CONSTRUCTIVE TYPE THEORY, AN APPETIZER

LAURA CROSILLA

Abstract. Recent debates in metaphysics have highlighted the significance of type theories, such as Simple Type Theory (STT), for our philosophical analysis. In this chapter, I present the salient features of a *constructive* type theory in the style of Martin-Löf, termed CTT. My principal aim is to convey the flavour of this rich, flexible and sophisticated theory and compare it with STT. I especially focus on the forms of quantification which are available in CTT. A further aim is to argue that a comparison between a plurality of theories is beneficial to the philosophical analysis. We may, for example, discover helpful features of one theory that we may want to *import* into another context, thus enriching our repertoire of formal tools. Or, through comparison with a less well-known theory, we may gain a better understanding of the characteristics of the theories we are familiar with. As argued in this chapter, CTT has much to offer in all of these respects.

Keywords: Martin-Löf type theory, Simple type theory, dependent types, predicativity, unrestricted quantification, higher-order metaphysics.

1. Introduction

Recent debates in metaphysics have made it absolutely clear that the instruments we employ to analyse philosophical phenomena are not neutral, in the sense that the outcomes of our philosophical analysis may depend on the resources which are available to us. In particular, this is so for the *formal theories* we employ to underpin our philosophical analysis. For example, proponents of higher-order metaphysics have argued that a shift from first-order to higher-order theorising can help us resolve or even dissolve philosophical disputes that have become intractable within a first-order context.[1] Furthermore, different higher-order theories may make available importantly different expressive resources, hence impacting the outcome of one's philosophical analysis.[2]

This is a pre-print of a chapter for the volume *Higher-Order Metaphysics*, edited by Peter Fritz and Nicholas K. Jones, forthcoming with OUP.

[1]See, for instance, (Williamson 2003, Williamson 2013, Jones 2016, Fritz & Goodman 2017, Goodman 2017, Jones 2018, Jones 2019, Trueman 2020).

[2]See e.g. (Florio & Jones 2021).

Key to the most recent debates in metaphysics is a substantial broadening of the spectrum of formal resources ordinarily employed by the philosopher, adding to first-order theories also powerful type theories such as, for example, Simple Type Theory (STT). With this chapter, I wish to encourage a further expansion of the formal tools we employ in our philosophical enquiry. For this purpose, I present the salient features of an intuitionistic theory, **Constructive Type Theory** (CTT), and compare it with STT. By exploring new theories and comparing them with better-known formalisms, we are bound to enrich the philosophical analysis, for example, by making more perspicuous the role and the nature of the formal instruments we employ. An additional benefit is that we may fruitfully *import* resources from one formal context to another, therefore expanding the repertoire of formal tools available to the philosopher. For example, CTT's so-called *dependent types* and *universes* are powerful constructs that can enrich other contexts as well. As further argued in this chapter, CTT's spirit harmonizes well with the primitivist approach that characterises much of contemporary higher-order metaphysics. There are, however, significant differences between CTT and STT, so much so that even a succinct introduction to CTT would require substantial space and involve a high level of technicality. My strategy is therefore to present selected features of CTT, focusing on those aspects of the theory that I think will be most helpful to the higher-order metaphysician. As I go along, I provide references to the relevant literature for the interested reader.

2. Constructive Type Theory: Types and propositions

Constructive Type Theory designates a family of rich and sophisticated intuitionistic theories which are increasingly popular in mathematics and computer science due to their expressive capabilities and computational character. Recently they have been profitably applied to linguistics.[3] The main focus of this chapter is the constructive type theory proposed by Per Martin-Löf as a *foundation of constructive mathematics*, that is, of mathematics carried out in intuitionistic logic.[4] No system of constructive type theory has reached the status of main reference system, in part due to the numerous applications of this theory and in part as Martin-Löf has proposed an *open-ended* theory: new ways of forming types can, and have been, added over time, provided that they can be justified by the same constructive perspective as the core. In this chapter, I sketch the main traits of a type theory,

[3]See (Ranta 1994, Chatzikyriakidis & Luo 2021).
[4]See, for example, (Martin-Löf 1975, Martin-Löf 1984).

CTT, which is close to the system presented in the book *Intuitionistic type theory* (Martin-Löf 1984).[5]

As the current literature is primarily mathematical, and so are the available examples, my presentation differs in a number of respects from standard philosophical introductions to STT. Furthermore, as I focus on the expressive capabilities of STT and CTT, I am mainly interested in syntactic features of these type theories. However, I follow standard practice and use expressions such as "element" of a type for what in the formal systems is more precisely a term (or an expression) of that type and indulge in harmless use and mention confusion. This is to ensure consistency with the relevant literature, to which I often refer to complement my discussion. Furthermore, due to space constraints, my exposition simplifies CTT as much as possible, largely omitting its philosophical interpretations.[6]

2.1. **Kinds of Types: a comparison between STT and CTT.** In STT there are two ground types, the type of *individuals* and the type of formulas or, to use the terminology of CTT, *propositions*, often denoted by e and t, respectively.[7] Typically one also thinks of the type e as the syntactic category of singular terms or 'the type of objects', and of the type t as the type of truth values.[8] New types can be formed by applying the following rule: if A and B are types, then $A \to B$ is a type. Employing the terminology of CTT, we may also say that in STT new types are formed by applying the *type constructor* "$\to$", which given two types A and B produces the *function type* $A \to B$. In the mathematical literature, the elements of $A \to B$ are also thought of as *functions* taking elements of A to elements of B, where the notion of function is primitive and not to be thought of in set-theoretic terms. Crucially, in STT one has also logical connectives and quantifiers obeying the rules of *classical logic* and impredicative *comprehension principles* (e.g. in the form of β-conversion).

[5]For book-length presentations of constructive type theory, see (Nordström, Petersson & Smith 1990, Sommaruga 2000, Primiero 2007, Granström 2011). For short introductory texts, see (Crosilla 2006, Rahman, McConaughey, Klev & Clerbout 2018, Dybjer & Palmgren 2020).

[6]For the philosophy of constructive type theory and for applications of this theory to philosophy see especially the work by Per Martin-Löf, Göran Sundholm and Ansten Klev.

[7]The philosophical literature presents us with a number of variants of simple type theory. Here I consider a functional presentation of simple type theory as in (Benzmüller & Andrews 2019) and (Dorr 2016, p. 87). Furthermore, to ease the comparison between simple and constructive type theories, I employ capital letters for meta-variables ranging over types (rather than lower case Greek letters), in agreement with the conventions in use in CTT.

[8]See, for instance, (Dorr 2016, p. 50) and (Benzmüller & Andrews 2019).

Function types are particularly important as their elements enable us to express predicates. For example, a predicate applying to a singular term can be expressed in STT by a function belonging to the function type $e \to t$.

Consider, for example, the sentence *John is a child*. This can be expressed by applying a function ***child*** of type $e \to t$ to an individual ***John*** of type e, giving as output a proposition, ***child(John)***, of type t. When considering applications of STT to metaphysics, it is important to note that in STT a function such as ***child*** is of a different, *higher* type compared to the type to which an individual such as ***John*** belongs. This is fundamental as it makes it possible to employ STT to express higher-order proposals for the resolution (or dissolution) of long-standing metaphysical disputes as mentioned in the Introduction. Another significant point to bear in mind is that STT has the resources not only to express higher-order entities such as the function ***child***, but also to *quantify* over them. For example, we may express sentences such as: *there is a property that John has* or *John has some property*, where the quantifiers range over higher-order entities. I will get back to the central issue of which kinds of quantification are available in STT and in CTT in Section 4.

In CTT, we also have types of individuals and propositions, and we may form new types by applying type constructors to types. However, a major difference with STT is that there is no all-encompassing type of individuals, but there are (infinitely) *many ground types of individuals*. Typical examples of ground types are the type of natural numbers, $\mathbf{N}$, and the finite types, such as, for example, the type $\mathbf{N}_2$ which contains two (canonical) elements, 0_2 and 1_2. For applications outside mathematics, one may also consider non-mathematical types that act as ground types of individuals. For example, to express statements such as *John is a child*, one may consider a type of children (see Section 3.2).

A further difference with STT is that new types are formed by applying not just one but *a number of* type constructors. Prominent ones are: Σ (dependent disjoint union), Π (dependent Cartesian product), I (identity), $+$ (disjoint union of two types). For example, given types A and B we may form the dependent Cartesian product of A and B, written $\Pi(A, B)$ (see Section 3.1). The function type constructor "$\to$" is also available in CTT, as it is a special case of the dependent Cartesian product.

Another substantial difference between STT and CTT is that the former is classical and impredicative, while the latter is intuitionistic and predicative, that is, intuitionistic rather than classical logic holds in CTT and quantification is restricted in certain respects. We will look into each of these

characteristics, intuitionism and predicativity, in Sections 2.4 and 4. In sections 2.2 and 2.3, we explore further CTT's notions of type and proposition, and in Section 3 we consider examples of types.

2.2. **Types as domains of quantification.** The principal kind of types we consider in this chapter are also called *sets* in (Martin-Löf 1984). They are not to be thought of as sets in axiomatic set theories (such as **ZFC**), as they are essentially typed, namely each element belongs to a type. In fact, Martin-Löf suggests to see his sets as a generalisation of Russell's *ranges of significance of propositional functions*.[9] To avoid terminological confusion with sets in axiomatic set theory, in this chapter I call Martin-Löf's sets simply *types*.[10]

Types in CTT are defined through *rules* which explain *how to construct their elements* and *when two such elements are equal*. The rules thus bring out the constructive character of CTT, as they describe a generative process of construction of the types from their elements. In fact, the rules lay out how to construct the so-called *canonical* elements of a type, namely its typical elements, and also give an identity criterion for them.[11] The connection between types, understood in this way, and Russell's ranges of significance of propositional functions becomes clearer when we consider Martin-Löf's further suggestion to think of types as *domains of quantification*: "it makes sense to quantify over them" (Martin-Löf 1984, p. 22). Presumably the thought is that since a type is defined through rules which thoroughly spell out what the elements of a type are and when two of them are equal it *makes sense* to talk of ***all*** or ***some*** of the elements of that type, namely quantification over types is meaningful. For example, as there are exhaustive rules prescribing how to form the natural numbers and how to identify two of them, quantification over the type of natural numbers is meaningful. However, we can also think of entities that are in some sense *too large* to be types and over which quantification becomes problematic. For example, in CTT there is no "type of all types", as we cannot specify once and for all all the rules defining types.[12]

[9]See e.g. (Martin-Löf 1975, p. 76) and (Martin-Löf 1984, p. 22).

[10] Martin-Löf (Martin-Löf 1984, p. 21) distinguishes the kind of types we consider in this chapter, that he calls *sets*, from a more general notion of type, that he calls *category* or *type*. See Part III of (Nordström et al. 1990), especially p. 137.

[11]I discuss the notion of canonical element at the end of this subsection.

[12]See (Martin-Löf 1984, p. 87) and Section 4. One may wonder whether it makes even sense to consider a type collecting types. As it turns out, this is a fruitful idea that motivates the concept of universe (see Section 4).

Let us consider in more detail how types are introduced in CTT. Each type is introduced through four kind of rules: ***formation***, ***introduction***, ***elimination*** and ***equality***. The *formation* rules explain how to form a new type. For example, the formation rule for the natural numbers states that there is a type $\mathbf{N}$ of natural numbers. There are then formation rules that allow us to form new types from given ones. For example, given types A and B also $A+B$, $A \times B$ and $A \to B$ are types. Furthermore, there are formation rules for so-called *dependent types*, which will be discussed in Section 3.1.

For example, the formation rule for the *disjoint union* of two types, A and B, is as follows:

$+$ - Formation

$$\frac{A \; type \qquad B \; type}{A{+}B \; type}$$

where A *type* signifies that A is a type. The rule explains that the *constructor* '$+$' applies to two types, A and B, and gives as a result a new type, $A+B$, the disjoint union of A and B.

The introduction and elimination rules are particularly important. The purpose of the *introduction* rules is to explain what is a type by explaining what are its *canonical* elements, that is, the typical elements of that type. They also explain when two such elements are equal. For example, the $\mathbf{N}$-introduction rules are as follows:

$\mathbf{N}$-Introduction

$$0 : \mathbf{N} \qquad\qquad \frac{a : \mathbf{N}}{suc(a) : \mathbf{N}}$$

For each rule introducing a canonical element, there is also a rule explaining when two canonical elements are equal. For example, we have that the element 0 of $\mathbf{N}$ is equal to itself, and that if a and b are equal elements of $\mathbf{N}$, so are $suc(a)$ and $suc(b)$, that is:

$$0 = 0 : \mathbf{N} \qquad\qquad \frac{a = b : \mathbf{N}}{suc(a) = suc(b) : \mathbf{N}}$$

The introduction rules for $+$ explain how to form the disjoint union of two types, by explaining what are its canonical elements:

$+$ - Introduction

$$\frac{a : A}{\mathbf{inl}(a) : A{+}B} \qquad\qquad \frac{b : B}{\mathbf{inr}(b) : A{+}B}$$

where **inl** and **inr** are primitive constants that produce a canonical element of $A+B$ and also indicate if it comes from A (**l**eft) or from B (**r**ight).[13]

[13] I omit the corresponding equality rules. As the elimination and equality rules for both $\mathbf{N}$ and $+$ are rather complex they are omitted.

The *elimination* rules explain how to use the elements of a type, that is, they explain how to define functions on the type defined by the introduction rules (Martin-Löf 1984, p. 24). For example, in the case of the type **N**, the elimination rule gives simultaneously the principle of mathematical induction and definition by recursion over the natural numbers. Finally, the *equality* rules furnish the relevant identity criteria for the elements of a type, by showing how a function defined through the elimination rule acts on the canonical elements of the type (as introduced by the introduction rules).

The notion of ***canonical element*** is an important feature of CTT, which distinguishes it from type theories such as, for example, STT. The distinction between canonical and non-canonical elements of a type is particularly important for mathematical types, such as the infinite type **N**, which may be thought of as generated through rules which specify its canonical elements. The introduction rules for **N** introduce its canonical elements: 0 and $suc(a)$ for a a natural number (possibly non-canonical). The type **N**, however, contains also non-canonical natural numbers, that is, natural numbers that are not of the form 0 or $suc(a)$ (for a a natural number). For example, $5+7$ is not of the form $suc(a)$ (for a in **N**), but it should also be considered a natural number since we can calculate $5 + 7$ and obtain a number which is in canonical form. This justifies taking $5 + 7$ as *non-canonical* element of **N**. We may also think of canonical and non-canonical elements of **N**, for example, $suc(11)$ and $5 + 7$, as different presentations of a natural number, somehow resembling different Fregean senses denoting one and the same object.[14]

The notion of canonical element plays a key role in the so-called *meaning explanations* of CTT (Martin-Löf 1982, Martin-Löf 1984). These are an informal, *pre-mathematical* semantics of CTT which explains the meaning of a CTT-statement by showing how the terms occurring in it can be brought into canonical form. Although formal semantics, such as realizability and set-theoretic interpretations, are available for CTT, Martin-Löf introduced his meaning explanations to give a *direct* semantics for CTT, without presupposing other mathematical theories. It is tempting to see here a similarity of intent with primitivist interpretations of higher-order logic in metaphysics.[15]

One may wonder whether the notion of canonical element, which plays an important role for CTT as a foundation for mathematics, is also available and helpful in applications of CTT to everyday situations. In the case of

[14]See (Naibo & Tranchini 2023).

[15]See Martin-Löf (Martin-Löf 1982, p. 176) and (Williamson 2003, p. 459).

small, finite types the rules of introduction may be read as introducing a name for each canonical element of that type and specifying an identity criteria. It is less clear, however, how to introduce canonical elements for types such as, for example, *child* or *tree*. In such cases, if the distinction between canonical and non-canonical elements is not deemed helpful, one may prefer to employ a more permissive formulation of CTT that makes use of a less specific notion of type, which does not require canonical elements for all types.[16]

2.3. **Propositions.** In STT, in addition to a type of individuals, there is also a type of formulas or, we may say, propositions, namely the type t. ***Propositions*** are likewise a key component of CTT, where they play roughly a similar role as propositions in STT.[17] There is, however, a significant difference with propositions in STT, since in STT a proposition is an *element* of the type t, while in CTT a proposition *is a type* and therefore *may have elements*. In fact, as further clarified in Section 2.4, in CTT propositions *are identified with types*, this being a distinguishing characteristics of this type theory.

As propositions are types, they may have elements. What are the elements of a proposition (if any)? The elements of a proposition are its proofs (if the proposition has a proof). Indeed "a proposition is defined by laying down what counts as a proof of the proposition" (Martin-Löf 1984, p. 11). More precisely, a proposition is introduced through rules that specify what is a *canonical* proof of that proposition, and when two such proofs are equal. For example, given propositions A and B, we can form the proposition $A \wedge B$, the conjunction of A and B. If a is a proof of A and b a proof of B, we can combine these proofs to obtain a canonical proof of $A \wedge B$, namely the ordered pair $\langle a, b \rangle$. The notion of canonical proof is here akin to the notion of canonical element that we saw in Section 2.2. The key idea is to include as proofs of a proposition also non-canonical proofs, provided that they are inferred in a cogent way. For example, a proof of $A \wedge B$ which reaches the conclusion $A \wedge B$, but whose last rule is not a conjunction introduction, but, say, modus ponens, is also to be counted as a proof of $A \wedge B$.

The higher-order metaphysician may find the thought that a proposition has elements puzzling. Furthermore, one may worry that if elements of

[16] See footnote 10 for references and (Ranta 1994, Section 2.26) for discussion.

[17] In the philosophical literature on CTT it is often stressed that formulas are distinct from propositions, as they are "the formalistic counterparts of propositions" (Sundholm 1986, p. 498). CTT's notion of proposition is inspired by the German philosophical tradition including, for instance, Bolzano.

a proposition are proofs, CTT describes a mind-dependent domain and is therefore unsuitable as a framework for a general metaphysical theory. Due to space constraints, I cannot discuss the complex question of how to best interpret CTT's notions of proposition and proof. However, arguably the formal system itself leaves entirely open the question of the interpretation of these notions: a proof is simply a term belonging to a type and a proposition is a type. CTT makes no further claim about the nature of the entities these terms and types denote, including the issue of their dependence or independence from us.[18] My preferred way of thinking about propositions in CTT is as having further "structure" or as carrying additional information, their proofs, compared with propositions in STT. This mathematical structure can be omitted when not required, but may be useful in certain contexts; for example, it is key to applications of CTT to proof assistants and also to linguistics, where it allows for an elegant treatment of anaphora and generalized quantifiers.[19]

The crucial point to bear in mind in the present context is that the interplay between propositions and their proofs is intimately bound up with the *constructive* character of CTT: proofs of propositions are intuitionistic rather than classical. The interaction between propositions and their proofs is clarified by the so-called ***Curry-Howard correspondence*** (also known as 'propositions–as–types' isomorphism), which is the focus of the next section.

2.4. **Curry–Howard correspondence.** The Curry–Howard correspondence is motivated by the observation that there is a *structural similarity* between formal systems of intuitionistic predicate logic on the one side and constructive type theories on the other side. The correspondence relates formulas in the intuitionistic predicate calculus with types in a constructive type theory, and proofs of such formulas with elements of those types. Although the Curry-Howard correspondence is characteristic of a number of constructive type theories, it takes a particularly strong form in CTT, where it is not just a correspondence but an isomorphism: in CTT types and propositions are *identified*.

[18]Recently Martin-Löf has offered an attractive interpretation of both propositions and proofs in non-epistemic terms, so much so that proofs are often referred to as "proof-objects". This interpretation of proofs contrasts with earlier work by Martin-Löf which was closer to traditional approaches to constructivism. See (Martin-Löf 1998) and (Prawitz 2012) for discussion. See also Sundholm's interpretation of proofs of propositions as truth-makers (Sundholm 1994, Klev 2016).

[19]See, for example, (Sundholm 1986, Sundholm 1989, Ranta 1994, Bekki 2014, Chatzikyriakidis & Luo 2021).

The Curry-Howard correspondence is often seen as giving precise *mathematical content* to the rather vague constructive explanation of the logical constants that goes under the name of Brouwer-Heyting-Kolmogorov (BHK) interpretation. The BHK interpretation gives an inductive procedure for constructing intuitionistic proofs of a complex formula in terms of the proofs of its components. The Curry–Howard correspondence builds on this aspect of the BHK interpretation and gives a rigorous characterisation of such proofs which are identified with elements of CTT's types.

Let us see how the correspondence works in a simple case. From a constructive perspective, a proof of an implication $A \supset B$ can be thought of as a method or, more precisely, as a *function* that transforms a proof of A into a proof of B. For example, a proof of $A \supset A$ is the identity function that takes a proof of A to itself. Now, in CTT the proposition A is a type whose elements are proofs of A, or, as it is usually referred to in the literature, A is *the type of its proofs*. Similarly for B. Therefore an intuitionistic *proof* of the implication $A \supset B$ is a *function* which given arguments in A (the type of proofs of proposition A) produces values in B (the type of proofs of proposition B). It is then natural to identify the proposition $A \supset B$ with the type $A \to B$ (the type of all functions from type A to type B) and an intuitionistic proof of the implication $A \supset B$ with a function belonging to the type $A \to B$.

Similar considerations explain the working of the Curry-Howard correspondence for the other propositional connectives. To extend the correspondence to the predicate calculus, we need types that exhibit the typical variability of a quantifier over a domain: *dependent types* (see Section 3.1).

The significance of the Curry-Howard correspondence for CTT cannot be overestimated. The correspondence is important first of all for practical reasons: proofs in intuitionistic logic are mapped into elements of types in CTT, and these have a natural *computational* interpretation. For example, an intuitionistic proof of a statement of the form *for every natural number n there exists a natural number m such that $m = 2n$* is now identified with a typed expression in CTT which describes a step-by-step *procedure* that given n computes m. This can be translated (in a thoroughly automatic way) into a *computer program* which given n as input produces m as output. The correspondence thus enables important applications of CTT to real-life computer programming.

The correspondence is also important because, as further discussed in Section 4.1, in the context of CTT, it enforces a rejection of so-called impredicative definitions. It therefore has important consequences for the kind

of quantification that is available in CTT. Before looking at the latter issue, we need to introduce the notion of dependent types, which enables us to express quantifiers in CTT.

3. Judgements, hypothetical reasoning and dependent types

CTT's types are introduced through rules in natural deduction style, with premises and a conclusion, like the rules in Section 2.2. A distinguishing feature of CTT is that the premises and the conclusions of its rules are not formulas or propositions, as in typical presentations of natural deduction, but ***judgements***.[20] The notion of judgement plays a significant role in CTT, as it offers additional sophisticated expressive resources, such as the possibility to define dependent types. In this section, I first clarify how judgements differ from propositions and then sketch how they can be used to express dependent types.

In CTT there are four fundamental judgement forms:

Judgement	Meaning
A *type*	A is a type
$a : A$	a is an element of the type A
$A = B$	A and B are equal types
$a = b : A$	a and b are equal elements of the type A

Given the identification of propositions with types, each judgement form also admits a *propositional reading*. For example, 'A *type*' may be read as 'A *prop*', which signifies 'A *is a proposition*'. The judgement '$a : A$', for some a in A may also be read as 'A *is true*'. This reading is less informative than a judgement of the form '$a : A$', as it does not carry the additional information on which proof of A is being considered. It is therefore only used when all we need to know is that A has a proof, but we are not interested in particular proofs of A.

For example, a propositional reading of the constructor + gives the connective $\vee$, and + - Formation (see page 6) becomes:

$\vee$ - Formation

$$\frac{A \; prop \qquad B \; prop}{A \vee B \; prop}$$

This tells us that if A and B are propositions, then so is also $A \vee B$. Here A *prop*, B *prop* and $A \vee B$ *prop* are all *judgements*, while A, B and $A \vee B$ are *propositions*, rather than judgements. Note that the *connective* '$\vee$' applies

[20] I only focus on the formal aspects of judgements. See (Martin-Löf 1985) for a detailed discussion of the notion of judgement, its similarities with Frege's notion and its historical roots. See also (Sundholm 1986, p. 498) for discussion.

to two *propositions* and gives as a result a *proposition*; it does not apply to judgements.

Under a propositional reading, + - Introduction gives the natural deduction rules for disjunction introduction:

$\vee$ - Introduction

$$\frac{A\ true}{A \vee B\ true} \qquad\qquad \frac{B\ true}{A \vee B\ true}$$

Similarly, the elimination rule for + gives us the natural deduction rule $\vee$-Elimination.

Other type constructors in CTT can be given a propositional reading and we obtain in this way all the connectives of propositional intuitionistic logic with their natural deduction rules. To express quantifiers we need dependent types, whose formulation requires a more sophisticated notion of judgement, which is discussed next.

3.1. **Hypothetical judgements and dependent types.** The judgements introduced so far are independent from assumptions: they are categorical. CTT also features *hypothetical judgements*, which depend on assumptions.

A simple example of hypothetical judgement is the following:

$$(\star) \qquad B(x)\ type\ (x : A),$$

which asserts that $B(x)$ is a type under the hypothesis that x is in A. The meaning of this hypothetical judgement is given by considering the judgements that result if we substitute any element of A for x:

(i) $B(a)$ is a type, for a in A,

(ii) $B(a)$ and $B(b)$ are equal types, for a and b equal elements of A. This second judgement expresses the functional behaviour of B over A and we thus call B a *family of types depending on A*.

The key observation is that in $(\star)$ B may **depend** on an element of A, that is, depending on which element of A we plug in for x in $B(x)$, we may obtain *different types* – say for a, b, c different elements of the type A, $B(a)$, $B(b)$, $B(c)$ may be different types. This marks an important difference with STT, where free variables may occur in expressions for elements of a type but not in expressions involving types, such as $B(x)$. When considering propositions, a family $B(x)$ of types depending on A may be read as a family $B(x)$ of propositions depending on A, written $B(x)\ prop\ (x : A)$. This is called a *propositional function* over A, as it may be thought of as a function that when applied to an element a of A gives as result a proposition $B(a)$.

To see how dependent families of types work, let us consider a simple every-day example. Let P be the type of people. We can then use the

following judgement to expresses in CTT the relation *admires* between two people:

$$admires(x,y)\ type \qquad (x:P,\ y:P).$$

Here the value of the propositional function admires(x,y) (over the type P) varies as we substitute different elements of P for x and y.

A hypothetical judgement such as ($\star$) can be used to introduce the dependent Cartesian product and the dependent disjoint union, whose propositional reading give us the universal and the existential quantifiers, respectively. Let us see how the universal quantifier can be read off from the *dependent Cartesian product*.

Given a type A and a family, $B(x)$, of types depending on A, we can form a new type, the dependent Cartesian product $\Pi(A, B)$. The *elements* of $\Pi(A, B)$ are *functions* which take an element a of type A as input and give an element of type $B(a)$ as output. It is important to distinguish $\Pi(A, B)$ from a family of types $B(x)$ depending on A: $\Pi(A, B)$ is a type, with certain functions as its canonical elements, while $B(x)$ $(x : A)$ is better thought of as a collection (rather than a type) of types indexed by A.

We can give a propositional reading of $\Pi(A, B)$, yielding $(\forall x : A)B(x)$. The introduction and elimination rules for the product then give rise to the rules $\forall$-Introduction and $\forall$-Elimination in natural deduction. Note that in this way we endow the universal quantifier with a constructive interpretation: proofs of $(\forall x : A)B(x)$ are *functions* that map an element a of A to a proof of $B(a)$. When B does not depend on A, the dependent Cartesian product yields the function space $A \to B$ and, under the Curry-Howard correspondence, ***implication***, $A \supset B$. In a similar way, one introduces a type for the so-called ***disjoint union*** of a family of types (with constructor Σ). Under the Curry-Howard correspondence this gives the existential quantifier and also conjunction, as special non-dependent case.

3.2. **An example.** Let us consider how we may formalize the sentence *John is a child* in CTT. Given CTT's rich type structure, we have a number of options. The most common option is to express *is a child* by a *type predication*, introducing a type, say C, for child. As mentioned in Section 2.2, a prominent interpretation of CTT sees types as domains of quantification. Furthermore, we may want to distinguish between different roles of predication: to circumscribe or single out a domain of quantification or to express a property of certain elements of a given domain. Common nouns such as *child* are then naturally read as singling out a domain of quantification

rather than expressing a property of some individual.[21] In this case *John is a child* can be expressed as *John is in* C, written $John : C$, with the type predication playing the role of the copula.

Another option is to use a *propositional function* to express the predicate *is a child*. This suggests that we see this predication as expressing a property of some elements of a type. For example, we may take *child* to be a propositional function depending on the type P of people, that is, we make the following judgement: $child(x)$ *prop* $(x : P)$.

Finally, as function types are available in CTT, a further option is to proceed as in STT, and take *is a child* as a function, say f, that to an element of, say, the type P of people assigns a value in a suitable type. This option requires us to have suitable types for the domain and range of the function f. One option is to use an extension of CTT with a new powerful type constructor, a so-called universe, which acts as a type of propositions (see Section 4), and take the range of f to be such a universe.

This example witnesses the flexibility of CTT. Each of the options I have sketched is importantly different from the others from a philosophical perspective and only a careful analysis can clarify which option is the most appropriate in a given context. It is important to note, however, that all of these options have in common that they distinguish sharply between the categories to which the individual *John* and the predication *is a child* belong. For example, in the first option *John* and *child* belong to different categories altogether, as *John* is an *element* of a type and *child* is a *type*. In the second option, *is a child* is expressed by a propositional function over P, which is a family of types, and hence of a different category from that of *John*, that is an element of the type P. In the third option, *John* and *is a child* belong to distinct types: a ground type of individuals for *John* and a higher function type for *is a child*. In other terms, in all of these options there is a kind of *incommensurability* between the the individual *John* and whatever has been taken to codify in CTT the predicate *is a child*. This suggests that it should be possible to employ CTT to express in full precision a variety of philosophical responses to troubling metaphysical phenomena, that argue for the incommensurability between different entities such as individuals, properties, propositions etc.[22]

[21] See (Sundholm 1986, Ranta 1994, Chatzikyriakidis & Luo 2017) and (Bekki & Mineshima 2017) for critical discussion. See (Klev 2018, Klev 2017) for a philosophical discussion.

[22] See, for instance, (Jones 2016, Klev 2018, Trueman 2020). See also (Klev 2018), for a higher-order solution of the "concept horse paradox" that makes use of type predication, rather than functions, to express predicates such as "is a concept".

3.3. **Equality.** CTT's treatment of identity, termed *equality* in (Martin-Löf 1984), is sophisticated and flexible. Here I can only point to the most essential aspects.[23] We have seen that in CTT we have judgements and propositions, therefore we have two kinds of equality: ***judgemental equality*** and ***propositional equality***. Judgemental equality is the primary notion of equality and figures in two of the four fundamental judgements: $A = B$ (for types A and B) and $a = b : A$ (for a and b elements of type A). Its behaviour is determined by CTT's rules. For example, the rules that define $\mathbf{N}$ explain when two distinct expressions, say $5 + 7$ and $suc(11)$ of type $\mathbf{N}$, are equal. Judgemental equality expresses a *local* form of identity, since each type comes equipped with its own equality relation specified by its defining rules. In this respect CTT differs importantly from other theories which come equipped with a general identity relation, such as extensionality.

In CTT we cannot combine judgements by applying connectives to them to obtain more complex judgements, since connectives apply to propositions rather than judgements. Consequently, if we wish to express a complex statement obtained by combining an equality statement with some other statement by means of a connective, we need first to express the equality statement as a proposition. To this aim, CTT has a constructor, $\mathbf{I}$, that allows us to form new types expressing equality statements. One may also say that $\mathbf{I}$ enables us to *internalise* equality judgements *at the level of propositions*. Given a type A, and elements a and b of A, $\mathbf{I}(A, a, b)$ is a new type (or proposition) which asserts that the two elements a and b of A are equal. Contrary to judgemental equality, which is local, propositional equality allows us to express identity statements across the type-theoretic hierarchy in a uniform manner, making use of just one type constructor, $\mathbf{I}$, that can be applied to any type and its elements.[24]

4. Quantification

In Section 3.1, we have seen that dependent types allow us to express quantified statements of the form $(\forall x : A)B(x)$ and $(\exists x : A)B(x)$, where A is a type. We have also seen that given types A and B, $A \to B$ is a type, as the function type is a special case of the dependent Cartesian product. We can therefore quantify over function types such as $A \to B$, use them to

[23] See (Klev 2022) for a detailed discussion and (Dybjer & Palmgren 2020) for a self-contained description.

[24] The interplay between judgemental and propositional identity has been thoroughly studied as it relates to important variants of CTT, so-called intensional and extensional type theories. See (Maietti & Sambin 2005, Maietti 2009) for a variant of CTT that brings together within one framework intensional and extensional aspects.

construct new types, and so on. An element of a function type such as, for example, a function from the natural numbers to the natural numbers, is a *higher-order* entity. Consequently, in CTT we can represent some higher-order entities and quantify over their types. However, in STT there is also a more powerful kind of quantification, as quantifiers may range over the type t of *all propositions*. Quantification of this kind figures prominently within higher-order metaphysics; it is therefore important to clarify whether it is available in CTT. Higher order theories such as STT have also played a significant role within debates over unrestricted quantification. In the following, I briefly discuss each of these issues and rise some questions, leaving to another occasion a through analysis of each point.

4.1. **Quantification and predicativity.** In CTT propositions are not elements of a distinguished type of propositions but types. Quantifying over a type of all propositions would then amount to quantifying over a type of all types. One may wonder whether it makes sense to consider a type whose elements are types. As we will see shortly, when carefully executed, the idea of introducing a type collecting other types can enrich CTT with new expressive capabilities. However, care is needed. An early version of CTT featuring a *type of all types* turned out to be inconsistent, as realised by Girard (Girard's paradox).[25] As in CTT propositions are identified with types, this rules out quantification over a type of all propositions.

Note that if we could quantify over a type of all propositions, we could define new propositions in terms of the totality of all propositions. Such definitions are paradigmatic examples of *impredicative* definitions, as they define an entity, say A, in terms of a totality to which A belongs. Martin-Löf's way out of Girard's paradox was to ensure that types are defined *predicatively*, that is, avoiding impredicative definitions. In other terms, types are defined from the bottom up, namely, from their elements, as explained in Section 2.2 and 3.1. This bottom-up construction rules out impredicative definitions and thus also the problematic type of all types. However, it also limits in some respects the expressive capabilities of CTT compared with stronger impredicative type theories.[26] In fact, the form of CTT I have considered so

[25]See (Girard 1972) and (Coquand 1986, Coquand 1989) for insightful analysis of Girard's paradox. In particular, Coquand clarifies the significance of the strong form of Curry-Howard correspondence that CTT implements.

[26]For example, the Calculus of Constructions (Coquand & Huet 1986) suitably weakens the Curry-Howard correspondence to enable impredicativity. See also (Crosilla 2016, Crosilla 2022*a*, Crosilla 2022*b*) for more on the variant of predicativity that is characteristic of constructive type theory.

far is rather weak also for mathematical purposes. For this reason, Martin-Löf began a process of expansion of CTT with new type constructors, thus making available new domains of quantification. One may say that the fact that CTT is predicative is compensated by the addition of new powerful type constructions.

Of particular interest in our context are extensions of CTT by *universes*.[27] Universes play the role of transfinite types in type theory and their introduction amounts to a "reflection principle which roughly speaking says whatever we are used to doing with [types] can be done inside a universe" (Martin-Löf 1975, p. 83). We may think of a universe as collecting into a new type (codes for) all the types constructed so far. To avoid paradoxical situations, a universe cannot belong to itself, but (a code for it) belongs to a *higher* universe. For example, we may introduce a first universe, U_1, and then a *next* universe, U_2, which contains (a code for) the universe U_1. As universes gather together (codes for) types that can be thought of as "already constructed", the resulting substantial expansion of CTT is in agreement with the general outlook of type theory as built up from below, or predicative. It also witnesses the open-ended character of type theory, as a new kind of types is added to an initial CTT, therefore expanding its expressive capabilities.

The crucial point for the present analysis is that the first universe U_1 can now be seen as a (higher-level) *type of propositions* and enable us to express *quantification over all propositions* – all the propositions constructed prior to introducing the universe. This can be iterated indefinitely, by adding further universes. In the mathematical case, the combination of universes with other powerful constructions, such as inductive types, produces strong and expressive theories.[28] More work needs to be carried out to see if constructions of this kind suffice to express the forms of quantification that figure prominently in contemporary metaphysics.

4.2. **Unrestricted quantification.** Higher-order theories such as STT have been hold to offer a hospitable environment for unrestricted quantification. It is often claimed that when a philosopher states, for example, that *everything is physical*, they intend to say that absolutely everything there is,

[27] See (Martin-Löf 1975, Martin-Löf 1984, Palmgren 1992).

[28] Rathjen (Rathjen 2005) argues that such theories are strong enough to prove the consistency of a classical theory in which virtually *all ordinary mathematics* can be carried out.

nothing excluded, is physical, that is, they have in mind an all-inclusive domain. It has been argued that in standard first-order contexts the assumption of such unrestricted domains gives rise to paradoxical situations, which are, however, averted by employing higher-order theories (Williamson 2003). Recently, Florio and Jones (Florio & Jones 2021) have given a new analysis of unrestricted quantification in simple type theory that takes inspiration from Russell's notion of range of significance of a propositional function. As Martin-Löf explicitly refers to Russell's ranges of significance of propositional functions, of which his types are meant to be generalisations, Florio and Jones' analysis is particularly promising in our context. In the reminder of this section, I briefly sketch an approach to unrestricted quantification in STT broadly within the spirit of (Florio & Jones 2021) and carry out a first few steps towards its extension to CTT.[29] My aim is not to argue for unrestricted quantification, rather to explore CTT's capabilities and show the fruitfulness of a comparison between STT and CTT under the light of Florio and Jones' analysis.

Florio and Jones's analysis builds around the claim that there is an intimate connection between unrestricted quantification and the structure of meaningful predicability. The latter is further taken to line up with syntactic restrictions on predication as expressed in *simple type theory*. Given this assumption, it is natural to wonder whether *different formulations* of simple type theory have an impact on the availability or not of unrestricted quantification. Florio and Jones' conclusion is that unrestricted quantification is available in two formulations of simple type theory they consider, strict and cumulative type theories, but not in a third more permissive theory. To reach this conclusion, they proceed as follows.

As they take unrestricted quantification to be quantification over an unrestricted domain, Florio and Jones begin by explicating the notion of unrestricted domain. A key insight informs their analysis: an *unrestricted domain contains absolutely everything relevant in principle to the truth or falsity of a generalisation.* In other terms, true universal quantification over an unrestricted domain "precludes there from being absolutely any counterexamples whatsoever" (Florio & Jones 2021, p. 49). But when is a domain unrestricted in this sense? To answer this question Florio and Jones

[29]Florio and Jones examine three variants of simple type theory, none of which coincides with our STT. However, as most of what they say on strict type theory carries over to STT, in my presentation I adapt their discussion to STT. Another difference with (Florio & Jones 2021) is that their analysis makes use of semantic notions, such as interpretations, framed within simple type theory. As more work is needed to transpose these semantic notions to CTT, I here simply omit semantic considerations to focus on CTT itself.

introduce the notion of *Russellian domain*. This notion is best explained by considering a simple predication, say

(1) $\forall x F(x)$,

which states that predicate F, of type $e \to t$, holds of every individual x, that is, of every x of type e.[30] A domain is *Russellian for* (1), if and only if *it coincides with F's range of significance*, namely, with all the entities that can be meaningfully said to be F. The thought is that what cannot be meaningfully said to be F cannot be meaningfully said not to be F either, and therefore cannot be a counterexample to a generalisation involving F (Florio & Jones 2021, p. 52). Therefore Florio and Jones claim that Russellian domains play the role of unrestricted domains: a *domain is unrestricted relative to some predication if and only if it is Russellian relative to it*. For example, in the case of (1), given the type structure of STT, F can be meaningfully applied only to entities of type e. Therefore no entity of a different type, such as, for example, a property of individuals, can be in F's range of significance and provide a counterexample to (1).

The next question is whether there are in fact unrestricted domains in STT for some predication. To answer this question positively, Florio and Jones observe that every individual is self-identical and hence we may employ comprehension to define a property, say $u^{(e\to t)}$, that holds of all and only the individuals. This property may be regarded as an unrestricted domain for predications such as (1), as it clearly satisfies the conditions for being a Russellian domain for such predications. Similar considerations allow us to define universal properties analogous to $u^{(e\to t)}$ that play the role of unrestricted domains for predications of higher orders.

Let us now try to apply Florio and Jones' analysis to CTT. We want to see if unrestricted quantification is available in CTT, that is, if there is a domain, say D, and a predication such that D is unrestricted relative to that predication – in the sense that it is Russellian for it. As constructive type theorists usually regard types as domains of quantification, a straightforward approach to adapting Florio and Jones' analysis to CTT is to take domains to be types. Then the question can be rephrased as follows: is there a type D and a predication, $(\forall x : D)B(x)$, such that D is Russellian for that predication? Here D is Russellian for $(\forall x : D)B(x)$ if it coincides with B's range of significance. Now, for each type, A, for each element a of A, we have that $a = a$. Hence for each a in A we can prove that the proposition $\mathbf{I}(A, a, a)$, expressing the judgement that the element a of A is

[30] Types are omitted for ease of reading.

self-identical, is true. Therefore $(\forall x : A)\,\mathbf{I}(A, x, x)$ is true. But the type A is Russellian for the predication $(\forall x : A)\,\mathbf{I}(A, x, x)$, since $\mathbf{I}(A, x, x)(x : A)$ can be meaningfully applied to all and only the elements of A. According to Florio and Jones' analysis, this means that there is a domain in CTT that is unrestricted for some predication, in fact, *every* type D is unrestricted for the predication *everything in D is self-identical*. Hence unrestricted quantification in the sense of Florio and Jones is available in CTT.

While both CTT and STT satisfy Florio and Jones' criteria for unrestricted quantification, there are striking dissimilarities between these theories' type-theoretic structures. In CTT, contrary to STT, there is no type of all individuals, rather, there are infinitely many types of individuals, each equipped with its own identity criteria. It is then natural to ask whether these theories' type-theoretic structures have an impact on each theory's ability to express the kinds of quantification that occur in philosophical enquiry. Consider a two-element type, say the type $\mathbf{N}_2$, with canonical elements 0_2 and 1_2. If I say (in CTT) *everything in* $\mathbf{N}_2$ *is self-identical* and (now in STT) *everything in u is self-identical* it is tempting to think that the range of the second quantifier is somehow *wider* than that of the first quantifier. This is not to say that the first domain is restricted nor that the second domain contains all the elements of the first one. Types $\mathbf{N}_2$ and u are not only distinct but belong to different theories altogether, so that we cannot directly compare them in any meaningful way. The naive thought is rather that it seems that $\mathbf{N}_2$ will not in general play the role of an all-inclusive domain within our philosophical theorising in the same way as the type of all individuals, u.

Let us go back to the philosophical example we started with, the sentence *everything is physical*. Let us call this sentence s. One may perhaps argue that as STT's type structure lines up with the structure of predication, unrestricted quantification over domains such as u make available the kind of generality required by sentences such as s. The domain u, one may argue, is an all-inclusive domain of individuals. However, it seems that CTT's wealth of separate and unconnected domains of quantification will not provide us with an all-inclusive domain of quantification as required by s.[31]

A possible way out of this difficulty would be to add a new type, say *thing*, to CTT, which plays a similar role as e in STT. It is, however, difficult to imagine how such a type could be introduced in CTT, as types are defined by specifying their (canonical) elements and an identity criteria. What would be the identity criteria for a type *thing*? A more promising strategy is to

[31] See also (Parsons 2006, p. 215).

bring together all the types of CTT into an all-inclusive domain, that is, to introduce a *universe*. Given a universe, U_1, we can prove that for every type A (whose code is) in U_1, for every element x of A, $\mathbf{I}(A, x, x)$. The universe is therefore a Russellian domain for this predication, as it coincides with its range of significance. The universe would also seem to be the kind of all-inclusive domain that can help us in our philosophical inquiry in a similar way as e.

An important point to note is that CTT with a universe has additional expressive capabilities compared with STT, as we can now quantify not only over individual types but over the types themselves. We can, for example, say: every type in the universe is so-and-so. One may say that a universe enables a view from *above* of the types, rather than from within, and consequently offers substantial new expressive capabilities.

5. Conclusions

In this chapter, we have seen some of the most significant features of CTT, among which, for example, the Curry-Howard correspondence, the role of dependent types for quantification and the powerful addition of universes. I hope to have shown the fruitfulness of a comparison between STT and CTT, which has the potential to increase our understanding of the formal tools we employ and their role within philosophy. The comparison may also suggest to *import* new instruments within our best-known theories.

Acknowledgments

The research leading to this article has received funding from the Norwegian Research Council, grant *Infinity and Intensionality: a New Synthesis*, project number 314435. I am grateful to the editors of this volume, Peter Fritz and Nick K. Jones, and to Salvatore Florio for very helpful comments to earlier versions of this chapter that prompted extensive improvements. I also profited from comments by Øystein Linnebo and the Infinity and Intensionality discussion group.

References

Bekki, D. (2014), Representing anaphora with dependent types, *in* N. Asher & S. Soloviev, eds, 'Logical Aspects of Computational Linguistics', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 14–29.

Bekki, D. & Mineshima, K. (2017), *Context-Passing and Underspecification in Dependent Type Semantics*, Studies in Linguistics and Philosophy, Springer Science and Business Media B.V., pp. 11–41.

Benzmüller, C. & Andrews, P. (2019), Church's Type Theory, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Summer 2019 edn, Metaphysics Research Lab, Stanford University.

Chatzikyriakidis, S. & Luo, Z. (2017), On the Interpretation of Common Nouns: Types Versus Predicates, *in* S. Chatzikyriakidis & Z. Luo, eds, 'Modern perspectives in type-theoretical semantics', Vol. 98, Springer.

Chatzikyriakidis, S. & Luo, Z. (2021), *Formal Semantics in Modern Type Theories*, John Wiley and Sons, Incorporated, Newark.

Coquand, T. (1986), An analysis of Girard's Paradox, *in* 'Proceedings of the Symposium on Logic in Computer Science (LICS '86), Cambridge, Massachusetts, USA, June 16-18, 1986', IEEE Computer Society, pp. 227–236.

Coquand, T. (1989), Metamathematical investigations of a calculus of constructions, Technical report, INRIA.

Coquand, T. & Huet, G. (1986), The calculus of constructions, Technical Report RR-0530, INRIA.

Crosilla, L. (2006), 'Constructive notions of set. Part I. Sets in Martin-Löf type theory', *Annali del Dipartimento di Filosofia, Nuova serie XI* pp. 347–387.

Crosilla, L. (2016), Constructivity and Predicativity: Philosophical foundations, PhD thesis, School of Philosophy, Religion and the History of Science, University of Leeds.

Crosilla, L. (2022*a*), 'The entanglement of logic and set theory, constructively', *Inquiry* **65**(6), 638–659.

Crosilla, L. (2022*b*), Predicativity and constructive mathematics, *in* G. O. S. Boscolo & C. Ternullo, eds, 'Objects, Structures, and Logics', Springer International Publishing, Cham.

Dorr, C. (2016), 'To Be F Is To Be G', *Philosophical Perspectives* **30**(1), 39–134.

Dybjer, P. & Palmgren, E. (2020), Intuitionistic Type Theory, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', Summer 2020 edn, Metaphysics Research Lab, Stanford University.

Florio, S. & Jones, N. K. (2021), 'Unrestricted quantification and the structure of type theory', *Philosophy and Phenomenological Research* **102**(1), 44–64.

Fritz, P. & Goodman, J. (2017), 'Counting incompossibles', *Mind* **126**(504), 1063–1108.

Girard, J. (1972), Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur, PhD thesis, These d'Etat, Paris VII.

Goodman, J. (2017), 'Reality is not structured', *Analysis* **77**(1), 43–53.

Granström, J. G. (2011), *Treatise on Intuitionistic Type Theory*, Dordrecht, Netherland: Springer.

Jones, N. K. (2016), 'A higher-order solution to the problem of the concept horse', *Ergo, an Open Access Journal of Philosophy* **3**.

Jones, N. K. (2018), 'Nominalist realism', *Noûs* pp. 808–835.

Jones, N. K. (2019), 'Propositions and cognitive relations', *Proceedings of the Aristotelian Society* **119**(2), 157–178.

Klev, A. (2016), 'Truthmaker semantics: Fine versus Martin-Löf', *Logica Yearbook 2016* pp. 87–108.

Klev, A. (2017), 'Identity and Sortals (and Caesar)', *Erkenntnis* **82**(1), 1–16.

Klev, A. (2018), 'The concept horse is a concept', *Review of Symbolic Logic* **11**(3), 547–572.

Klev, A. (2022), 'Identity in Martin-Löf Type Theory', *Philosophy Compass* **17**(2), e12805.

Maietti, M. E. (2009), 'A minimalist two–level foundation for constructive mathematics', *Annals of Pure and Applied Logic* **160**(3), 319–354.

Maietti, M. E. & Sambin, G. (2005), Toward a minimalist foundation for constructive mathematics, *in* L. Crosilla & P. Schuster, eds, 'From Sets and Types to Topology and Analysis: Towards Practicable Foundations for Constructive Mathematics', Vol. 48 of *Oxford Logic Guides*, Oxford University Press.

Martin-Löf, P. (1975), An intuitionistic theory of types: predicative part, *in* H. E. Rose & J. C. Shepherdson, eds, 'Logic Colloquium 1973', North–Holland, Amsterdam.

Martin-Löf, P. (1982), Constructive mathematics and computer programming, *in* L. J. Choen, ed., 'Logic, Methodology, and Philosophy of Science VI', North–Holland, Amsterdam.

Martin-Löf, P. (1984), *Intuitionistic type theory*, Vol. 1 of *Studies in Proof Theory*, Bibliopolis.

Martin-Löf, P. (1985), On the Meanings of the Logical Constants and the Justification of the Logical Laws, *in* 'Atti degli Incontri di Logica Matematica, Siena', Vol. 2, pp. 203–281. Reprinted in Nordic Journal of Philosophical Logic 1, 11–60.

Martin-Löf, P. (1998), Truth and Knowability: on the Principles C and K of Michael Dummett, *in* G. H. Dales & G. Oliveri, eds, 'Truth in Mathematics', Clarendon Press, Oxford, pp. 105–114.

Naibo, A. & Tranchini, L. (2023), Jugmental and definitional equality from a Fregean perspective, *in* A. Klev, ed., 'The Architecture and Archaeology of Modern Logic. Studies dedicated to Göran Sundholm', Springer.

Nordström, B., Petersson, K. & Smith, J. M. (1990), *Programming in Martin-Löf's Type Theory: an introduction*, Clarendon Press.

Palmgren, E. (1992), 'Type-theoretic interpretation of iterated, strictly positive inductive definitions', *Arch Math Logic* **32**, 75–99.

Parsons, C. (2006), The problem of absolute universality, *in* A. Rayo & G. Uzquiano, eds, 'Absolute Generality', Oxford University Press, pp. 203–19.

Prawitz, D. (2012), *Truth and Proof in Intuitionism*, Springer Netherlands, Dordrecht, pp. 45–67.

Primiero, G. (2007), *Information and Knowledge: A Constructive Type-Theoretical Approach*, Springer.

Rahman, S., McConaughey, Z., Klev, A. & Clerbout, N. (2018), A brief introduction to constructive type theory, *in* R. S., M. Z., K. A. & C. N., eds, 'Immanent Reasoning or Equality in Action. Logic, Argumentation & Reasoning (Interdisciplinary Perspectives from the Humanities and Social Sciences)', Vol. 18, Springer.

Ranta, A. (1994), *Type-Theoretical Grammar*, Oxford, England: Oxford University Press on Demand.

Rathjen, M. (2005), 'The constructive Hilbert program and the limits of Martin–Löf type theory', *Synthese* **147**, 81–120.

Sommaruga, G. (2000), *History and Philosophy of Constructive Type Theory*, Dordrecht, Netherland: Springer.

Sundholm, G. (1986), Proof theory and meaning, *in* 'Handbook of Philosophical Logic', Vol. III, D. Reidel Publishing Company, pp. 471–506.

Sundholm, G. (1989), 'Constructive generalized quantifiers', *Synthese* **79**(1), 1–12.

Sundholm, G. (1994), 'Existence, Proof and Truth-Making: A Perspective on the Intuitionistic Conception of Truth', *Topoi* **13**, 117–126.

Trueman, R. (2020), *Properties and Propositions: The Metaphysics of Higher-Order Logic*, Cambridge: Cambridge University Press.

Williamson, T. (2003), Everything, *in* J. Hawthorne & D. Zimmerman, eds, 'Philosophical Perspectives 17: Language and Philosophical Linguistics', Blackwell, Boston and Oxford.

Williamson, T. (2013), *Modal Logic as Metaphysics*, Oxford University Press, Oxford.

Laura Crosilla, Department of Philosophy, IFIKK, University of Oslo, Postboks 1020, Blindern, 0315 Oslo, Norway. Laura.Crosilla@ifikk.uio.no