PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 68(82) (2000), 1–19

THE MAXIMALITY OF THE TYPED LAMBDA CALCULUS AND OF CARTESIAN CLOSED CATEGORIES

Kosta Došen and Zoran Petrić

Communicated by Žarko Mijajlović

Abstract. From the analogue of Böhm's Theorem proved for the typed lambda calculus, without product types and with them, it is inferred that every cartesian closed category that satisfies an equality between arrows not satisfied in free cartesian closed categories must be a preorder. A new proof is given here of these results, which were obtained previously by Richard Statman and Alex K. Simpson.

1. Introduction

In [7] we have shown that every cartesian category that satisfies an equality between arrows not satisfied in cartesian categories freely generated by sets of objects must be a preorder; i.e., arrows with the same source and target must be equal. In this paper we give a new proof of the result of [16, Theorem 1] that cartesian closed categories are maximal in the same sense. This means that all equalities between arrows not assumed for the axiomatization of cartesian, or cartesian closed categories, are equivalent with each other. Each of them entails all the other equalities.

It should be stressed that the equalities in question are in the language of free cartesian categories, or free cartesian closed categories, and their satisfaction is taken to be universal with respect to objects; i.e., atomic symbols for objects are assumed to be variables, and the equalities are said to be satisfied when they hold for every assignment of objects to these variables.

The maximality of cartesian categories we have proved previously cannot be inferred from the maximality of cartesian closed categories we are working on in this paper, because not every cartesian category need be closed. So the latter result cannot be simply dubbed a "generalization" of the former. These results are

1991 *Mathematics Subject Classification.* Primary 03B40, 18D15; Secondary 18A15, 03G30.

independent, since the inference of the latter result from the former is impossible too.

The proof of the maximality of cartesian closed categories we are going to present here is more demanding than the proof of [7]. This new proof is based on the analogue of Böhm's Theorem for the typed lambda calculus, a version of which was established in [19, Theorem 2]. We prove this analogue first for the typed lambda calculus with only functional types, in a way different from Statman's, and from that we pass to the analogue of Böhm's Theorem for the typed lambda calculus with product types added. The proof of the latter analogue reduces essentially to the proof of the former. These analogues of Böhm's Theorem cannot be deduced from the proof of Böhm's Theorem for the untyped lambda calculus.

To pass from the analogue of Böhm's Theorem for the typed lambda calculus with product types to the maximality result for cartesian closed categories we rely on the categorial equivalence between typed lambda calculuses and cartesian closed categories, whose discovery is due to Lambek (see [11, I.11], and references in the Historical Perspective and Historical Comments on Part I of that book). This fundamental equivalence, coupled with the understanding of cartesian closed categories as theories of deduction in intuitionistic logic, expresses what is usually called the Curry-Howard correspondence, but which, with more fairness, could be called *the Curry-Lambek-Howard correspondence*. (The Curry-Howard correspondence is often called an isomorphism, but the term *isomorphism* is more problematic than the looser term *correspondence*. If typed lambda terms are just used as codes for natural-deduction proofs, then there is presumably an isomorphism between the codes and the things coded, but no independent algebraic description is given of the things coded. If such an independent description is given with the language of cartesian closed categories, then we fall upon Lambek's *equivalence* of categories, and not upon an *isomorphism* of categories.)

The maximality of cartesian and cartesian closed categories is analogous to the property of the classical propositional calculus called *Post completeness*. That this calculus is Post complete means that if we add to it a new axiom schema in the language of this calculus, then we can prove every formula. Böhm's Theorem in the lambda calculus, or rather its immediate corollaries, are sometimes termed "Post completeness".

The equational theory of Boolean algebras is also maximal, i.e. Post complete. If we add to this theory a new equality in the language of the theory, then we can deduce $1 = 0$ and every other equality. The maximality of cartesian and cartesian closed categories is analogous to this maximality of Boolean algebras. Only in equalities we must take care of types, while in Boolean algebras there is only one type. Another difference is that in Boolean algebras the equalities producing the extension may involve variable terms, while in our extensions of the equational theories of cartesian and cartesian closed categories we envisage only equalities with constant arrow terms; variables occur only at the level of types, i.e. the level of objects.

The import of the maximality of cartesian closed categories for logic is that, in the implication-conjunction fragment of intuitionistic logic, the choice of equalities between deductions induced by $\beta\eta$ normalization in natural deduction is optimal. These equalities, which correspond to the equalities of cartesian closed categories, are wanted, and no equality is missing, because any further equality would lead to collapse: all deductions with the same premises and conclusion would be equal. The import of the maximality of cartesian categories for conjunctive logic is the same.

Although the results of this paper were already established in [19] and [16], our proof is different, and we hope it might shed some new light on the matter. For his proof, Simpson relies essentially, among other things, upon a syntactic result of [18, Theorem 3]; for a proof of this theorem see [14]), which reduces types in equalities to a particular type, whereas we rely on a different result from the same paper [18, Theorem 2], proved previously in [17, Theorem 2], which is a finite-model property for the typed lambda calculus. Our approach provides an alternative proof of the type-reducing result of [18, Theorem 3].

An analogue of Böhm's Theorem in the typed lambda calculus without product types is proved in [19, Theorem 2], without mentioning Böhm's Theorem. Statman has even a semantic notion of consistent extension, rather than a syntactic notion, such as we have, following Böhm. (The two notions happen to be equivalent, however.) Our analogues of Böhm's Theorem in the the typed lambda calculus, with and without product types, are closer to standard formulations of this theorem, and our proof is different from Statman's, which, as Simpson's proof, relies on the type-reducing result of [18, Theorem 3]. There are, however, some analogies in the general spirit of these proofs.

The possibility of proceeding as we do is indicated briefly in [16, last paragraph of section 5]. Simpson says: "It is an interesting fact that an alternative direct proof of Theorem 3 is possible using a typed version of the Böhm-out technique [1, Chapter 10]. The details are beyond the scope of this paper." (Simpson's Theorem 3 amounts to our Maximality Corollary in Section 6 below.) We don't know what Böhm-out technique Simpson had in mind, but he assured us his approach is different from ours. Anyway, we couldn't find such a technique by imitating [1]. Our technique has some intrinsic difficulties, but presumably not more than the technique of [19]. Our presentation takes a little bit more space because we have tried to help the reader by going into more details. These details, which were beyond the scope of Simpson's paper, fall exactly within the scope of ours.

2. Böhm's Theorem

Böhm's Theorem in the untyped lambda calculus says that if a and b are two different lambda terms in $\beta\eta$ normal form, and c and d are arbitrary lambda terms, then one can construct terms $h_1, \dots, h_n$, $n \geq 0$, and find variables $x_1, \dots, x_m$, $m \geq 0$, such that

$$(\lambda_{x_1 \dots x_m} a) h_1 \dots h_n = c,$$
$$(\lambda_{x_1 \dots x_m} b) h_1 \dots h_n = d$$

are provable in the β lambda calculus (see [1, Chapter 10, §4, Theorem 10.4.2]; [4, Chapter 11F, §8, Theorem 5]; [10, Chapitre V, Théorème 2]; we know the original paper of Böhm [3] only from references). As a corollary of this theorem one obtains that if a and b are two lambda terms having a normal form such that $a = b$ is not provable in the $\beta\eta$ lambda calculus and this calculus is extended with $a = b$, then one can prove every equality in the extended calculus.

We will demonstrate first the analogue of Böhm's Theorem in the typed lambda calculus with only functional types. The standard proof of Böhm's Theorem, which may be found in the books cited above, cannot be transferred to the typed case. At crucial places it introduces lambda terms that cannot be appropriately typed. For example, for $\lambda_{xy}xy$ and $\lambda_{xy}x(xy)$ (i.e., the Church numerals for 1 and 2) with x of type $p \to p$ and y of type p there is no appropriate permutator of type $p \to p$ with whose help these two terms can be transformed into terms with a head original head normal form (see [1, Chapter 10, §3]). A more involved example is given with the terms $\lambda_x x\lambda_y(x\lambda_z y)$ and $\lambda_x x\lambda_y(x\lambda_z z)$ with x of type $(p \to p) \to p$ and y and z of type p (we deal with these two typed terms in the Example of Section 6).

One cannot deduce our analogue of Böhm's Theorem for the typed lambda calculus from Böhm's Theorem for the untyped lambda calculus. The typed calculus has a more restricted language and does not allow everything permitted in the untyped case. Conversely, one cannot deduce Böhm's Theorem for the untyped lambda calculus from our typed version of this theorem. Our result covers only cases where a and b are typable by the same type.

3. The typed lambda calculus

The formulation of the typed lambda calculus with only functional types we rely on is rather standard (see, for example, [1, Appendix 1], or [9]). However, we sketch this formulation briefly, to fix notation and terminology.

Types are defined inductively by a nonempty set of *atomic types* and the clause "if A and B are types, then $(A \to B)$ is a type". For atomic types we use the schematic letters $p, q, r, \dots, p_1, \dots$, and for all types we use the schematic letters $A, B, C, \dots, A_1, \dots$ We write A^p_B for the result of substituting B for p in A. (*Substitution* means as usual *uniform replacement*.)

Terms are defined inductively in a standard manner. We have infinitely many *variables* of each type, for which we use the schematic letters $x, y, z, \dots, x_1, \dots$ For arbitrary terms we use the schematic letters $a, b, c, \dots, a_1, \dots$ That a term a is of type A is expressed by $a : A$. However, for easier reading, we will not write types inside terms, but will specify the types of variables separately. For application we use the standard notation, with the standard omitting of parentheses. For lambda abstraction we will write λ_x with subscripted x, instead of λx (this way we can do without dots in $\lambda_x x$, which is otherwise written $\lambda x.x$). We abbreviate $\lambda_{x_1} \dots \lambda_{x_n} a$ by $\lambda_{x_1 \dots x_n} a$, as usual. We write a^x_b for the result of substituting b for x in a, provided b is free for x in a.

If a is a term, let a *type-instance* of a be obtained by substituting some types for the atomic types in the variables of a.

A *formula* of the typed lambda calculus Λ is of the form $a = b$ where a and b are terms of the same type.

The calculus Λ of $\beta\eta$ equality is axiomatized with the usual axioms

$$(\beta) \qquad (\lambda_x a)b = a^x_b, \quad \text{provided } b \text{ is free for } x \text{ in } a,$$

$$(\eta) \qquad \lambda_x ax = a, \qquad \text{provided } x \text{ is not free in } a,$$

and the axioms and rules for equality, i.e. $a = a$ and the rule of replacement of equals. It is not usually noted that the equality of α conversion can be proved from the remaining axioms as follows:

$$\begin{aligned} \lambda_x a &= \lambda_y(\lambda_x a)y, \text{ by } (\eta), \\ &= \lambda_y a^x_y, \quad \text{ by } (\beta), \end{aligned}$$

where y is a variable not occurring in a.

4. Lambda terms for P-functionals

Let P be a finite ordinal. In what follows an interesting P will be greater than or equal to the ordinal 2. The set of *P-types* is defined inductively by specifying that P is a P-type and that if A and B are P-types, then $A \to B$, i.e. the set of all functions with domain A and codomain B, is a P-type. Symbols for P-types are types with a single atomic type P. It is clear that for P nonempty a P-type cannot be named by two different P-type symbols.

An element of a P-type is called a *P-functional.* It is clear that every P-functional is finite (i.e., its graph is a finite set of ordered pairs) and that in every P-type there are only finitely many P-functionals. For P-functionals we use the Greek letters $\varphi, \psi, \dots, \varphi_1, \dots$

Our aim is to define for every P-functional a closed term defining it, in a sense to be made precise. But before that we must introduce a series of preliminary definitions. In these definitions we take that the calculus Λ is built over types with a single atomic type, which we call p.

Let the type A_0 be p and let the type A_{n+1} be $A_n \to A_n$. For $i \geq 0$, let the type N_i be A_{i+2}, i.e. $(A_i \to A_i) \to (A_i \to A_i)$.

Let $x^0(y)$ be y and let $x^{n+1}(y)$ be $x(x^n(y))$. The terms $[n]_i$, called *Church numerals of type* N_i, are defined by

$$[n]_i =_{\mathrm{def}} \lambda_{xy} x^n(y)$$

for $x : A_{i+1}$ and $y : A_i$.

For x, y and z all of type N_i, $u : A_{i+1}$, and v and w of type A_i, let

$$C_i =_{\mathrm{def}} \lambda_{xyzuv} x(\lambda_w zuv)(yuv).$$

These are conditional function combinators, because in the calculus Λ one can prove

$$C_i[n]_i ab = \begin{cases} a & \text{if } n = 0 \\ b & \text{if } n \neq 0 \end{cases}$$

For $x : N_{i+1}$, y and z of type A_{i+1}, and u and v of type A_i, let

$$R_i =_{\mathrm{def}} \lambda_{xy} x(\lambda_{zu} y(zu))(\lambda_v v).$$

These combinators reduce the types of numerals; namely, in Λ one can prove

$$R_i[n]_{i+1} = [n]_i.$$

For x and y of type N_{i+1}, let the exponentiation combinators be defined by

$$E_i =_{\mathrm{def}} \lambda_{xy} x(R_i y).$$

In Λ one can prove

$$E_i[n]_{i+1}[m]_{i+1} = [m^n]_i.$$

For $E_i ab$ we use the abbreviation b^a.

For x and y of type N_i, $z : A_{i+1}$ and $u : A_i$, let the addition and multiplication combinators be defined by

$$S_i =_{\mathrm{def}} \lambda_{xyzu} xz(yzu),$$
$$M_i =_{\mathrm{def}} \lambda_{xyzu} x(yz)u.$$

In Λ one can prove

$$S_i[n]_i[m]_i = [n+m]_i,$$
$$M_i[n]_i[m]_i = [n \cdot m]_i.$$

For $M_i ab$ we use the abbreviation $a \cdot b$.

For x, y and z of type N_i, and $u : N_{i+1}$, let the pairing and projection combinators be defined by

$$\Pi_i =_{\mathrm{def}} \lambda_{xyz} C_i zxy,$$
$$\pi_i^1 =_{\mathrm{def}} \lambda_u u[0]_i,$$
$$\pi_i^2 =_{\mathrm{def}} \lambda_u u[1]_i.$$

In Λ one can prove

$$\pi_i^1(\Pi_i ab) = a,$$
$$\pi_i^2(\Pi_i ab) = b.$$

For $x : N_{i+1}$ and $y : N_{i+3}$, let

$$T_i =_{\mathrm{def}} \lambda_x \Pi_i(S_i[1]_i(\pi_i^1 x))(\pi_i^1 x),$$
$$H_i =_{\mathrm{def}} \lambda_y y T_i(\Pi_i[0]_i[0]_i),$$
$$P_i =_{\mathrm{def}} \lambda_y \pi_i^2(H_i y).$$

The terms T_i and H_i are auxiliary, while the terms P_i are predecessor combinators, because, for $n \geq 1$, one can prove in Λ

$$P_i[n]_{i+3} = [n-1]_i,$$
$$P_i[0]_{i+3} = [0]_i.$$

Typed terms corresponding to all the terms C_i, R_i, up to P_i, may be found in [2] (cf. [15] and [8]).

For x and y of type N_i, $z : A_{i+1}$, and u and v of type A_i, let

$$Z_{i+1} =_{\mathrm{def}} \lambda_{xyzu} x(\lambda_v yzu)(zu).$$

These combinators raise the types of numerals for 0 and 1; namely, in Λ one can prove

$$Z_{i+1}[0]_i = [0]_{i+1},$$
$$Z_{i+1}[1]_i = [1]_{i+1}.$$

The equality (η) is essential to prove this.

For $x : N_i$, let

$$D_i^0 =_{\mathrm{def}} \lambda_x C_i x[0]_i[1]_i$$

and for $k \geq 1$ and $i \geq 3k$ let

$$D_i^k =_{\mathrm{def}} \lambda_x C_i x[1]_i Z_i(Z_{i-1}(Z_{i-2}(D_{i-3}^{k-1}(P_{i-3}x)))).$$

These combinators check whether a numeral stands for k; namely, for $n \geq 0$, one can prove in Λ

$$D_i^k[n]_i = \begin{cases} [0]_i & \text{if } n = k \\ [1]_i & \text{if } n \neq k. \end{cases}$$

For every P-type symbol A, let A^i be the type obtained from A by substituting N_i for P. Now we are ready to define for every P-functional $\varphi \in A$ a closed term $\varphi^\lambda : A^i$.

Take a P-functional $\varphi \in A$, where A is $B_1 \to (\ldots \to (B_k \to P)\ldots)$. By induction on the complexity of the P-type symbol A we define a natural number $\kappa(\varphi)$ and for every $i \geq \kappa(\varphi)$ we define a term $\varphi^\lambda : A^i$.

If A is P, then φ is an ordinal in P. Then $\kappa(n) = 0$ and $n^\lambda : N_i$ is $[n]_i$ for every $i \geq 0$.

Suppose $k \geq 1$ and B_1 is $B \to (C \to P)$. It is enough to consider this case, which gives the gist of the proof. When B_1 is $C_1 \to (C_2 \to \ldots (C_l \to P)\ldots)$ for l different from 2 we proceed analogously, but with more notational complications if $l \geq 3$. For $B = \{\beta_1, \ldots, \beta_m\}$ and $C = \{\gamma_1, \ldots, \gamma_r\}$, by the induction hypotheses, we have defined $\kappa(\beta_1)$, ..., $\kappa(\beta_m)$, $\kappa(\gamma_1)$, ..., $\kappa(\gamma_r)$, for every $i \geq \kappa(\beta_1)$ we have defined β_1^λ, and analogously for β_2, ..., β_m, γ_1, ..., γ_r. For $B_1 = \{\psi_1, \ldots, \psi_q\}$, let $\varphi(\psi_j) = \xi_j \in B_2 \to (\ldots \to (B_k \to P)\ldots)$. (Note that φ is not necessarily one-one.) By the induction hypothesis, we have defined $\kappa(\xi_1), \ldots, \kappa(\xi_q)$, for every $i \geq \kappa(\xi_1)$ we have defined ξ_1^λ, and analogously for $\xi_2, \ldots, \xi_q$.

Let now

$$\begin{array}{llll}
(\psi_1(\beta_1))(\gamma_1) = d_1 \in P, & (\psi_1(\beta_2))(\gamma_1) = d_{r+1} \in P, & \ldots & (\psi_1(\beta_m))(\gamma_1) = d_{(m-1)r+1} \in P \\
(\psi_1(\beta_1))(\gamma_2) = d_2 \in P, & (\psi_1(\beta_2))(\gamma_2) = d_{r+2} \in P, & \ldots & (\psi_1(\beta_m))(\gamma_2) = d_{(m-1)r+2} \in P \\
\vdots & \vdots & & \vdots \\
(\psi_1(\beta_1))(\gamma_r) = d_r \in P, & (\psi_1(\beta_2))(\gamma_r) = d_{2r} \in P, & \ldots & (\psi_1(\beta_m))(\gamma_r) = d_{mr} \in P
\end{array}$$

Let $n_1 = 2^{d_1} \cdot 3^{d_2} \cdot \ldots \cdot p_{mr}^{d_{mr}}$, where p_{mr} is the mr-th prime number. Analogously, we obtain the natural numbers $n_2, \ldots, n_q$, all different, that correspond to $\psi_2, \ldots, \psi_q$.

We can now define $\kappa(\varphi)$ as

$\max\{3\cdot\max\{n_1,\dots,n_q\}+1,\kappa(\beta_1),\dots,\kappa(\beta_m),\kappa(\gamma_1),\dots,\kappa(\gamma_r),\kappa(\xi_1),\dots,\kappa(\xi_q)\}$.

For every $i\geq\kappa(\varphi)$ and for $x_1:B_1^i$, let the term t be defined as

$$[2]_i^{x_1\beta_1^\lambda\gamma_1^\lambda}\cdot[3]_i^{x_1\beta_1^\lambda\gamma_2^\lambda}\cdot\ldots\cdot[p_{mr}]_i^{x_1\beta_m^\lambda\gamma_r^\lambda}:N_{i-1}.$$

For $x_2:B_2^i,\dots,x_k:B_k^i$, let

$$\begin{aligned}
Q_1 &=_{\mathrm{def}} C_i(Z_i(D_{i-1}^{n_1}t))(\xi_1^\lambda x_2\dots x_k)Q_2,\\
Q_2 &=_{\mathrm{def}} C_i(Z_i(D_{i-1}^{n_2}t))(\xi_2^\lambda x_2\dots x_k)Q_3,\\
&\vdots\\
Q_{q-1} &=_{\mathrm{def}} C_i(Z_i(D_{i-1}^{n_{q-1}}t))(\xi_{q-1}^\lambda x_2\dots x_k)(\xi_q^\lambda x_2\dots x_k).
\end{aligned}$$

We can now, finally, define φ^λ as $\lambda_{x_1\dots x_k}Q_1$.

Next we define by induction on the complexity of the P-type symbol A, when a P-functional $\varphi\in A$ is *i-defined* by a term $a:A^i$.

We say that a closed term $a:N_i$ *i-defines* an ordinal $n\in P$ iff in Λ we can prove $a=[n]_i$.

For a P-functional $\varphi\in B\rightarrow C$ we say that $a:B^i\rightarrow C^i$ *i-defines* φ iff, for every $\psi\in B$ and every $b:B^i$, if b i-defines ψ, then $ab:C^i$ i-defines $\varphi(\psi)\in C$.

We can now prove the following lemma.

LEMMA 4.1. *For every $i\geq\kappa(\varphi)$, the P-functional $\varphi\in A$ is i-defined by $\varphi^\lambda:A^i$.*

PROOF. We proceed by induction on the complexity of the P-type symbol A. The case when A is P is trivial.

Let now A be of the form $B_1\rightarrow(\dots\rightarrow(B_k\rightarrow P)\dots)$ for $k\geq1$, let $B_1=\{\psi_1,\dots,\psi_q\}$, and let everything else be as in the inductive step of the definition of φ^λ. Suppose $b_1:B_1^i$ i-defines ψ_1. We have to check that $\varphi^\lambda b_1$ i-defines $\varphi(\psi_1)=\xi_1$.

By the induction hypothesis we have that $\beta_1^\lambda,\dots,\beta_m^\lambda,\gamma_1^\lambda,\dots,\gamma_r^\lambda,\xi_1^\lambda,\dots,\xi_q^\lambda$ i-define $\beta_1,\dots,\beta_m,\gamma_1,\dots,\gamma_r,\xi_1,\dots,\xi_q$, respectively. Then we have

$$\begin{aligned}
\varphi^\lambda b_1 &= (\lambda_{x_1\dots x_k}C_i(Z_i(D_{i-1}^{n_1}t))(\xi_1^\lambda x_2\dots x_k)Q_2)b_1\\
&= \lambda_{x_2\dots x_k}C_i(Z_i(D_{i-1}^{n_1}t_{b_1}^{x_1}))(\xi_1^\lambda x_2\dots x_k)(Q_2)_{b_1}^{x_1}.
\end{aligned}$$

For the closed term $t_{b_1}^{x_1}$ we have

$$t_{b_1}^{x_1}=[2]_i^{b_1\beta_1^\lambda\gamma_1^\lambda}\cdot[3]_i^{b_1\beta_1^\lambda\gamma_2^\lambda}\cdot\ldots\cdot[p_{mr}]_i^{b_1\beta_m^\lambda\gamma_r^\lambda}.$$

It follows by the induction hypothesis that $b_1\beta_1^\lambda\gamma_1^\lambda$ i-defines d_1, which means that in Λ we can prove $b_1\beta_1^\lambda\gamma_1^\lambda=[d_1]_i$. We proceed analogously with the other exponents. So in Λ we can prove $t_{b_1}^{x_1}=[n_1]_{i-1}$. Hence in Λ we have $D_{i-1}^{n_1}t_{b_1}^{x_1}=[0]_{i-1}$, and we conclude that

$$\begin{aligned}
\varphi^\lambda b_1 &= \lambda_{x_2\dots x_k}\xi_1^\lambda x_2\dots x_k\\
&= \xi_1^\lambda,\ \text{by } (\eta).
\end{aligned}$$

So $\varphi^\lambda b_1$ i-defines ξ_1.

Suppose now $b_2 : B_1^i$ i-defines ψ_2. Then in Λ we have

$$\varphi^\lambda b_2 = \lambda_{x_2 \dots x_k} C_i(Z_i(D_{i-1}^{n_1} t_{b_2}^{x_1}))(\xi_1^\lambda x_2 \dots x_k)(C_i(Z_i(D_{i-1}^{n_2} t_{b_2}^{x_1}))(\xi_2^\lambda x_2 \dots x_k)(Q_3)_{b_2}^{x_1}).$$

Since in Λ we can prove $t_{b_2}^{x_1} = [n_2]_{i-1}$, we can also prove $D_{i-1}^{n_1} t_{b_2}^{x_1} = [1]_{i-1}$, and we conclude that

$$\varphi^\lambda b_2 = \lambda_{x_2 \dots x_k} C_i(Z_i(D_{i-1}^{n_2}[n_2]_{i-1}))(\xi_2^\lambda x_2 \dots x_k)(Q_3)_{b_2}^{x_1}.$$

Finally, we obtain as above that $\varphi^\lambda b_2$ i-defines ξ_2. We proceed analogously for $\psi_3, \dots, \psi_q$. $\square$

This lemma does not mean that we can i-define all P-functionals simultaneously for some i. But we can always find such an i for finitely many P-functionals.

5. P-models

A model based on $P = \{0, \dots, h-1\}$, with $h \geq 2$, for the calculus Λ built over types with a single atomic type p will be defined as in [9].

An *assignment* is a function f assigning to a variable $x : A$ of Λ a functional $f(x)$ in the P-type A_P^p, where A_P^p is obtained from A by substituting P for p. For an assignment f and a variable y, the assignment f_α^y is defined by

$$f_\alpha^y(x) = \begin{cases} \alpha, & \text{if } x \text{ is } y \\ f(x), & \text{if } x \text{ is not } y. \end{cases}$$

If F is the set of all P-functionals, then the *P-model* is a pair $\langle F, V \rangle$ such that V maps the pairs (a, f), with a a term and f an assignment, into F. We write $V_{a,f}$ instead of $V(a, f)$. The function V must satisfy the conditions

$$V_{x,f} = f(x),$$
$$V_{ab,f} = V_{a,f}(V_{b,f}),$$
$$\text{for } x : A \text{ and } \alpha : A_P^p, V_{\lambda_x a, f}(\alpha) = V_{a, f_\alpha^x}.$$

There is exactly one such function V.

Let $a : A$ be a term such that $x_1 : A_1, \dots, x_n : A_n$ are all the variables, both free and bound, occurring in a. Let f be an assignment, and for every $j \in \{1, \dots, n\}$ let b_j i-define $f(x_j)$. Finally, let $\underline{a}$ be the type-instance of a obtained by substituting N_i for p. The type of $\underline{a}$ is $(A_P^p)^i$. Then we can prove the following lemma.

LEMMA 5.1. *The term* $\underline{a}\dfrac{x_1 \dots x_n}{b_1 \dots b_n}$ *i-defines* $V_{a,f}$.

The proof proceeds by a straightforward induction on the complexity of the term a.

Of course, when a is closed, $V_{a,f}$ does not depend on f, and has the same value for all assignments f. So, for closed terms a, we can write V_a instead of $V_{a,f}$, and we shall do so from now on.

As an immediate corollary of Lemma 5.1 we obtain the following lemma.

LEMMA 5.2. *If a is closed, then $\underline{a}$ i-defines V_a.*

6. The maximality of the typed lambda calculus

We are now ready to prove our analogue of Böhm's Theorem for the typed lambda calculus Λ, which is not necessarily built over types with a single atomic type.

THEOREM 6.1. *If a and b are of the same type and $a = b$ is not provable in Λ, then for every two terms c and d of the same type one can construct type-instances a' and b' of a and b, respectively, and terms $h_1, \ldots, h_n$, $n \geq 0$, and also find variables $x_1, \ldots, x_m$, $m \geq 0$, such that*

$$(\lambda_{x_1 \ldots x_m} a')h_1 \ldots h_n = c,$$
$$(\lambda_{x_1 \ldots x_m} b')h_1 \ldots h_n = d$$

are provable in Λ.

PROOF. Let a_1 and b_1 be type-instances of a and b, respectively, obtained by substituting p for all atomic types. It is easy to see that $a = b$ is provable in Λ iff $a_1 = b_1$ is provable in Λ.

Let $x_1, \ldots, x_m$ be all the free variables in a_1 or b_1. Then since $a_1 = b_1$ is not provable in Λ, the equality $\lambda_{x_1 \ldots x_m} a_1 = \lambda_{x_1 \ldots x_m} b_1$ is not provable in Λ. Let a_2 be $\lambda_{x_1 \ldots x_m} a_1$ and let b_2 be $\lambda_{x_1 \ldots x_m} b_1$.

It follows from a theorem of [17, Theorem 2, p. 187] and [18, Theorem 2, p. 21] that if $a_2 = b_2$ is not provable in Λ, then there exists a P-model $\langle F, V \rangle$ such that $V_{a_2} \neq V_{b_2}$. Soloviev's and Statman's theorem doesn't mention exactly P-models, which are based on the full type structure built over an ordinal P, but instead it mentions completely analogous models based on the full type structure built over a finite set S.

We can always name the elements of S by ordinals so that S becomes an ordinal P. Moreover, for every two distinct elements s_1 and s_2 of S we can always name the elements of S so that s_1 is named by 0 and s_2 is named by 1. This means that the elements of S can always be named by elements of P so that in the P-model $\langle F, V \rangle$ above there are P-functionals $\varphi_1, \ldots, \varphi_k$, $k \geq 0$, such that

$$((V_{a_2}(\varphi_1))(\varphi_2)) \ldots (\varphi_k) = 0,$$
$$((V_{b_2}(\varphi_1))(\varphi_2)) \ldots (\varphi_k) = 1.$$

Take an even $i \geq \max\{\kappa(\varphi_1), \ldots, \kappa(\varphi_k)\}$. By Lemma 4.1, the closed terms $\varphi_1^\lambda, \ldots, \varphi_k^\lambda$ i-define $\varphi_1, \ldots, \varphi_k$, respectively. By Lemma 5.2, the term $\underline{a}_2$ i-defines V_{a_2} and $\underline{b}_2$ i-defines V_{b_2}. It follows that in Λ we can prove $\underline{a}_2 \varphi_1^\lambda \ldots \varphi_k^\lambda = [0]_i$ and $\underline{b}_2 \varphi_1^\lambda \ldots \varphi_k^\lambda = [1]_i$.

For $x : A_i$, $y : A_{i-1}$ and $z : A_{i-2}$ we can prove in Λ

$$[0]_i(\lambda_{xyz} yz)(\lambda_{yz} z) = [0]_{i-2},$$
$$[1]_i(\lambda_{xyz} yz)(\lambda_{yz} z) = [1]_{i-2}.$$

So there are closed terms $c_1, \dots, c_i$ such that in Λ we can prove

$$\underline{a}_2 \varphi_1^\lambda \dots \varphi_k^\lambda c_1 \dots c_i = [0]_0,$$
$$\underline{b}_2 \varphi_1^\lambda \dots \varphi_k^\lambda c_1 \dots c_i = [1]_0.$$

Let the left-hand sides of these two equalities be a_3 and b_3, respectively.

Take now c and d of type A and take the type-instances a_4 and b_4 of a_3 and b_3, respectively, obtained by substituting A for p. For $u : A$ we can prove in Λ

$$a_4(\lambda_u d)c = c,$$
$$b_4(\lambda_u d)c = d.$$

The terms a_4 and b_4 are of the form $(\lambda_{x_1 \dots x_n} a')h_1 \dots h_{k+i}$ and $(\lambda_{x_1 \dots x_n} b')h_1 \dots h_{k+i}$. If $(N_i)_A^p$ is obtained by substituting A for p in N_i, then a' is a type-instance of a obtained by substituting $(N_i)_A^p$ for every atomic type. □

Since the procedure for constructing $h_1, \dots, h_n$ in the proof of Theorem 6.1 can be pretty involved, it may be useful to illustrate this procedure with an example. For this example we take two terms unequal in Λ that we mentioned in Section 2 (this is the more involved of the examples given there).

Example. Let a and b be $\lambda_x x \lambda_y (x \lambda_z y)$ and $\lambda_x x \lambda_y (x \lambda_z z)$, respectively, with $x : (p \to p) \to p$, $y : p$ and $z : p$. Since all the atomic types of a and b are already p, and since these two terms are closed, we have that a_2 is a and b_2 is b.

The P-model falsifying $a = b$ has $P = \{0, 1\}$ and $P \to P = \{\psi_1, \psi_2, \psi_3, \psi_4\}$, where

$$\psi_1(0) = \psi_1(1) = 0,$$
$$\psi_2(0) = \psi_2(1) = 1,$$
$$\psi_3(0) = 0, \quad \psi_3(1) = 1,$$
$$\psi_4(0) = 1, \quad \psi_4(1) = 0.$$

For $\varphi \in (P \to P) \to P$ defined by

$$\varphi(\psi_1) = 1, \quad \varphi(\psi_2) = \varphi(\psi_3) = \varphi(\psi_4) = 0$$

we have $V_a(\varphi) = 0$ and $V_b(\varphi) = 1$.

Then

$$\begin{array}{ll} n_1 = 2^0 \cdot 3^0 = 1 & \text{corresponds to } \psi_1, \\ n_2 = 2^1 \cdot 3^1 = 6 & \text{corresponds to } \psi_2, \\ n_3 = 2^0 \cdot 3^1 = 3 & \text{corresponds to } \psi_3, \\ n_4 = 2^1 \cdot 3^0 = 2 & \text{corresponds to } \psi_4, \end{array}$$

and $\kappa(\varphi) = 19$. For every $i \geq 19$ and for $x_1 : N_i \to N_i$, the term t is defined as $[2]_i^{x_1[0]_i} \cdot [3]_i^{x_1[1]_i} : N_{i-1}$. The term φ^λ is defined as

$$\lambda_{x_1} C_i(Z_i(D_{i-1}^1 t))[1]_i (C_i(Z_i(D_{i-1}^6 t))[0]_i (C_i(Z_i(D_{i-1}^3 t))[0]_i [0]_i)).$$

The terms $\underline{a}$ and $\underline{b}$ are like a and b with $x : (N_{20} \to N_{20}) \to N_{20}$, $y : N_{20}$ and $z : N_{20}$, and let i in φ^λ be 20. Then in Λ we can prove $\underline{a}\varphi^\lambda = [0]_{20}$ and $\underline{b}\varphi^\lambda = [1]_{20}$.

The remaining steps in the construction of a_3 and b_3 are straightforward, and we shall not pursue this example further.

By taking that for x and y of the same type the term c is $\lambda_{xy}x$ and d is $\lambda_{xy}y$, we obtain the following refinement of Theorem 6.1.

THEOREM 6.2. *If a and b are of the same type and $a = b$ is not provable in Λ, then for every two terms e and f of the same type one can construct type-instances a' and b' of a and b, respectively, and closed terms $h_1, \ldots, h_l$, $l \geq 0$, and also find variables $x_1, \ldots, x_m$, $m \geq 0$, such that*

$$(\lambda_{x_1 \ldots x_m} a')h_1 \ldots h_l ef = e,$$
$$(\lambda_{x_1 \ldots x_m} b')h_1 \ldots h_l ef = f$$

are provable in Λ.

It is clear that if a and b are closed, we need not mention in this theorem the variables $x_1, \ldots, x_m$ and we can omit the λ-abstraction $\lambda_{x_1 \ldots x_m}$.

Although our proof of Theorem 6.1 relies on the equality (η) at some key steps (as we noted in connection with the combinator Z_{i+1}), it is possible to derive a strengthening of this theorem, as well as of Theorem 6.2, where Λ is replaced by Λ_β, which is Λ minus (η) and plus the equality of α conversion. We learned how to obtain this strengthening from Alex Simpson.

First note that if a term a is in both contracted and expanded $\beta\eta$ normal form, and $a = b$ in Λ, then $a = b$ in Λ_β. For if $a = b$ in Λ, then, since a is in contracted $\beta\eta$ normal form, there is a term a' such that b β-reduces to a' and a' η-reduces by contractions to a. But then, since a is also in expanded $\beta\eta$ normal form, a' must be the same term as a. So $a = b$ in Λ_β.

Then, as we did to derive Theorem 6.2, take in Theorem 6.1 that c is $\lambda_{xy}x$ and d is $\lambda_{xy}y$ for x and y of atomic type p. The terms c and d are then in both contracted and expanded $\beta\eta$ normal form, and hence it is easy to infer Simpson's strengthening mentioned above by instantiating p with an arbitrary type.

To formulate below a corollary of Theorem 6.1 we must explain what it means to extend Λ with a new axiom. Let a and b be of type A, and let a' and b' be type-instances of a and b respectively. Then assuming $a = b$ as a new axiom in Λ means assuming also $a' = b'$. In other words, $a = b$ is assumed as an axiom schema, atomic types being understood as schematic letters. The postulate (β) and (η) are also assumed as axiom schemata, in the same sense. We could as well add to Λ a new rule of substitution for atomic types. The calculus Λ is closed under this substitution rule (i.e., this rule is admissible, though not derivable from the other rules). And any extension of Λ we envisage should be closed under this rule. The rule of substitution of types says that atomic types are variables.

We can now state the following corollary of Theorem 6.1.

MAXIMALITY COROLLARY. *If $a = b$ is not provable in Λ, then in Λ extended with $a = b$ we can prove every formula $c = d$.*

7. The typed lambda calculus with product types

We want to demonstrate next the analogue of Böhm's Theorem in the typed $\beta\eta$ lambda calculus with product types, i.e. with surjective pairing, projections and a constant of terminal type, by reducing it to our analogue of Böhm's Theorem for the typed lambda calculus Λ. The idea of this reduction is inspired by [6, Chapter 4.1], [17, pp. 180ff] and [20].

In the typed lambda calculus with product types, types include an atomic constant type T and the type-forming operation $\times$ besides $\rightarrow$. Terms now include an atomic constant term $k : \mathrm{T}$. Moreover, for every term $a : A \times B$ we have the terms $p^1 a : A$ and $p^2 a : B$, and for all terms $a : A$ and $b : B$ we have the term $\langle a, b\rangle : A \times B$.

The typed lambda calculus $\Lambda_\times$ of $\beta\eta$ equality is axiomatized with the postulates for Λ extended with the axioms

$$\begin{array}{lll} (\times\beta) & p^1\langle a,b\rangle = a, & p^2\langle a,b\rangle = b, \\ (\times\eta) & \langle p^1 c, p^2 c\rangle = c, & \\ (\mathrm{T}) & \text{for } x : \mathrm{T}, & x = k. \end{array}$$

8. Product normal form of types of $\Lambda_\times$

Consider the following reductions of types, which consist in replacing subtypes of the form on the left-hand side by subtypes of the form on the right-hand side:

redexes	*contracta*
$A \rightarrow (B_1 \times B_2)$	$(A \rightarrow B_1) \times (A \rightarrow B_2)$
$(A_1 \times A_2) \rightarrow B$	$A_1 \rightarrow (A_2 \rightarrow B)$
$A \times (B \times C)$	$(A \times B) \times C$
$A \rightarrow \mathrm{T}$	T
$\mathrm{T} \rightarrow B$	B
$A \times \mathrm{T}$	A
$\mathrm{T} \times A$	A

A type of $\Lambda_\times$ is in *product normal form* iff it does not have subtypes that are redexes. If $\times_{i=1}^{1} A_i$ is A_1 and $\times_{i=1}^{n+1} A_i$ is $(\times_{i=1}^{n} A_i) \times A_{n+1}$, then a type A of $\Lambda_\times$ in product normal form is either of the form $\times_{i=1}^{n} A_i$ with every A_i a type of Λ or A is simply T.

Every type A of $\Lambda_\times$ can be reduced by the reductions above to a unique product normal form A^π. This follows from the Church-Rosser property of these reductions. These reductions are also strongly normalizing. This can be proved by assigning uniformly to atomic types, including T, natural numbers greater than or equal to 2, which will be their *complexity measure*. The complexity measure $c(C)$ of a type

C is computed according to

$$c(A \times B) = (c(A)+1)c(B),$$
$$c(A \to B) = c(B)^{c(A)}.$$

Then all reductions decrease the complexity measure. (The measure of complexity of [6, Chapter 4.1, Proposition 4.1.2] does not work, and we couldn't extract a suitable measure from [17].)

A term $a : A \to B$ is an *isomorphism* iff there is a term $a' : B \to A$ such that for $x : A$ and $y : B$ we can prove $a'(ax) = x$ and $a(a'y) = y$. We have the following lemma.

LEMMA 8.1. *For every type A one can construct an isomorphism $h : A \to A^\pi$.*

For the proof see [6, Chapter 1.9, Theorem 1.9.9].

Let a^ν be the expanded $\beta\eta$ normal form of a term a of $\Lambda_\times$ (sometimes also called *long normal form*; see [21], Chapter 6.5, and [6], Chapter 2). This normal form is unique, according to [5] (see also references in [6]).

Let $\Pi_{i=1}^{1} a_i$ be a_1 and let $\Pi_{i=1}^{n+1} a_i$ be $\langle \Pi_{i=1}^{n} a_i, a_{n+1}\rangle$. We can easily prove the following lemmata.

LEMMA 8.2. *Let a be a closed term of type A^π for some A. Then a^ν is either of the form $\Pi_{i=1}^{n} a_i$ with a_i a term of Λ, or a^ν is k.*

For that we rely on the fact that the type of a subterm of a^ν must be a subtype of A^π, which in logic is called the *subformula property*.

LEMMA 8.3. *For a and b of type A and $h : A \to A^\pi$ an isomorphism, if $a = b$ is not provable in $\Lambda_\times$, then*

(1) *$(ha)^\nu$ is of the form $\Pi_{i=1}^{n} a_i$ for a_i a term of Λ,*
(2) *$(hb)^\nu$ is of the form $\Pi_{i=1}^{m} b_i$ for b_i a term of Λ,*
(3) *$n = m$,*
(4) *one can find an $i \in \{1, \dots, n\}$ such that $a_i = b_i$ is not provable in Λ.*

PROOF. Clauses 1 and 2 follow by Lemma 8.2, since a and b are not of type T. Otherwise, $a = b = k$ is provable in $\Lambda_\times$. Clause 3 follows from the fact that ha and hb are of the same type.

For clause 4 we have that if $a_i = b_i$ is provable in Λ, then $a_i = b_i$ is provable in $\Lambda_\times$. So if for every $i \in \{1, \dots, n\}$ we had $a_i = b_i$ provable in Λ, then $(ha)^\nu = (hb)^\nu$ would be provable in $\Lambda_\times$, which would imply that $a = b$ is provable in $\Lambda_\times$. □

9. The maximality of $\Lambda_\times$

For $a : \times_{i=1}^{n} A_i$ we define $\pi^i a : A_i$ as a for $n = 1$, and for $n > 1$,

$$\pi^i a = \begin{cases} p^2 a & \text{if } i = n \\ \pi^i p^1 a & \text{if } i < n. \end{cases}$$

We can now state the analogue of Böhm's Theorem for $\Lambda_\times$.

THEOREM 9.1. *If a and b are of the same type and $a = b$ is not provable in $\Lambda_\times$, then for every two terms c and d of the same type of $\Lambda_\times$ one can construct type-instances a' and b' of a and b, respectively, and terms $h, h_1, \dots, h_n$, $n \geq 0$, and also find variables $x_1, \dots, x_m$, $m \geq 0$, and a natural number i such that*

$$\pi^i(h\lambda_{x_1\dots x_m}a')h_1\dots h_n = c,$$
$$\pi^i(h\lambda_{x_1\dots x_m}b')h_1\dots h_n = d$$

are provable in $\Lambda_\times$.

The proof is obtained by applying Lemma 8.3 and Theorem 6.1.

It would be possible to prove Theorem 9.1 directly, without passing through Theorem 6.1, by applying Soloviev's version of Soloviev-Statman's theorem, which is given for $\Lambda_\times$ (see [17, Theorem 2, p. 187]. This would yield a different form for the equalities of Theorem 9.1.

There is an analogue of Theorem 6.2 obtained by refining Theorem 9.1, and for closed terms a and b of $\Lambda_\times$, the variables $x_1, \dots, x_m$ are not mentioned. We can, of course, prove an analogue of Theorem 9.1 for the typed lambda calculus with nonempty product types, i.e. without terminal type T. Finally, we can draw from Theorem 9.1 the Maximality Corollary where Λ is replaced by $\Lambda_\times$.

In the last part of this work we shall apply the following version of Theorem 9.1.

THEOREM 9.2. *If a and b are closed terms of the same type and $a = b$ is not provable in $\Lambda_\times$, then one can construct type-instances a' and b' of a and b, respectively, and closed terms $h, h_1, \dots, h_l$, $l \geq 0$, and also find a natural number i such that*

$$\pi^i(ha')h_1\dots h_l(p^1x)(p^2x) = p^1x,$$
$$\pi^i(hb')h_1\dots h_l(p^1x)(p^2x) = p^2x$$

are provable in $\Lambda_\times$.

10. Free cartesian closed categories

The equational calculus CCC of cartesian closed categories is introduced as follows. *Object terms* of CCC are the types of $\Lambda_\times$. For *arrow terms* of CCC we use the schematic letters f, g, h, $\dots$, f_1, $\dots$, and we indicate by $f : A \vdash B$, where A and B are types, that A is the *source* and B the *target* of f (we use $\vdash$ instead of the usual $\to$, which we have reserved for a type operation). We say that $A \vdash B$ is the *arrow type* of f.

Arrow terms are defined inductively starting from the *atomic arrow terms*

$$\mathbf{1}_A : A \vdash A,$$
$$\boldsymbol{p}^1_{A,B} : A \times B \vdash A, \quad \boldsymbol{p}^2_{A,B} : A \times B \vdash B,$$
$$\boldsymbol{\varepsilon}_{A,B} : (A \to B) \times A \vdash B,$$
$$\boldsymbol{k}_A : A \vdash \mathrm{T},$$

with the help of the partial operations on arrows

$$\frac{f : A \vdash B \qquad g : B \vdash C}{g \circ f : A \vdash C}$$

$$\frac{f : C \vdash A \qquad g : C \vdash B}{\langle\!\langle f, g \rangle\!\rangle : C \vdash A \times B}$$

$$\frac{f : C \times A \vdash B}{\mathbf{\Gamma}_{C,A} f : C \vdash A \rightarrow B}$$

The *formulae* of CCC are equalities $f = g$ where the arrow terms f and g have the same arrow type.

The axioms of CCC are

$$f = f,$$
$$f \circ \mathbf{1}_A = \mathbf{1}_B \circ f = f\,, \quad h \circ (g \circ f) = (h \circ g) \circ f,$$
$$\boldsymbol{p}^1_{A,B} \circ \langle\!\langle f, g \rangle\!\rangle = f\,, \quad \boldsymbol{p}^2_{A,B} \circ \langle\!\langle f, g \rangle\!\rangle = g,$$
$$\langle\!\langle \boldsymbol{p}^1_{A,B} \circ h, \boldsymbol{p}^2_{A,B} \circ h \rangle\!\rangle = h,$$
$$\boldsymbol{\varepsilon}_{A,B} \circ \langle\!\langle \mathbf{\Gamma}_{C,A} f \circ \boldsymbol{p}^1_{A \rightarrow B,A}\,,\ \boldsymbol{p}^2_{A \rightarrow B,A} \rangle\!\rangle = f,$$
$$\mathbf{\Gamma}_{C,A}(\boldsymbol{\varepsilon}_{A,B} \circ \langle\!\langle g \circ \boldsymbol{p}^1_{C,A}\,,\ \boldsymbol{p}^2_{C,A} \rangle\!\rangle) = g,$$
$$\text{for } f : A \vdash \mathrm{T}\,, \quad f = \boldsymbol{k}_A,$$

and its inference rules are replacement of equals and substitution of types for atomic types. (Substitution of types says that atomic types are variables; cf. the remarks on substitution of types at the end of Section 6.)

We speak of *the* calculus CCC, but, as a matter of fact, there are many such calculuses obtained by varying the generating set of atomic types. When we say that a cartesian closed category $\mathcal{K}$ is a model of CCC, we mean that the arrow terms $\mathbf{1}_A$ of CCC are interpreted by the unit arrows of $\mathcal{K}$, that the arrow terms $\boldsymbol{p}^1_{A,B}$ and $\boldsymbol{p}^2_{A,B}$ of CCC are interpreted by the projection arrows of $\mathcal{K}$, etc.

The calculus CCC does not have arrow-term variables, and hence its models are not necessarily cartesian closed categories. They need not even be categories. This calculus catches only the "canonical-arrow fragment" of cartesian closed categories.

The term model of CCC is a free cartesian closed category. This category is isomorphic to the category $\mathcal{C}(\Lambda_\times)$, engendered by $\Lambda_\times$, which we will define by slightly varying the approach of [11, I.11]; (see also [12]). The objects of $\mathcal{C}(\Lambda_\times)$ are again the types of $\Lambda_\times$, and the arrows are equivalence classes $[\lambda_x a] = \{b : A \rightarrow B \mid b$ is a closed term of $\Lambda_\times$ and $b = \lambda_x a$ is provable in $\Lambda_\times\}$. The arrow type of $[\lambda_x a]$ is $A \vdash B$.

The CCC structure of $\mathcal{C}(\Lambda_\times)$ is defined by

$$\mathbf{1}_A = [\lambda_x x],\ \text{ for } x : A,$$
$$\boldsymbol{p}^1_{A,B} = [\lambda_x p^1 x]\,, \quad \boldsymbol{p}^2_{A,B} = [\lambda_x p^2 x]\,,\ \text{ for } x : A \times B,$$
$$\boldsymbol{\varepsilon}_{A,B} = [\lambda_x p^1 x(p^2 x)]\,,\ \text{ for } x : (A \to B) \times A,$$
$$\boldsymbol{k}_A = [\lambda_x k]\,,\ \text{ for } x : A,$$
$$[a] \circ [b] = [\lambda_x a(bx)],$$
$$\langle\!\langle [a], [b] \rangle\!\rangle = [\lambda_x \langle ax, bx \rangle],$$
$$\boldsymbol{\Gamma}_{C,A}[a] = [\lambda_{xy} a \langle x, y \rangle].$$

11. The maximality of cartesian closed categories

We can now prove the following theorem.

MAXIMALITY OF CCC. *If the formula $f = g$ of CCC is not provable in CCC, then every cartesian closed category that is a model of CCC extended with $f = g$ is a preorder.*

PROOF. Suppose that for $f, g : A \vdash B$ the equality $f = g$ is not provable in CCC. So one can construct closed terms $\lambda_x a$ and $\lambda_x b$ of $\Lambda_\times$ of type $A \to B$, with $x : A$, such that we have $[\lambda_x a] = f$ and $[\lambda_x b] = g$, and $\lambda_x a = \lambda_x b$ is not provable in $\Lambda_\times$. By Theorem 9.2, one can construct type-instances $\lambda_{x'} a'$ and $\lambda_{x'} b'$ of $\lambda_x a$ and $\lambda_x b$, respectively, and closed terms $h, h_1, \ldots, h_l$, $l \geq 0$, such that for $y : p \times p$,

$$\lambda_y(\lambda_z \pi^i(hz)h_1 \ldots h_l(p^1 y)(p^2 y))\lambda_{x'} a' = \lambda_y p^1 y,$$
$$\lambda_y(\lambda_z \pi^i(hz)h_1 \ldots h_l(p^1 y)(p^2 y))\lambda_{x'} b' = \lambda_y p^2 y$$

are provable in $\Lambda_\times$.

Let $\mathcal{K}$ be a cartesian closed category that is a model of CCC plus $f = g$. Then there is a cartesian closed functor F from $\mathcal{C}(\Lambda_\times)$ to $\mathcal{K}$ such that $F(p) = C$, for C an arbitrary object of $\mathcal{K}$. If $\pi^i(hz)h_1 \ldots h_l(p^1 y)(p^2 y)$ is abbreviated by c, then we have in $\mathcal{K}$

$$F([\lambda_y(\lambda_z c)\lambda_{x'} a']) =$$
$$\boldsymbol{\varepsilon}_{F(A') \to F(B'), C} \circ \langle\!\langle F([\lambda_{yz} c]), \boldsymbol{\Gamma}_{C \times C, F(A')}(F([\lambda_{x'} a']) \circ \boldsymbol{p}^2_{C \times C, F(A')}) \rangle\!\rangle$$

and the analogous equality obtained by replacing a' by b'. Since in $\mathcal{K}$ we have $F([\lambda_{x'} a']) = F([\lambda_{x'} b'])$, we obtain in $\mathcal{K}$

$$F([\lambda_y p^1 y]) = F([\lambda_y p^2 y]),$$

i.e., $\boldsymbol{p}^1_{C,C} = \boldsymbol{p}^2_{C,C}$. Then for $h_1, h_2 : E \vdash C$ in $\mathcal{K}$ we have

$$\boldsymbol{p}^1_{C,C} \circ \langle\!\langle h_1, h_2 \rangle\!\rangle = \boldsymbol{p}^2_{C,C} \circ \langle\!\langle h_1, h_2 \rangle\!\rangle,$$

i.e., $h_1 = h_2$. $\square$

To prove the Maximality of CCC one could also use the Maximality Corollary for $\Lambda_\times$ and the soundness of $\Lambda_\times$ with respect to CCC models (see [13, Theorem 4.2, p. 310]).

Note that the construction of a', b', h, $h_1, \dots, h_n$ and i in Theorem 9.2, as well as in our other analogues of Böhm's Theorem, is in principle effective, though it may be quite involved. (This relies on the effectiveness of Soloviev's and Statman's theorem mentioned in the proof of Theorem 6.1.) The construction of $\lambda_x a$ and $\lambda_x b$ out of f and g in the proof of the Maximality of CCC is also effective, and derivations in $\Lambda_\times$ made of equalities between closed terms, on which we can rely in the proof of the Maximality of CCC, can be transformed effectively into derivations in CCC. All this entails that we have a constructive method to derive $h_1 = h_2$ from $f = g$ in the proof of the Maximality of CCC.

The Maximality of CCC makes it possible to generalize Soloviev's and Statman's theorem we have used in the proof of Theorem 6.1. (Čubrić has in [5] a related theorem about the existence of a faithful cartesian closed functor from free cartesian closed categories with free arrows, i.e. from models of CCC extended with arrow-term variables, into the category of sets.) This generalization says that every equality not provable in CCC can be falsified in any cartesian closed category that is not a preorder. Soloviev and Statman envisage as a falsifying category only the category of finite sets. Our previous result of [7] yields an analogous statement for cartesian categories. All these matters about generalizing Soloviev's and Statman's theorem (as well as Čubrić's) are treated clearly, systematically and with much insight in [16].

References

[1] H.P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, Amsterdam, 1981, revised edition 1984.

[2] E. Barendsen, *Representation of Logic, Data Types and Recursive Functions in Typed Lambda Calculi*, Doctoraal Scripte, Faculteit Wiskunde en Informatica, Katholicke Universiteit Nijmegen, 1989.

[3] C. Böhm, *Alcune proprietà delle forme β-η-normali nel λ-K-calcolo*, Pubblicazioni dell'Istituto per le Applicazioni del Calcolo, Rome, **696** (1968), 19 pp.

[4] H.B. Curry, J.R. Hindley and J.P. Seldin, *Combinatory Logic, Volume* II, North-Holland, Amsterdam, 1972.

[5] D. Čubrić, *Embedding of a free cartesian closed category into the category of sets*, J. Pure Appl. Algebra **126** (1998), 121–147.

[6] R. Di Cosmo, *Isomorphism of Types: From λ-Calculus to Information Retrieval and Language Design*, Birkhaüser, Boston, 1995.

[7] K. Došen and Z. Petrić, *The maximality of cartesian categories*, preprint, Rapport IRIT 97–42 (1997).

[8] S. Fortune, D. Leivant and M. O'Donnel, *The expressiveness of simple and second-order type structures*, J. ACM **30** (1983), 151–185.

[9] H. Friedman, *Equality between functionals*, in: R. Parikh ed., *Logic Colloquium '73*, Lecture Notes in Math. **453**, Springer-Verlag, Berlin, 1975, 22–37.

[10] J.-L. Krivine, *Lambda-calcul: Types et modèles*, Masson, Paris, 1990 (English translation, Ellis Horwood, 1993).

[11] J. Lambek and P.J. Scott, *Introduction to Higher-Order Categorical Logic*, Cambridge University Press, Cambridge, 1986.

[12] G.E. Mints, *Category theory and proof theory* (in Russian), in: *Aktual'nye voprosy logiki i metodologii nauki*, Naukova Dumka, Kiev, 1980, 252–278 (English translation, with permuted title, in: G.E. Mints, *Selected Papers in Proof Theory*, Bibliopolis, Naples, 1992).

[13] J.C. Mitchell and P.J. Scott, *Typed lambda models and cartesian closed categories*, in: J.W. Gray and A. Scedrov eds, *Categories in Computer Science and Logic*, Contemp. Math. **92**, American Mathematical Society, Providence, 1989, 301–316.

[14] J.G. Riecke, *Statman's 1-Section Theorem*, Inform. and Comput. **116** (1995), 294–303.

[15] H. Schwichtenberg, *Definierbare Funktionen im Lambda-Kalkül mit Typen*, Arch. math. Logik Grundlagenforsch. **17** (1976), 113–114. (We know this paper only from references.)

[16] A.K. Simpson, *Categorical completeness results for the simply-typed lambda-calculus*, in: M. Dezani-Ciancaglini and G. Plotkin eds, *Typed Lambda Calculi and Applications (Edinburgh, 1995)*, Lecture Notes in Comput. Sci. **902**, Springer-Verlag, Berlin, 1995, 414–427.

[17] S.V. Soloviev, *The category of finite sets and cartesian closed categories* (in Russian), Zapiski nauchn. sem. LOMI **105** (1981), 174–194 (English translation in J. Soviet Math. **22**, 1983, 1387–1400).

[18] R. Statman, *Completeness, invariance and λ-definability*, J. Symbolic Logic **47** (1982), 17–26.

[19] R. Statman, *λ-definable functionals and $\beta\eta$-conversion*, Arch. Math. Logik Grundlagenforsch. **23** (1983), 21–26.

[20] A.S. Troelstra, *Strong normalization for typed terms with surjective pairing*, Notre Dame J. Formal Logic **27** (1986), 547–550.

[21] A.S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press, Cambridge, 1996.

Acknowledgements. We would like to thank Alex Simpson for reading a previous version of this paper, and for making a very helpful suggestion (noted in Section 6). We are also grateful to Slobodan Vujošević for his careful reading of the text and for his comments.

Matematički institut SANU
Kneza Mihaila 35
11001 Beograd, p.p. 367
Yugoslavia
and
University of Toulouse III
IRIT
31062 Toulouse cedex
France

kosta@mi.sanu.ac.yu

(Received 14 06 1999)
(Revised 07 10 1999)

Matematički institut SANU
Kneza Mihaila 35
11001 Beograd, p.p. 367
Yugoslavia

zpetric@mi.sanu.ac.yu