense that it yields equal types when applied to equal elements, it follows from $f = f' : F$ and $M(x) \text{ set}[x : F]$ that $M(f)$ and $M(f')$ are equal types. Consequently, from the requirement that $M(f)$ be true, we immediately get that also $M(f')$ is true.

The use of subset formation for an arbitrary property F (e.g. *banknote*) and a *privative* modifier M (e.g. *forged*) is not entirely correct, however. To preserve the constructive interpretation also for the case of privative modification, it is required that the meaning of $M(F) \text{ set}$ be given by some (canonical) $M(f)$. By using standard subset formation, the modifier type M will yield a subset of the set of *canonical* F 's. Since a *privative* modifier M is intended as a modification procedure that changes entirely the range of its input, an alteration is needed. Because a forged banknote is not a banknote in the first place, the privative modifier *forged* cannot be interpreted as a propositional function from the canonical set of banknotes to one of its (canonical) subsets. For this reason, one needs to define privative modification as an extreme version of subsection. The obvious intuition is that the basic argument $F \text{ set}$ needs to be modified whenever used as an input of the privative modifier M in a way that allows us to turn every $x : F$ into an element of the function from F to the empty set. The first step towards obtaining such a procedure is to define appropriate constructions of the empty set and of the function from a set to the empty one, returning the empty set of elements in that set. The empty set is introduced by declaring the following constants:²³

²² See [17], §1.8. For an analysis of functions and types and the reference of abstract terms, see [18].

²³ Cf. [15], p.21.

$$\begin{array}{c} \{\}: set; \\ case_{\{\}} : El(Z(x)) [Z:(\{\}) set, x: El(\{\})]. \end{array}$$

The first constant simply declares the collection with no elements to be a set; the case step gives the empty set of Z 's elements, by applying a set Z to any element x on condition that Z be an element of the collection of empty sets, and x an element of any set in that collection. Both of these constructions are crucial to the formulation of the privative modifier M . The idea is to use a canonical type declaration F set and to apply a modifier M to any $x:F$, under conditions that $x:El(\{\})$ and $F:El(\{\})$ set. By this, we do not mean to construct (in the standard way) an arbitrary empty set, nor to show a (constructively inadmissible) canonical element for not- F . For the canonical constructive empty set does not allow distinguishing among different empty sets (which is what we need, if we interpret privative modification as a construction of the empty set); and there is constructively no way to give a definitional procedure for a negative type such as *the set of non-banknotes*, because its conditions cannot be canonically specified, in case such set should include everything that does *not* satisfy the conditions for being a banknote. Instead, we give the appropriate assertion conditions for a function that takes any element in the set of banknotes to the complement of such a set, because in this case it is completely specified what the conditions for its input are, and the function only requires that those conditions remain (entirely or partially) unsatisfied. Formally:

Privative Subset Formation Rule

$$\frac{F \text{ set} \quad M(x) \text{ set}[F:(El(\{\})) \text{ set}; x:El(F(f))]}{\{x:F \mid M(x)\} \text{ set}}$$

This construction defines a function M over the set F ; the result is not a canonical subset of F , for given any x in F as its input, $M(x)$ returns a set of functions to the empty set. The apparent mismatch between F in the first premise and its occurrence in the context of the second premise is easily explained: the first premise declares a type which, by the given case formation rule for the empty set, is taken as valid input for the type of elements in the empty set and used as a condition for the second premise. In the latter, F (a function) is employed as the argument of a function application rule: namely, M is the function and $x:El(F(f))$ gives the input. Nothing needs to be said explicitly about M , provided the needed information is contained in the context under which M is a valid construction.²⁴

²⁴ We consider $El(\{\})$ in the second premise as not entirely arbitrary, instead it contains an object of the type F defined by the first premise: hence one might require that f be not only an arbitrary object in $\{\}$, but, more specifically, an object in the set $\{x:\{\} \mid F(x)\}$. This makes any (standard) restriction over F impossible. The second premise needs to be taken conditionally, where its conditions are not meant to be interpreted

When the Privative Subset Formation Rule is applied to the example of *forged banknote*, one starts from the set of banknotes and, by applying the appropriate conditions on that set, one wishes to obtain the empty set of banknotes:

$$\frac{\text{banknote set} \quad \text{forged}(x) \text{ set}[\text{banknote}:(El\{\}) \text{ set}; b:El(\text{banknote}(x))]}{\{x:\text{banknote} \mid \text{forged}(x)\} \text{ set}}$$

It is essential, therefore, to operate with *typed* empty sets.

Privative modification treated as output of the empty-set function lays down the distinction between the output of $M(F)$ – for M some privative modifier like *forged* and F an argument, e.g. *banknote* – and any other empty set: what is the difference between constructing the empty set of banknotes in terms of the set of forged banknotes and any other way of constructing a set none of whose elements is a banknote? This problem is constructively solved by putting forward an appropriate equality rule governing $M(F)$ with respect to the set F :

Equality Rule on Sets

$$\frac{F \text{ set} \quad F = F' \text{ set} \quad M(x) \text{ set}[F = F' : (El\{\}) \text{ set}; f:El(F = F'(x))]}{\{x:F = F' \mid M(x)\} \text{ set}}$$

By this rule, for any equivalent set taken as argument of the modifier, the same empty set is obtained. For any set G with its own constructor $g \neq f : F$ the modifier $M(x) \text{ set}[x : G]$ shall return a different empty set (namely, the empty set of G 's, different from the empty set of F 's). This obviously allows defining the difference between $M(F)$ (*forged banknotes*) and G (*railroads*, say) as empty sets of banknotes in a different sense: the former will, strictly speaking, be the set of function constructors from the set of banknotes to the empty set; the second set will contain no constructor of the set of banknotes at all, hence being empty with respect to any such individual.

The introduction rule instantiates the procedure which, starting from a typed object, returns a privatively modified one:

Introduction Rule

$$\frac{f:F \quad m:M(f)[F:(El\{\}) \text{ set}; f:El(F(f))]}{f:\{x:F \mid M(x)\}}$$

in terms of subsection. The context in which the modifier M is applied to f requires that F be an element in the collection of empty sets; then, f is declared one arbitrary element in this empty set, and finally the set obtained by functional application $F(f)$ is considered. This gives the empty set of F 's, restricting all arbitrary elements of the empty set to those obtained by only considering functions from f s to the definitional set, in turn declared empty. The formulation of the second premise is therefore conditional on the requirement of an empty set F , and that whenever we consider an $M(f)$ we know it leads to an empty set of F 's. The crucial point is precisely not to introduce a subset of F 's, but a set of functions satisfied by an empty argument.

where F can be taken to be the set of banknotes and f an instance of that set, and M the modifier *forged*.

In the introduction rule one starts from the premise that a canonical element f in the set F is given; provided $M(f)$ is true, i.e. there is a canonical element m of the set of functions from F to the empty set, we know that f will yield a canonical element in the set of modified F 's when taken as the argument of the empty-set function of $M(F)$. By the associated equality rule, if $f = f'$ are elements in F , and if there is an m such that $M(f)$ is true, f and f' will yield canonical elements in the set of modified F 's; and from $f = f' : F$ and $m : M(f)$ it follows that $m : M(f')$. Notice that according to the constructive requirement on the introduction rule, in order to form the set of modified F 's, one needs to know at least one instance $m : M(f)$, and because the latter relies on a function applied to f , it is a further presupposition that f be known. For example, in the case of *forged banknote*, in order to display or recognise a forged banknote one needs to be able to lay down the conditions for knowing what a banknote is.

The set of rules is rounded off by an appropriate elimination rule, which makes one able to specify how to extract a modified property from its corresponding set. Formulating an elimination rule for the subset theory is a notoriously difficult matter. It is impossible to give in constructive type theory an elimination rule that captures the way one has introduced elements in a subset, because there is no explicit construction of the element $m : M(f)$ for a standard subset $\{x : F \mid M(x)\}$.²⁵ In the case of privative modification, the elimination rule is supposed to formalise the procedure which, starting from an element of a privatively modified property (*forged banknote*, say), will return another modified element defined over the former; this means that variables will occur bound in the second construction. The informal meaning of the elimination rule is to enable positive predication for privatively modified entities. Saying that a banknote can be identified by ascertaining that it reacts to ultra-violet lamps emitting light at around 365 nanometres²⁶ can be rephrased by saying that a forged banknote will fail to react to uv-lamps emitting light at around 365 nanometres; similarly, one may want to state of a false friend that he or she is a seasoned liar, or that sham jewellery is an “abomination [. . .], a lie, a pretension”.²⁷ In the following, let Δ abbreviate the conditions on a privatively modified set as given by the second premise in its introduction rule. In the corresponding elimination rule, one starts from an instance of a privatively modified property $M(f)$ satisfying $x : \{f : F \mid M(f)[\Delta]\}$; then, another function $f'(x)$ of type $M'(x)$; by substituting f in the free occurrences of x in $M'(x)$, one concludes that $f'(f)$ is an element of the newly modified type $M'(f)$:

²⁵ Cf. [22] for a full explanation, the solution proposed and the consequences for the deductive power of the theory.

²⁶ Pamphlet of the Bank of England, downloadble at http://www.bankofengland.co.uk/banknotes/kyb_lo_res.pdf.

²⁷ From the *Routledge Manual of Etiquette*, 2007, p. 175.

Elimination Rule

$$\frac{f:\{x:F \mid M(x)[\Delta]\} \quad f'(x):M'(x)[x:F, m:M(x)]}{f'(f):M'(f)}$$

3.1 Degrees of Modification

Standard typing rules do not as yet say anything relevant about the sense in which modification comes in degrees, given that there are different sorts of forged banknotes. For example, in the light of a description of a banknote as a green piece of polymer with a hologram printed on it, there are different ways in which a forged banknote may be forged: it may be a piece of polymer which is either not green or lacks the appropriate hologram, or it can be a green piece of something other than polymer with or without a hologram printed on it. All in all, an individual that lacks all three properties fails to qualify as a forged banknote. We shall explain these differences by introducing a formal notion of degrees of modification.

The use of *dependent* types has been shown to be crucial to the definition of the subset formation rule, both in its standard format and its privative variant. We want now to make a dependency relation explicit also for the argument of the modifier function, which will make it possible to differentiate among privatively modified F 's. Take

$$F \text{ set}[x_1:F_1, \dots, x_n:F_n]$$

to be the formal way of saying that F is a canonical set whenever each $x_i:F_i$ is a type-theoretical expression satisfied by an appropriate element $[x_i/f_i]$, where each F_i is a definitional property of F .²⁸ The rule for defining the privative modifier can be analytically formulated with respect to its application to the definitional properties F_i of F :

Dependent Privative Subset Formation

$$\frac{F \text{ set}[x_1:F_1, \dots, x_n:F_n] \quad M(x) \text{ set}[F_i:(El\{\}) \text{ set}; x:El(F_i(x))]}{\{x:F \mid M(x)\} \text{ set}}$$

²⁸ In the present treatment of type-theoretical predications, we are referring to standard types requiring a finitistic formulation of a dependency relation from a context of assumptions. In [12], a non-standard extension of intuitionistic type theory with infinite objects was introduced, which represents a generalization of the finitistic frame, relying on the latter for justification. The negation of predicates at one stage or more in the infinite dependent structure of contexts can be formulated in that frame in a way that resembles the notion of unsatisfied conditions introduced here. As mentioned in the Introduction, we are relying on the simplification that elements in the dependency context come without any priority relation.

where $1 \leq i \leq n$. This new rule says that $M(x)$ is a modified F in view of the empty set of F_i , for every $\bigvee F_i \in F$ up to $\bigwedge_{-1} F_i$ defining F , that is by privation with respect to some – up to all bar one – of its definitional properties.

Depending on the selection and combination of F_i , one obtains different *degrees of modification*. A standard recursive definition of the factorial of the integer n ²⁹ is used in the following for the standard combinatorial result of d elements extracted from n .³⁰ In the following we shall use n to indicate the number of F_i occurring in the dependency context of definitional properties of F , so that we shall call the degree d of modification M of a property F the number of n definitional properties of F with respect to which a privative modifier is applied. By the combinatorial result, the following can be easily stated:

- there will be n distinct modifications of degree $d = 1$, corresponding to the privation of $x : F$ with respect to F_i for some $i \in n$ in the set of conditions for F set;
- there will be a combinatorial number of distinct modifications of degree $d = i < n$ in view of the rule for C_n^i , corresponding to the privation of $x : F$ with respect to the union $\bigcup\{F_1, \dots, F_i\}$, $2 \leq i < n - 1$ in the set of conditions for F set.

Following this rule, an individual determined by 10 properties will accommodate a total of 198,720 possible combinations of modification, counting all the modifications of one property, those of two properties and so on, up to counting 10 possible combinations of modification involving 9 properties (obtained by the calculation $3,628,800/362,880 = 10$). For a simple example, consider the definitional presentation of the set of banknotes introduced above, for which three different modifications of degree 1 are possible, making forged banknotes forged due to their being deprived of just one defining property:

$$\frac{\text{banknote set}[\text{polymer, green, hologram}]}{\text{forged}(x) \text{ set}[F_i : (\text{El}\{\}) \text{ set}; x : \text{El}(F_i(x))]} \\ \{x : \text{banknote} \mid \text{forged}(x)\} \text{ set}$$

where F_i is a variable for any of the properties of being made of polymer, of being green or of having a hologram. A modification of degree 2 would take into account two defining properties; as a result, an instance of the following constructor would be a forged banknote by failing to be made of green polymer (or any other combination):

$$\frac{\text{banknote set}[\text{polymer, green, hologram}]}{\text{forged}(x) \text{ set}[F_{i,j} : \text{El}\{\}) \text{ set}; x : \text{El}(F_{i,j}(x))]} \\ \{x : \text{banknote} \mid \text{forged}(x)\} \text{ set}$$

where again $F_{i,j}$ instantiate two defining properties.

²⁹ $n! = 1$, if $n = 0$ and $n(n - 1)$, if $n \geq 1$.

³⁰ $C_n^d = n! / d! \cdot (n - d)!$.

3.2 Iteration of Modifiers

The formulation of degrees of modification enables us to make comparisons among different instances of the same modified type. In particular, it enables us to express, in the metatheory, that a particular modified set is at a certain degree of approximation to its original counterpart. In the case of *forged banknote*, a privative modification of degree 1 will be a closer approximation to *banknote* than will a privative modification of degree 2. This squares with natural-language predicates like ‘is a well-made forged banknote’, whose use presupposes various degrees to which a forged banknote may succeed in passing for what it is a forgery of.

This remark leads directly to the next case we want to analyse, namely the iteration of modifiers. The modifier *well-made* needs to qualify *forged banknote*, otherwise one ends up with the infelicitous *((well-made forged) banknote)*.³¹ Whether *well-made* modifies *forged banknote* or *forged*, *well-made* is a subsecutive modifier, and we do not want to extract well-made forged banknotes from a set of banknotes. For the iteration to be such that, given a set of forged banknotes, one extracts only the well-made ones, one has to be sure that the construction of *(well-made (forged banknote))* uses a correct application of different subset formation rules.

Consider the by now well-known construction of *forged banknote* and let us abbreviate again the additional conditions on the privative subsection as Δ . Now the construction of *(well-made (forged banknote))* is of the following form:

$$\frac{\frac{\textit{banknote set} \quad \textit{forged}(x) \textit{ set}[\Delta]}{\{x:\textit{banknote} \mid \textit{forged}(x)\} \textit{ set}} \quad \textit{well-made}(x) \textit{ set}[x:\textit{banknote} \mid \textit{forged}(x)]}{\{x:\textit{banknote} \mid \textit{well-made} \times \textit{forged}(x)\} \textit{ set}}$$

This construction applies first the privative subset formation rule and then the standard subset formation rule to the resulting set of functions, thus obtaining the cartesian product of two families of functions over correctly defined sets.

On the other hand, the construction of *((well-made forged) banknote)* is an illegitimate one. The predicate ‘is a (well-made forged) banknote’ does not split the application of the modifiers into two steps, instead the formal construction combines via the cartesian product the standard subsecutive modifier and the privative subsecutive modifier. The resulting construction is ill-defined because the subsecutive modifier *well-made* has the categorical set *banknote* as its argument, whereas the privative modifier *forged* applies to functions defined over an empty set:

$$\frac{\textit{banknote set} \quad \textit{well-made}(x) \textit{ set}[x:\textit{banknote}] \times \textit{forged}(x) \textit{ set}[\Delta]}{\{x:\textit{banknote} \mid \textit{well-made}(x) \times \textit{forged}(x)[\Delta]\} \textit{ set}}$$

³¹ Brackets are used as scope indicators. Note that if *well-made* is to modify *forged*, then because the latter is a first-order modifier (modifying, as it does, a non-modifier), the former must be a higher-order modifier like, e.g., *very*. See [6] for discussion of higher-order modification.

A specific case of iteration of modifiers is the iteration of privative modifiers. This kind of iteration avoids the problem of the previous case, because in both cases the modifiers are privative, hence they share the same conditions. The iteration will give the cartesian product of the sets of functions that are arguments of the modifier. The following construction is an example of a formation rule regulating *burned forged banknote*:

$$\frac{\frac{\text{banknote set} \quad \text{forged}(x) \text{ set}[\Delta]}{\{x:\text{banknote} \mid \text{forged}(x)\} \text{ set}} \quad \text{burned}(x) \text{ set}[\text{forged}(x)[\Delta]]}{\{x:\text{banknote} \mid \text{forged} \times \text{burned}(x)[\Delta]\} \text{ set}}$$

Burned is privative because a burned F is not an F , though it originally was an F . Not all pairs of privative modifiers cancel each other out, such that a burned forged banknote would be a banknote. Furthermore, though both *forged* and *burned* are privative, their logical behaviour does not overlap entirely. In particular, “ a is a burned banknote” is an example of resultative predication³² while “ a is a forged banknote” is not. From a being a burned banknote, it follows that a is not a banknote (because a pile of ashes does not make a banknote), but it is presupposed that a started out as a banknote (otherwise there would have been no banknote to burn). So *burned* comes with a dynamic dimension that *forged* lacks: a forged banknote was never a banknote and only remains an approximation to one.³³

4 Realist Privative Modification

4.1 Predication of modified properties

A property is an intensional entity of type $((o\iota)\tau)\omega$, abbreviated ‘ $(o\iota)_{\tau\omega}$ ’, which is a function from worlds (ω) to functions from times (τ) to sets of individuals $((o\iota))$. A property modifier, by contrast, is an extensional entity, because it is not indexed to possible worlds. Instead it is a function-in-extension between two intensions. Since a property modifier is a function that takes one property to another, its type is $((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$. So in order to construct a modified property, the procedure of functional application (Composition) is called for:

$$[\text{modifier property}]$$

The predication of a property of an individual goes via two instances of functional application. First, the relevant property is extensionalized so as to obtain a set from a property. Second, the set is applied to the individual to obtain a truth-value. The philosophical motivation is that individuals exemplify empirical

³² See [2], p. 226ff.

³³ As for a being a well-made forged banknote, the degree to which a qualifies as being well-made is a reflection of the quality of the craftsmanship of the forgery.

properties only relative to worlds and times.³⁴ Schematically, predication is this Closure:

$$\lambda w \lambda t [property_{wt} a]$$

This Closure, which constructs a possible-world proposition (a function from worlds to functions from times to truth-values), would be the logical form of the sense of a sentence like, “*a* is a banknote”.

The schema of the predication of a modified property of *a* is this Closure:

$$\lambda w \lambda t [[modifier\ property]_{wt} a]$$

This Closure would be the logical form of the sense of a sentence like, “*a* is a forged banknote” or “*a* is a burned banknote”.

If the property constructed by $[modifier\ property]$ is itself modified, the resulting predication looks like this:

$$\lambda w \lambda t [[modifier' [modifier\ property]]_{wt} a]$$

This would be the form of, say, “*a* is a burned forged banknote” or “*a* is a well-made forged banknote”. In all three cases the semantic analysis culminates in the assignment of a propositional construction to a sentence as its sense.

4.2 The requisites of privation

True to its top-down approach, TIL accounts for a property like *being a forged banknote* in terms of other properties being ‘stacked upon it’, to wit, the set of properties that are individually necessary and jointly sufficient for an individual to have that property. Such a set is called the *essence* of the property in question, and each element is called a *requisite*.³⁵ The type of a requisite, when a relation-in-extension between two properties, is $(o(o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$, while the type of the essence of a property is $((o(o\iota)_{\tau\omega})(o\iota)_{\tau\omega})$: the *essence* function takes a property to the set of properties that are its requisites. Formally, *F* being of type $(o\iota)_{\tau\omega}$ and *p* ranging over the same type, these two constructions converge in the same set of properties:

$$[essence\ F] = \lambda p [Req\ p\ F]$$

The requisite relation is defined in the following manner. Let *X, Y* be intensional constructions such that *X, Y* are first-order constructions ranging over the type $(o\iota)_{\tau\omega}$ (i.e. *X, Y* are property variables) and let *x* range over ι .³⁶ Then:

³⁴ See [5] for details.

³⁵ See [1], §4.4. Requisites play pretty much the same role as do *presuppositions* in constructivism.

³⁶ See [1] §4.1, def. 4.1. See also §4.1 for *True*, which is the propositional property of being true at $\langle w, t \rangle$.

$$[Req YX] = \forall w \forall t [\forall x [[True_{wt} \lambda w \lambda t [X_{wt} x]] \rightarrow [True_{wt} \lambda w \lambda t [Y_{wt} x]]]]$$

Gloss *definiendum* as, “Y is a requisite of X”, and *definiens* as, “Necessarily, at every $\langle w, t \rangle$, whatever x instantiates X at $\langle w, t \rangle$ also instantiates Y at $\langle w, t \rangle$.”

Logically, privation comes down to, say, *being a banknote* and *being a forged banknote* having an empty intersection at every $\langle w, t \rangle$. This is obtained thus:

$$[Req \lambda w \lambda t \neg [banknote_{wt} x] [forged banknote]]$$

We say that the property constructed by *[forged banknote]* has, *inter alia*, the requisite property constructed by $\lambda w \lambda t \neg [banknote_{wt} x]$. This is to say that if, at some $\langle w, t \rangle$ or other, an individual x is in the extension of *[forged banknote]* then x is in the extension of the property constructed by $\lambda w \lambda t \neg [banknote_{wt} x]$.

Hence, the proposition that *not being a banknote* is a requisite of *being a forged banknote* is equivalent to the proposition constructed thus:³⁷

$$\forall w \forall t [\forall x [[forged banknote]_{wt} x] \rightarrow \neg [banknote_{wt} x]]$$

What is special about the sort of non-banknote that is not a tea mug, a railroad or a tax form, but a forged banknote? Given a $\langle w, t \rangle$, the set constructed by *[banknote_{wt}]* will have a complement in which we find tea mugs and all the rest, including forged banknotes, but the set constructed by *[[forged banknote]_{wt}]* will be a well-defined proper subset of that complement.³⁸ To define the notion of the subset of forged banknotes within the set of non-banknotes, we need to express that no forged banknote is a banknote and that some non-banknotes are forged banknotes:

$$\forall w \forall t [[[All [forged banknote]_{wt}] [\lambda x \neg [banknote_{wt} x]]] \wedge \\ [[Some [\lambda x \neg [banknote_{wt} x]]] \lambda x [[forged banknote]_{wt} x]]]$$

We invoke the quantifiers *All*, *Some*, here of type $((o(o\iota))(o\iota))$.³⁹ *All* is the function from the set constructed by $[F_{wt}]$ to the set of all those sets that contain the set constructed by $[F_{wt}]$ as a subset. *Some* is the function from the set constructed by $[F_{wt}]$ to the set of all those sets that share a non-empty intersection with the set constructed by $[F_{wt}]$.

In the Introduction we argued that a forged banknote is an (intended) approximation to a banknote. We also made the (simplistic) assumption that *being green* and *being made of polymer* exhaust *being a banknote*. Thus, one reason why a may be a forged banknote is because a , though being made of polymer,

³⁷ For the record, ‘ $\forall y$ ’ abbreviates ‘ $[\forall y]$ ’, y ranging over an arbitrary type α , \forall a function of type $(o(o\alpha))$, and ${}^0\forall$ being the Trivialization of this function.

³⁸ See [7] for a positive characterization of the proper subset of the complement of any set of F ’s containing forged F ’s.

³⁹ See [1] §1.4.3.

fails to be green. Therefore, at some $\langle w, t \rangle$, a may have some, though not all, of the properties making up the essence of being a banknote, q ranging over $(oi)_{\tau\omega}$:

$$\lambda w \lambda t \ [[[Some \lambda q [q_{wt} a]] [essence \textit{banknote}]] \wedge \neg[[All \lambda q [q_{wt} a]] [essence \textit{banknote}]]]$$

A forged banknote is any individual that is not a banknote and which is either made of polymer but fails to be green, or is green but fails to be made of polymer. If we add a third property, e.g. *having a hologram*, it becomes an option that a non-banknote may have either one or two of those three properties and, therefore, qualify as a forged banknote to a lower or higher degree. Degrees of modification would be captured in TIL by spelling out which of the requisite properties of being a banknote a given forged banknote possessed.

5 Conclusion and Further Research

Above we set out the philosophical and technical features of two different conceptions – one constructive, the other platonist – of what a procedural semantics for privative modification may look like. These two conceptions of privative modification are, however, only the first step toward dealing with modification in general within these two frameworks. Subjective and privative modification are the easiest two of the altogether four forms of modification mentioned in the Introduction. The modal/intensional variant, on the other hand, represents the most challenging case both philosophically and technically. For one thing, its very logic is far from being clear, since it is not sufficient to simply infer the classical tautology $Fa \vee \neg Fa$ from $(MF)a$. Future research will be devoted to extending both Martin-Löf’s Constructive Type Theory and Tichý’s Transparent Intensional Logic so as to include a worked-out semantics for intensional modifiers. In particular, the application of CTT to intensional modification will take its lead from [19] and [20], which present a modal type theory including syntactic rules for defining possibility judgements made from *open* assumptions.⁴⁰

References

1. Duží, M., Jespersen, B., Materna, P.: Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic. Logic, Epistemology and the Unity of Science, vol.17. Springer, Berlin (2010)
2. Jackendoff, R.: Semantic Structures. MIT Press, London, Cambridge (1990)

⁴⁰ The authors wish to thank the referee for helpful comments pertaining especially to §2. Bjørn Jespersen is indebted to Marie Duží for very helpful suggestions regarding §4.2. Giuseppe Primiero wishes to thank Bartosz Wieckowski for very helpful comments leading to clarifications in §3. The research reported herein was conducted while Bjørn Jespersen was affiliated with the Section of Philosophy, Delft University of Technology, The Netherlands. He gratefully acknowledges its financial support enabling him to participate at the LENLS VI conference.

3. Jespersen, B.: Significant sententialism in Transparent Intensional Logic and Martin-Löf's Type Theory. In: T. Childers, O. Majer (eds.), *The Logica Yearbook 2002*, 117-131 (2003)
4. Jespersen, B.: Explicit intensionalization, anti-actualism, and how Smith's murderer might not have murdered Smith. *Dialectica* 59, 285-314 (2005)
5. Jespersen, B.: Predication and extensionalization. *Journal of Philosophical Logic* 37, 479-99 (2008)
6. Jespersen, B.: Property modification and the rule of pseudo-detachment. (in submission)
7. Jespersen, B., Carrara, M.: Two conceptions of technical malfunction. (in submission)
8. Jespersen, B.: How hyper are hyperpropositions?. *Language and Linguistics Compass*, vol.4, pp.96-106 (2010)
9. Jespersen, B., Duží, M., Materna, P.: The *logos* of semantic structure. In: P. Stalmaszczyk (ed.), *Philosophy of Language and Linguistics. Volume I: The Formal Turn*. Ontos-Verlag, Farnkfurt, pp.85-101 (2010)
10. Martin-Löf, P.: Constructive mathematics and computer programming. In: J.J. Cohen et al. (eds.) *Sixth International Congress for Logic, Methodology and Philosophy of Science*, 153-175. North-Holland, Amsterdam (1982)
11. Martin-Löf, P.: *Intuitionistic Type Theory*. Bibliopolis, Naples (1984).
12. Martin-Löf, P.: Mathematics of infinity. In: P. Martin-Löf and G. Mints (eds.) *Colog 88, International Conference on Computer Logic. Lecture Notes in Computer Science* 417, 146-197. Springer Verlag (1990)
13. Montague, R.: English as a formal language. In B. Visentini et al. (eds.), *Linguaggi nella società e nella tecnica*. Milan, 189-224 (1970). Reprinted in R.H. Thomasson (ed.), *Formal Philosophy*. New Haven, London: Yale University Press (1974).
14. Nordström, B., Petersson, K. and Smith, J.: *Programming in Martin-Löf's Type Theory*. Oxford University Press, Oxford (1990)
15. Nordström, B., Petersson, K. and Smith, J.: *Martin-Löf's Type Theory*. In S. Abramsky, D. Gabbay and T.S.E. Maibaum (eds.), *Handbook of Logic in Computer Science*, 1-38. Oxford University Press, Oxford (2000)
16. Primiero, G.: The determination of reference in a constructive setting. *Giornale di Metafisica* 26, 483-502 (2004)
17. Primiero, G.: Information and Knowledge. A Constructive Type-Theoretical Approach. *Logic Epistemology and the Unity of Science*, vol.10. Springer, Berlin (2008)
18. Primiero, G.: Proceeding in abstraction. From concepts to types and the recent perspective on information. *History and Philosophy of Logic*, 30, 257-282 (2009)
19. Primiero, G.: Epistemic modalities. In: G. Primiero and S. Rahman (eds.) *Acts of Knowledge: History, Philosophy and Logic*, 207-231. College Publications, London (2009).
20. Primiero, G.: Constructive contextual modal judgements for reasoning from open assumptions. Technical Report 542, Centre for Logic and Philosophy of Science, Ghent University, 2009. (in Submission)
21. Ranta, A.: *Type Theoretical Grammar*. Oxford University Press, Oxford (1990)
22. Salvesen, A., Smith, J.: The strength of the subset type in Martin-Löf's Type Theory. In: *Proceedings of LICS'88*. IEEE, Edinburgh (1988)
23. Tichý, P.: *The Foundations of Frege's Logic*. De Gruyter, Berlin (1988)
24. Tichý, P.: *Collected Papers in Logic and Philosophy*. V. Svoboda, B. Jespersen and C. Cheyne (eds.). *Filosofia*, Czech Academy of Sciences, Prague; University of Otago Press, Dunedin (2004)