Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 174 (2007) 19–35
www.elsevier.com/locate/entcs

Modelling Generic Judgements

Ulrich Schöpp[1]

TCS, Institut für Informatik
Ludwig-Maximilians-Universität München
Oettingenstraße 67, D-80538 München, Germany

Abstract

We propose a semantics for the ∇-quantifier of Miller and Tiu. First we consider the case for classical first-order logic. In this case, the interpretation is close to standard Tarski-semantics and completeness can be shown using a standard argument. Then we put our semantics into a broader context by giving a general interpretation of ∇ in categories with binding structure. Since categories with binding structure also encompass nominal logic, we thus show that both ∇-logic and nominal logic can be modelled using the same definition of binding. As a special case of the general semantics in categories with binding structure, we recover Gabbay & Cheney's translation of FOλ^{∇} into nominal logic.

Keywords: Higher-Order Abstract Syntax, First-Order Logic, Model Theory, Categorical Logic

Introduction

Miller & Tiu [18] have introduced the logic FOλ^{∇} for reasoning about specifications in λ-tree syntax (a version of higher-order abstract syntax). The main new feature of FOλ^{∇} is the ∇-quantifier for the treatment of object-level eigenvariables. The design of FOλ^{∇} is based on a study of the proof-theory of first-order logic and the result is an elegant and simple calculus.

In addition to the proof theory for FOλ^{∇}, one may be interested in a model-theoretic semantics, e.g. [6,17,25]. A simple semantics is useful for explaining the logic and for understanding the meaning of formulae. It may also help in transferring technology from other, semantics-based, approaches. Nominal Logic [19], for example, solves a problem similar to that solved by FOλ^{∇}, but is based on a semantic approach. A clarification of the semantics of FOλ^{∇} should help in studying the connection of FOλ^{∇} to Nominal Logic, in order to answer questions such as whether the elegant proof theory of FOλ^{∇} can be used for Nominal Logic, or whether programming in the style of Fresh O'Caml [23,22] is possible with ∇. In general, a model-theoretic explanation of the ∇-quantifier should make it amenable

[1] Email: schoepp@tcs.ifi.lmu.de

doi:10.1016/j.entcs.2007.01.027

for inclusion in solutions that are based on the model-theory of first-order logic, an example being Brotherston's very general approach to cyclic proofs [2].

In this paper we give a simple and direct semantic interpretation of FOλ^{∇}. We focus on classical logic with its easy-to-describe Tarski semantics and show that well-known theory can be adapted to classical FOλ^{∇}. We focus on the classical case, as it is particularly simple to describe. In the second part of this paper we give a general framework for modelling FOλ^{∇}, which we hope will help in studying the relationships of models for FOλ^{∇}, such as those of Gabbay & Cheney [6] and Miculan & Yemane [17].

To explain the interpretation informally, recall from [18] that FOλ^{∇} is a first-order logic over simply-typed λ-calculus. Object-level syntax is represented by λ-tree types (HOAS). For instance, the type of λ-terms can be represented by a type Tm with two constants $app\colon Tm \times Tm \to Tm$ and $lam\colon (Tm \to Tm) \to Tm$. In addition to the usual existential and universal quantification, $\exists x\colon \tau.\, \varphi(x)$ and $\forall x\colon \tau.\, \varphi(x)$, FO$\lambda^{\nabla}$ also features the quantification $\nabla x\colon Tm.\, \varphi(x)$. In this paper we restrict the ∇-quantifier to range only over a subclass of types such as Tm, which we call λ-tree types. A similar restriction is imposed in [25]. Our restriction on ∇ is in line with restrictions imposed on the new-quantifier N of Gabbay & Pitts, which typically ranges over names only. For the semantics, we use a standard interpretation for the quantifiers $\exists$ and $\forall$. Our interpretation of $\nabla x\colon Tm.\, \varphi(x)$ may be expressed as: 'for all new *variables* $x\colon Tm$ the formula $\varphi(x)$ holds.' The quantification ranges only over variables, but not over terms of the form $app\langle x, y\rangle$ or $lam(f)$. The meaning of the formula $\nabla x\colon Tm.\, \varphi(x)$ may equivalently be expressed as: 'for some new variable $x\colon Tm$ the formula $\varphi(x)$ holds.'

To formalise this informal description of the semantics, we have to deal with a number of issues. First, object-level terms are represented using higher-order abstract syntax. To model the type Tm, and in particular the constant lam of type $(Tm \to Tm) \to Tm$, adequately, we must find a model in which the interpretation of the function space $(Tm \to Tm)$ contains just α-equivalence classes of object-level terms. We address this issue using an approach of Hofmann [12], who has shown that certain presheaves provide a canonical model in which such an interpretation of Tm is available. A second issue is that to model ∇ we want to explain quantification over variables only. The internal logic of the presheaf-category, in which the terms are interpreted, is not suitable for this purpose, since it can express only propositions that are closed under all substitutions. We address this issue by modelling the terms in a presheaf category, but taking the logic from a so-called category with binding structure. This approach of modelling terms in one place and using the logic from another place is also used by Hofmann [12]. We consider a number of different logics including Nominal Logic, in which case we recover the interpretation of FOλ^{∇} in Nominal Logic proposed by Gabbay & Cheney [6].

In Part I of this paper we spell out directly an interpretation of classical FOλ^{∇}. The aim of this part is to present the interpretation as simple as possible, without assuming knowledge of FOλ^{∇}. We show that the semantics is close to the standard Tarski semantics for classical logic and that the well-known completeness argument

goes through almost unchanged. The point of Part I is to stress that just a little change to the standard Tarski-semantics is enough for modelling the ∇-quantifier.

In Part II, we generalise the development from the first part by giving models of ∇ in *categories with binding structure* [20, Chapter 10], which capture a general definition of binding. Since Nominal Logic also fits into the framework of categories with binding structure, the development in Part II will therefore show that both ∇ and nominal logic can be modelled using the very same definition of binding.

1 Part I
Classical First-Order Logic with ∇

1.1 Simply typed λ-calculus

We fix the notation for a simply typed λ-calculus.

$$\begin{array}{rl} \text{Types:} & \tau := \text{base types} \mid \lambda\text{-tree types} \mid 1 \mid \tau \times \tau \mid \tau \to \tau \\ \text{Terms:} & M := x \mid c \mid * \mid \langle M, M\rangle \mid \pi_1(M) \mid \pi_2(M) \mid \lambda x{:}\tau.\, M \mid M\ M \\ \text{Contexts:} & \Sigma := \cdot \mid \Sigma,\, x\colon \tau \end{array}$$

Terms are identified up to renaming of bound variables. We assume constants c to be defined in a signature $\mathcal{S}$, and we assume the choice of $\mathcal{S}$ and the primitive types to be such that the types and terms are countable.

In addition to the normal base types, there is a second kind of base types, which we call λ-tree types. The intended use of λ-tree types is for higher-order abstract syntax encodings. We use ι to range over λ-tree types.

Contexts are subject to the usual convention that no variable may be declared more than once. A context that contains only declarations $x\colon \iota$ of λ-tree types is called a λ-tree context. We use σ to range over λ-tree contexts.

The typing judgement is defined by the following rules.

$$\frac{(x\colon \tau) \in \Sigma}{\Sigma \vdash x\colon \tau} \qquad \frac{(c\colon \tau) \in \mathcal{S}}{\Sigma \vdash c\colon \tau} \qquad \frac{}{\Sigma \vdash *\colon 1}$$

$$\frac{\Sigma \vdash M\colon \tau \qquad \Sigma \vdash N\colon \tau'}{\Sigma \vdash \langle M, N\rangle\colon \tau \times \tau'} \qquad \frac{\Sigma \vdash M\colon \tau \times \tau'}{\Sigma \vdash \pi_1(M)\colon \tau} \qquad \frac{\Sigma \vdash M\colon \tau \times \tau'}{\Sigma \vdash \pi_2(M)\colon \tau'}$$

$$\frac{\Sigma,\, x\colon \tau \vdash M\colon \tau'}{\Sigma \vdash \lambda x{:}\tau.\, M\colon \tau \to \tau'} \qquad \frac{\Sigma \vdash M\colon \tau \to \tau' \qquad \Sigma \vdash N\colon \tau}{\Sigma \vdash M\ N\colon \tau'}$$

A *substitution* $\rho\colon \Sigma \to \Sigma'$ is a function that assigns to each variable $(x\colon \tau)$ in Σ' a term $\Sigma \vdash M\colon \tau$. We use the following notation for substitutions:

$$[M_1/x_1] \ldots [M_n/x_n]\colon \Sigma \to (x_1\colon \tau_1, \ldots, x_n\colon \tau_n)$$

We use the letters ρ and θ to range over substitutions. An *order-preserving renaming* is a substitution of the following form (note the order):

$$[y_1/x_1] \ldots [y_n/x_n]\colon (y_1\colon \tau_1, \ldots, y_n\colon \tau_n) \to (x_1\colon \tau_1, \ldots, x_n\colon \tau_n)$$

We use the letters α and β to stand for an order-preserving renaming.

We write $\mathbf{T}$ for the category having contexts as objects and substitutions as morphisms, and we write $\mathbf{L}$ for the sub-category of $\mathbf{T}$ consisting of λ-tree contexts and substitutions between them.

The semantic structure corresponding to simply-typed λ-calculus is that of a Cartesian closed category with finite products. An interpretation of the syntax in such a category $\mathbf{C}$ is given by a functor $\|-\|\colon \mathbf{T} \to \mathbf{C}$ that preserves finite products and exponents.

In order to model higher-order syntax adequately, we will interpret the types as presheaves on $\mathbf{L}$, i.e. as functors $\mathbf{L}^{\mathrm{op}} \to \mathbf{Sets}$. This means that the interpretation $\|\tau\|$ of a type τ is a mapping that assigns to each λ-tree context σ a set $\|\tau\|\sigma$ and that assigns to each substitution $\theta\colon \sigma \to \sigma'$ and each element $x \in \|\tau\|\sigma'$ an element $x[\theta] \in \|\tau\|\sigma$ in such a way that the equations $x[id] = x$ and $x[\theta \circ \rho] = x[\theta][\rho]$ hold. A term $\Sigma \vdash M\colon \tau$ is interpreted as a natural transformation $\|M\|\colon \|\Sigma\| \to \|\tau\|$. This means that the interpretation of a term is given by, for each σ, a function $\|M\|_\sigma\colon \|\Sigma\|\sigma \to \|\tau\|\sigma$, such that these functions behave well with respect to substitution, that is $(\|M\|_{\sigma'}(x))[\theta] = \|M\|_\sigma(x[\theta])$ holds for all $\theta\colon \sigma \to \sigma'$ and all $x \in \|\Sigma\|\sigma'$.

For a small category $\mathbf{C}$, we write $\widehat{\mathbf{C}}$ for the category of functors $\mathbf{C}^{\mathrm{op}} \to \mathbf{Sets}$.

The point of using presheaves is that λ-tree syntax can be modelled adequately, as has been shown in [12]. Consider for example the λ-tree type Tm with constants *app* and *lam*, as described in the introduction. For the interpretation $\|Tm\|$ we choose the presheaf $\|Tm\|(\sigma) = \{M \mid \sigma \vdash M\colon Tm\}$ with the canonical substitution action. Now, the Cartesian closed structure on $\widehat{\mathbf{L}}$ is such that there are isomorphisms $\|Tm \times Tm\|\sigma \cong \|Tm\|\sigma \times \|Tm\|\sigma$ and $\|Tm \to Tm\|\sigma \cong \|Tm\|(\sigma, x\colon Tm)$. In particular, the interpretation of the function type $Tm \to Tm$ at stage σ consists just of terms with an additional variable x. Therefore, we can interpret the terms *app* and *lam* by the maps $\|Tm\| \times \|Tm\| \to \|Tm\|$ and $\|Tm \to Tm\| \to \|Tm\|$ given by $\langle M, N\rangle \mapsto app\langle M, N\rangle$ and $M \mapsto lam(\lambda x{:}Tm.\, M)$ respectively.

The reader unfamiliar with the presheaf semantics may find it instructive to think of the term model (up to α-equivalence), in which all types are interpreted by $\|\tau\|\sigma = \{M \mid \sigma \vdash M\colon \tau\}$.

1.2 Classical First-order Logic with ∇

Logical formulae in context Σ can be defined using a base type o and, for each relation symbol R, a constant $R\colon \tau_1 \to \cdots \to \tau_n \to \mathrm{o}$, as well as constants for the logical connectives $\neg\colon \mathrm{o} \to \mathrm{o}$ and $\vee, \wedge\colon \mathrm{o} \times \mathrm{o} \to \mathrm{o}$ and $\forall, \exists\colon (\tau \to \mathrm{o}) \to \mathrm{o}$ and $\nabla\colon (\iota \to \mathrm{o}) \to \mathrm{o}$ etc. Although these constants can be interpreted in our semantics, it is simpler to consider logical formulae as a separate entity with the evident inductive definition, which is the view we adopt in this section.

We define a sequent calculus for a classical logic with ∇. The sequents have the form $\Sigma \mid \Gamma \longrightarrow \Delta$, in which Σ is a context and Γ and Δ are (possibly infinite) sets of formulae in a local signature $\sigma \rhd A$. A local signature σ is a λ-tree context and $\sigma \rhd A$ in context Σ presupposes that A is a well-formed formula in context Σ, σ. We

General Rules

$$(\textsc{Axiom})\ \frac{}{\Sigma \mid \Gamma \longrightarrow \Delta}\ \Gamma \cap \Delta \neq \emptyset$$

$$(\textsc{Cut})\ \frac{\Sigma \mid \Gamma \longrightarrow \mathcal{B} \qquad \Sigma \mid \mathcal{B}, \Delta \longrightarrow \Phi}{\Sigma \mid \Gamma, \Delta \longrightarrow \Phi}$$

Logical Rules

$$(\bot\text{-L})\ \frac{}{\Sigma \mid \Gamma, \sigma \rhd \bot \longrightarrow \Delta} \qquad (\top\text{-R})\ \frac{}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd \top}$$

$$(\neg\text{-L})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A}{\Sigma \mid \Gamma, \sigma \rhd \neg A \longrightarrow \Delta} \qquad (\neg\text{-R})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A}{\Sigma \mid \Gamma, \sigma \rhd \neg A \longrightarrow \Delta}$$

$$(\wedge\text{-L})\ \frac{\Sigma \mid \Gamma, \sigma \rhd A, \sigma \rhd B \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd A \wedge B \longrightarrow \Delta} \qquad (\wedge\text{-R})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A \qquad \Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd B}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A \wedge B}$$

$$(\vee\text{-L})\ \frac{\Sigma \mid \Gamma, \sigma \rhd A \longrightarrow \Delta \qquad \Sigma \mid \Gamma, \sigma \rhd B \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd A \vee B \longrightarrow \Delta} \qquad (\vee\text{-R})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A, \sigma \rhd B}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A \vee B}$$

$$(\supset\text{-L})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A \qquad \Sigma \mid \Gamma, \sigma \rhd B \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd A \supset B \longrightarrow \Delta} \qquad (\supset\text{-R})\ \frac{\Sigma \mid \Gamma, \sigma \rhd A \longrightarrow \Delta, \sigma \rhd B}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A \supset B}$$

$$(\forall\text{-L})\ \frac{\Sigma, \sigma \vdash M : \tau \qquad \Sigma \mid \Gamma, \sigma \rhd A[M/x] \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd \forall x : \tau.\, A \longrightarrow \Delta} \qquad (\forall\text{-R})\ \frac{\Sigma, h \mid \Gamma \longrightarrow \Delta, \sigma \rhd A[h\ \sigma/x]}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd \forall x : \tau.\, A}$$

$$(\exists\text{-L})\ \frac{\Sigma, h \mid \Gamma, \sigma \rhd A[h\ \sigma/x] \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd \exists x : \tau.\, A \longrightarrow \Delta} \qquad (\exists\text{-R})\ \frac{\Sigma, \sigma \vdash M : \tau \qquad \Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd A[M/x]}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd \exists x : \tau.\, A}$$

$$(\nabla\text{-R})\ \frac{\Sigma \mid \Gamma, (\sigma, x : \iota) \rhd A \longrightarrow \Delta}{\Sigma \mid \Gamma, \sigma \rhd \nabla x : \iota.\, A \longrightarrow \Delta} \qquad (\nabla\text{-R})\ \frac{\Sigma \mid \Gamma \longrightarrow \Delta, (\sigma, x : \iota) \rhd A}{\Sigma \mid \Gamma \longrightarrow \Delta, \sigma \rhd \nabla x : \iota.\, A}$$

Fig. 1. Sequent calculus

identify statements $\sigma \rhd A$ up to bound renaming of variables in σ. One may think of $\sigma \rhd A$ as the formula $\nabla\sigma.\, A$. For $\cdot \rhd A$ we write just A.

The rules of the sequent calculus, which are given in Fig. 1, are a straightforward extension to classical logic of the rules in [18].

Just as in FOλ^{∇}, the ∇-quantifier commutes with all other logical connectives, i.e. one can prove equivalences of the form $(\nabla x : \iota.\, A \wedge B) \dashv\vdash (\nabla x : \iota.\, A) \wedge (\nabla x : \iota.\, B)$ and $(\nabla x : \iota.\, \exists y : \tau.\, A(x, y)) \dashv\vdash (\exists h : \iota \to \tau.\, \nabla x : \iota.\, A(x, h\ x))$ and likewise for all the other connectives. This property of ∇ makes it very similar to the new-quantifier N of Gabbay & Pitts [7]. The quantifier ∇ differs from N, however, in the fact that the following equivalences are *not* provable: $\nabla x : \iota.\, \nabla y : \iota.\, A(x, y) \dashv\vdash \nabla y : \iota.\, \nabla x : \iota.\, A(x, y)$ and $\forall x : \iota.\, A(x) \supset \nabla x : \iota.\, A(x)$ and $\nabla x : \iota.\, A(x) \supset \exists x : \iota.\, A(x)$. In Part II we will show that both ∇ and N are nevertheless instances of the same structure (Prop. 2.2).

We omit sequent rules for equality and definitions. Although these rules are important for practical reasoning and are non-trivial from a proof-theoretic perspective, for semantic purposes it is simpler to replace them by axiom-schemes.

1.3 Semantic Interpretation

To give a meaning to the logic, we must first interpret the underlying λ-calculus. To do this, we assume, for each base type τ, an object $\|\tau\|$ of $\widehat{\mathbf{L}}$. We extend this

assignment to an interpretation of the λ-calculus by the following clauses.

$$\begin{aligned}
\|\iota\|\sigma &= \mathbf{y}(x\colon \iota) \cong \{M \mid \sigma \vdash M\colon \iota\}, \text{ where } \iota \text{ is a } \lambda\text{-tree type} \\
\|1\|\sigma &= 1 \\
\|\tau \times \tau'\|\sigma &= \|\tau\|\sigma \times \|\tau'\|\sigma \\
\|\tau \to \tau'\|\sigma &= \widehat{\mathbf{L}}(\mathbf{y}(\sigma) \times \|\tau\|, \|\tau'\|)
\end{aligned}$$

Here, $\mathbf{y}(\sigma) = \mathbf{L}(-, \sigma)$ is the Yoneda-embedding. We note that, as a consequence of the Yoneda Lemma, there is an isomorphism $\|\iota \to \tau\|(\sigma) \cong \|\tau\|(\sigma,\, x\colon \iota)$ for all λ-tree types ι. We extend the interpretation to contexts by use of Cartesian products.

Propositions are interpreted as subsets closed under order-preserving renaming of variables. That is, a proposition A in context Σ is interpreted by, for each σ, a subset $A(\sigma) \subseteq \|\Sigma\|\sigma$, such that, for each σ, each $x \in A(\sigma)$ and each bijective variable renaming $\alpha\colon \sigma' \to \sigma$, we have $x[\alpha] \in A(\sigma')$. Note that propositions must be closed only under variable-renaming, not under all substitutions.

The interpretation of base-types, constants and relations is recorded in a structure $\mathfrak{A}$. It assigns to each base type τ an object $\|\tau\|$ in $\widehat{\mathbf{L}}$ and to each constant $c\colon \tau$ a morphism $\|1\| \to \|\tau\|$ in $\widehat{\mathbf{L}}$. It assigns to each relation symbol $R\colon \tau_1 \to \cdots \to \tau_n \to \mathrm{o}$ a proposition on $x_1\colon \tau_1, \ldots, x_n\colon \tau_n$, as described just above.

The interpretation of formulae is defined by the satisfaction relation $\sigma \Vdash_{\rho,\Sigma} A$, in which the stage σ is a λ-tree context, A is a formula in context Σ and ρ is a Σ-valuation at stage σ defined as follows: a Σ-*valuation* at stage σ is a function ρ mapping each variable $x\colon \tau$ in Σ to an element of $\|\tau\|(\sigma)$. For a term $\Sigma \vdash t\colon \tau$, we write $\rho(t)$ for the evident element of $\|\tau\|(\sigma)$.

$\sigma \Vdash_{\rho,\Sigma} R(t_1, \ldots, t_n)$ iff $\langle \rho(t_1), \ldots, \rho(t_n)\rangle \in \|R\|(\sigma)$
$\sigma \Vdash_{\rho,\Sigma} \bot$ never
$\sigma \Vdash_{\rho,\Sigma} \top$ always
$\sigma \Vdash_{\rho,\Sigma} \neg A$ iff not $\sigma \Vdash_{\rho,\Sigma} A$
$\sigma \Vdash_{\rho,\Sigma} A \wedge B$ iff $\sigma \Vdash_{\rho,\Sigma} A$ and $\sigma \Vdash_{\rho,\Sigma} B$
$\sigma \Vdash_{\rho,\Sigma} A \vee B$ iff $\sigma \Vdash_{\rho,\Sigma} A$ or $\sigma \Vdash_{\rho,\Sigma} B$
$\sigma \Vdash_{\rho,\Sigma} A \supset B$ iff $\sigma \Vdash_{\rho,\Sigma} A$ implies $\sigma \Vdash_{\rho,\Sigma} B$
$\sigma \Vdash_{\rho,\Sigma} \exists x\colon \tau.\, A$ iff there exists $e \in \|\tau\|(\sigma)$ such that $\sigma \Vdash_{\rho[e/x],(\Sigma,\, x\colon \tau)} A$
$\sigma \Vdash_{\rho,\Sigma} \forall x\colon \tau.\, A$ iff $\sigma \Vdash_{\rho[e/x],(\Sigma,\, x\colon \tau)} A$ for all $e \in \|\tau\|(\sigma)$
$\sigma \Vdash_{\rho,\Sigma} \nabla x\colon \iota.\, A$ iff $\sigma,\, x\colon \iota \Vdash_{\rho[x/x],\,(\Sigma,\, x\colon \iota)} A$

The interpretation of formulae is extended to local signatures by:

$$\sigma \Vdash_{\rho,\Sigma} (x_1\colon \iota_1, \ldots, x_n\colon \iota_n) \rhd A \iff^{\text{def}} \sigma \Vdash_{\rho,\Sigma} \nabla x_1\colon \iota_1.\, \ldots \nabla x_n\colon \iota_n.\, A$$

For a set of formulae Γ, we define $\sigma \Vdash_{\rho,\Sigma} \Gamma$ as an abbreviation for $\bigwedge_{G\in\Gamma}(\sigma \Vdash_{\rho,\Sigma} G)$. A formula A is *valid in* $\mathfrak{A}$ if, for all Σ-valuations ρ at the empty stage, $\cdot \Vdash_{\rho,\Sigma} A$ holds. A sequent $\Sigma \mid \Gamma \longrightarrow \Delta$ is *valid in* $\mathfrak{A}$ if, for all Σ-valuations ρ at the empty

stage, $\cdot \Vdash_{\rho,\Sigma} \Gamma$ implies the existence of a $D \in \Delta$ such that $\cdot \Vdash_{\rho,\Sigma} D$ holds. A structure $\mathfrak{A}$ is a *model* of a set of closed formulae Γ, if all $G \in \Gamma$ are valid in the interpretation relative to $\mathfrak{A}$. A formula (respectively sequent) is *valid* if it is valid in all structures $\mathfrak{A}$.

1.4 Examples

Lambda calculus and its induction principle. As an example of an induction principle that can be justified in the semantics, we consider the induction principle for the untyped λ-calculus Tm. The following induction schema is valid for all formulae P.

$$\Sigma \mid \begin{array}{l} \forall t, t' \colon Tm.\ P(t) \wedge P(t') \supset P(app\ \langle t, t' \rangle), \\ \forall f \colon (Tm \to Tm).\ (\forall t \colon Tm.\ P(t) \supset P(f\ t)) \supset P(lam\ f) \end{array} \longrightarrow \forall t \colon Tm.\ P(t)$$

That this induction schema is indeed valid follows because the definition of validity uses valuations at the empty stage, so that the quantification in this schema is over closed terms only. It is also possible to quantify over terms with free variables in σ by introducing a local signature σ.

$$\Sigma \mid \begin{array}{l} \sigma \rhd \forall t \colon Tm.\ isVar(t) \supset P(t), \\ \sigma \rhd \forall t, t' \colon Tm.\ P(t) \wedge P(t') \supset P(app\ t\ t'), \\ \sigma \rhd \forall f \colon (Tm \to Tm).\ (\forall t \colon Tm.\ P(t) \supset P(f\ t)) \supset P(lam\ f) \end{array} \longrightarrow \sigma \rhd \forall t \colon Tm.\ P(t)$$

Here, $isVar$ is the predicate expressing that t is a variable. It is interpreted by $\|isVar\|\sigma = \{x \mid \sigma \vdash x \colon Tm,\ x \text{ is a variable}\}$. Note that the interpretation of $isVar$ is closed under order-preserving renaming but not under all substitutions.

Finally, if we let

$$\Gamma_\sigma = \begin{array}{l} \sigma \rhd \forall t \colon Tm.\ isVar(t) \supset P(t), \\ \sigma \rhd \forall t, t' \colon Tm.\ P(t) \wedge P(t') \supset P(app\ t\ t'), \\ \sigma \rhd \forall f \colon (Tm \to Tm).\ (\nabla x \colon Tm.\, P(f\ x)) \supset P(lam\ f) \end{array}$$

and let Γ be the union of all Γ_σ for all σ, then the sequent $\Sigma \mid \Gamma \longrightarrow \sigma' \rhd \forall t \colon Tm.\ P(t)$ is valid for all σ'.

Standard classical logic. The well-known Tarski-style semantics for classical logic is a special case of the above definition. Consider the case where there are no λ-tree types. By the syntactic restrictions, no ∇-quantification is allowed in this case. Furthermore, $\widehat{\mathbf{L}}$ is just the category of sets. Hence, in this case, the above definition of the satisfaction relation coincides with the well-known standard interpretation of classical logic.

1.5 Soundness

For soundness, we first show that the ∇-quantifier commutes with all the other logical connectives. With this property, the proof of soundness is a straightforward induction on derivations. The soundness proof of Miculan & Yemane [17] is similar.

Lemma 1.1 *For any $\theta\colon \Sigma \to \Sigma'$, we have $\sigma \Vdash_{(\theta\circ\rho),\Sigma'} A \iff \sigma \Vdash_{\rho,\Sigma} A[\theta]$.*

Proof. By induction on the structure of the formula A. The only interesting case is the base case, which follows because we have $(\theta \circ \rho)(t) = \rho(t[\theta])$. □

We show that ∇ commutes with the quantifiers, omitting the similar cases for the other connectives for space-reasons.

Lemma 1.2 *The following equivalences hold.*

$$\begin{array}{lcl} \sigma \Vdash_{\rho,\Sigma} \nabla x\colon \iota.\, \exists y\colon \tau.\, A & \iff & \sigma \Vdash_{\rho,\Sigma} \exists h\colon \iota \to \tau.\, \nabla x\colon \iota.\, A[h\ x/y] \\ \sigma \Vdash_{\rho,\Sigma} \nabla x\colon \iota.\, \forall y\colon \tau.\, A & \iff & \sigma \Vdash_{\rho,\Sigma} \forall h\colon \iota \to \tau.\, \nabla x\colon \iota.\, A[h\ x/y] \end{array}$$

Proof. We consider the case for the existential quantifier. The other case is dual.

$$\begin{array}{l} \sigma \Vdash_{\rho,\Sigma} \nabla x\colon \iota.\, \exists y\colon \tau.\, A \iff \sigma,\, x\colon \iota \Vdash_{\rho[x/x],\Sigma,\, x\colon \iota} \exists y\colon \tau.\, A \\ \quad \iff \text{exists } e \in \|\tau\|(\sigma,\, x\colon \iota) \text{ with } \sigma,\, x\colon \iota \Vdash_{\rho[x/x,e/y],\Sigma,\, x\colon \iota,\, y\colon \tau} A \end{array}$$

Consider the substitution $[h\ x/y]\colon (\Sigma,\, h\colon (\iota \to \tau),\, x\colon \iota) \to (\Sigma,\, x\colon \iota,\, y\colon \tau)$. By use of the isomorphism $i\colon \|\tau\|(\sigma,\, x\colon \iota) \cong \|\iota \to \tau\|\sigma$, we obtain from e the element $i(e)$ of $\|\iota \to \tau\|\sigma$. The interpretation of application in $\widehat{\mathbf{L}}$ is such that we have $[h\ x/y] \circ \rho[i(e)/h, x/x] = \rho[x/x, e/y]$. Using the above lemma, we can continue as follows.

$$\begin{array}{l} \text{exists } e \in \|\tau\|(\sigma,\, x\colon \iota) \text{ with } \sigma,\, x\colon \iota \Vdash_{\rho[x/x,e/y],\Sigma,\, x\colon \iota,\, y\colon \tau} A \\ \quad \iff \text{exists } e \in \|\tau\|(\sigma,\, x\colon \iota) \text{ with } \sigma,\, x\colon \iota \Vdash_{\rho[i(e)/h,x/x],\Sigma,\, h\colon (\iota\to\tau),\, x\colon \iota} A[h\ x/y] \\ \quad \iff \text{exists } e' \in \|\iota \to \tau\|\sigma \text{ with } \sigma,\, x\colon \iota \Vdash_{\rho[e'/h,x/x],\Sigma,\, h\colon (\iota\to\tau),\, x\colon \iota} A[h\ x/y] \\ \quad \iff \text{exists } e' \in \|\iota \to \tau\|\sigma \text{ with } \sigma \Vdash_{\rho[e'/h],\Sigma,\, h\colon (\iota\to\tau)} \nabla x\colon \iota.\, A[h\ x/y] \\ \quad \iff \sigma \Vdash_{\rho,\Sigma} \exists h\colon (\iota \to \tau).\, \nabla x\colon \iota.\, A[h\ x/y] \end{array}$$

□

Using this lemma, we obtain by induction on derivations:

Proposition 1.3 (Soundness) *Any derivable sequent is valid.*

1.6 Completeness

In this section we show that the well-known completeness argument for first-order logic, see e.g. [5,11], goes through almost unchanged.

Definition 1.4

(i) A *theory* T is a set of closed formulae such that $(\cdot \mid T \longrightarrow A)$ implies $A \in T$

(ii) A theory T is syntactically consistent if $\cdot \mid T \longrightarrow \bot$ is not derivable.

(iii) A theory T is a *Henkin Theory* if, for each closed formula $\exists x\colon \tau.\, A(x)$, there exists a constant $\cdot \vdash c\colon \tau$ such that $(\exists x\colon \tau.\, A(x) \supset A(c)) \in T$ holds.

Lemma 1.5 *Let T be a syntactically consistent theory for the signature $\mathcal{S}$. There exists a signature $\mathcal{S}^*$ extending $\mathcal{S}$ with countably many new constants and a theory T^* for the signature $\mathcal{S}^*$, such that the following hold:*

(i) *T^* is conservative over T;*

(ii) *T^* is a Henkin theory.*

Notice that even though we are extending the signature with closed witnesses only, by $\|\tau\|(\sigma) \cong \|\sigma \to \tau\|(\cdot)$ we can reach all stages of the presheaves. Moreover, suppose we have $\nabla\sigma.\, \exists x\colon \tau.\, A(x)$. This is equivalent to $\exists h\colon \iota \to \tau.\, \nabla\sigma.\, A(h\ \sigma)$, so that we have a witness c making $\nabla\sigma.\, A(c\ \sigma)$ true.

Lemma 1.6 *For any syntactically consistent theory T, there exists a maximally consistent extension $T^* \supseteq T$ for which the following hold:*

(i) *$A \notin T^*$ iff $\neg A \in T^*$;*

(ii) *$A \wedge B \in T^*$ iff $A \in T^*$ and $B \in T^*$;*

(iii) *$A \vee B \in T^*$ iff $A \in T^*$ or $B \in T^*$;*

(iv) *$A \supset B \in T^*$ iff $\{\neg A, B\} \cap T^* \neq \emptyset$;*

Lemma 1.7 *If T^* is a maximally consistent extension of T and T is a Henkin theory then T^* is also a Henkin theory.*

Lemma 1.8 *Any syntactically consistent set Γ has a model.*

Proof. Let T be the theory axiomatised by Γ and extend it to a maximally consistent Henkin theory T^*. We define a model from T^*. Base types are interpreted by $\|\tau\|(\sigma) = \{M \mid \sigma \vdash M\colon \tau\}$ and the presheaf action is given by substitution. A relation $R\colon \tau_1 \to \cdots \to \tau_n \to o$ is interpreted by

$$\|R\|(\sigma) = \{\langle t_1, \ldots, t_n\rangle \mid \nabla\sigma.\, R(t_1, \ldots, t_n) \in T^*\} \tag{1}$$

Notice that this presheaf $\|R\|$ only has to be closed under variable renaming, not under all substitutions. That it is indeed closed follows because $\nabla\sigma.\, R(t_1, \ldots, t_n)$ is closed under α-conversion.

It remains to show that this definition does indeed define a model of T^*. It suffices to show the equivalence $\cdot \Vdash_{\rho,\Sigma} A \iff A[\rho] \in T^*$ for all stages σ, all contexts Σ, all formulae A in context Σ and all Σ-valuations ρ at stage σ. The assertion follows from this, since, by letting Σ be the empty context, it can be seen that each formula $A \in T^*$ is valid in the above model.

The proof of the equivalence goes by induction on the number of logical connectives other than ∇ in a formula A. Since we have the well-known equivalences of

classical logic, we can restrict our attention to the connectives $\neg$, $\vee$ and $\exists$. We proceed by case-distinction on the outermost connective in A. We show the base-case and the case for the existential quantifier. The other cases are similar.

- A is $\nabla\sigma.\, R(t_1, \ldots, t_n)$. Let $\rho' = \rho[\sigma/\sigma]$. Then we have:

$$\begin{aligned} \cdot \Vdash_{\rho,\Sigma} \nabla\sigma.\, R(t_1, \ldots, t_n) &\iff \sigma \Vdash_{\rho',(\Sigma,\sigma)} R(t_1, \ldots, t_n) \\ &\iff \langle \rho'(t_1), \ldots, \rho'(t_n)\rangle \in \|R\|(\sigma) \\ &\iff \nabla\sigma.\, R(t_1[\rho'], \ldots, t_n[\rho']) \in T^* \\ &\iff (\nabla\sigma.\, R(t_1, \ldots, t_n))[\rho] \in T^* \end{aligned}$$

- A is $\nabla\sigma.\, B$, where B is neither of the form $\nabla x\colon \iota.\, C$ nor $R(t_1, \ldots, t_n)$. We consider the representative case where B is $\exists x\colon \tau.\, B'$. Since the formula A is provably equivalent to $\exists h.\, \nabla\sigma.\, B[h\ \sigma/x]$, this case is handled by that for $\exists$ below.
- A is $\exists x\colon \tau.\, B$. From left to right, $\cdot \Vdash_{\rho,\Sigma} \exists x\colon \tau.\, B$ gives $e \in \|\tau\|\cdot$ with $\cdot \Vdash_{\rho[e/x],(\Sigma,\, x\colon \tau)} B$. By induction hypothesis this implies $B[\rho[e/x]] \in T^*$. Using rule $\exists$-R, we derive $\cdot \mid T^* \longrightarrow (B[\rho][e/x]) \supset ((\exists x\colon \tau.\, B)[\rho])$. By maximality of T^*, we get $(\exists x\colon \tau.\, B)[\rho] \in T^*$.

  From right to left, suppose $(\exists x\colon \tau.\, B)[\rho] \in T^*$. Since T^* is a Henkin theory, there exists $c \in \|\tau\|\cdot$ such that $((\exists x\colon \tau.\, B)[\rho] \supset B[\rho][e/x]) \in T^*$. By modus ponens and maximality, $B[\rho][e/x] \in T^*$. By induction hypothesis, $\cdot \Vdash_{\rho[e/x],(\Sigma,\, x\colon \tau)} B$. By definition of $\Vdash$, we get the required $\cdot \Vdash_{\rho,\Sigma} \exists x\colon \tau.\, B$.

□

Using this lemma, completeness now follows by a standard argument [5,11].

Proposition 1.9 (Completeness) *Any valid sequent* $\cdot \mid \Gamma \longrightarrow A$ *is derivable.*

2 Part II
Models in Categories with Binding Structure

In Part I of this paper we have given a simple model of classical first-order logic with ∇ that is very close to classical Tarski-semantics. In the second part we generalise this result by giving a general notion of model in *categories with binding structure*. We describe these models in the language of categorical logic, which we assume the reader to be familiar with, see e.g. [14] for an introduction.

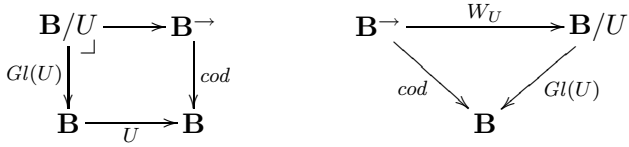
To simplify the presentation, we assume from now on that there is only a single λ-tree type Tm. In this case, the objects of $\mathbf{L}$ may be identified with finite sets. We write Tm also for the presheaf in $\mathbf{L}$ that interprets the type Tm.

2.1 Categories with Binding Structure

Categories with binding structure [20, Chapter 10] axiomatise binding in a general way. The definition of binding structure can be seen as a direct formalisation of the statement: 'Working with an α-equivalence class is the same as working with a

freshly named instance'. This statement, expressing that two modes of working are equivalent, is being formalised directly as an equivalence of two categories (Def. 2.1).

Let $\mathbf{B}$ be a category with finite limits and consider its internal language, i.e. the codomain fibration on $\mathbf{B}$. The aim is to explain that constructions with ordinary judgements in the internal language of $\mathbf{B}$ are essentially the same as constructions with judgements that may make use of a fresh name. To formalise 'judgements with a fresh name' we use the glueing construction, see e.g. [24]. Given an endofunctor U on $\mathbf{B}$, consider the pullback as in the left diagram below. In this diagram, $\mathbf{B}^{\to}$ has as objects the morphisms of $\mathbf{B}$, and a morphism in $\mathbf{B}^{\to}$ from $f\colon A \to B$ to $g\colon C \to D$ is a pair $\langle u\colon A \to C, v\colon B \to D\rangle$ of $\mathbf{B}$-morphisms for which $v \circ f = g \circ u$ holds. The functor *cod* maps objects to their codomain and morphisms $\langle u, v\rangle$ to v. The category $\mathbf{B}/U$ has as objects the $\mathbf{B}$-morphisms of the form $f\colon A \to UB$. Its morphisms from $f\colon A \to UB$ to $g\colon C \to UD$ are pairs $\langle u\colon A \to C, v\colon B \to D\rangle$ of $\mathbf{B}$-morphisms for which $Uv \circ f = g \circ u$ holds. The functor $Gl(U)$ maps $f\colon A \to UB$ to B and $\langle u, v\rangle$ to v. Both *cod* and $Gl(U)$ are fibrations. There is an canonical functor $W_U\colon \mathbf{B}^{\to} \to \mathbf{B}/U$ making the triangle on the right below commute. Specifically, W_U maps an object $f\colon A \to B$ to $Uf\colon UA \to UB$.



We use the glued fibration $Gl(- \otimes V)$ for talking about judgements with a fresh name, where $\otimes$ is a monoidal structure and V is an object of $\mathbf{B}$. The intuition is that $A \otimes B$ consists of pairs whose components do not share names, and V is an object of names. Then, the functor $W_{(-\otimes V)}$ adds a fresh name to a judgement. We write short W_V for it.

With this notation, the definition of a category with binding structure is simple:

Definition 2.1 A *category with binding structure* is a triple $(\mathbf{B}, \otimes, V)$ consisting of a category $\mathbf{B}$ with finite limits, a monoidal structure $\otimes$ on $\mathbf{B}$ and an object V of $\mathbf{B}$, such that the functor $W_{(-\otimes V)}$ is an equivalence of fibrations.

Instances of categories with binding structure are given in [20, Chapter 10]. A prime example is the category of nominal sets (also known as the Schanuel topos, or FM-Sets) [7], where $A \otimes B = \{\langle a, b\rangle\colon A \times B \mid a\#b\}$ and V is the set of atoms.

To understand the motivation for Def. 2.1, recall that in categorical logic existential quantification is modelled by a left adjoint to weakening and, dually, universal quantification is modelled by a right adjoint to weakening. The functor W_V can be thought of as a non-standard 'weakening' functor. Since it is an equivalence, there is a functor H that is both left and right adjoint to W_V. By the view of quantifiers as adjoints to weakening, H can be viewed as a non-standard quantifier that is both an existential and a universal quantifier at the same time. It can be shown that H directly generalises the new-quantifier N of Gabbay & Pitts. The defining feature of H

is that it preserves all categorical constructions, e.g. $\mathrm{H}(A \Rightarrow B) \cong (\mathrm{H}A) \Rightarrow (\mathrm{H}B)$, which follows because H is part of an equivalence. This suggests a relation to ∇, which has the same defining feature, e.g. $\nabla x.\,(A \supset B) \cong (\nabla x.\,A) \supset (\nabla x.\,B)$.

In [21] and [20], the structure of categories with binding structure used as the basis of a dependent type theory.

Categories with binding structure have rich structure, see [20, Chapter 10]. The properties we use in this paper are given by the next three propositions from [20].

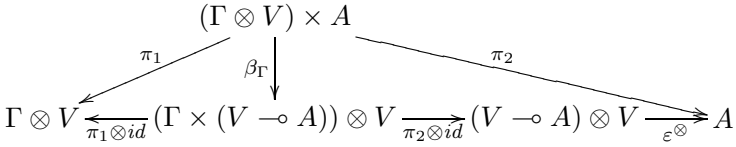
Proposition 2.2 *For each category with binding structure $(\mathbf{B}, \otimes, V)$, there exists a functor $\mathsf{M}\colon \mathbf{Sub}(\Gamma \otimes V) \to \mathbf{Sub}(\Gamma)$ that is left and right adjoint to $W_V\colon \mathbf{Sub}(\Gamma) \to \mathbf{Sub}(\Gamma \otimes V)$.*

In the statement of this proposition we have used the fact that W_V, being a right adjoint, preserves monomorphisms and so restricts to a functor on subobjects.

Proposition 2.3 *In each category with binding structure $(\mathbf{B}, \otimes, V)$, the functor $(-) \otimes V$ has a right adjoint $V \multimap (-)$, which itself has a further right adjoint. We write $\varepsilon^{\otimes}$ for the co-unit of the adjunction $(-) \otimes V \dashv V \multimap (-)$.*

If $\mathbf{B}$ is the category of nominal sets then $(V \multimap X)$ is (isomorphic to) the abstraction set $[\mathbb{A}]X$ of Gabbay & Pitts. The application map $(V \multimap X) \otimes V \to X$ amounts to the concretion operation $x@a$. The binding operation, mapping $a \in \mathbb{A}$ and $x \in X$ to $a.x \in [\mathbb{A}]X$ with $a\#(a.x)$ and $(a.x)@a = x$, is given by β in the next proposition.

Proposition 2.4 *In each category with binding structure $(\mathbf{B}, \otimes, V)$, there is a natural transformation $\beta\colon ((-) \otimes V) \times A \to ((-) \times (V \multimap A)) \otimes V$ making the following diagram commute for all objects Γ.*



This proposition has its origin in the work of Menni [16].

2.2 Modelling ∇ in a Category with Binding Structure

As in Part I, we interpret the λ-calculus in $\widehat{\mathbf{L}}$. As outlined in the introduction, the internal logic of this category is not appropriate for modelling a logic with the ∇-quantifier. Instead, we use the internal logic of a category with binding structure $\mathbf{B}$, which we transfer to $\widehat{\mathbf{L}}$ by reindexing along a functor $F\colon \widehat{\mathbf{L}} \to \mathbf{B}$, as in the following change-of-base situation. This idea has been used with different categories by Hofmann [12] to model the Theory of Contexts [13], see also [3].

$$\begin{array}{ccc} \mathbf{E} & \longrightarrow & \mathbf{Sub}(\mathbf{B}) \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle sub} \\ \widehat{\mathbf{L}} & \xrightarrow{F} & \mathbf{B} \end{array} \tag{2}$$

We need the subobject logic on $\mathbf{B}$ to be strong enough to provide a model for at least first-order logic. Hence, we assume *sub* to be a first-order fibration:

Definition 2.5 A fibration $q\colon \mathbf{E} \to \mathbf{B}$ is a *first-order fibration* if the following hold.

(i) Each fibre $\mathbf{E}_\Gamma$ is a preorder with finite products $(\top, \wedge)$, coproducts $(\bot, \vee)$ and exponents $(\supset)$.

(ii) For each map $u\colon \Gamma \to \Delta$, the reindexing functor u^* has a left adjoint $\exists_u$ that satisfies the Beck-Chevalley and Frobenius conditions.

(iii) For each map $u\colon \Gamma \to \Delta$, the reindexing functor u^* has a right adjoint $\forall_u$ that satisfies the Beck-Chevalley condition.

We refer to e.g. [14] for a definition of the Beck-Chevalley and Frobenius conditions.

To model ∇-logic in the fibration p defined in (2), we use the following structure.

Definition 2.6 A ∇-*model* consists of a category with binding structure $(\mathbf{B}, \otimes, V)$ and a functor $F\colon \widehat{\mathbf{L}} \to \mathbf{B}$ with the following additional structure.

(i) The subobject fibration on $\mathbf{B}$ is a first-order fibration.

(ii) The functor F preserves finite limits and has a right-adjoint. We use the notation $\delta_{A,B}\colon FA \times FB \to F(A \times B)$ for the natural isomorphism witnessing product-preservation of F.

(iii) There are a distinguished morphism $\eta\colon V \to F\,Tm$ and natural transformations $\imath\colon FA \otimes FB \to FA \times FB$ and $\theta\colon (V \multimap FA) \to F(Tm \Rightarrow A)$ in $\mathbf{B}$ for which the following diagram commutes.

$$\begin{array}{ccccc} (V \multimap FA) \otimes V & \xrightarrow{\theta \otimes id} & F(Tm \Rightarrow A) \otimes V & \xrightarrow{f_{Tm \Rightarrow A}} & F((Tm \Rightarrow A) \times Tm) \\ {\scriptstyle F\varepsilon^{\otimes}}\downarrow & & & & \downarrow{\scriptstyle F\varepsilon} \\ FA & = & & = & FA \end{array}$$

Here, $f_{Tm \Rightarrow A}$ is part of the natural transformation $f\colon F(-) \otimes V \to F((-) \times Tm)$ defined by $F\Gamma \otimes V \xrightarrow{\imath} F\Gamma \times V \xrightarrow{id \times \eta} F\Gamma \times F\,Tm \xrightarrow{\delta} F(\Gamma \times Tm)$.

Using η we can map abstract variables in V into the object $F\,Tm$, which encodes the object-level syntax. The morphism θ and thes diagram relate binding in the category with binding structure to higher-order abstract syntax.

Next we study the structure of the fibration p in (2). We may assume that the pullback (2) is constructed such that the fibre $\mathbf{E}_A$ over an object A in $\widehat{\mathbf{L}}$ is the partial order $\mathbf{Sub}(FA)$ of subobjects on FA in $\mathbf{B}$. Given $u\colon \Gamma \to \Delta$ in $\widehat{\mathbf{L}}$, the reindexing functor $u^*\colon \mathbf{E}_\Delta \to \mathbf{E}_\Gamma$ for p is given by $(Fu)^*\colon \mathbf{Sub}(F\Delta) \to \mathbf{Sub}(F\Gamma)$ for *sub*. It is well-known that the structure of a first-order fibration is preserved under reindexing along a functor F that preserves finite limits and has a right adjoint [12].

Proposition 2.7 *The functor $p\colon \mathbf{E} \to \widehat{\mathbf{L}}$ is a first-order fibration.*

We remark that this proposition can be extended to higher-order logic [12], and all the concrete models that we consider in this paper are indeed models of higher-

order logic. This means that it is straightforward to extend our results to defining an interpretation of the type o and to validating higher-order logic.

Definition 2.8 For each object Γ of $\widehat{\mathbf{L}}$, define the functor $\nabla_\Gamma \colon \mathbf{E}_{\Gamma \times Tm} \to \mathbf{E}_\Gamma$ to be the functor $И f_\Gamma^* \colon \mathbf{Sub}(F(\Gamma \times Tm)) \to \mathbf{Sub}(F\Gamma)$.

This definition captures the quantification over some/any fresh variable of type Tm. Since $u^* \nabla_\Delta = \nabla_\Gamma (u \times id)^*$ holds for all $u \colon \Gamma \to \Delta$ in $\widehat{\mathbf{L}}$, it is justified to write just ∇.

Now we come to the central property of ∇: it commutes with existential and universal quantification. This is given by the following generalisation of Lemma 1.2.

Lemma 2.9 *For each Γ in $\widehat{\mathbf{L}}$, we have $\nabla \exists_A = \exists_{Tm \Rightarrow A} \nabla s^*$ and $\nabla \forall_A = \forall_{Tm \Rightarrow A} \nabla s^*$, where s is the map $\langle \pi_1 \times id, \varepsilon \circ (\pi_2 \times id) \rangle \colon (\Gamma \times (Tm \Rightarrow A)) \times Tm \to (\Gamma \times Tm) \times A$.*

We remark that to prove this lemma, we make essential use of the binding map β, for moving from a quantification over A to one over $(Tm \Rightarrow A)$.

With the evident translation of formulae in the structure of the first-order fibration p, we now have the following soundness result for the intuitionistic version of the sequent calculus. With Lemma 2.9, the proof is a straightforward induction.

Proposition 2.10 *For any sequent $\Sigma \mid \Gamma \longrightarrow A$ provable in the intuitionistic sequent calculus, there is a morphism $\|\Gamma\| \to \|A\|$ in $\mathbf{E}_{\|\Sigma\|}$.*

In the next section we give concrete examples of ∇-models. These examples do, in fact, all validate classical logic.

2.3 Instances of the Interpretation

Linear Species. The semantics in Part I is an explication of the interpretation in a ∇-model of linear species. The category of linear species [1] is the presheaf category $\widehat{\mathbf{C}^{\mathrm{op}}}$, where $\mathbf{C}$ is the category of finite totally ordered sets with order-preserving bijections. Along the lines of [20, Prop. 10.3.13], $\widehat{\mathbf{C}^{\mathrm{op}}}$ can be seen to be a category with binding structure. The object V is the presheaf with $V\{x\} = \{x\}$ and $V\sigma = \emptyset$ if $|\sigma| \neq 1$. The set $(A \otimes B)\sigma$ consists of pairs $\langle x \in A\sigma_1, y \in B\sigma_2 \rangle$ with $\sigma = \sigma_1, \sigma_2$. Then, the set $(V \multimap A)\sigma$ is isomorphic to $A(\sigma, x)$.

To obtain a ∇-model in $\widehat{\mathbf{C}^{\mathrm{op}}}$, we take $F \colon \widehat{\mathbf{L}} \to \widehat{\mathbf{C}^{\mathrm{op}}}$ to be the canonical inclusion, arising because each presheaf in $\widehat{\mathbf{L}}$ is all the more a presheaf in $\widehat{\mathbf{C}^{\mathrm{op}}}$. For the map $\eta \colon V \to FTm$, we take the inclusion function that maps the variable $x \in V\{x\}$ to the term $x \colon Tm$ in $(FTm)\{x\} = \{t \mid x \colon Tm \vdash t \colon Tm\}$. Finally, we define the map $\theta \colon (V \multimap FA) \to F(Tm \to A)$ to be the isomorphism arising from $(V \multimap FA)\sigma \cong FA(\sigma, x)$ in $\widehat{\mathbf{C}^{\mathrm{op}}}$ and $(Tm \to A)\sigma \cong A(\sigma, x)$ in $\widehat{\mathbf{L}}$. Because the functor F is so simple, the conditions of Def. 2.6 are straightforward to verify.

A convenient way of working with the internal language of a topos is the Kripke-Joyal semantics, see e.g. [15]. When spelled out for the linear species model, the Kripke-Joyal semantics specialises exactly to the satisfaction relation $\Vdash$ from Sect. 1.3 (Theorem VI.7.1 of [15]). We should say, however, that the notion of validity in Sect. 1.3 differs from the standard notion of validity from categorical logic.

Although Part I can be adapted for the standard notion of validity, we have opted for the non-standard definition, since it matches better with existing work on FOλ^{∇}.

Species. Very similar to the model in linear species is that of ordinary species of structures. In this case, $\mathbf{B}$ is the presheaf category $\widehat{\mathbf{D}^{\mathrm{op}}}$, where $\mathbf{D}$ is the category of finite sets and all bijections. This model differs from the model in linear species in that the local λ-tree contexts are unordered. As a consequence, this model validates the equivalence $\nabla x.\, \nabla y.\, B(x,y) \dashv\vdash \nabla y.\, \nabla x.\, B(x,y)$. Just as for linear species, the interpretation in $\widehat{\mathbf{D}^{\mathrm{op}}}$ can be described in the style of Part I. We expect that the completeness argument can be adapted to this case.

Nominal Sets. The prototypical instance of a category with binding structure is the category of nominal sets $\mathbf{S}$, also known as the Schanuel topos, see [20, Chapter 10]. The category $\mathbf{S}$ provides an instance of Def. 2.6 if we take $F\colon \widehat{\mathbf{L}} \to \mathbf{S}$ to be the composite of the inclusion functor $I\colon \widehat{\mathbf{L}} \to \widehat{\mathbf{I}^{\mathrm{op}}}$, where $\mathbf{I}$ is the category of finite sets and injections, with the sheafification functor $\mathbf{a}\colon \widehat{\mathbf{I}^{\mathrm{op}}} \to \mathbf{S}$ with respect to the atomic topology on $\mathbf{I}^{\mathrm{op}}$. We refer to e.g. [10,15] for more information on this situation.

We use the notation of Gabbay & Pitts [7] to describe the structure of $\mathbf{S}$. Specifically, we take V to be the object of atoms $\mathbb{A}$, and we use the freshness monoidal structure $A \otimes B = \{\langle a \in A, b \in B\rangle \mid a \# b\}$. In $\mathbf{S}$, the functor $L(X) = \mathbb{A} + (X \times X) + [\mathbb{A}]X$ has an initial algebra $[var, app, lam]\colon L(T) \to T$ that can be used to represent untyped λ-terms. It can be shown that FTm is isomorphic to T, by observing that ITm is already a sheaf. We take $\eta\colon V \to FTm$ to be the map $V = \mathbb{A} \stackrel{var}{\to} T \cong FTm$. We have to omit the definition of θ for space reasons.

The interpretation of ∇ in $\mathbf{S}$ provides a translation of ∇-logic to Nominal Logic, since the internal logic of $\mathbf{S}$ is (a version of) Nominal Logic. The translation is similar to that described by Gabbay & Cheney in [6]. In particular, the definition of the functor ∇ in Def. 2.8 is such that $\nabla x.\, \varphi(x,y)$ is interpreted as $\mathit{И} n.\, \|\varphi\|(\eta(n), y)$, which agrees with the interpretation in [6]. As observed by Gabbay & Cheney, the interpretation in $\mathbf{S}$ is not complete: it validates $\forall x.\, \varphi \supset \nabla x.\, \varphi$ and $\nabla x.\, \varphi \supset \exists x.\, \varphi$.

3 Conclusion and Further Work

We have explained a simple sound and complete model for classical FOλ^{∇}, and we have worked towards identifying the essential structure of ∇ by giving an abstract model in categories with binding structure. We have shown that ∇ and $\mathit{И}$ can be modelled by the same concept of binding.

In further work, the semantics of intuitionistic FOλ^{∇} should be studied. A starting point in this direction is Cheney's sound and complete translation from FOλ^{∇} into intuitionistic nominal logic [4] together with Gabbay's complete model for intuitionistic nominal logic [8]. Perhaps the semantics from Part I can also be generalised directly to a Kripke-style model.

Regarding other related work, we conjecture that the model for FOλ^{∇} of Miculan & Yemane [17] fits in the general construction of Part II, but the details remain

to be worked out. Finally, a-logic, proposed by Gabbay & Gabbay [9], has an informal explanation that appears to be quite similar to our explanation of FOλ^{∇}, and it would be interesting to make precise the relationship.

Acknowledgement

I thank Ian Stark, Marino Miculan, James Cheney and Lennart Beringer for discussions and the referees for their comments. In particular, Ian suggested to use presheaves over finite sets with bijections.

References

[1] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*. Cambridge University Press, 1997.

[2] J. Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *TABLEAUX'05*, volume 3702 of *LNCS*, pages 78–92. Springer-Verlag, 2005.

[3] A. Bucalo, M. Hofmann, F. Honsell, M. Miculan, I. Scagnetto. Consistency of the Theory of Contexts. In *Journal of Functional Programming*, 16(3):327–395, 2006

[4] J.R. Cheney. A simpler proof theory for nominal logic. In *FOSSACS 2005*, number 3441 in LNCS, pages 379–394. Springer-Verlag, 2005.

[5] D. van Dalen. *Logic and Structure*. Springer Verlag, Berlin, 1983.

[6] M.J. Gabbay and J.R. Cheney. A sequent calculus for nominal logic. In *LICS 2004*, pages 139–148. IEEE Computer Society Press, 2004.

[7] M.J. Gabbay and A.M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.

[8] Murdoch J. Gabbay. Fresh Logic. Pending publication, July 2003.

[9] Murdoch J. Gabbay and Michael J. Gabbay. a-logic. In *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 1. College Publications, 2005.

[10] F. Gadducci, M. Miculan, and U. Montanari. About permutation algebras, (pre)sheaves and named sets. In *Higher Order and Symbolic Computation*, 2006.

[11] H. Herre. Logik. Lecture Notes, University of Leipzig, 1999.

[12] M. Hofmann. Semantical analysis of higher-order abstract syntax. In *LICS'99*, 1999.

[13] F. Honsell, M. Miculan, and I. Scagnetto. An axiomatic approach to metareasoning about nominal algebras in HOAS. In *ICALP01*, 2001.

[14] B. Jacobs. *Categorical Logic and Type Theory*. Elsevier Science, 1999.

[15] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer-Verlag, 1992.

[16] M. Menni. About $\forall$-quantifiers. *Applied Categorical Structures*, 11(5):421–445, 2003.

[17] M. Miculan and K. Yemane. A unifying model of variables and names. In *FOSSACS'05*, volume 3441 of *LNCS*, pages 170–186. Springer-Verlag, 2005.

[18] D. Miller and A. Tiu. A proof theory for generic judgments. *ACM Transactions on Computational Logic*, 6(4):749–783, 2005.

[19] A.M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[20] U. Schöpp. *Names and Binding in Type Theory*. PhD thesis, University of Edinburgh, 2006.

[21] U. Schöpp and I. Stark. A dependent type theory with names and binding. In *CSL'04*, volume LNCS of *3210*, pages 235–249. Springer-Verlag, 2004.

[22] M.R. Shinwell. *The Fresh Approach: functional programming with names and binders*. PhD thesis, University of Cambridge, 2005.

[23] M.R. Shinwell and A.M. Pitts. On a monadic semantics for freshness. *Theoretical Computer Science*, 342:28–55, 2005.

[24] P. Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999.

[25] A. Tiu. A logic for reasoning about generic judgements. In *LFMTP'06*, 2006.