A Minimal Propositional Type Theory

Mark Kaminski[1,2] Gert Smolka[1]

July 23, 2008

Abstract

Propositional type theory, first studied by Henkin, is the restriction of simple type theory to a single base type that is interpreted as the set of the two truth values. We show that two constants (falsity and implication) suffice for denotational and deductive completeness. Denotational completeness means that every value of the full set-theoretic type hierarchy can be described by a closed term. Deductive completeness is shown for a sequent-based proof system that extends a propositional natural deduction system with lambda conversion and Boolean replacement.

Keywords: higher-order logic, propositional type theory, deductive completeness

1 Introduction

Propositional type theory, first studied by Henkin [4], is the restriction of simple type theory [3, 1] to a single base type that is interpreted as the set of the two truth values. Functional types are interpreted as the full set-theoretic function spaces. As logical constants Henkin takes all identity predicates, which suffices to express the propositional connectives and the quantifiers. His deductive system is a Hilbert system whose inference rules are β and replacement of equals with equals. Henkin first shows that his language is denotationally complete, meaning that every value of the full set-theoretic type hierarchy can be denoted by a closed term. He then exploits denotational completeness

[1]Programming Systems Lab, Saarland University, Campus E1 3, 66123 Saarbrücken, Germany. E-mail: {kaminski,smolka}@ps.uni-sb.de

[2]Corresponding author. Fax: +49-681-3025615

to show deductive completeness. Deciding validity of formulas in Henkin's language requires nonelementary time [7].

It turns out that only two constants, falsity and implication, suffice for denotational completeness [6]. This result raises the question for a likewise minimal proof system. We answer this question in this paper. As a basis we take a sequent-based natural deduction system for propositional logic with falsity and implication. We show that with two additional rules one obtains a complete deduction system. The first rule incorporates lambda conversion (α, β, η). The second rule provides for Boolean replacement of equals with equals where variable capture is admissible in some cases.

In a previous paper [6], we give a complete proof system for propositional type theory with falsity and implication. This system is an instance of an equational proof system for the pure lambda calculus. As such it is far from minimal since it provides for equations and replacement at all types. In contrast, the system of the present paper has equations and replacement only for the type of truth values.

The paper is organized as follows. We start by defining the syntax and semantics of our language. Then we show denotational completeness. Finally we establish a sequent-based proof system and show its completeness.

2 Terms, Formulas, and Validity

We assume familiarity with the simply typed lambda calculus (see, e.g., [5]). *Types* (σ, τ, ρ) are obtained from a single *base type* B (for bool) according to $\sigma ::= B \mid \sigma\sigma$. We think of $\sigma\tau$ as the type of functions from σ to τ. *Terms* (s, t, u) are obtained from *names* (x, y, z, f, g) according to $s ::= x \mid \lambda x.s \mid ss$. We assume a *typing relation* $s : \sigma$ satisfying the following properties:

1. For every term s there is at most one type σ such that $s : \sigma$.
2. For every type σ there are infinitely many names x such that $x : \sigma$.
3. For all x, s, σ, τ: $\lambda x.s : \sigma\tau \iff x : \sigma \wedge s : \tau$.
4. For all s, t, σ: $s\,t : \sigma \iff \exists\tau\colon\ s : \tau\sigma \wedge t : \tau$.

A term σ is *well-typed* if there is a type σ such that $s : \sigma$. We only consider well-typed terms. We use Λ to denote the set of all well-typed terms. We omit parentheses according to $\sigma\tau\rho \rightsquigarrow \sigma(\tau\rho)$ and $stu \rightsquigarrow (st)u$.

Terms of type B are called *formulas*. We fix two names $\bot\colon B$ and $\to\colon BB$ and call them *constants*. All other names are called *variables*. In a term $\lambda x.s$, the bound name x must be a variable. We use $\mathcal{V}s$ to denote the set of all variables that *occur free in* s. A term s is *closed* if $\mathcal{V}s = \emptyset$.

Contexts are obtained according to $C ::= [] \mid \lambda x.C \mid C\,s \mid s\,C$. The notation $C[s]$ describes the term obtained by replacing the hole $[]$ of C with s (capturing is ok). We assume a *substitution operation* s^x_t that yields for s, x, t a term that can be obtained from s by *capture-free substitution* of t for x, possibly after renaming of bound variables. *Lambda equivalence* $\sim_\lambda$ is the least equivalence relation on Λ that satisfies the following properties:

(α) $\lambda x.s \sim_\lambda \lambda y.s^x_y$ if $y \notin \mathcal{V}s$

(β) $(\lambda x.s)t \sim_\lambda s^x_y$

(η) $\lambda x.sx \sim_\lambda s$ if $x \notin \mathcal{V}s$

(γ) if $s \sim_\lambda t$, then $C[s] \sim_\lambda C[t]$

It is easy to see that α is subsumed by the other properties. A term s is a *subterm* of a term t if there exists a context C such that $C[s] = t$. A *β-redex* is a term of the form $(\lambda x.s)t$. A term is *β-normal* if none of its subterms is a β-redex. The following fact is well-known [5].

Proposition 1 *For every term s there exists a lambda equivalent term t such that t is β-normal and satisfies $\mathcal{V}t \subseteq \mathcal{V}s$.*

An *interpretation* is a function $\mathcal{I}$ that maps every type to a nonempty set and every name $x : \sigma$ to an element of $\mathcal{I}\sigma$. We require $\mathcal{I}B = \{0,1\}$, $\mathcal{I}\bot = 0$ and $\mathcal{I}(\to)ab = \mathsf{if}\ a{=}0\ \mathsf{then}\ 1\ \mathsf{else}\ b$ for all $a, b \in \{0,1\}$. We will only consider standard interpretations, that is, interpretations that map a functional type $\sigma\tau$ to the set of all total functions from $\mathcal{I}\sigma$ to $\mathcal{I}\tau$.

Every interpretation $\mathcal{I}$ can be extended uniquely to a function $\hat{\mathcal{I}}$ that maps every term $s : \sigma$ to an element of $\mathcal{I}\sigma$ and treats applications and abstractions as one would expect. An interpretation *$\mathcal{I}$ satisfies a formula s* if $\hat{\mathcal{I}}s = 1$. A formula is *valid* if it is satisfied by every interpretation.

3 Denotational Completeness

We fix an interpretation $\mathcal{B}$. We have $\mathcal{I}\sigma = \mathcal{B}\sigma$ for every interpretation $\mathcal{I}$ and every type σ. Moreover, we have $\hat{\mathcal{I}}s = \hat{\mathcal{B}}s$ for every closed term s and every

interpretation $\mathcal{I}$. We will show that our language is *denotationally complete*, that is, for every type σ and every value $a \in \mathcal{B}\sigma$ there is a closed term $s : \sigma$ such that $\hat{\mathcal{B}}s = a$.

It is well-known that implication and falsity can express the usual propositional connectives:

$$\begin{array}{lcllcl} \top & := & \bot \to \bot & \qquad s \vee t & := & (s \to t) \to t \\ \neg s & := & s \to \bot & \qquad s \wedge t & := & \neg(\neg s \vee \neg t) \\ & & & \qquad s \equiv t & := & (s \to t) \wedge (t \to s) \end{array}$$

Note that Boolean equivalence $s \equiv t$ is Boolean identity (i.e., $\mathcal{I}s = \mathcal{I}t$ iff $\mathcal{I}$ satisfies $s \equiv t$). Notationally, we assume the operator precedence $\equiv$, $\to$, $\vee$, $\wedge$, $\neg$ where $\neg$ binds strongest.

To show denotational completeness, we will define a family of *quote functions* $\downarrow_\sigma : \mathcal{B}\sigma \to \Lambda_0^\sigma$ where Λ_0^σ is the set of all closed terms of type σ. The quote functions will satisfy $\hat{\mathcal{B}}(\downarrow_\sigma a) = a$ for all $a \in \mathcal{B}\sigma$ and all types σ.

The quote functions are defined by recursion on types. The definition of the basic quote function $\downarrow_B$ is straightforward. To explain the definition of the other quote functions, we consider the special case $\downarrow_{\sigma B}$. We start with

$$\downarrow_{\sigma B}(a) \;=\; \lambda x. \bigvee_{\substack{b \in \mathcal{B}\sigma \\ ab=1}} x \doteq_\sigma (\downarrow_\sigma b)$$

It remains to define a closed term $\doteq_\sigma$ that denotes the identity predicate for $\mathcal{B}\sigma$. If $\sigma = B$, $\lambda xy.\, x \equiv y$ does the job. If $\sigma = \sigma_1\sigma_2$, we rely on recursion and define

$$\doteq_\sigma \;=\; \lambda fg. \bigwedge_{a \in \mathcal{B}\sigma_1} f(\downarrow_{\sigma_1} b) \doteq_{\sigma_2} g(\downarrow_{\sigma_1} b)$$

Figure 1 shows the full definition of the quote functions. A disjunction with an empty index set denotes $\bot$, and a conjunction with an empty index set denotes $\top$. The notations $\doteq_\sigma$ and $\forall_\sigma$ defined in the figure will be used in the following. We write $\forall_\sigma x.\, s$ for $\forall_\sigma(\lambda x.s)$. The notational operator $\doteq_\sigma$ will be used with a precedence higher than $\neg$ (i.e, $\neg\, s \doteq_\sigma t$ stands for $\neg(s \doteq_\sigma t)$). The following results are all straightforward consequences of the definitions in Figure 1.

Proposition 2 *The terms $\downarrow_\sigma a$, $\forall_\sigma$ and $\doteq_\sigma$ are closed.*

Proposition 3 *Let σ be a type, $f \in \mathcal{B}(\sigma B)$, and $a, b \in \mathcal{B}\sigma$. Then:*

1. $\hat{\mathcal{B}}(\downarrow_\sigma a) = a$

$$\downarrow_{\sigma_1 \ldots \sigma_n B} a \;:=\; \lambda x_1 \ldots x_n. \bigvee_{\substack{\langle b_i \in \mathcal{B}\sigma_i \rangle \\ a b_1 \ldots b_n = 1}} \bigwedge_{1 \le j \le n} x_j \doteq_{\sigma_j} (\downarrow_{\sigma_j} b_j) \qquad \text{D}\downarrow$$

$$\forall_\sigma \;:=\; \lambda f. \bigwedge_{a \in \mathcal{B}\sigma} f(\downarrow_\sigma a) \qquad \text{D}\forall$$

$$\doteq_B \;:=\; \lambda xy.\ x \equiv y \qquad \text{D}\doteq$$

$$\doteq_{\sigma\tau} \;:=\; \lambda fg.\ \forall_\sigma(\lambda x.\ fx \doteq_\tau gx) \qquad \text{D}\doteq$$

$\langle b_i \in \mathcal{B}\sigma_i \rangle$ stands for $(b_1, \ldots, b_n) \in \mathcal{B}\sigma_1 \times \cdots \times \mathcal{B}\sigma_n$

Figure 1: Quote Functions $\downarrow_\sigma$ and Notations $\forall_\sigma$ and $\doteq_\sigma$

2. $\hat{\mathcal{B}}(\forall_\sigma) f = 1 \iff \forall c \in \mathcal{B}\sigma\colon \;\; fc = 1$
3. $\hat{\mathcal{B}}(\doteq_\sigma) ab = 1 \iff a = b$

Note that statement (1) of Proposition 3 implies that our language is denotationally complete. We state this important fact explicitly.

Proposition 4 (Denotational Completeness) *Let σ be a type and $a \in \mathcal{B}\sigma$. Then there is a closed term s such that $\hat{\mathcal{B}}s = a$.*

4 Proof System

A *sequent* is a pair $A \Rightarrow s$ where A is a finite set of formulas and s is a formula. The letter A will always denote a finite set of formulas. We write $\mathcal{V}A$ for the set of all variables that occur free in at least one formula in A. An interpretation $\mathcal{I}$ *satisfies* A if it satisfies every formula in A. A sequent $A \Rightarrow s$ is *valid* if every interpretation that satisfies A also satisfies s. A context C *captures* a variable x if the hole of C is in the scope of a binder λx. A context C is *admissible for* A if it does not capture any variable in $\mathcal{V}A$.

Figure 2 defines a sequent-based proof system for our language. The first five rules (Triv, Weak, Ded, MP, DN) are well-known from propositional logic. We refer to the proof system obtained with these rules as the *propositional*

$$\text{Triv}\ \frac{}{A, s \Rightarrow s} \qquad \text{Weak}\ \frac{A \Rightarrow t}{A, s \Rightarrow t} \qquad \text{Ded}\ \frac{A,\ s \Rightarrow t}{A \Rightarrow s \to t}$$

$$\text{MP}\ \frac{A \Rightarrow s \to t \qquad A \Rightarrow s}{A \Rightarrow t} \qquad \text{DN}\ \frac{A \Rightarrow \neg\neg s}{A \Rightarrow s}$$

$$\text{Lam}\ \frac{A \Rightarrow s}{A \Rightarrow t}\ s \sim_\lambda t \qquad \text{BR}\ \frac{A \Rightarrow s \equiv t \qquad A \Rightarrow C[s]}{A \Rightarrow C[t]}\ \text{C admissible for A}$$

Figure 2: Proof System

subsystem. The propositional subsystem differs from a pure propositional system in that the propositional variables may be instantiated with any term of type B. The rule Lam incorporates lambda equivalence. The final rule BR provides for replacement with respect to Boolean equations. A replacement may capture variables of the equation if they don't occur in the assumptions. A sequent is *deducible* if it is derivable with the proof rules. A formula s is *deducible* if the sequent $\emptyset \Rightarrow s$ is deducible.

Proposition 5 (Soundness) *Every deducible sequent is valid.*

Proof It suffices to show that every instance of every proof rule is sound, that is, that the conclusion is valid if all the premises are valid. This is obvious for the propositional rules and well-known for Lam. The soundness of BR can be shown by induction on the context C. □

Let us look at an example illustrating that the completeness of the proof system is not obvious. Consider the formula $f(f(fx)) \equiv fx$, where $x : B$ and $f : BB$ are variables. The formula is valid. Checking this claim is easy since there are only 4 functions of type BB (negation, the identity function, and the two constant functions). But for non-experts, a proof of the formula in our proof system is not obvious.

A *propositional formula* is a formula s that can be obtained according to $s ::= x \mid \bot \mid s \to s$ where x serves as a placeholder for variables. A *tautology* is a valid propositional formula. A formula is *tautologous* if it is a substitution instance of a tautology.

Proposition 6 (Taut) *Every tautologous formula is deducible.*

Proof We take it for granted that the propositional subsystem can deduce every tautology. Since the instances of the propositional proof rules are closed under substitution of variables, propositional proof trees are closed under substitution of variables. Hence the claim follows. □

We use $\vdash$ to denote the set of all deducible sequents. Since sequents are pairs, $\vdash$ is a binary relation. We write $A \vdash s$ if the sequent $A \Rightarrow s$ is deducible, and $\vdash s$ if the formula s is deducible. The next proposition states properties of $\vdash$ that we will use in the following.

Proposition 7

Ded $A \vdash s \to t \iff A, s \vdash t$

Cut $A, s_1, \ldots, s_n \vdash s \;\land\; A \vdash s_1 \;\land\; \ldots \;\land\; A \vdash s_n \implies A \vdash s$

And $s_1, s_2 \vdash s_1 \land s_2$

Ref $A \vdash s \equiv s$

Sub $A \vdash s \implies A^x_t \vdash s^x_t$

Proof The derivation of Ded and Cut is straightforward. And follows with Taut and Ded since $x \to y \to x \land y$ is a tautology. Ref follows with Taut since $x \equiv x$ is a tautology. Because of Ded it suffices to show Sub for $A = \emptyset$. Let $\vdash s$. Then $\vdash s \equiv \top$ by Taut and BR since $x \equiv (x \equiv \top)$ is a tautology. By Ref an BR we obtain $\vdash (\lambda x.s)t \equiv (\lambda x.\top)t$. Thus $\vdash s^x_t \equiv \top$ by Lam. Hence $\vdash s^x_t$ by Taut and BR. □

Proposition 8 *Every closed and β-normal formula is propositional.*

Proof By induction on the size of formulas. Let s be a β-normal and closed formula. Then $s = x s_1 \ldots s_n$ where $s_1, \ldots, s_n$ are all closed and β-normal. Since s is closed, either $x = \bot$ or $x = \to$. If $x = \bot$, then $n = 0$ and hence s is propositional. If $x = \to$, then $n = 2$ and the claim follows by the induction hypothesis applied to s_1 and s_2. □

Proposition 9 (Closed Completeness)
Every closed and valid formula is deducible.

Proof By Proposition 1, Lam, and Soundness it suffices to show the claim for closed, valid, β-normal formulas. By Proposition 8 we know that such formulas are tautologies. Now the claim follows with Taut. □

5 Deductive Completeness

We say that $\vdash$ is *complete* if every valid formula is deducible. By Ded, completeness of $\vdash$ implies that every valid sequent is deducible. For every type σ, we define three properties:

All$_\sigma$ For all $f : \sigma B$ and $x : \sigma$ it holds: $\forall_\sigma f \vdash fx$

Enum$_\sigma$ For all $x : \sigma$ it holds: $\vdash \bigvee_{a \in \mathcal{B}\sigma} (\downarrow_\sigma a) \doteq_\sigma x$

Rep$_\sigma$ For all A and all formulas $s \doteq_\sigma t$ and $C[s]$ such that the context C is admissible for A, it holds: If $A \vdash s \doteq_\sigma t$ and $A \vdash C[s]$, then $A \vdash C[t]$.

We will show that the properties hold for all types.

Lemma 10 *If All$_\sigma$ holds for all types σ, then $\vdash$ is complete.*

Proof Assume All$_\sigma$ holds for all types σ. Let s be a valid formula. We show $\vdash s$. Let $\mathcal{V}s = \{x_1, \dots, x_n\}$. Then $\forall x_1 \dots \forall x_n.s$ is closed and valid. Hence, $\vdash \forall x_1 \dots \forall x_n.s$ by Proposition 9. The claim now follows by repeated application of All$_\sigma$, Sub, Lam, and Cut. □

Lemma 11 $\vdash \downarrow_\tau (ab) \doteq_\sigma (\downarrow_{\sigma\tau} a)(\downarrow_\sigma b)$

Proof Since the formula is closed, by Proposition 9 it suffices to show that it is valid. This holds since $\hat{\mathcal{B}}(\downarrow_\tau (ab)) = ab = (\hat{\mathcal{B}}(\downarrow_{\sigma\tau} a))(\hat{\mathcal{B}}(\downarrow_\sigma b)) = \hat{\mathcal{B}}((\downarrow_{\sigma\tau} a)(\downarrow_\sigma b))$ by Proposition 3. □

Lemma 12 *Let I and J be finite sets and $x_{i,j} : B$ be a variable for all $i \in I$, $j \in J$. Moreover, let $[I \to J]$ be the set of all total functions $I \to J$. Then:*

$$\vdash \bigwedge_{i \in I} \bigvee_{j \in J} x_{i,j} \equiv \bigvee_{\varphi \in [I \to J]} \bigwedge_{i \in I} x_{i,\varphi i}$$

Proof Let s and t be the left and the right term of the equivalence in question, respectively, and let $\mathcal{I}$ be an interpretation. The claim follows by Taut if we can show that $\hat{\mathcal{I}}s = 1$ iff $\hat{\mathcal{I}}t = 1$. Let $\hat{\mathcal{I}}s = 1$. Then for every $i \in I$ there exists a $j \in J$ such that $\mathcal{I}(x_{i,j}) = 1$. Hence there exists a function $\varphi \in [I \to J]$ such that $\mathcal{I}(x_{i,\varphi i}) = 1$ for every $i \in I$. Hence $\hat{\mathcal{I}}t = 1$. The other direction follows analogously. □

Lemma 13 *Let σ be a type. If Enum_σ and Rep_σ hold, then All_σ holds.*

Proof Assume that Enum_σ and Rep_σ hold. Let $f : \sigma B$, $x : \sigma$, and $a \in \mathcal{B}\sigma$. By Taut we have:

$$\vdash \left(\bigwedge_{b \in \mathcal{B}\sigma} f(\downarrow_\sigma b) \right) \to f(\downarrow_\sigma a)$$

Hence, by D$\forall$, Lam, and Weak:

$$(\downarrow_\sigma a) \doteq_\sigma x \vdash \forall_\sigma f \to f(\downarrow_\sigma a)$$

Hence, by Triv, Rep_σ, and Ded:

$$\vdash (\downarrow_\sigma a) \doteq_\sigma x \to \forall_\sigma f \to fx$$

Since $a \in \mathcal{B}\sigma$ was chosen freely, we have by And and Cut:

$$\vdash \bigwedge_{a \in \mathcal{B}\sigma} ((\downarrow_\sigma a) \doteq_\sigma x \to \forall_\sigma f \to fx)$$

Now, by Taut and MP, it follows:

$$\vdash \left(\bigvee_{a \in \mathcal{B}\sigma} (\downarrow_\sigma a) \doteq_\sigma x \right) \to \forall_\sigma f \to fx$$

The claim follows by Enum_σ and MP. □

Lemma 14 *Enum_σ and Rep_σ hold for all types σ.*

Proof By induction on σ.

We first show Enum_σ. Let $x : \sigma$. If $\sigma = B$, the claim follows by Taut. Otherwise, let $\sigma = \sigma_1\sigma_2$. By Enum_{σ_2} (induction hypothesis) and Sub, we obtain

$$\vdash \bigvee_{c \in \mathcal{B}\sigma_2} (\downarrow_{\sigma_2} c) \doteq_{\sigma_2} x(\downarrow_{\sigma_1} b)$$

for every $b \in \mathcal{B}\sigma_1$. Hence, by And:

$$\vdash \bigwedge_{b \in \mathcal{B}\sigma_1} \bigvee_{c \in \mathcal{B}\sigma_2} (\downarrow_{\sigma_2} c) \doteq_{\sigma_2} x(\downarrow_{\sigma_1} b)$$

By Lemma 12, Sub, and BR, this yields:

$$\vdash \bigvee_{a \in \mathcal{B}\sigma} \bigwedge_{b \in \mathcal{B}\sigma_1} (\downarrow_{\sigma_2} (ab)) \doteq_{\sigma_2} x(\downarrow_{\sigma_1} b)$$

By repeated application of Lemma 11 and Rep_{σ_2} (induction hypothesis), we obtain

$$\vdash \bigvee_{a\in\mathcal{B}\sigma} \bigwedge_{b\in\mathcal{B}\sigma_1} (\downarrow_\sigma a)(\downarrow_{\sigma_1} b) \doteq_{\sigma_2} x(\downarrow_{\sigma_1} b)$$

which is Enum_σ up to D$\doteq$, D$\forall$ and Lam.

Next we show Rep_σ. Let $A \vdash s \doteq_\sigma t$ and $A \vdash C[s]$, and let C be admissible for A. We show $A \vdash C[t]$. If $\sigma = B$, the claim is immediate by BR. Otherwise, let $\sigma = \sigma_1\sigma_2$. By Triv, D$\doteq$, and Lam we have:

$$s \doteq_\sigma t \vdash \forall_{\sigma_1} y.\ sy \doteq_{\sigma_2} ty$$

for some variable $y \notin \mathcal{V}(s \doteq_\sigma t)$. By the induction hypothesis and Lemma 13, we have All_{σ_1}. By Sub, Lam, Weak, and Cut we then obtain:

$$s \doteq_\sigma t \vdash sy \doteq_{\sigma_2} ty$$

Hence, by Weak, the assumption $A \vdash s \doteq_\sigma t$, and Cut:

$$A \vdash sy \doteq_{\sigma_2} ty$$

Since $A \vdash C[s]$ and $y \notin \mathcal{V}s$, we have by Lam:

$$A \vdash C[\lambda y.sy]$$

Since the context $C[\lambda y.[]]$ is admissible for A and Rep_{σ_2} holds by the induction hypothesis, we have:

$$A \vdash C[\lambda y.ty]$$

The claim follows by Lam. □

Theorem 15 (Deductive Completeness) $\vdash$ *is complete.*

Proof Follows by Lemmas 14, 13, and 10. □

References

[1] Andrews, P. B. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, 2nd ed., vol. 27 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.

[2] Benzmüller, C. E., Brown, C. E., Siekmann, J., and Statman, R., Eds. *Festschrift in Honor of Peter B. Andrews on His 70th Birthday*. Studies in Logic and the Foundations of Mathematics. IFCoLog. To appear.

[3] Church, A. A formulation of the simple theory of types. *J. Symb. Log. 5*, 1 (1940), 56–68.

[4] Henkin, L. A theory of propositional types. *Fund. Math. 52* (1963), 323–344.

[5] Hindley, J. R., and Seldin, J. P. *Lambda-Calculus and Combinators: an Introduction*, 2nd ed. Cambridge University Press, 2008.

[6] Kaminski, M., and Smolka, G. A finite axiomatization of propositional type theory in pure lambda calculus. Tech. rep., Saarland University, 2008. To appear in [2].

[7] Vorobyov, S. The most nonelementary theory. *Inf. Comput. 190*, 2 (2004), 196–219.