Math. Struct. in Comp. Science (2002), *vol.* 12, *pp.* 771–795.
DOI: 10.1017/S0960129502003766 Printed in the United Kingdom

A proof-irrelevant model of Martin-Löf's logical framework

DANIEL FRIDLENDER[†]

FaMAF, Universidad Nacional de Córdoba,
Ciudad Universitaria, 5000 Córdoba, Argentina
Email: fridlend@hal.famaf.unc.edu.ar
URL: www.cs.famaf.unc.edu.ar/fridlend/

Received 18 March 2001; revised 14 February 2002

We extend the proof-irrelevant model defined in Smith (1988) to the whole of Martin-Löf's logical framework. The main difference here is the existence of a type whose objects themselves represent types rather than proof-objects. This means that the model must now be able to distinguish between objects with different degree of relevance: those that denote proofs are irrelevant whereas those that denote types are not. In fact a whole hierarchy of relevance exists.
Another difference is the higher level of detail in the formulation of the formal theory, such as the explicit manipulation of contexts and substitutions. This demands an equally detailed definition of the model, including interpreting contexts and substitutions.
We are thus led to a whole reformulation of the proof-irrelevant model. We present a model that is built up from an arbitrary model of the untyped lambda calculus. We also show how to extend it when the logical framework itself is enlarged with inductive definitions. In doing so, a variant of Church numerals is introduced.
As in Smith (1988), the model can only be defined in the absence of universes, and it is useful to obtain an elementary proof of consistency and to prove the independence of Peano's fourth axiom.

1. Introduction

Type theory, as studied here, is a system invented by Martin-Löf for formalising constructive mathematics. Different stages in the development of the system gave rise to several formulations of type theory, see, for instance, Martin-Löf (1975), Martin-Löf (1984), Nordström *et al.* (1990) and Martin-Löf (1992).

For the presentations of type theory in Martin-Löf (1975) and Martin-Löf (1984), Smith defined a model in Smith (1988) in which types are interpreted as truth values, and in which all the instances of a dependent type receive the same interpretation. The latter means that proof-objects play no role in the interpretation of the types in which they

[†] Some of the work presented in this paper was developed during the author's affiliation to BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation.

occur. In other words, they are irrelevant to the interpretation of types. For that reason, this model is sometimes said to be *proof-irrelevant*.

The importance of proof-irrelevant models is that they often make it evident that a given proposition is not provable in type theory. In Smith (1988) they are used, for instance, to give an elementary proof that Peano's fourth axiom is not provable, and hence, to give an elementary proof of consistency of type theory.

As mentioned in Smith (1988), the main limitation of proof-irrelevant models is that they cannot be defined in the presence of universes, which make Peano's fourth axiom provable.

The proof-irrelevant model in Smith (1988) is defined in a direct way by the equations

$$\begin{array}{rclcrcl} [\![\mathrm{N}]\!] & = & \textbf{true} & & [\![\forall x \in A.\ B]\!] & = & [\![A]\!] \Rightarrow [\![B]\!] \\ [\![\bot]\!] & = & \textbf{false} & & [\![\exists x \in A.\ B]\!] & = & [\![A]\!] \wedge [\![B]\!] \\ [\![A_1 \vee A_2]\!] & = & [\![A_1]\!] \vee [\![A_2]\!] & & [\![a =_A b]\!] & = & [\![A]\!], \end{array}$$

where N denotes the type of natural numbers and $\bot$ the empty type. The variable x may occur free in B, but since variables denote proof-objects, they will eventually be ignored by the interpretation, as in the case of equality.

Intuitively, dependent types having inhabited instances are interpreted as **true**. For example, $x =_{\mathrm{N}} y$ is interpreted as **true** because the instance $0 =_{\mathrm{N}} 0$ is inhabited. The surprising consequence of this model is that it is impossible to have a dependent type B such that both B and its negation (defined in terms of $\forall$ and $\bot$) have inhabited instances. Therefore, since $x =_{\mathrm{N}} y$ has an inhabited instance, its negation does not, which means that Peano's fourth axiom is not provable.

In more recent formulations of type theory (Nordström *et al.* 1990) Martin-Löf put forward a *logical framework* in which there is a type called Set whose objects represent types. This implies a complete revision of the interpretation above, in which free variables were ignored. Variables for objects of type Set do not represent proofs, they represent types that are relevant and will not be ignored. Types like Set, whose objects are relevant, and like N, whose objects are not, can be combined to generate a variety of types whose objects will have different degrees of relevance.

In Martin-Löf (1992), Martin-Löf gave a very precise formulation of the logical framework, the *theory of types with explicit substitution*, which was fully described in Tasistro (1997). Contexts and substitutions, which were treated informally in the previous formulations, are now defined by the rules of the formal system. Substitutions are explicit rather than metaoperations on the expressions. Section 2 is a summary of this system.

This is the formulation of the logical framework for which we define the proof-irrelevant model here. The high level of detail of the formulation of the system makes our proofs tedious but it also makes it possible to ensure the correctness of the model without relying on informal assumptions about the formal system. A merit of our model is that it is defined for the exact formulation in Martin-Löf (1992) without making compromises such as replacing abstraction *à la* Curry by abstraction *à la* Church, and that proof-irrelevance is achieved in a non-trivial and rather subtle way (see Section 5 for further discussion of this).

In order to define the proof-irrelevant model, Section 3 first presents an abstract

construction in terms of an arbitrary extensional model of the untyped lambda calculus. The desired model is then obtained by specialising this construction.

The model thus defined can be mechanically extended to interpret inductive definitions in the style of Dybjer (Dybjer 1994). In doing so, a variant of Church numerals is used to validate in the model the equality rules with which such inductive definitions are equipped. This is illustrated through examples in Section 4.

In the literature, models of type theory are often defined by interpreting only well-typed expressions. This is obtained by requiring the model to have a rather elaborate structure, sometimes as complex as that of type theory itself. See, for instance, Hofmann (1997) and Dybjer (1996), and Martin-Löf's Tarski-style semantics of type theory (Gaspes 1997). In contrast to this, the structure of our model is simple, each syntactic category is interpreted separately and the interpretation is defined for all the expressions, even those that are ill typed. Miquel's model (Miquel 2000) has this in common with our model but, aiming as it does to interpret impredicativity, universes, intersection types and subtyping, it involves complex constructions in terms of coherence spaces, stable functions and inaccessible cardinals. In contrast to this, our model is built elementarily from the notion of an extensional model of the lambda calculus.

2. Type theory

We give a concise presentation of the formulation of type theory of Martin-Löf (1992). It consists of a large number of forms of judgment and inference rules. The former are displayed in Figure 1; the latter, in Figures 4 to 13. See Tasistro (1997) for complete explanations of the intuitive meaning of the forms of judgment, and for justifications of the rules of inference.

Γ context	Γ is a context
$\Gamma \preceq \Delta$	Γ is a subcontext of Δ
α type $[\Gamma]$	α is a type under Γ
$\alpha_1 = \alpha_2$ type $[\Gamma]$	α_1 and α_2 are equal types under Γ
$a : \alpha$ $[\Gamma]$	a is an object of type α under Γ
$a_1 = a_2 : \alpha$ $[\Gamma]$	a_1 and a_2 are equal objects of type α under Γ
β : $\alpha \rightarrow$ type $[\Gamma]$	β is a family of types over α under Γ
$\beta_1 = \beta_2$: $\alpha \rightarrow$ type $[\Gamma]$	β_1 and β_2 are equal families of types over α under Γ
δ : $\Gamma \rightarrow \Delta$	δ is a substitution for Δ under Γ
$\delta_1 = \delta_2$: $\Gamma \rightarrow \Delta$	δ_1 and δ_2 are equal substitutions for Δ under Γ

Fig. 1. Forms of judgment

The abbreviations that we use for the forms of judgment are listed in Figure 1. A context Γ declares variables that can then be used when making judgments under Γ. A substitution δ for Δ under Γ assigns well-typed objects under Γ to the variables declared in Δ. A family of types β over α is a type indexed by the objects of α.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

There are expressions of five different syntactic categories: $\mathscr{C}$, $\mathscr{T}$, $\mathscr{O}$, $\mathscr{F}$ and $\mathscr{S}$. As is customary, we call expressions in those categories *pre*context, *pre*type, *pre*object, *pre*family or *pre*substitution, respectively, unless there is a derivation of the corresponding judgment proving that the expression in question is indeed a context, type, object, family of types or substitution. Figure 2 establishes the convention of names for meta-variables ranging over expressions in each of the syntactic categories. In addition to that, x, y, $x_1, x_2, \ldots$, range over an infinite set $\mathscr{V}$ of variables, though in Section 4, for convenience, we name variables more liberally.

precontexts:	$\Gamma, \Delta, \Theta, \Phi \in \mathscr{C}$
pretypes:	$\alpha, \alpha_1, \alpha_2 \in \mathscr{T}$
preobjects:	$a, a_1, a_2, f, g \in \mathscr{O}$
prefamilies:	$\beta, \beta_1, \beta_2 \in \mathscr{F}$
presubstitutions:	$\delta, \delta_1, \delta_2, \gamma, \theta \in \mathscr{S}$

Fig. 2. Name conventions

Γ	:=	$() \mid (\Gamma, x : \alpha)$
α	:=	$\mathrm{Set} \mid \beta\, a \mid \alpha \to \beta \mid \alpha\delta$
a	:=	$x \mid a_1\, a_2 \mid [x]a \mid a\delta$
β	:=	$\mathrm{El} \mid [x]\alpha \mid \beta\delta$
δ	:=	$() \mid (\delta, x{=}a) \mid \delta_1\delta_2$

Fig. 3. Abstract syntax

Figure 3 describes the expressions by means of a grammar. Precontexts are sequences of variable declarations. Prefamilies are expressions denoting functions that return pretypes. They can be built up by abstracting a variable from a pretype ($[x]\alpha$) or by substituting in a prefamily ($\beta\delta$). Pretypes can be obtained by applying a prefamily to a preobject ($\beta\, a$), by forming the pretype of dependent functions ($\alpha \to \beta$) or by substituting in a pretype ($\alpha\delta$). Presubstitutions are built up from the empty substitution by extending with assignments of preobjects to variables, and by composing with other presubstitutions ($\delta_1\delta_2$). Preobjects are lambda terms with explicit substitutions. They are built up from variables by application ($a_1\, a_2$), abstraction ($[x]a$) and substitution ($a\delta$).

In addition, there are a basic pretype Set and a basic prefamily El. The former denotes the type whose objects are inductively defined sets. The latter denotes the following family of types indexed by the objects of Set. For each set given as an object a in Set, $\mathrm{El}\, a$ is the type whose objects are the elements of the set a. For example, the set N of natural numbers will have type Set and every natural number will have type $\mathrm{El}\, \mathrm{N}$.

Observe that all prefamilies have arity 1 since El has arity 1 and in the prefamily $[x]\alpha$, α is a pretype. Notice also that substitution is not assumed as a meta-operation on expressions but rather presubstitutions are explicit in the syntax and they are to be manipulated by the rules of inference exclusively.

The symbol '$\to$' is being used in two very different ways. In Figure 3 it is used for building pretypes. In Figure 1, however, it has exactly the same status as words like context and type, it is only used for the sake of brevity. There are no such things as type, $\alpha \to \mathsf{type}$ or $\Gamma \to \Delta$ in the abstract syntax.

In the pretype $\alpha \to \beta$, β is a prefamily rather than a pretype. Intuitively, an object of type $\alpha \to \beta$ will be a function f such that $f\, a$ has type $\beta\, a$ for every object a of type α. Such an f is a dependent function, the type of its result depends on its argument. For example, any object of type Set $\to$ El is a dependent function, a function that, when applied to a set, returns an element of that set. That is, a choice function.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$$1: \frac{}{() \text{ context}} \qquad 2: \frac{\Gamma \text{ context} \quad \alpha \text{ type } [\Gamma]}{(\Gamma, x:\alpha) \text{ context}} \ (x \notin \Gamma)$$

Fig. 4. Rules for contexts

$$1: \frac{\Gamma \text{ context}}{x:\alpha\ [\Gamma]} \ (x:\alpha \in \Gamma) \qquad 2: \frac{\Gamma \preceq \Delta \quad a:\alpha\ [\Gamma]}{a:\alpha\ [\Delta]}$$

$$3: \frac{\alpha_1{=}\alpha_2 \text{ type } [\Gamma] \quad a:\alpha_2\ [\Gamma]}{a:\alpha_1\ [\Gamma]} \qquad 4: \frac{a:\alpha\ [\Delta] \quad \delta : \Gamma \to \Delta}{a\delta:\alpha\delta\ [\Gamma]}$$

$$5: \frac{f:\alpha\to\beta\ [\Gamma] \quad a:\alpha\ [\Gamma]}{f\,a:\beta\,a\ [\Gamma]} \qquad 6: \frac{a:\alpha_1\ [(\Gamma, x:\alpha_2)]}{[x]a:\alpha_2\to[x]\alpha_1\ [\Gamma]} \ (x \notin \Gamma)$$

Fig. 5. Rules for objects

$$1: \frac{\Gamma \text{ context}}{() : \Gamma \to \Gamma} \qquad 2: \frac{\delta : \Gamma \to \Delta \quad a:\alpha\delta\ [\Gamma]}{(\delta, x{=}a) : \Gamma \to (\Delta, x:\alpha)} \ (x \notin \Delta)$$

$$3: \frac{\theta : \Delta \to \Theta \quad \delta : \Gamma \to \Delta}{\theta\delta : \Gamma \to \Theta} \qquad 4: \frac{\delta : \Gamma \to \Delta \quad \Gamma \preceq \Theta}{\delta : \Theta \to \Delta} \qquad 5: \frac{\delta : \Gamma \to \Delta \quad \Theta \preceq \Delta}{\delta : \Gamma \to \Theta}$$

Fig. 6. Rules for substitutions

$$1: \frac{}{\text{Set type } [()]} \qquad 2: \frac{\Gamma \preceq \Delta \quad \alpha \text{ type } [\Gamma]}{\alpha \text{ type } [\Delta]} \qquad 3: \frac{\alpha \text{ type } [\Delta] \quad \delta : \Gamma \to \Delta}{\alpha\delta \text{ type } [\Gamma]}$$

$$4: \frac{\beta : \alpha \to \text{type } [\Gamma] \quad a:\alpha\ [\Gamma]}{\beta\,a \text{ type } [\Gamma]} \qquad 5: \frac{\alpha \text{ type } [\Gamma] \quad \beta : \alpha \to \text{type } [\Gamma]}{\alpha \to \beta \text{ type } [\Gamma]}$$

Fig. 7. Rules for types

$$1: \frac{}{\text{El} : \text{Set} \to \text{type } [()]} \qquad 2: \frac{\Gamma \preceq \Delta \quad \beta : \alpha \to \text{type } [\Gamma]}{\beta : \alpha \to \text{type } [\Delta]}$$

$$3: \frac{\alpha_1{=}\alpha_2 \text{ type } [\Gamma] \quad \beta : \alpha_2 \to \text{type } [\Gamma]}{\beta : \alpha_1 \to \text{type } [\Gamma]} \qquad 4: \frac{\beta : \alpha \to \text{type } [\Delta] \quad \delta : \Gamma \to \Delta}{\beta\delta : \alpha\delta \to \text{type } [\Gamma]}$$

$$5: \frac{\alpha_1 \text{ type } [(\Gamma, x:\alpha_2)]}{[x]\alpha_1 : \alpha_2 \to \text{type } [\Gamma]} \ (x \notin \Gamma)$$

Fig. 8. Rules for families

$$1: \frac{\Delta \text{ context}}{() \preceq \Delta} \qquad 2: \frac{\Gamma \preceq \Delta}{(\Gamma, x:\alpha) \preceq \Delta} \ (x:\alpha \in \Delta, x \notin \Gamma)$$

Fig. 9. Rules for subcontexts

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$$1: \frac{\Gamma \preceq \Delta \quad a_1{=}a_2 : \alpha\ [\Gamma]}{a_1{=}a_2 : \alpha\ [\Delta]} \qquad 2: \frac{\alpha_1{=}\alpha_2\ \mathsf{type}\ [\Gamma] \quad a_1{=}a_2 : \alpha_2\ [\Gamma]}{a_1{=}a_2 : \alpha_1\ [\Gamma]}$$

$$3: \frac{a_1{=}a_2 : \alpha\ [\Delta] \quad \delta_1{=}\delta_2 \ :\ \Gamma \to \Delta}{a_1\delta_1{=}a_2\delta_2 : \alpha\delta_1\ [\Gamma]}$$

$$4: \frac{a : \alpha\ [\Delta] \quad \delta \ :\ \Gamma \to \Delta \quad \gamma \ :\ \Phi \to \Gamma}{(a\delta)\gamma{=}a(\delta\gamma) : \alpha(\delta\gamma)\ [\Phi]} \qquad 5: \frac{a : \alpha\ [\Gamma]}{a(){=}a : \alpha\ [\Gamma]}$$

$$6: \frac{\delta \ :\ \Gamma \to \Delta \quad a : \alpha\delta\ [\Gamma]}{x(\delta, x{=}a){=}a : \alpha\delta\ [\Gamma]}\ (x \notin \Delta) \qquad 7: \frac{f{=}g : \alpha \to \beta\ [\Gamma] \quad a_1{=}a_2 : \alpha\ [\Gamma]}{f\ a_1{=}g\ a_2 : \beta\ a_1\ [\Gamma]}$$

$$8: \frac{f : \alpha \to \beta\ [\Gamma] \quad a : \alpha\ [\Gamma] \quad \gamma \ :\ \Phi \to \Gamma}{(f\ a)\gamma{=}(f\gamma)\,(a\gamma) : (\beta\ a)\gamma\ [\Phi]}$$

$$9: \frac{a : \alpha_1\ [(\Gamma, x : \alpha_2)] \quad \gamma \ :\ \Phi \to \Gamma \quad a' : \alpha_2\gamma\ [\Phi]}{(([x]a)\gamma)\ a'{=}a(\gamma, x{=}a') : \alpha_1(\gamma, x{=}a')\ [\Phi]}\ (x \notin \Gamma)$$

$$10: \frac{f\ x{=}g\ x : \beta\ x\ [(\Gamma, x : \alpha)]}{f{=}g : \alpha \to \beta\ [\Gamma]}\ (x \notin \Gamma) \qquad \text{reflexivity, symmetry and transitivity}$$

Fig. 10. Rules for equal objects

$$1: \frac{\delta_1 \ :\ \Gamma \to () \quad \delta_2 \ :\ \Gamma \to ()}{\delta_1{=}\delta_2 \ :\ \Gamma \to ()} \qquad 2: \frac{\delta_1{=}\delta_2 \ :\ \Gamma \to \Delta \quad x\delta_1{=}x\delta_2 : \alpha\delta_1\ [\Gamma]}{\delta_1{=}\delta_2 \ :\ \Gamma \to (\Delta, x : \alpha)}\ (x \notin \Delta)$$

$$3: \frac{\delta_1{=}\delta_2 \ :\ \Gamma \to \Delta \quad \Gamma \preceq \Theta}{\delta_1{=}\delta_2 \ :\ \Theta \to \Delta} \qquad 4: \frac{\delta_1{=}\delta_2 \ :\ \Gamma \to \Delta \quad \Theta \preceq \Delta}{\delta_1{=}\delta_2 \ :\ \Gamma \to \Theta}$$

$$5: \frac{\theta_1{=}\theta_2 \ :\ \Delta \to \Theta \quad \delta_1{=}\delta_2 \ :\ \Gamma \to \Delta}{\theta_1\delta_1{=}\theta_2\delta_2 \ :\ \Gamma \to \Theta}$$

$$6: \frac{\theta \ :\ \Delta \to \Theta \quad \delta \ :\ \Gamma \to \Delta \quad \gamma \ :\ \Phi \to \Gamma}{(\theta\delta)\gamma{=}\theta(\delta\gamma) \ :\ \Phi \to \Theta}$$

$$7: \frac{\delta \ :\ \Gamma \to \Delta}{\delta(){=}\delta \ :\ \Gamma \to \Delta} \qquad 8: \frac{\delta \ :\ \Gamma \to \Delta}{()\delta{=}\delta \ :\ \Gamma \to \Delta}$$

$$9: \frac{\delta \ :\ \Gamma \to \Delta \quad a : \alpha\delta\ [\Gamma] \quad \gamma \ :\ \Phi \to \Gamma}{(\delta, x{=}a)\gamma{=}(\delta\gamma, x{=}a\gamma) \ :\ \Phi \to (\Delta, x : \alpha)}\ (x \notin \Delta)$$

$$10: \frac{\delta \ :\ \Gamma \to \Delta \quad a : \alpha\delta\ [\Gamma]}{(\delta, x{=}a){=}\delta \ :\ \Gamma \to \Delta}\ (x \notin \Delta) \qquad \text{reflexivity, symmetry and transitivity}$$

Fig. 11. Rules for equal substitutions

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$$1: \frac{\Gamma \preceq \Delta \quad \alpha_1{=}\alpha_2 \text{ type } [\Gamma]}{\alpha_1{=}\alpha_2 \text{ type } [\Delta]} \qquad 2: \frac{\alpha_1{=}\alpha_2 \text{ type } [\Delta] \quad \delta_1{=}\delta_2 \ : \ \Gamma \to \Delta}{\alpha_1\delta_1{=}\alpha_2\delta_2 \text{ type } [\Gamma]}$$

$$3: \frac{\alpha \text{ type } [\Delta] \quad \delta \ : \ \Gamma \to \Delta \quad \gamma \ : \ \Phi \to \Gamma}{(\alpha\delta)\gamma{=}\alpha(\delta\gamma) \text{ type } [\Phi]} \qquad 4: \frac{\alpha \text{ type } [\Gamma]}{\alpha(){=}\alpha \text{ type } [\Gamma]}$$

$$5: \frac{\beta_1{=}\beta_2 \ : \ \alpha \to \text{type } [\Gamma] \quad a_1{=}a_2 : \alpha \ [\Gamma]}{\beta_1\, a_1{=}\beta_2\, a_2 \text{ type } [\Gamma]}$$

$$6: \frac{\beta \ : \ \alpha \to \text{type } [\Gamma] \quad a : \alpha \ [\Gamma] \quad \gamma \ : \ \Phi \to \Gamma}{(\beta\, a)\gamma{=}(\beta\gamma)\,(a\gamma) \text{ type } [\Phi]}$$

$$7: \frac{\alpha_1 \text{ type } [(\Gamma, x : \alpha_2)] \quad \gamma \ : \ \Phi \to \Gamma \quad a : \alpha_2\gamma \ [\Phi]}{(([x]\alpha_1)\gamma)\, a{=}\alpha_1(\gamma, x{=}a) \text{ type } [\Phi]} \ (x \notin \Gamma)$$

$$8: \frac{\alpha_1{=}\alpha_2 \text{ type } [\Gamma] \quad \beta_1{=}\beta_2 \ : \ \alpha_1 \to \text{type } [\Gamma]}{\alpha_1 \to \beta_1{=}\alpha_2 \to \beta_2 \text{ type } [\Gamma]}$$

$$9: \frac{\alpha \text{ type } [\Gamma] \quad \beta \ : \ \alpha \to \text{type } [\Gamma] \quad \gamma \ : \ \Phi \to \Gamma}{(\alpha \to \beta)\gamma{=}\alpha\gamma \to \beta\gamma \text{ type } [\Phi]}$$

reflexivity, symmetry and transitivity

Fig. 12. Rules for equal types

$$1: \frac{\Gamma \preceq \Delta \quad \beta_1{=}\beta_2 \ : \ \alpha \to \text{type } [\Gamma]}{\beta_1{=}\beta_2 \ : \ \alpha \to \text{type } [\Delta]}$$

$$2: \frac{\alpha_1{=}\alpha_2 \text{ type } [\Gamma] \quad \beta_1{=}\beta_2 \ : \ \alpha_2 \to \text{type } [\Gamma]}{\beta_1{=}\beta_2 \ : \ \alpha_1 \to \text{type } [\Gamma]}$$

$$3: \frac{\beta_1{=}\beta_2 \ : \ \alpha \to \text{type } [\Delta] \quad \delta_1{=}\delta_2 \ : \ \Gamma \to \Delta}{\beta_1\delta_1{=}\beta_2\delta_2 \ : \ \alpha\delta_1 \to \text{type } [\Gamma]}$$

$$4: \frac{\beta \ : \ \alpha \to \text{type } [\Delta] \quad \delta \ : \ \Gamma \to \Delta \quad \gamma \ : \ \Phi \to \Gamma}{(\beta\delta)\gamma{=}\beta(\delta\gamma) \ : \ \alpha(\delta\gamma) \to \text{type } [\Phi]}$$

$$5: \frac{\beta \ : \ \alpha \to \text{type } [\Gamma]}{\beta(){=}\beta \ : \ \alpha \to \text{type } [\Gamma]} \qquad 6: \frac{\beta_1\, x{=}\beta_2\, x \text{ type } [(\Gamma, x : \alpha)]}{\beta_1{=}\beta_2 \ : \ \alpha \to \text{type } [\Gamma]} \ (x \notin \Gamma)$$

reflexivity, symmetry and transitivity

Fig. 13. Rules for equal families

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

We write $x : \alpha \in \Gamma$ for '$x : \alpha$ is among the variable declarations of Γ', and, also, $x \in \Gamma$ for '$x : \alpha \in \Gamma$ for some α'. Since we are only concerned with formal aspects of type theory, here we will omit explanations of the rules of inference displayed in Figures 4 to 13. The omitted justifications can be found in Tasistro (1997). For brevity, we also omit obvious premises, for example the premises Γ context, Δ context and α type $[\Gamma]$ are implicit in Rule 2 of Figure 5.

3. The model

If we use Λ to denote the set of untyped lambda terms (writing $[x]a$ for abstraction), we have $\Lambda \subset \mathscr{O}$. We show in this section that every extensional model of Λ can be extended to obtain an interpretation of $\mathscr{O}$, $\mathscr{S}$, $\mathscr{T}$, $\mathscr{F}$ and $\mathscr{C}$. Finally, we prove that the interpretation is indeed a model of type theory.

There are several different notions of models of the lambda calculus in the literature (Barendregt 1984; Hindley and Seldin 1986). For our purposes, it is enough to consider extensional models. Observe that the symbol '$\rightarrow$' is also used in the metalanguage to build the standard set-theoretic function space between two sets.

Definition 3.1. Given a set $\mathscr{D}$, a valuation is a function $\rho : \mathscr{V} \rightarrow \mathscr{D}$. The set of all the valuations is denoted by **Val**. An extensional model of the lambda calculus is a triple $(\mathscr{D}, \cdot, [\![\,]\!])$ where $\mathscr{D}$ is a set with at least two elements, '$\cdot$' is a binary operation over $\mathscr{D}$, and $[\![\,]\!]$ is an interpretation function $[\![\,]\!] : \Lambda \rightarrow \mathbf{Val} \rightarrow \mathscr{D}$ such that the following hold:

1 if $d_1 \cdot e = d_2 \cdot e$ for all $e \in \mathscr{D}$, then $d_1 = d_2$, (extensionality)
2 $[\![x]\!]_\rho = \rho(x)$,
3 $[\![a_1\, a_2]\!]_\rho = [\![a_1]\!]_\rho \cdot [\![a_2]\!]_\rho$,
4 $[\![[x]a]\!]_\rho \cdot d = [\![a]\!]_{(\rho, x=d)}$,

where $(\rho, x{=}d)$ is the valuation that assigns d to x and $\rho(y)$ to all $y \neq x$.

The set Λ with α, β and η rules is an extensional model of the lambda calculus. That is, taking $\mathscr{D} = \Lambda$ and $\cdot$ to be application, it is possible to define an interpretation function $[\![\,]\!]$ satisfying the conditions in the definition above. This is an essentially trivial statement but it is cumbersome to prove, since a precise definition of $[\![\,]\!]$ requires careful considerations regarding substitutions and names of variables. The proof that $\mathscr{D}$ has at least two elements, however, relies on the Church–Rosser property of the lambda calculus.

We assume hereafter that an extensional model $(\mathscr{D}, \cdot, [\![\,]\!])$ of the lambda calculus is given. We use $c, d, e, c_1, d_1, e_1, c_2, d_2, e_2, \ldots$ to denote elements of $\mathscr{D}$, and ρ, ρ_1 and ρ_2 to denote elements of **Val**. The following proposition says that the only relevant information provided by ρ when interpreting a lambda term a are the values that it assigns to the free variables of a. The set of free variables of a is denoted by $\mathscr{V}(a)$.

Proposition 3.2. Given $a \in \Lambda$ and $\rho_1, \rho_2 \in \mathbf{Val}$, if for all $x \in \mathscr{V}(a)$, $\rho_1(x) = \rho_2(x)$, then $[\![a]\!]_{\rho_1} = [\![a]\!]_{\rho_2}$.

Proof. The proof is by induction on the syntax of lambda terms. The cases in which the lambda term is a variable or an application are easy. When it is an abstraction, the proof follows by extensionality. □

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$[\![a]\!] : \mathbf{Val} \to \mathscr{D}$	$[\![\alpha]\!] : \mathbf{Val} \to Pow(\mathscr{D})$	$[\![\Gamma]\!] \in Pow(\mathbf{Val})$
$[\![\delta]\!] : \mathbf{Val} \to \mathbf{Val}$	$[\![\beta]\!] : \mathbf{Val} \to \mathscr{D} \to Pow(\mathscr{D})$	$\mathbf{Val} = \mathscr{V} \to \mathscr{D}$

Fig. 14. Semantic categories

$$
\begin{array}{lll}
[\![a\delta]\!]_\rho = [\![a]\!]_{[\![\delta]\!]_\rho} & [\![\mathrm{Set}]\!]_\rho = \mathbf{Set} & [\![\mathrm{El}]\!]_\rho = \mathbf{El} \\
 & [\![\beta\, a]\!]_\rho = [\![\beta]\!]_\rho([\![a]\!]_\rho) & [\![[x]\alpha]\!]_\rho = d \mapsto [\![\alpha]\!]_{(\rho, x=d)} \\
[\![()]\!]_\rho = \rho & [\![\alpha \to \beta]\!]_\rho = [\![\alpha]\!]_\rho \overset{\mathscr{D}}{\to} [\![\beta]\!]_\rho & [\![\beta\delta]\!]_\rho = [\![\beta]\!]_{[\![\delta]\!]_\rho} \\
[\![\theta\delta]\!]_\rho = [\![\theta]\!]_{[\![\delta]\!]_\rho} & [\![\alpha\delta]\!]_\rho = [\![\alpha]\!]_{[\![\delta]\!]_\rho} & [\![()]\!] = \mathbf{Val} \\
[\![(\delta, x{=}a)]\!]_\rho = ([\![\delta]\!]_\rho, x{=}[\![a]\!]_\rho) & & [\![(\Gamma, x : \alpha)]\!] = \{\rho \in [\![\Gamma]\!] \mid \rho(x) \in [\![\alpha]\!]_\rho\}
\end{array}
$$

Fig. 15. Interpretation of elements of $\mathscr{O}$, $\mathscr{S}$, $\mathscr{T}$, $\mathscr{F}$ and $\mathscr{C}$

3.1. *Interpreting $\mathscr{O}$, $\mathscr{S}$, $\mathscr{T}$, $\mathscr{F}$ and $\mathscr{C}$*

It is possible to extend the interpretation function to the whole of $\mathscr{O}$, yielding a function $[\![a]\!] : \mathbf{Val} \to \mathscr{D}$. This involves interpreting presubstitutions as well, since they occur in preobjects. We will also give interpretations to pretypes, prefamilies and precontexts. The semantic categories are summarised in Figure 14 and the equations defining the interpretation in Figure 15.

The interpretation of $a\delta$ in ρ is the interpretation of a in the new valuation $[\![\delta]\!]_\rho$; and similarly for $\theta\delta$, $\alpha\delta$ and $\beta\delta$. The notation '$(,)$' denotes different things on the left-hand and right-hand sides of the equation for $(\delta, x{=}a)$, namely the extension of a presubstitution and a valuation, respectively.

Intuitively, pretypes are collections of preobjects, and prefamilies, functions mapping preobjects to pretypes. We realise this by interpreting – in a given valuation – pretypes as subsets of $\mathscr{D}$, and prefamilies as functions mapping elements in $\mathscr{D}$ to such subsets.

We do not need to make **Set** and **El** precise at this stage. It is enough to assume that the former is a subset of $\mathscr{D}$ and the latter a function from $\mathscr{D}$ to $Pow(\mathscr{D})$. In the equation defining $[\![\beta\, a]\!]_\rho$, $[\![\beta]\!]_\rho([\![a]\!]_\rho)$ denotes the application (in the metalanguage) of the function $[\![\beta]\!]_\rho$ to the element $[\![a]\!]_\rho$ of $\mathscr{D}$. For $[\![\alpha \to \beta]\!]_\rho$ we define for all subsets $\mathscr{X}$ of $\mathscr{D}$ and all functions $\mathscr{Y}$ from $\mathscr{D}$ to $Pow(\mathscr{D})$, the subset of $\mathscr{D}$

$$\mathscr{X} \overset{\mathscr{D}}{\to} \mathscr{Y} = \{d \in \mathscr{D} \mid \forall e \in \mathscr{X}.\ d \cdot e \in \mathscr{Y}(e)\},$$

which expresses type dependency in the interpretation. For $[\![[x]\alpha]\!]_\rho$ we take the function that maps d into the set $[\![\alpha]\!]_{(\rho, x=d)}$.

Precontexts are used to declare variables denoting arbitrary preobjects of certain pretypes. We realise this by interpreting a precontext Γ as the set of all the valuations that respect Γ's assignments of pretypes to variables. That is, $[\![\Gamma]\!] \subseteq \mathbf{Val}$ is the set of those valuations ρ that for all $x : \alpha \in \Gamma$ assign an element of the set $[\![\alpha]\!]_\rho$ to x.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

3.2. *Preliminary properties*

For each Γ we define an equivalence relation: given ρ_1 and ρ_2, we write $\rho_1 =_\Gamma \rho_2$ for $\forall x \in \Gamma.\ \rho_1(x) = \rho_2(x)$. The following propositions can be proved easily by induction on the syntax of precontexts: we omit the proofs here.

Proposition 3.3. For all Γ, ρ, d, x, if $x \notin \Gamma$, then $(\rho, x{=}d) =_\Gamma \rho$.

Proposition 3.4. For all Γ, ρ, x, α, if $x : \alpha \in \Gamma$ and $\rho \in [\![\Gamma]\!]$, then $\rho(x) \in [\![\alpha]\!]_\rho$.

3.3. *Statement of the model*

The following theorem states that the interpretation defined above is a model of type theory. There is a remarkable similarity between the statement of the theorem and Martin-Löf's explanation of the meaning of the forms of judgment in Tasistro (1997).

Theorem 3.5. For all Γ, Δ, α, α_1, α_2, a, a_1, a_2, β, β_1, β_2, δ, δ_1, δ_2, the following hold:

1 If Γ context is derivable, then for all $\rho_1, \rho_2 \in$ **Val** such that $\rho_1 =_\Gamma \rho_2$, if $\rho_2 \in [\![\Gamma]\!]$, then $\rho_1 \in [\![\Gamma]\!]$.
2 If $\Gamma \preceq \Delta$ is derivable, then:
   (a) $[\![\Delta]\!] \subseteq [\![\Gamma]\!]$.
   (b) for all $\rho_1, \rho_2 \in$ **Val**, if $\rho_1 =_\Delta \rho_2$, then $\rho_1 =_\Gamma \rho_2$.
3 If α type $[\Gamma]$ is derivable, then for all $\rho_1, \rho_2 \in$ **Val**, if $\rho_1 =_\Gamma \rho_2$, then $[\![\alpha]\!]_{\rho_1} = [\![\alpha]\!]_{\rho_2}$.
4 If $\alpha_1 = \alpha_2$ type $[\Gamma]$ is derivable, then for all $\rho \in$ **Val**, $[\![\alpha_1]\!]_\rho = [\![\alpha_2]\!]_\rho$.
5 If $a : \alpha$ $[\Gamma]$ is derivable, then:
   (a) for all $\rho \in [\![\Gamma]\!]$, $[\![a]\!]_\rho \in [\![\alpha]\!]_\rho$.
   (b) for all $\rho_1, \rho_2 \in$ **Val**, if $\rho_1 =_\Gamma \rho_2$, then $[\![a]\!]_{\rho_1} = [\![a]\!]_{\rho_2}$.
6 If $a_1 = a_2 : \alpha$ $[\Gamma]$ is derivable, then for all $\rho \in$ **Val**, $[\![a_1]\!]_\rho = [\![a_2]\!]_\rho$.
7 If $\beta : \alpha \to$ type $[\Gamma]$ is derivable, then for all $\rho_1, \rho_2 \in$ **Val**, if $\rho_1 =_\Gamma \rho_2$, then $[\![\beta]\!]_{\rho_1} = [\![\beta]\!]_{\rho_2}$.
8 If $\beta_1 = \beta_2 : \alpha \to$ type $[\Gamma]$ is derivable, then for all $\rho \in$ **Val**, $[\![\beta_1]\!]_\rho = [\![\beta_2]\!]_\rho$.
9 If $\delta : \Gamma \to \Delta$ is derivable, then:
   (a) for all $\rho \in [\![\Gamma]\!]$, $[\![\delta]\!]_\rho \in [\![\Delta]\!]$.
   (b) for all $\rho_1, \rho_2 \in$ **Val**, if $\rho_1 =_\Gamma \rho_2$, then $[\![\delta]\!]_{\rho_1} =_\Delta [\![\delta]\!]_{\rho_2}$.
10 If $\delta_1 = \delta_2 : \Gamma \to \Delta$ is derivable, then for all $\rho \in$ **Val**, $[\![\delta_1]\!]_\rho =_\Delta [\![\delta_2]\!]_\rho$.

A priori we might have to prove all the statements in the formulation of Theorem 3.5 simultaneously by induction on the derivations, considering all the rules of the calculus collected in Figures 4 to 13. Fortunately, it is possible to separate them into different small groups of statements. In every group the statements are proved simultaneously by induction on the derivations, and only a few of those rules need to be considered. We make each group into a lemma, and we have to prove the lemmas in a certain order. Figure 16 describes the organisation of the proof of Theorem 3.5.

In statements 5(a) and 9(a) (and hence, in Theorem 3.15) ρ only ranges over $[\![\Gamma]\!]$. In all the remaining statements and lemmas, ρ, ρ_1 and ρ_2 range over the whole **Val**. This technical subtlety is essential to the organisation of the proof as in Figure 16.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

	Items of Theorem 3.5									
Lemma 3.6		2 (b)								
Lemma 3.7					5 (b)				9 (b)	
Lemma 3.8				3			7			
Lemma 3.9	1									
Lemma 3.12						6				10
Lemma 3.13				4			8			
Lemma 3.14		2 (a)								
Theorem 3.15					5 (a)				9 (a)	

Fig. 16. Proof of Theorem 3.5

Lemma 3.6. For all Γ, Δ, ρ_1, ρ_2, if $\Gamma \preceq \Delta$ is derivable, then $\rho_1 =_\Delta \rho_2$ implies $\rho_1 =_\Gamma \rho_2$.

Proof. The proof is by induction on the derivation of $\Gamma \preceq \Delta$. Any such derivation ends with one of the rules in Figure 9. If that is Rule 1, the lemma is trivial, since any two valuations are related in $=_{()}$. If it is Rule 2, we assume $\rho_1 =_\Delta \rho_2$. By the induction hypothesis, we get $\rho_1 =_\Gamma \rho_2$, and as $x \in \Delta$, we get $\rho_1(x) = \rho_2(x)$. Hence, $\rho_1 =_{(\Gamma, x : \alpha)} \rho_2$. □

Lemma 3.7. For all Γ, Δ, α, a, δ, ρ_1, ρ_2, if $\rho_1 =_\Gamma \rho_2$, then:

— If $a : \alpha\ [\Gamma]$ is derivable, then $[\![a]\!]_{\rho_1} = [\![a]\!]_{\rho_2}$.
— If $\delta\ :\ \Gamma \to \Delta$ is derivable, then $[\![\delta]\!]_{\rho_1} =_\Delta [\![\delta]\!]_{\rho_2}$.

Proof. The proof is by induction on the derivation. Any derivation of $a : \alpha\ [\Gamma]$ ends with one of the rules in Figure 5, and any derivation of $\delta\ :\ \Gamma \to \Delta$, with one of those in Figure 6.

If the last rule is one of those in Figure 5, then the only non-trivial case is when it is Rule 6. We prove $[\![[x]a]\!]_{\rho_1} = [\![[x]a]\!]_{\rho_2}$ by extensionality, just as in the proof of Proposition 3.2.

The remaining cases follow mechanically from the definition of the interpretation in Definition 3.1 and Figure 15, the hypothesis $\rho_1 =_\Gamma \rho_2$ and the induction hypotheses available in every particular case, as well as Lemma 3.6 in the case of Rule 2.

If the last rule applied is one of those in Figure 6, then in all the cases the proof is automatic using the definition of the interpretation, the hypothesis $\rho_1 =_\Gamma \rho_2$ and the induction hypotheses, as well as Lemma 3.6 in the case of Rules 4 and 5. □

Lemma 3.8. For all Γ, α, β, ρ_1, ρ_2, if $\rho_1 =_\Gamma \rho_2$, then:

— If α type $[\Gamma]$ is derivable, then $[\![\alpha]\!]_{\rho_1} = [\![\alpha]\!]_{\rho_2}$.
— If $\beta\ :\ \alpha \to$ type $[\Gamma]$ is derivable, then $[\![\beta]\!]_{\rho_1} = [\![\beta]\!]_{\rho_2}$.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

Proof. The proof is by induction on the derivation. Any derivation of α type $[\Gamma]$ ends with one of the rules in Figure 7 and any derivation of $\beta : \alpha \to$ type $[\Gamma]$, with one of those in Figure 8.

If the last rule applied is one of those in Figure 7, then in all the cases the proof is straightforward using the definition of the interpretation in Figure 15, the hypothesis $\rho_1 =_\Gamma \rho_2$ and the induction hypotheses at our disposal in each case, as well as Lemma 3.6 in the case of Rule 2 and Lemma 3.7 in the case of Rules 3 and 4.

If the last rule is one of those in Figure 8, then the only non-trivial case is when it concerns Rule 5. To prove that $[\![[x]\alpha_1]\!]_{\rho_1}$ and $[\![[x]\alpha_1]\!]_{\rho_2}$ are equal functions from $\mathscr{D}$ to $Pow(\mathscr{D})$, we prove that for any d the following holds

$$[\![[x]\alpha_1]\!]_{\rho_1}(d) = [\![[x]\alpha_1]\!]_{\rho_2}(d).$$

This follows from $(\rho_1, x{=}d) =_{(\Gamma, x : \alpha_2)} (\rho_2, x{=}d)$ by the definition of interpretation and the induction hypothesis.

The remaining cases follow automatically from the definition of the interpretation in Figure 15, the hypothesis $\rho_1 =_\Gamma \rho_2$ and the induction hypotheses available in every particular case, as well as Lemma 3.6 in the case of Rule 2 and Lemma 3.7 in the case of Rule 4. □

Lemma 3.9. For all Γ, ρ_1, ρ_2, if Γ context is derivable, then $\rho_1 =_\Gamma \rho_2$ and $\rho_2 \in [\![\Gamma]\!]$ imply $\rho_1 \in [\![\Gamma]\!]$.

Proof. The proof is by induction on the derivation. Any derivation of Γ context ends with one of the rules in Figure 4. If that is Rule 1, the proof is trivial. If it is Rule 2, we assume that $\rho_1 =_{(\Gamma, x : \alpha)} \rho_2$, that is, $\rho_1 =_\Gamma \rho_2$ and $\rho_1(x) = \rho_2(x)$. Assume also that $\rho_2 \in [\![(\Gamma, x : \alpha)]\!]$, that is, $\rho_2 \in [\![\Gamma]\!]$ and $\rho_2(x) \in [\![\alpha]\!]_{\rho_2}$. The former implies $\rho_1 \in [\![\Gamma]\!]$ by the induction hypothesis. Lemma 3.8 implies $[\![\alpha]\!]_{\rho_1} = [\![\alpha]\!]_{\rho_2}$. Therefore

$$\rho_1(x) = \rho_2(x) \in [\![\alpha]\!]_{\rho_2} = [\![\alpha]\!]_{\rho_1},$$

and hence $\rho_1 \in [\![(\Gamma, x : \alpha)]\!]$. □

Corollary 3.10. For all Γ, ρ, d, x, if Γ context is derivable, then $x \notin \Gamma$ and $\rho \in [\![\Gamma]\!]$ imply $(\rho, x{=}d) \in [\![\Gamma]\!]$.

Proof. The statement is an easy consequence of Lemma 3.9 and Proposition 3.3. □

Corollary 3.11. For all Γ, ρ, d, x, α, if Γ context and α type $[\Gamma]$ are derivable, then $x \notin \Gamma$, $\rho \in [\![\Gamma]\!]$ and $d \in [\![\alpha]\!]_\rho$ imply $(\rho, x{=}d) \in [\![(\Gamma, x : \alpha)]\!]$.

Proof. By Corollary 3.10, $(\rho, x{=}d) \in [\![\Gamma]\!]$. By Proposition 3.3, $(\rho, x{=}d) =_\Gamma \rho$. Then, by Lemma 3.8, $[\![\alpha]\!]_{(\rho, x=d)} = [\![\alpha]\!]_\rho$. Thus,

$$(\rho, x{=}d)(x) = d \in [\![\alpha]\!]_\rho = [\![\alpha]\!]_{(\rho, x=d)},$$

and hence $(\rho, x{=}d) \in [\![(\Gamma, x : \alpha)]\!]$. □

Lemma 3.12. For all Γ, Δ, α, a_1, a_2, δ_1, δ_2, ρ, the following hold:

— If $a_1 = a_2 : \alpha$ $[\Gamma]$ is derivable, then $[\![a_1]\!]_\rho = [\![a_2]\!]_\rho$.
— If $\delta_1 = \delta_2 : \Gamma \to \Delta$ is derivable, then $[\![\delta_1]\!]_\rho =_\Delta [\![\delta_2]\!]_\rho$.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

Proof. The proof is by induction on the derivation. Any derivation of $a_1=a_2 : \alpha\ [\Gamma]$ ends with one of the rules in Figure 10 and any derivation of $\delta_1=\delta_2\ :\ \Gamma \to \Delta$ with one of those in Figure 11.

When the derivation ends with one of the rules in Figure 10 in most cases the proof is routine using the induction hypotheses and the definition of the interpretation in Definition 3.1 and Figure 15.

The exceptions to this are the cases of Rules 3, 9 and 10. In the case of Rule 3, the induction hypothesis on the derivation of $\delta_1=\delta_2\ :\ \Gamma \to \Delta$ yields $[\![\delta_1]\!]_\rho =_\Delta [\![\delta_2]\!]_\rho$. From this and the implicit premise $a_1 : \alpha\ [\Delta]$ we get

$$[\![a_1]\!]_{[\![\delta_1]\!]_\rho} = [\![a_1]\!]_{[\![\delta_2]\!]_\rho}$$

by Lemma 3.7 applied to the preobject a_1 and the valuations $[\![\delta_1]\!]_\rho$ and $[\![\delta_2]\!]_\rho$. Finally, the induction hypothesis on the derivation of $a_1=a_2 : \alpha\ [\Delta]$ when applied to $[\![\delta_2]\!]_\rho$ yields

$$[\![a_1]\!]_{[\![\delta_2]\!]_\rho} = [\![a_2]\!]_{[\![\delta_2]\!]_\rho}.$$

When the last rule applied is Rule 9, the proof follows from

$$[\![[x]a]\!]_{[\![\gamma]\!]_\rho} \cdot [\![a']\!]_\rho = [\![a]\!]_{([\![\gamma]\!]_\rho,\, x=[\![a']\!]_\rho)},$$

which holds by Definition 3.1.

Finally, when the last rule applied is Rule 10, we want to prove $[\![f]\!]_\rho = [\![g]\!]_\rho$. By extensionality, we take an arbitrary d, and prove $[\![f]\!]_\rho \cdot d = [\![g]\!]_\rho \cdot d$. As $x \notin \Gamma$, by Proposition 3.3 $(\rho, x=d)=_\Gamma \rho$. Thus, by Lemma 3.7,

$$[\![f]\!]_{(\rho,\, x=d)} = [\![f]\!]_\rho \qquad [\![g]\!]_{(\rho,\, x=d)} = [\![g]\!]_\rho$$

since $f : \alpha \to \beta\ [\Gamma]$ and $g : \alpha \to \beta\ [\Gamma]$ are implicit premises of Rule 10. Now, by the induction hypothesis applied to $(\rho, x=d)$, we obtain $[\![f]\!]_\rho \cdot d = [\![g]\!]_\rho \cdot d$.

When the derivation ends with one of the rules in Figure 11, in most cases the proof is direct using the definition of interpretation and the induction hypotheses, as well as Lemma 3.6 in the case of Rule 4 and the definitions of $=_{()}$ and $=_{(\Delta, x\,:\,\alpha)}$ in the case of Rules 1 and 2.

If the derivation ends with Rule 5, the situation is entirely analogous to the case of Rule 3 of Figure 10. Finally, if the derivation ends with Rule 10, the proof follows from Proposition 3.3, since $x \notin \Delta$. □

Lemma 3.13. For all Γ, α_1, α_2, β_1, β_2, ρ the following hold:

— If $\alpha_1=\alpha_2$ type $[\Gamma]$ is derivable, then $[\![\alpha_1]\!]_\rho=[\![\alpha_2]\!]_\rho$.
— If $\beta_1=\beta_2\ :\ \alpha \to$ type $[\Gamma]$ is derivable, then $[\![\beta_1]\!]_\rho=[\![\beta_2]\!]_\rho$.

Proof. The proof is by induction on the derivation. Any derivation of $\alpha_1=\alpha_2$ type $[\Gamma]$ ends with one of the rules in Figure 12, and any derivation of $\beta_1=\beta_2\ :\ \alpha \to$ type $[\Gamma]$ with one of those in Figure 13.

When the last rule of the derivation is one of those in Figure 12 the proof is always direct from the definition of the interpretation in Figure 15 and the induction hypotheses, except in the cases that concern Rules 2 or 5.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

In the former, we use, in addition, Lemmas 3.12 and 3.8 on the implicit premise α_1 type $[\Delta]$, to obtain a proof analogous to the proof of the case of Rule 3 of Figure 10 in the proof of Lemma 3.12. For the latter, we use Lemma 3.12.

Most of the cases in which the last rule of the derivation is one of those in Figure 13 have a direct proof via the use of the definition of the interpretation and the induction hypotheses. Apart from the case of Rule 3, which is provable in very much the same way as the case of Rule 2 in Figure 12, the only non-trivial case is the one of Rule 6.

To prove that $[\![\beta_1]\!]_\rho = [\![\beta_2]\!]_\rho$, we take an arbitrary d and prove that $[\![\beta_1]\!]_\rho(d) = [\![\beta_2]\!]_\rho(d)$. As $x \notin \Gamma$, by Proposition 3.3, we get $(\rho, x{=}d) =_\Gamma \rho$. Thus, by Lemma 3.8,

$$[\![\beta_1]\!]_{(\rho, x=d)} = [\![\beta_1]\!]_\rho \qquad [\![\beta_2]\!]_{(\rho, x=d)} = [\![\beta_2]\!]_\rho \ ,$$

since $\beta_1 \ : \ \alpha \to \mathsf{type}\ [\Gamma]$ and $\beta_2 \ : \ \alpha \to \mathsf{type}\ [\Gamma]$ are implicit premises of Rule 6. By the induction hypothesis applied to $(\rho, x{=}d)$, we get $[\![\beta_1]\!]_\rho(d) = [\![\beta_2]\!]_\rho(d)$. □

Lemma 3.14. For all Γ, Δ, if $\Gamma \preceq \Delta$ is derivable, then $[\![\Delta]\!] \subseteq [\![\Gamma]\!]$.

Proof. The proof is by induction on the derivation. Any derivation of $\Gamma \preceq \Delta$ ends with one of the two rules in Figure 9. In the case of the first rule, the proof is trivial. For the second, take $\rho \in [\![\Delta]\!]$. By the induction hypothesis, $\rho \in [\![\Gamma]\!]$. As $x : \alpha \in \Delta$, by Proposition 3.4, $\rho(x) \in [\![\alpha]\!]_\rho$. Hence $\rho \in [\![(\Gamma, x : \alpha)]\!]$. □

Theorem 3.15. For all Γ, Δ, α, a, δ, ρ, if $\rho \in [\![\Gamma]\!]$, the following hold:

— if $a : \alpha\ [\Gamma]$ is derivable, then $[\![a]\!]_\rho \in [\![\alpha]\!]_\rho$.
— if $\delta \ : \ \Gamma \to \Delta$ is derivable, then $[\![\delta]\!]_\rho \in [\![\Delta]\!]$.

Proof. The proof is by induction on the derivation. Any derivation of $a : \alpha\ [\Gamma]$ ends with one of the rules in Figure 5 and any derivation of $\delta \ : \ \Gamma \to \Delta$, with one of those in Figure 6.

If the last rule used is one of those in Figure 5, we can analyse the different possibilities:

1: This is easy by Proposition 3.4.
2: By Lemma 3.14, $\rho \in [\![\Gamma]\!]$ and then, by the induction hypothesis, $[\![a]\!]_\rho \in [\![\alpha]\!]_\rho$.
3: By the induction hypothesis, $[\![a]\!]_\rho \in [\![\alpha_2]\!]_\rho$. Lemma 3.13 implies that $[\![\alpha_1]\!]_\rho = [\![\alpha_2]\!]_\rho$, thus $[\![a]\!]_\rho \in [\![\alpha_1]\!]_\rho$.
4: The induction hypothesis on $\delta \ : \ \Gamma \to \Delta$ yields $[\![\delta]\!]_\rho \in [\![\Delta]\!]$. By the induction hypothesis on $a : \alpha\ [\Delta]$ applied to $[\![\delta]\!]_\rho$, we get $[\![a\delta]\!]_\rho \in [\![\alpha\delta]\!]_\rho$.
5: By the induction hypotheses, we have $[\![a]\!]_\rho \in [\![\alpha]\!]_\rho$, and $[\![f]\!]_\rho \in [\![\alpha]\!]_\rho \xrightarrow{\mathscr{D}} [\![\beta]\!]_\rho$, that is, for all $d \in [\![\alpha]\!]_\rho$, $[\![f]\!]_\rho \cdot d \in [\![\beta]\!]_\rho(d)$. In particular, when d is $[\![a]\!]_\rho$, we get $[\![f\ a]\!]_\rho \in [\![\beta\ a]\!]_\rho$.
6: We have to verify that $[\![[x]a]\!]_\rho \cdot d \in [\![\alpha_1]\!]_{(\rho, x=d)}$ holds for every $d \in [\![\alpha_2]\!]_\rho$. Since, by Definition 3.1, $[\![[x]a]\!]_\rho \cdot d = [\![a]\!]_{(\rho, x=d)}$, by the induction hypothesis, it is enough to prove that $(\rho, x{=}d) \in [\![(\Gamma, x : \alpha_2)]\!]$, which follows from Corollary 3.11.

If the last rule in the derivation is one of those in Figure 6 it is always immediate using the definition of the interpretation and the induction hypotheses, as well as Lemma 3.14 in the cases of Rules 4 and 5, and Corollary 3.11 in the case of Rule 2. □

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

3.4. *Consistency*

Since by Proposition 3.2 the valuation ρ has no significance when evaluating a closed lambda term a, we are entitled to use a itself to denote $[\![a]\!]_\rho$. From now on we will use $\top$ to denote $[x][y]x$ and $\bot$ to denote $[x][y]y$. These are different elements of $\mathscr{D}$, since otherwise we would have that $d_1 = \top \cdot d_1 \cdot d_2 = \bot \cdot d_1 \cdot d_2 = d_2$ for all $d_1, d_2 \in \mathscr{D}$. But Definition 3.1 does not allow $\mathscr{D}$ to be a singleton.

To obtain a proof-irrelevant model of the logical framework, we only need to define **Set** and **El** since their definition was postponed from Section 3.1. Therefore, we define:

$$\begin{array}{rcl} \mathbf{Set} & = & \{\top, \bot\} \\ \mathbf{El} & = & d \mapsto \{e \in \mathscr{D} \mid d = \top\}. \end{array}$$

Observe that $\mathbf{El}(\top) = \mathscr{D}$ and $\mathbf{El}(\bot) = \{\}$. Observe also that for all d and e the following equivalences hold:

$$e \in \mathbf{El}(d) \iff \mathbf{El}(d) \text{ is inhabited} \iff \mathbf{El}(d) = \mathscr{D} \iff d = \top.$$

This observation will be generalised by Remark 4.1.

An immediate consequence of the definitions of **Set** and **El** is that the logical framework is consistent, that is, that there is no preobject a such that

$$a : \mathrm{Set} \to \mathrm{El}\ [()]$$

is derivable. By Theorem 3.15, it is enough to show that for every environment ρ, the interpretation of $\mathrm{Set} \to \mathrm{El}$ in ρ is $\{\}$:

$$[\![\mathrm{Set} \to \mathrm{El}]\!]_\rho = \mathbf{Set} \xrightarrow{\mathscr{D}} \mathbf{El} = \{d \in \mathscr{D} \mid \forall e \in \mathbf{Set}.\ d \cdot e \in \mathbf{El}(e)\} = \{\},$$

since, if there were such a d, we would have $d \cdot \bot \in \mathbf{El}(\bot) = \{\}$, which is impossible.

3.5. *Proof irrelevance*

With these definitions of **Set** and **El** we achieve proof-irrelevance. By this we mean that the interpretation of a family of sets f over a given set a ($a : \mathrm{Set}$ and $f : \mathrm{El}\ a \to [x]\mathrm{Set}$) will not actually vary with the interpretation of the elements of the set a. On the contrary, the interpretation of f in the model represents a constant function.

In effect, a being a set, it would be interpreted as a $d_1 \in \mathbf{Set}$. The family f would be interpreted as a d such that for all $e_1 \in \mathbf{El}(d_1)$, $d \cdot e_1 \in \mathbf{Set}$. Corollary 3.19 below says that such a d represents a constant function. First we introduce the notion of **Set**-sequences.

Definition 3.16. Given $n \geqslant 0$ and $d_1, \ldots, d_n$, a sequence $e_1, \ldots, e_n$ is an **El**-sequence for $d_1, \ldots, d_n$ if $e_1 \in \mathbf{El}(d_1), \ldots, e_n \in \mathbf{El}(d_n \cdot e_1 \cdot \ldots \cdot e_{n-1})$. A sequence $d_1, \ldots, d_n$ is a **Set**-sequence if for all $1 \leqslant i \leqslant n$ and all **El**-sequences $e_1, \ldots, e_{i-1}$ for $d_1, \ldots, d_{i-1}$, $d_i \cdot e_1 \cdot \ldots \cdot e_{i-1} \in \mathbf{Set}$. The set of **El**-sequences for $d_1, \ldots, d_n$ is denoted $\mathbf{El}(d_1, \ldots, d_n)$.

Many sequences can easily be turned into **Set**-sequences. For example, from $d_1, d_2 \in \mathbf{Set}$ we obtain a **Set**-sequence d_1, d by defining $d = \top \cdot d_2$. Thus, we can restrict our attention to the notion of **Set**-sequence in Definition 3.16 without losing generality.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

Theorem 3.17. For all d, if for all $e_1, \ldots, e_n$, $d \cdot e_1 \cdot \ldots \cdot e_n \in \mathbf{Set}$, then either for all $e_1, \ldots, e_n$, $d \cdot e_1 \cdot \ldots \cdot e_n = \top$, or for all $e_1, \ldots, e_n$, $d \cdot e_1 \cdot \ldots \cdot e_n = \bot$.

Proof. Take any $c_1, \ldots, c_n$ (for example $c_i = d$) and assume without loss of generality that $d \cdot c_1 \cdot \ldots \cdot c_n = \top$. Taking any $d_1, \ldots, d_n$, we want to show that $d \cdot d_1 \cdot \ldots \cdot d_n = \top$ as well. Let i be the greatest such that $d \cdot d_1 \cdot \ldots \cdot d_{i-1} \cdot c_i \cdot \ldots \cdot c_n = \top$. If $i = n + 1$, we obtain what we require. Otherwise, $i \leqslant n$, which will lead to a contradiction. Define

$$e = ([y][x_1] \ldots [x_{i-1}][x_{i+1}] \ldots [x_n][x_i] y \ x_1 \ \ldots \ x_n) \cdot d \cdot d_1 \cdot \ldots \cdot d_{i-1} \cdot c_{i+1} \cdot \ldots \cdot c_n.$$

We have $e \cdot c_i = \top$, $e \cdot d_i = \bot$ and for all e_i, we have $e \cdot e_i \in \mathbf{Set}$. This is impossible. The proof of that fact is very similar to the proof of a theorem attributed to Scott on page 144 of Barendregt (1984). It uses the fact that all the elements of $\mathscr{D}$ have a fixpoint, which can be obtained, for instance, using the fixpoint combinator $Y = [y](([x] y \ x \ x) \ ([x] y \ x \ x)) \in \mathscr{D}$.

Define $c = ([y][x_1][x_2][x] y \ x \ x_2 \ x_1) \cdot e \cdot c_i \cdot d_i$. It satisfies $c \cdot c_i = d_i$, $c \cdot d_i = c_i$, and $c \cdot e_i \in \{c_i, d_i\}$ for all e_i. Thus the fixpoint of c must be c_i or d_i. In either case it yields $c_i = d_i$, which is impossible since $\top \neq \bot$. □

Corollary 3.18. For all **Set**-sequences $d_1, \ldots, d_n$, $\mathbf{El}(d_1, \ldots, d_n)$ is either $\mathscr{D}^n$ or $\{\}$.

Proof. The proof is by induction on n. If $n = 0$, then $\mathbf{El}()$ is the singleton set whose element is the empty sequence, that is, $\mathscr{D}^0$.

Assume now that the corollary is true for n. We must show that it is also true for $n+1$. If $\mathbf{El}(d_1, \ldots, d_n)$ is empty, then so is $\mathbf{El}(d_1, \ldots, d_{n+1})$. If it is $\mathscr{D}^n$, by Theorem 3.17 there are two possibilities for d_{n+1}: either $d_{n+1} \cdot e_1 \cdot \ldots \cdot e_n = \top$ for all $e_1, \ldots, e_n$, or $d_{n+1} \cdot e_1 \cdot \ldots \cdot e_n = \bot$ for all $e_1, \ldots, e_n$. In the former case $\mathbf{El}(d_1, \ldots, d_{n+1})$ is $\mathscr{D}^{n+1}$, and in the latter it is $\{\}$. □

Notice that if $\mathbf{El}(d_1, \ldots, d_n) = \mathscr{D}^n$, then for all $0 \leqslant i \leqslant n$, $\mathbf{El}(d_1, \ldots, d_i) = \mathscr{D}^i$.

Corollary 3.19 (Proof irrelevance). For all **Set**-sequence $d_1, \ldots, d_{n+1}$, the element d_{n+1} represents a constant function, that is, either

— for all **El**-sequence $e_1, \ldots, e_n$ for $d_1, \ldots, d_n$, $d_{n+1} \cdot e_1 \cdot \ldots \cdot e_n = \top$, or
— for all **El**-sequence $e_1, \ldots, e_n$ for $d_1, \ldots, d_n$, $d_{n+1} \cdot e_1 \cdot \ldots \cdot e_n = \bot$.

Proof. By Corollary 3.18, $\mathbf{El}(d_1, \ldots, d_{n+1})$ is either $\mathscr{D}^{n+1}$ or $\{\}$. In the former case, the first item holds, and in the latter, the second. □

4. Inductive definitions

In this section we show how to extend the definition of the proof-irrelevant model to inductively defined sets in the scheme of Dybjer (1994). It is possible to determine a general method to interpret every inductive definition that follows that scheme, but we prefer to avoid a too formal presentation hoping that a few examples will be illustrative enough.

Each inductive definition consists of adding a new preobject for each constant and new rules of two kinds: for objects and for equality between objects. In order to extend the model, we must define the interpretation of the new constants and revise the proof of

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

Theorem 3.5 for the extended system. The only items that need revision are 5 (a) and 6, since only rules for objects and for equality between objects are incorporated. Item 5 (b) will continue to hold trivially because the interpretation of the new constants will be independent of the valuation ρ. This also explains why we drop the subindex ρ, below, denoting the interpretation of the new constants directly using $[\![\forall]\!]$, $[\![\lambda]\!]$, $[\![\forall_E]\!]$, and so on.

Thus, we have to check that the interpretation of each new constant belongs to the interpretation of its type, and that in every new equality the preobjects involved are interpreted as the same element of $\mathscr{D}$.

The reader will notice that the new constants are interpreted in two different ways depending on whether they denote set formers or not. Those that denote set formers are given interpretations such that, not only do they belong to the interpretation of their types, but they also make the interpretation of the types of the remaining constants of the definition become $\mathscr{D}$. Thus, regardless of what interpretation the remaining constants are given, they will belong trivially to the interpretation of their types. These constants are interpreted by means of a variant of Church's numerals like the one used in Parigot (1988) in such a way that the equalities introduced in the definition hold in the model.

4.1. *Preliminaries*

In order to stress the intuitive reading of $\top$ and $\bot$ as Boolean values, we will write $a_1 \Rightarrow a_2$ for $a_1\ a_2\ \top$, $a_1 \wedge a_2$ for $a_1\ a_2\ \bot$ and $a_1 \vee a_2$ for $a_1\ \top\ a_2$. Similarly, $d_1 \Rightarrow d_2$, $d_1 \wedge d_2$ and $d_1 \vee d_2$ stand for $d_1 \cdot d_2 \cdot \top$, $d_1 \cdot d_2 \cdot \bot$ and $d_1 \cdot \top \cdot d_2$, respectively. These clearly behave as Boolean expressions, for example, for all $d_1, d_2 \in \mathbf{Set}$, $d_1 \Rightarrow d_2 \in \mathbf{Set}$ and $d_1 \Rightarrow d_2 = \top$ iff $d_2 = \top$ whenever $d_1 = \top$.

From Corollary 3.18, it is possible to derive the following useful remark.

Remark 4.1. For all **Set**-sequence $d_1, \ldots, d_n$, all $e_1, \ldots, e_n$ and all e^i_j for $0 < j \leqslant i < n$, the following are equivalent:

— $e_1, \ldots, e_n$ is an **El**-sequence for $d_1, \ldots, d_n$,
— $\mathbf{El}(d_1, \ldots, d_n)$ is inhabited,
— $\mathbf{El}(d_1, \ldots, d_n) = \mathscr{D}^n$,
— $d_1 \wedge d_2 \cdot e^1_1 \wedge \ldots \wedge d_n \cdot e^{n-1}_1 \cdot \ldots \cdot e^{n-1}_{n-1} = \top$,
— $d_1 \wedge d_2 \cdot e^1_1 \wedge \ldots \wedge d_n \cdot e^{n-1}_1 \cdot \ldots \cdot e^{n-1}_{n-1} \neq \bot$,
— $d_1 = d_2 \cdot e^1_1 = \ldots = d_n \cdot e^{n-1}_1 \cdot \ldots \cdot e^{n-1}_{n-1} = \top$,
— for all $0 < i \leqslant n$, $d_i \cdot e^{i-1}_1 \cdot \ldots \cdot e^{i-1}_{i-1} \neq \bot$.

Proof. The equivalence of the first three items is an obvious consequence of Corollary 3.18. To prove that the fourth and fifth items are equivalent to the third one, we prove that $\phi_n = \top$ if $\mathbf{El}(d_1, \ldots, d_n) = \mathscr{D}^n$, and $\phi_n = \bot$ otherwise, where ϕ_n stands for $d_1 \wedge d_2 \cdot e^1_1 \wedge \ldots \wedge d_n \cdot e^{n-1}_1 \cdot \ldots \cdot e^{n-1}_{n-1}$. This can be proved by reasoning very much as in the proof of Corollary 3.18: induction on n followed by an analysis of the two cases for $\mathbf{El}(d_1, \ldots, d_n)$ in the inductive step, followed by considering the two possibilities for d_{n+1} in the case when $\mathbf{El}(d_1, \ldots, d_n) = \mathscr{D}^n$. A similar trick can be used to prove the equivalence between the last two items and the third one. □

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

The truth of each of the last items does not depend on the particular choice of $e_1^1, \ldots, e_{n-1}^{n-1}$, since these do not occur in the first three items. We replace them by '$*$' whenever we find it convenient to ignore their precise value. Thus, '$*$' stands for arbitrary closed expressions and $x\ *_1 \ldots *_n$ says that x is applied to n such expressions. Taking $\Sigma_n = [x_1 \ldots x_n]x_1 \wedge \ldots \wedge x_n\ *_1 \ldots *_{n-1}$, Remark 4.1 implies that for all **Set**-sequences $d_1, \ldots, d_n$, we have $\Sigma_n \cdot d_1 \cdot \ldots \cdot d_n \in \mathbf{Set}$. Thus, it states that $\Sigma_n \cdot d_1 \cdot \ldots \cdot d_n$ can then be handled as a Boolean value, which is $\bot$ iff $d_i \cdot *_1 \cdot \ldots \cdot *_{i-1} = \bot$ for some i. Defining $\Pi_n = [x_1 \ldots x_{n+1}]\Sigma_n x_1 \ldots x_n \Rightarrow x_{n+1}\ *_1 \ldots *_n$, it also follows that $\Pi_n \cdot d_1 \cdot \ldots \cdot d_{n+1}$ denotes a Boolean value whenever $d_1, \ldots, d_{n+1}$ is a **Set**-sequence, since it is then equal to $\Sigma_n \cdot d_1 \cdot \ldots \cdot d_n \Rightarrow \Sigma_{n+1} \cdot d_1 \cdot \ldots \cdot d_{n+1}$. That Boolean value is $\top$ iff $d_{n+1} \cdot *_1 \cdot \ldots \cdot *_n = \top$ follows from $d_1 = \ldots = d_n \cdot *_1 \cdot \ldots \cdot *_{n-1} = \top$.

Observe that $\bot, d$ is a **Set**-sequence for all d. Thus, in spite of Remark 4.1, it is possible to have a **Set**-sequence $d_1, \ldots, d_n$ such that for some i, $d_i \cdot *_1 \cdot \ldots \cdot *_{i-1} \notin \mathbf{Set}$. But then, we would necessarily have $j < i$ such that $d_j \cdot *_1 \cdot \ldots \cdot *_{j-1} = \bot$.

4.2. *Interpretation of inductive sets*

From now on we will omit occurrences of the prefamily El in pretypes of the form El a since they can always be inferred: whenever a preobject is placed in a position corresponding to a pretype, the prefamily El is implicitly applied to it. We write $(x{:}\alpha_1)\alpha_2$ for the pretype $\alpha_1 \to [x]\alpha_2$, $(x{:}\alpha_1; y{:}\alpha_2)\alpha_3$ for $(x{:}\alpha_1)(y{:}\alpha_2)\alpha_3$ and $(x, y{:}\alpha_1)\alpha_2$ for $(x{:}\alpha_1; y{:}\alpha_1)\alpha_2$.

Proposition 4.2. For all pretypes $(x_1{:}\alpha_1; \ldots; x_n{:}\alpha_n)\alpha$ the following holds.

$$d \in [\![(x_1{:}\alpha_1; \ldots; x_n{:}\alpha_n)\alpha]\!]\rho \iff (\forall d_i \in [\![\alpha_i]\!]_{\rho_{i-1}})_{i\in\{1\ldots n\}}\ d \cdot d_1 \cdot \ldots \cdot d_n \in [\![\alpha]\!]_{\rho_n}$$

where $\rho_0 = \rho$ and $\rho_{i+1} = (\rho_i, x_{i+1}{=}d_{i+1})$.

Proof. The proof is by induction on n. □

Each inductive definition introduces new constants of certain types. For each such constant c and type T_c we have to define the interpretation $[\![c]\!]$, and verify that $[\![c]\!] \in [\![T_c]\!]$ holds and that certain equations for the constant c hold for $[\![c]\!]$. The type T_c of c and the equations for c are specified by means of new rules for objects and for equality between objects. In the presentation of the rules the occurrences of the empty context are omitted.

4.3. *Empty set*

This extension consists of adding the preobjects `Bot` and `abort` and the rules in Figure 17.

$$\frac{}{\texttt{Bot} : \text{Set}} \qquad \frac{}{\texttt{abort} : (C{:}(p{:}\texttt{Bot})\text{Set}; p{:}\texttt{Bot})C\ p}$$

Fig. 17. Rules for the empty set

Define $[\![\texttt{Bot}]\!] = \bot \in \mathbf{Set} = [\![T_{\texttt{Bot}}]\!]$. By Proposition 4.2, $d \in [\![T_{\texttt{abort}}]\!]$ if and only if

$$(\forall e_1 \in \mathbf{El}([\![\texttt{Bot}]\!])\ d_1 \cdot e_1 \in \mathbf{Set}) \wedge d_2 \in \mathbf{El}([\![\texttt{Bot}]\!]) \Longrightarrow d \cdot d_1 \cdot d_2 \in \mathbf{El}(d_1 \cdot d_2),$$

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

which is always true because, by Remark 4.1, $d_2 \in \mathbf{El}([\![\mathtt{Bot}]\!])$ iff $\top = [\![\mathtt{Bot}]\!] = \bot$. Thus, $[\![T_{\mathtt{abort}}]\!] = \mathscr{D}$, so $[\![\mathtt{abort}]\!] \in [\![T_{\mathtt{abort}}]\!]$ regardless of the value of $[\![\mathtt{abort}]\!]$. To be consistent with the remaining examples of inductive definitions, we take $[\![\mathtt{abort}]\!] = [C][p]p$.

This example already illustrates the use of Proposition 4.2 and Remark 4.1, which will be used implicitly in the remaining examples. The latter, for example, would in addition allow us to rewrite the implication above as

$$[\![\mathtt{Bot}]\!], d_1 \ \mathbf{Set}\text{-sequence} \wedge [\![\mathtt{Bot}]\!] = \top \Longrightarrow d_1 \cdot * = \top,$$

and further to

$$[\![\mathtt{Bot}]\!], d_1 \ \mathbf{Set}\text{-sequence} \Longrightarrow \Pi_1 \cdot [\![\mathtt{Bot}]\!] \cdot d_1 = \top,$$

the truth of which can now be computed.

4.4. *Singleton set*

For this extension we add the preobjects Unit, unit and Unit_E and the rules in Figure 18.

$$\frac{}{\mathrm{Unit} : \mathrm{Set}} \qquad \frac{}{\mathrm{unit} : \mathrm{Unit}} \qquad \frac{}{\mathrm{Unit}_E : (C : (p : \mathrm{Unit})\mathrm{Set}; x : C\ \mathrm{unit}; p : \mathrm{Unit})C\ p}$$

$$\frac{\Gamma\ \mathrm{context}}{\mathrm{Unit}_E\ C\ x\ \mathrm{unit} = x : C\ \mathrm{unit}\ [\Gamma]}\ C, x \in \Gamma \text{ with expected type}$$

Fig. 18. Rules for the singleton set

Define $[\![\mathrm{Unit}]\!] = \top \in \mathbf{Set} = [\![T_{\mathrm{Unit}}]\!]$. We now have

$$[\![\mathrm{unit}]\!] \in \mathscr{D} = \mathbf{El}(\top) = \mathbf{El}([\![\mathrm{Unit}]\!]) = [\![T_{\mathrm{unit}}]\!]$$

regardless of the value of $[\![\mathrm{unit}]\!]$. Finally, $d \in [\![T_{\mathrm{Unit}_E}]\!]$ if and only if

$$\top, d_1 \ \mathbf{Set}\text{-sequence} \wedge d_1 \cdot * = \top \Longrightarrow d_1 \cdot * = \top,$$

which is trivially true. Thus, $[\![T_{\mathrm{Unit}_E}]\!] = \mathscr{D}$, so $[\![\mathrm{Unit}_E]\!] \in [\![T_{\mathrm{Unit}_E}]\!]$ regardless of the value of $[\![\mathrm{Unit}_E]\!]$. We define $[\![\mathrm{unit}]\!]$ and $[\![\mathrm{Unit}_E]\!]$ so that the equation

$$[\![\mathrm{Unit}_E]\!] \cdot d_1 \cdot d_2 \cdot [\![\mathrm{unit}]\!] = d_2$$

holds. For example $[\![\mathrm{unit}]\!] = [x]x$ and $[\![\mathrm{Unit}_E]\!] = [C][x][p]p\,x$.

4.5. *Universal quantifier*

This extension consists of adding the preobjects $\forall$, λ and $\forall_E$ and the rules in Figure 19.

Define $[\![\forall]\!] = \Pi_1$. We must check that it belongs to the interpretation of $T_\forall$:

$$[\![\forall]\!] \in [\![T_\forall]\!] \quad \Longleftrightarrow \quad d_1, d_2 \ \mathbf{Set}\text{-sequence} \Longrightarrow \Pi_1 \cdot d_1 \cdot d_2 \in \mathbf{Set},$$

which is true. In addition, for all d we have $d \in [\![T_\lambda]\!]$ if and only if

$$d_1, d_2 \ \mathbf{Set}\text{-sequence} \wedge (d_1 = \top \Rightarrow d_2 \cdot * = \top) \Longrightarrow [\![\forall]\!] \cdot d_1 \cdot d_2 = \top,$$

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$$\frac{}{\forall : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set})\mathrm{Set}} \qquad \frac{}{\lambda : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set}; u{:}(a{:}A)U\ a)\forall\ A\ U}$$

$$\frac{}{\forall_E : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set}; C{:}(p{:}\forall\ A\ U)\mathrm{Set}; x{:}(u{:}(a{:}A)U\ a)C\ (\lambda\ A\ U\ u); p{:}\forall\ A\ U)C\ p}$$

$$\frac{\Gamma \text{ context}}{\forall_E\ A\ U\ C\ x\ (\lambda\ A\ U\ u) {=} x\ u\ :\ C\ (\lambda\ A\ U\ u)\ [\Gamma]}\ A, U, C, x, u \in \Gamma \text{ with expected type}$$

Fig. 19. Rules for universal quantifier

which, is equivalent to

$$d_1, d_2\ \mathbf{Set}\text{-sequence}\ \wedge \Pi_1 \cdot d_1 \cdot d_2 = \top \Longrightarrow \Pi_1 \cdot d_1 \cdot d_2 = \top,$$

which is trivially true. This means that $[\![T_\lambda]\!] = \mathscr{D}$, hence $[\![\lambda]\!] \in [\![T_\lambda]\!]$ regardless of the value of $[\![\lambda]\!]$. Similarly, $d \in [\![T_{\forall_E}]\!]$ is equivalent to

$$\left.\begin{array}{l} d_1, d_2\ \mathbf{Set}\text{-sequence}\ \wedge \\ \Pi_1 \cdot d_1 \cdot d_2, d_3\ \mathbf{Set}\text{-sequence}\ \wedge \\ \Pi_1 \cdot d_1 \cdot d_2 = \top \Rightarrow d_3 \cdot * = \top \end{array}\right\} \Longrightarrow \Pi_1 \cdot (\Pi_1 \cdot d_1 \cdot d_2) \cdot d_3 = \top,$$

which is true. Thus $[\![T_{\forall_E}]\!] = \mathscr{D}$, so $[\![\forall_E]\!] \in [\![T_{\forall_E}]\!]$ regardless of the value of $[\![\forall_E]\!]$. We define $[\![\lambda]\!]$ and $[\![\forall_E]\!]$ so that the equation

$$[\![\forall_E]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot ([\![\lambda]\!] \cdot d_1 \cdot d_2 \cdot e_1) = d_4 \cdot e_1$$

holds. For example, $[\![\lambda]\!] = [A][U][u][x]x\,u$ and $[\![\forall_E]\!] = [A][U][C][x][p]p\,x$.

4.6. *Disjunction*

This extension consists of adding the preobjects `Or`, `l`, `r` and `when` and the rules in Figure 20.

$$\frac{}{\mathtt{Or} : (A, B{:}\mathrm{Set})\mathrm{Set}} \qquad \frac{}{\mathtt{l} : (A, B{:}\mathrm{Set}; a{:}A)\mathtt{Or}\ A\ B} \qquad \frac{}{\mathtt{r} : (A, B{:}\mathrm{Set}; b{:}B)\mathtt{Or}\ A\ B}$$

$$\frac{}{\mathtt{when} : (A, B{:}\mathrm{Set}; C{:}(p{:}\mathtt{Or}\ A\ B)\mathrm{Set}; x{:}(a{:}A)C\ (\mathtt{l}\ A\ B\ a); y{:}(b{:}B)C\ (\mathtt{r}\ A\ B\ b); p{:}\mathtt{Or}\ A\ B)C\ p}$$

$$\frac{\Gamma \text{ context}}{\mathtt{when}\ A\ B\ C\ x\ y\ (\mathtt{l}\ A\ B\ a) {=} x\ a\ :\ C\ (\mathtt{r}\ A\ B\ a)\ [\Gamma]}\ A, B, C, x, y, a \in \Gamma \text{ with expected type}$$

$$\frac{\Gamma \text{ context}}{\mathtt{when}\ A\ B\ C\ x\ y\ (\mathtt{r}\ A\ B\ b) {=} y\ b\ :\ C\ (\mathtt{r}\ A\ B\ b)\ [\Gamma]}\ A, B, C, x, y, b \in \Gamma \text{ with expected type}$$

Fig. 20. Rules for disjunction

Define $[\![\mathtt{Or}]\!] = [A][B]A \vee B \in [\![T_{\mathtt{Or}}]\!]$. For all d,

$$d \in [\![T_{\mathtt{l}}]\!] \iff d_1, d_2 \in \mathbf{Set} \wedge d_1 = \top \Rightarrow d_1 \vee d_2 = \top,$$

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

which is true. Thus $[\![T_{\mathtt{1}}]\!] = \mathscr{D}$, so $[\![\mathtt{1}]\!] \in [\![T_{\mathtt{1}}]\!]$ regardless of the value of $[\![\mathtt{1}]\!]$. The same applies for $[\![\mathtt{r}]\!]$. Also, $d \in [\![T_{\mathtt{when}}]\!]$ if and only if

$$\left.\begin{array}{l} d_1, d_2 \in \mathbf{Set} \wedge \\ d_1, d_3 \ \mathbf{Set}\text{-sequence} \wedge d_2, d_3 \ \mathbf{Set}\text{-sequence} \wedge \\ \Pi_1 \cdot d_1 \cdot d_3 = \top \wedge \Pi_1 \cdot d_2 \cdot d_3 = \top \end{array}\right\} \Longrightarrow \left\{\begin{array}{c} d_1 = \top \Rightarrow d_3 \cdot * = \top \\ \wedge \\ d_2 = \top \Rightarrow d_3 \cdot * = \top. \end{array}\right.$$

Here we have used, on the left-hand side, the fact that $\forall e_1.\ d_1 \vee d_2 = \top \Rightarrow d_3 \cdot e_1 \in \mathbf{Set}$ is equivalent to $(\forall e_1.\ d_1 = \top \Rightarrow d_3 \cdot e_1 \in \mathbf{Set}) \wedge (\forall e_1.\ d_2 = \top \Rightarrow d_3 \cdot e_1 \in \mathbf{Set})$, if $d_1, d_2 \in \mathbf{Set}$. This yielded that d_1, d_3 and d_2, d_3 are **Set**-sequences. Similarly, on the right-hand side, we have used the fact that $d_1 \vee d_2 = \top \Rightarrow d_3 \cdot * = \top$ is equivalent to $(d_1 = \top \Rightarrow d_3 \cdot * = \top) \wedge (d_2 = \top \Rightarrow d_3 \cdot * = \top)$.

The implication above is trivially true. Thus $[\![T_{\mathtt{when}}]\!] = \mathscr{D}$, and hence $[\![\mathtt{when}]\!] \in [\![T_{\mathtt{when}}]\!] = \mathscr{D}$ regardless of the value of $[\![\mathtt{when}]\!]$. We define $[\![\mathtt{1}]\!]$, $[\![\mathtt{r}]\!]$ and $[\![\mathtt{when}]\!]$ so that the equations

$$[\![\mathtt{when}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot d_5 \cdot ([\![\mathtt{1}]\!] \cdot d_1 \cdot d_2 \cdot e_1) = d_4 \cdot e_1$$
$$[\![\mathtt{when}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot d_5 \cdot ([\![\mathtt{r}]\!] \cdot d_1 \cdot d_2 \cdot e_1) = d_5 \cdot e_1$$

hold: $[\![\mathtt{1}]\!] = [A][B][a][x][y]x\,a$, $[\![\mathtt{r}]\!] = [A][B][b][x][y]y\,b$ and

$$[\![\mathtt{when}]\!] = [A][B][C][x][y][p]p\,x\,y.$$

4.7. *Existential quantifier*

This extension consists of adding the preobjects $\exists$, $\mathtt{pr}$ and $\exists_E$ and the rules in Figure 21.

$$\frac{}{\exists : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set})\mathrm{Set}} \qquad \frac{}{\mathtt{pr} : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set}; a{:}A; u{:}U\,a)\exists\,A\,U}$$

$$\frac{}{\exists_E : (A{:}\mathrm{Set}; U{:}(a{:}A)\mathrm{Set}; C{:}(p{:}\exists\,A\,U)\mathrm{Set}; x{:}(a{:}A; u{:}U\,a)C\,(\mathtt{pr}\,A\,U\,a\,u); p{:}\exists\,A\,U)C\,p}$$

$$\frac{\Gamma\ \text{context}}{\exists_E\,A\,U\,C\,x\,(\mathtt{pr}\,A\,U\,a\,u) = x\,a\,u : C\,(\mathtt{pr}\,A\,U\,a\,u)\ [\Gamma]}$$

Fig. 21. Rules for existential quantifier: conditions on Γ omitted

Define $[\![\exists]\!] = \Sigma_2 \in [\![T_{\exists}]\!]$. It is easy to prove that $[\![\mathtt{pr}]\!] \in [\![T_{\mathtt{pr}}]\!]$ and $[\![\exists_E]\!] \in [\![T_{\exists_E}]\!]$ regardless of the values of $[\![\mathtt{pr}]\!]$ and $[\![\exists_E]\!]$. We define $[\![\mathtt{pr}]\!]$ and $[\![\exists_E]\!]$ so that the equation

$$[\![\exists_E]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot ([\![\mathtt{pr}]\!] \cdot d_1 \cdot d_2 \cdot e_1 \cdot e_2) = d_4 \cdot e_1 \cdot e_2$$

holds. For example $[\![\mathtt{pr}]\!] = [A][U][a][u][x]x\,a\,u$ and $[\![\exists_E]\!] = [A][U][C][x][p]p\,x$.

4.8. *Natural numbers*

This extension consists of adding the preobjects $\mathtt{N}$, 0, $\mathtt{s}$, and $\mathtt{Nrec}$ and the rules in Figure 22.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

$$\frac{}{\mathtt{N} : \mathrm{Set}} \quad \frac{}{0 : \mathtt{N}} \quad \frac{}{\mathtt{s} : (n{:}\mathtt{N})\mathtt{N}} \quad \frac{}{\mathtt{Nrec} : (C{:}(n{:}\mathtt{N})\mathrm{Set}; x{:}C\,0; y{:}(n{:}\mathtt{N}; i{:}C\,n)C\,(\mathtt{s}\,n); p{:}\mathtt{N})C\,p}$$

$$\frac{\Gamma \text{ context}}{\mathtt{Nrec}\,C\,x\,y\,0{=}x : C\,0\ [\Gamma]} \qquad \frac{\Gamma \text{ context}}{\mathtt{Nrec}\,C\,x\,y\,(\mathtt{s}\,n){=}y\,n\,(\mathtt{Nrec}\,C\,x\,y\,n) : C\,(\mathtt{s}\,n)\ [\Gamma]}$$

Fig. 22. Rules for natural numbers: conditions on Γ omitted

Define $[\![\mathtt{N}]\!] = \top \in \mathbf{Set} = [\![T_{\mathtt{N}}]\!]$. Obviously, $[\![T_0]\!] = [\![T_{\mathtt{s}}]\!] = [\![T_{\mathtt{Nrec}}]\!] = \mathscr{D}$. We define $[\![0]\!]$, $[\![\mathtt{s}]\!]$ and $[\![\mathtt{Nrec}]\!]$ so that the equations

$$[\![\mathtt{Nrec}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot [\![0]\!] = d_2$$
$$[\![\mathtt{Nrec}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot ([\![\mathtt{s}]\!] \cdot e_1) = d_3 \cdot e_1 \cdot ([\![\mathtt{Nrec}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot e_1)$$

hold. For example, $[\![0]\!] = [x][y]x$, $[\![\mathtt{s}]\!] = [n][x][y]y\,n\,(n\,x\,y)$, and $[\![\mathtt{Nrec}]\!] = [C][x][y][p]p\,x\,y$.

These numerals were used in Parigot (1988).

4.9. *Equality*

This extension consists of adding the preobjects `Eq`, `eq`, and `peel` and the rules in Figure 23.

$$\frac{}{\mathtt{Eq} : (A{:}\mathrm{Set}; a, b{:}A)\mathrm{Set}} \qquad \frac{}{\mathtt{eq} : (A{:}\mathrm{Set}; a{:}A)\mathtt{Eq}\,A\,a\,a}$$

$$\frac{}{\mathtt{peel} : (A{:}\mathrm{Set}; C{:}(a, b{:}A; p{:}\mathtt{Eq}\,A\,a\,b)\mathrm{Set}; x{:}(a{:}A)C\,a\,a\,(\mathtt{eq}\,A\,a); a, b{:}A; p{:}\mathtt{Eq}\,A\,a\,b)C\,a\,b\,p}$$

$$\frac{\Gamma \text{ context}}{\mathtt{peel}\,A\,C\,x\,a\,a\,(\mathtt{eq}\,A\,a){=}x\,a : C\,a\,a\,(\mathtt{eq}\,A\,a)\ [\Gamma]}$$

Fig. 23. Rules for equality: conditions on Γ omitted

Define $[\![\mathtt{Eq}]\!] = [A][a][b]A$. Clearly, $[\![\mathtt{Eq}]\!] \in [\![T_{\mathtt{Eq}}]\!]$ and $[\![T_{\mathtt{eq}}]\!] = \mathscr{D}$. Also, $d \in [\![T_{\mathtt{peel}}]\!]$ if and only if

$$\left.\begin{array}{l} d_1, [x]\top, [x][y]\top, d_2 \text{ } \mathbf{Set}\text{-sequence} \wedge \\ \Pi_3 \cdot d_1 \cdot [x]\top \cdot [x][y]\top \cdot d_2 = \top \wedge \\ d_1 = \top \end{array}\right\} \Longrightarrow d_2 \cdot * \cdot * \cdot * = \top.$$

In order to satisfy the hypotheses of Remark 4.1, we have turned d_1, d_2 (which, because of the arity of d_2, is not a **Set**-sequence) into the **Set**-sequence $d_1, [x]\top, [x][y]\top, d_2$. The implication above holds, so $[\![T_{\mathtt{peel}}]\!] = \mathscr{D}$. We define $[\![\mathtt{eq}]\!]$ and $[\![\mathtt{peel}]\!]$ so that the equation

$$[\![\mathtt{peel}]\!] \cdot d_1 \cdot d_2 \cdot d_3 \cdot d_4 \cdot d_4 \cdot ([\![\mathtt{eq}]\!] \cdot d_1 \cdot d_4) = d_3 \cdot d_4$$

holds, for instance $[\![\mathtt{eq}]\!] = [A][a][x]x\,a$ and $[\![\mathtt{peel}]\!] = [A][C][x][a][b][p]p\,x$.

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

4.10. *Peano's fourth axiom*

Peano's fourth axiom states that zero is not equal to the successor of any natural number. It can be stated as follows:

$$(n{:}\mathtt{N}; h{:}\mathtt{Eq}\,\mathtt{N}\,0\,(\mathtt{s}\,n))\mathtt{Bot}.$$

We denote this type T_{p4}. We show that Peano's fourth axiom is not provable in this version of type theory by showing that $[\![T_{\mathrm{p4}}]\!] = \{\}$. For all d, we have $d \in [\![T_{\mathrm{p4}}]\!]$ iff

$$[\![\mathtt{N}]\!] = \top \wedge [\![\mathtt{Eq}]\!] \cdot [\![\mathtt{N}]\!] \cdot [\![\mathtt{0}]\!] \cdot ([\![\mathtt{s}]\!] \cdot *) = \top \Longrightarrow [\![\mathtt{Bot}]\!] = \top,$$

which is trivially false since the antecedent holds and the consequent does not. Hence Peano's fourth axiom is not provable.

4.11. *Reflecting proof irrelevance*

Moreover, it is easy to see that $[\![(A{:}\mathrm{Set}; a, b{:}A)\mathtt{Eq}\,A\,a\,b]\!] = \mathscr{D}$, which means that it would be possible to interpret a constant

$$\mathrm{irrelevance}{:}(A{:}\mathrm{Set}; a, b{:}A)\mathtt{Eq}\,A\,a\,b.$$

Therefore, having such a constant would maintain consistency, reflect proof-irrelevance at the level of the theory, and imply the negation of Peano's fourth axiom.

5. Conclusion

We have presented a detailed and elementary proof of consistency of Martin-Löf's theory of types for the exact formulation in Martin-Löf (1992), without easing the task by introducing modifications into (or making informal assumptions about) the theory. The level of detail of the presentation is such that this paper could be used as a basis for a direct formalisation in a proof assistant.

We have shown that the model provides an elementary proof of consistency of the theory, and a proof that Peano's fourth axiom is not provable. We have also given a technique for interpreting inductive definitions by using a variant of Church numerals.

We have thus extended Smith's proof-irrelevant model (Smith 1988) to the whole logical framework. This can be compared to Gentzen's extension (Gentzen 1936) of Hilbert and Ackermann's elementary proof of consistency (Hilbert and Ackermann 1928) to the simple theory of types. In both cases the extension implies going from a first-order to a higher-order language, but Gentzen's proof is for a language with a predefined hierarchy of types, whereas in our language, the type system is more complex, the types themselves are defined within the language.

There are in principle two ways of achieving proof-irrelevance in a model of type theory. The first method, which is simpler and more direct, consists of interpreting all the proof-objects by the same denotation, and all the types by Boolean values. This is essentially how it was done in Smith (1988). The second method, which is non-trivial and subtler, allows different proof-objects to receive different denotations, but the denotation

https://doi.org/10.1017/S0960129502003766 Published online by Cambridge University Press

of predicates cannot distinguish between proof-objects, not even between proof-objects having different denotations. This is how proof-irrelevance was obtained in this paper.

We defined our model by interpreting precontexts, pretypes, preobjects, prefamilies and presubstitutions separately rather than interpreting whole derivations. For this reason, there was no hope of using the direct method mentioned above, since that would imply that $[x][y]x$ and $[x][y]y$ should receive, at the same time, identical denotations as objects of type $(x, y : \mathtt{N})\mathtt{N}$, and different denotations as objects of type $(x, y : \mathrm{Set})\mathrm{Set}$. Notice that this problem appears as a consequence of having abstraction *à la* Curry and large types such as Set in the theory.

This paper is an improved and simplified version of a part of my thesis (Fridlender 1997), which presents a concrete model in terms of the lambda calculus itself rather than in terms of an arbitrary extensional model of it. The thesis presents in addition more examples than were given in Section 4.

Acknowledgments

I am deeply indebted to Thierry Coquand, my thesis supervisor. He gave me invaluable guidance then, and ideas and encouragement to produce this improved version now.

I am also grateful to an anonymous referee from whose remarks this paper and myself have benefitted greatly. In particular, they contributed to a clearer conclusion and drew my attention to the possibility of reflecting proof-irrelevance at the level of the theory.

References

Barendregt, H. (1984) *The Lambda Calculus*, North-Holland.

Dybjer, P. (1994) Inductive families. *Formal Aspects of Computing* **6** 440–465.

Dybjer, P. (1996) Internal type theory. In: TYPES'95. *Springer-Verlag Lecture Notes in Computer Science* 120–134.

Fridlender, D. (1997) *Higman's Lemma in Type Theory*, Ph. D. thesis, Department of Computing Science at University of Göteborg and Chalmers University of Technology.

Gaspes, V. (1997) *A Type Theoretical Analysis of Some Aspects of Programming Languages*, Ph.D. thesis, Department of Computing Science at Chalmers University of Technology and University of Göteborg.

Gentzen, G. (1936) Die Widerspruchsfreiheit der Stufenlogik. *Mathematische Zeitschrift* **41** (3) 357–366.

Hilbert, D. and Ackermann, W. (1928) *Grundzüge der Theoretischen Logik*, Springer-Verlag.

Hindley, J. and Seldin, J. (1986) *Introduction to Combinators and λ-Calculus*, Cambridge University Press.

Hofmann, M. (1997) Syntax and semantics of dependent types. In: Pitts, A. and Dybjer, P. (eds.) *Semantics and Logics of Computation*, Cambridge University Press 79–130.

Martin-Löf, P. (1975) An Intuitionistic Theory of Types: Predicative Part. In: Rose, H. E. and Shepherdson, J. C. (eds.) *Logic Colloquium 1973*, North-Holland 73–118.

Martin-Löf, P. (1984) *Intuitionistic Type Theory*, Bibliopolis, Napoli.

Martin-Löf, P. (1992) Substitution Calculus (Series of lectures). (A complete presentation of the calculus can be found in Tasistro (1997).)

Miquel, A. (2000) A model for impredicative type systems, universes, intersection types and subtyping. *LICS*.

Nordström, B., Petersson, K. and Smith, J. (1990) *Programming in Martin-Löf's Type Theory. An Introduction*, Oxford University Press.

Parigot, M. (1988) Programming with proofs: a second order type theory. In: Ganzinger, H. (ed.) ESOP'88 – 2nd European Symposium on Programming. *Springer-Verlag Lecture Notes in Computer Science* **300** 145–159.

Smith, J. (1988) The Independence of Peano's Fourth Axiom from Martin-Löf's Type Theory without Universes. *Journal of Symbolic Logic* **53** 840–845.

Tasistro, A. (1997) *Substitution, Record Types and Subtyping in Type Theory, with Applications to the Theory of Programming*, Ph.D. thesis, Department of Computing Science at Chalmers University of Technology and University of Göteborg.