ELSEVIER Journal of Pure and Applied Algebra 126 (1998) 121–147

JOURNAL OF PURE AND APPLIED ALGEBRA

Embedding of a free cartesian-closed category into the category of sets

Djordje Čubrić [a,b]

[a] *University of Cambridge, DPMMS, 16 Mill Lane, Cambridge, UK, CB2 1SB*
[b] *Matematički Institut, Kneza Mihaila 35/I, p.f.367, 11001 Belgrade, Yugoslavia*

Communicated by M. Barr; received 11 February 1993; revised 4 May 1993

Abstract

We show that every free cartesian closed category can be faithfully mapped to the category of sets. For that we use a Church–Rosser property of the appropriate typed lambda calculus. © 1998 Elsevier Science B.V.

1991 Math. Subj. Class.: Primary 18D15, 18B15, 18A15; secondary 03F05, 03B40

1. Introduction

The goal of this paper is to prove the following theorem.

Theorem 1.1. *Let $\mathscr{C}$ be a free cartesian closed category. Then there exists a faithful structure preserving functor $F : \mathscr{C} \to Set$.*

Informally, a free cartesian closed category is a cartesian closed category freely generated by objects and arrows between generated objects.

Some consequences of the above result are that various extensions of cartesian closed structure do not impose additional equalities among arrows:let $I : \mathscr{C} \to \mathscr{B}(\mathscr{C})$ be the canonical map from a free cartesian closed category $\mathscr{C}$ to the free Boolean topos $\mathscr{B}(\mathscr{C})$ generated by $\mathscr{C}$ then I is faithful. But perhaps more important is that it confirms our intuition that cartesian closed categories indeed axiomatize the cartesian closed structure of sets. (In "everyday practice" it means that a diagram commutes in every cartesian closed category if and only if it commutes in Set.)

* Correspondence address: University of Cambridge, DPMMS, 16 Mill Lane, Cambridge CB2 1SB, UK. E-mail: cubric@triples.math.mcgill.ca.

PII S0022-4049(96)00137-5

Concerning the history of Theorem 1.1, we note that the problem whether it was true had been raised, along with analogous problem involving monoidal closed categories, by M. Barr and others, many years ago. Ref. [16] gives a proof that for every two arrows in a free cartesian closed category without free arrows there is a structure-preserving functor into the category of finite sets which distinguishes these two arrows; in the proof the (then unrepaired) Mints' reductions are used.

Theorem 1.1 is formally analogous to results in [15, 6], each of which gives representations, in the form of structure-preserving functors, of cartesian-closed and richer structures in certain toposes; in place of faithfulness, other conditions are imposed on the representation. The methods in this paper are quite different from these of [15] or [6].

The above theorem is proved using connection between cartesian closed categories and typed λ-calculi for the "early history" of the subject; see the comments in [10]. A key technical step is that in a free cartesian closed category one can faithfully adjoin infinitely many maps $1 \to C$ for every object C. This is shown with the help of a system of reductions suggested by Mints. Unfortunately the original paper contains some mistakes, as Harnik pointed out to us, see Remark 4.17; since we think that these reductions are very interesting on their own right, we repair Mints' proof (of confluence as well as normalization). Also, an important ingredient in the proof is a variant of Friedman's completeness result for (a variant of) typed λ-calculus.

We obtained the main theorem in spring 1990 and I gave a talk on that at a McGill seminar organized by Prof. Lambek. However, I was using Mints' result without noticing this mistake in it. In December 1991 I corrected these mistakes in Mints' paper and distributed almost the same version of the paper in March 1992 to some people at McGill University. Since the end of July, beginning of August 1992 the paper was available from an "ftp-site" as announced on two e-mail lists (under the name "On free CCC"). The only mathematical changes are two additional corollaries about Mints' reductions – Corollaries 4.15 and 4.16 which are immediate consequences of our main result about Mints' reductions, i.e. Proposition 4.3. Corollary 4.15 is the main result in [1] – a paper which has our paper as a reference. Also, independently, Jay [8] gives a different proof of strong normalization for a system in which every type had a closed normal term – property not available in general.

Michael Makkai told me that the above result should be true and suggested some of the tools in proving it. I want to thank him and Victor Harnik for collaboration.

So, first we explain again that cartesian closed categories are "the same" as typed λ-calculi. A reader familiar with [10] or [13] can freely skip the following section (the difference in the presentation is given in Remark 3.3).

2. Basic information

We are going to work with the following definitions.

Definition 2.1 (*Cartesian closed category*). A category $\mathscr{C}$ is cartesian closed (ccc) if there is an object, denoted by 1, and for every two objects $A, B \in \mathscr{C}$ there is an object, denoted $A \times B$ and one object, denoted A^B and there are the following arrows:

- $0_A \in \hom(A,1)$, $\pi_{A,B} \in \hom(A \times B, A)$, $\pi'_{A,B} \in \hom(A \times B, B)$,
- $\varepsilon_{A,B} \in \hom(A^B \times B, A)$;

and the following operations on homsets:

- $\hom(C,A) \times \hom(C,B) \xrightarrow{\langle,\rangle} \hom(C, A \times B)$,
- $\hom(A \times B, C) \xrightarrow{*} \hom(A, C^B)$

(the operations would have to have indexes, but since they are uniquely determined by their arguments we omit them). These (constants and) operations have to satisfy the following equations:

$$\begin{array}{llll} (T) & f = 0_A, & & \\ (Pr_1) & \pi_{A_1,A_2}\langle f_1, f_2\rangle = f_1, & (Pr_2) & \pi'_{A_1,A_2}\langle f_1, f_2\rangle = f_2, \\ (SP) & \langle \pi_{A,B}g, \pi'_{A,B}g\rangle = g, & & \\ (B) & \varepsilon_{A,B}\langle h^*\pi_{C,B}, \pi'_{C,B}\rangle = h, & (H) & (\varepsilon_{A,B}\langle k\pi_{C,B}, \pi'_{C,B}\rangle)^* = k \end{array}$$

for every arrow $f \in \hom(A,1)$, $f_i \in \hom(C,A_i)$, $g \in \hom(C, A \times B)$, $h \in \hom(C \times B, A)$ and $k \in \hom(C, A^B)$. When space for superscripts is needed we may write π_1, π_2 instead of π, π'.

As we can see, we will work with cartesian closed categories with already chosen structure and with functors preserving the chosen structure "on the nose", just as in [10]. However, the main result holds for the "ordinary" free cartesian closed categories as well – every such is equivalent to a free cartesian closed category with a chosen structure.

We will use the following abbreviations: $\vec{A} = A_1 \times \cdots \times A_n$ when the brackets are nested on the left; also if $\vec{B}$ is a subsequence of $\vec{A}$ then $\pi^{\vec{B}}_{\vec{A}} : \vec{A} \to \vec{B}$ denotes the canonical projection.

Definition 2.2 (*Typed λ-calculus*). A typed λ-calculus is a formal system which consists of three classes: Types, Terms and Equations. They have to satisfy the following conditions:

Types: Types are freely generated from a set of basic types-sorts and the following rules: $1 \in$Types, $A, B \in$Types then $A \times B, A^B \in$Types.

Terms: For each type A we have countably many variables of type A (we denote them as x_i^A or $x_i : A$) and they are terms, also for every type we may have a set of basic constants of this type and they are terms; the other terms are generated as follows: $* : 1$ is a term, and then if $a : A_1 \times A_2$, $a_i : A_i$ (i=1,2), $f : A^B$, $b : B$ are terms then $\pi(a) : A_1$, $\pi'(a) : A_2$, $\langle a_1, a_2\rangle : A_1 \times A_2$, $(f`b) : A$, $\lambda x^A.b : B^A$ are terms. (We can notice that every term has uniquely assigned type.) The notions of free (bounded) variable in a term t are standard, $FV(t)$ will denote the set of the free variables in t.

Equations: They always have the following form $s =_X t$ where $s, t \in$Terms and X is a set of (typed) variables such that $FV(s) \cup FV(t) \subseteq X$.

Convention: When $FV(s) \cup FV(t) = X$ we often omit X in $s =_X t$. Also, typing is omitted whenever convenient.

The following expressions are equations (we call them axioms of λ-calculus):

$$(T)\quad f^1 = *,$$
$$(Pr_1)\ \pi_{A_1,A_2}(\langle f_1, f_2\rangle) = f_1, \qquad (Pr_2)\ \pi'_{A_1,A_2}(\langle f_1, f_2\rangle) = f_2,$$
$$(SP)\ \langle \pi_{A,B}(g), \pi'_{A,B}(g)\rangle = g,$$
$$(\beta)\quad (\lambda x^A.h`r) = h(r/x^A), \qquad (\eta)\quad \lambda x^B.(k`x^B) = k \quad x^B \notin FV(k)$$

for every term $f : 1$, $f_i : A_i$, $g : A \times B$, $h : B$, $r : A$ and $k : A^B$ ($h(r/x)$ denotes the substitution of r instead of all free occurrences of x in $h(x)$ but first taking care of clashes of variables – so we are all the time working under α-congruence since it is possible to do that naively as in untyped λ-calculus and it is safe for our purposes).

Equations are obtained also by the following rules (we also say that proofs are formed from the axioms and the following rules):

$$(R)\ \frac{}{t =_X t} \quad (S)\ \frac{s =_X t}{t =_X s} \quad (Tran)\ \frac{r =_X s \quad s =_Y t}{r =_{X \cup Y} t},$$

$$(\xi)\ \frac{t =_{X\cup\{x\}} s}{\lambda x.t =_X \lambda x.s} \quad (Sub')\ \frac{a^B =_X b^B \quad s^{A^B} =_Y t^{A^B}}{(s`a) =_{X\cup Y} (t`b)}.$$

The need for having indexed equations-contexts will be explained later. We can have some other basic types (sorts) and some other basic terms (constants). A set of equations added to the above system we will call a *λ-theory*.

In the presence of (Pr_i) and $(Tran)$ one can see that the reflexivity (rule (R)) is not needed. Also, it is a simple exercise to see that the following rules are derivable (the usual care about clashes of variables is needed for the second rule):

$$(W)\ \frac{t =_X s}{t =_{X\cup Y} s} \quad (Sub)\ \frac{a^B =_X b^B \quad s =_{Y\cup\{x^B\}} t}{s(a/x) =_{X\cup Y} t(b/x)}.$$

The following expression $(x_1^{A_1}, \ldots, x_n^{A_n} \rhd t)$ called term with context is going to be used often, it will always satisfy $FV(t) \subseteq \{x_1^{A_1}, \ldots, x_n^{A_n}\}$. And one more piece of notation: The symbol $t \equiv s$ is used to denote that t and s are equal as strings (but again up to α-congruence).

Definition 2.3 (*Interpretation, model*). An *interpretation* M of a language L in a cartesian closed category $\mathscr{C}$ is a function which assigns objects to basic types (sorts); if the language L has some basic constants it is assumed that the category $\mathscr{C}$ had them prescribed in advance (as the arrows from the terminal object). Then the interpretation assigns arrows to terms as follows (using induction on complexity of terms):

- $M(x_1^{A_1}, \ldots, x_n^{A_n} \rhd x_i) = \pi_{\vec{A}}^{A_i}$.
- $M(\vec{x}{:}\vec{A} \rhd *) = 0_{\vec{A}}$. If the context were empty then we would have $M(\ \rhd *) = 1_1$.
- $M(\vec{x}{:}\vec{A} \rhd c) = c0_{\vec{A}}$ (here c is a constant). Also we could have empty context, then $M(\ \rhd c) = c$.

- $M(\vec{x}:\vec{A} \rhd \pi_i(t)) = \pi_i M(\vec{x}:\vec{A} \rhd t),\ \ i = 1, 2.$
- $M(\vec{x}:\vec{A} \rhd \langle t_1, t_2\rangle) = \langle M(\vec{x}:\vec{A} \rhd t_1), M(\vec{x}:\vec{A} \rhd t_2)\rangle.$
- $M(\vec{x}:\vec{A} \rhd (t_1`t_2)) = \varepsilon\langle M(\vec{x}:\vec{A} \rhd t_1), M(\vec{x}:\vec{A} \rhd t_2)\rangle.$
- $M(\vec{x} : \vec{A} \rhd \lambda y^B.t) = (M(\vec{x} : \vec{A}, y : B \rhd t))^*$, if $\vec{x} : \vec{A}$ were not there we would have $M(\ \rhd \lambda y^B.t) = (M(y^B \rhd t)\pi'_{1,M(B)})^*$.

Model of a λ-theory T is an interpretation such that all equations from T are preserved.

For an interpretation $M : L \to \mathscr{C}$ (model $M : T \to \mathscr{C}$) and for a cartesian closed functor $F : \mathscr{C} \to \mathscr{D}$ by $F \circ M$ we denote the interpretation $N : L \to \mathscr{D}$ (model $N : T \to \mathscr{D}$) defined as follows: on basic types $N(A) = F(M(A))$ and on basic constants $N(c) = F(M(c))$. Now it is easy to see that the first equation is actually true for all types and that the second equations generalize to the terms with contexts i.e. $N(x : A \rhd t) = F(M(x : A \rhd t))$. So indeed N is an interpretation of L. That N is also a model (if M is one) will follow from the soundness below.

Now we can show soundness of our interpretation but before that we have to give a useful technical lemma which can be proved by induction on the complexity of terms:

Lemma 2.4. *Every interpretation M satisfies the following*:

1. $M(z^{A_1 \times \cdots \times A_n} \rhd f(\pi_1(z), \ldots, \pi_n(z))) = M(x_1^{A_1}, \ldots, x_n^{A_n} \rhd f(x_1, \ldots, x_n))$.

If $\vec{y} : \vec{B}$ is not free in t then:

2. $M(\vec{x}:\vec{A}, \vec{y}:\vec{B} \rhd t) = M(\vec{x}:\vec{A} \rhd t)\pi^{M(\vec{A})}_{M(\vec{A}), M(\vec{B})}$;
3. $M(\vec{x}:\vec{A} \rhd f(\vec{x}, (g(\vec{x}))/y^B)) = M(\vec{x}:\vec{A}, y:B \rhd f(\vec{x}, y^B))\langle 1_{\vec{A}}, M(\vec{x}:\vec{A} \rhd g(\vec{x}))\rangle$.

Proposition 2.5 (Soundness). *Let T be a λ-theory. Let M be a model of T in a cartesian closed category. Then*:

$$\text{if } T \vdash f =_X g \text{ then } M(X \rhd f) = M(X \rhd g).$$

Proof. As usual, this can be proved by induction on the complexity of proofs. □

Remark 2.6. Without contexts we would not have soundness – it would be provable (using (Sub'), β and $(Tran)$):

$$\lambda x^X.f = \lambda x^X.g \vdash f = g$$

and every interpretation in *Set* which maps X to empty set is a model of the left side but does not have to be of the right side. However, using the rules with contexts we get "only"

$$\lambda x^X.f = \lambda x^X.g \vdash f =_{FV(f,g) \cup \{x\}} g$$

and the above interpretation is a model for both sides.

Definition 2.7 (*Internal language*). To every cartesian closed category $\mathscr{C}$ we can associate a λ-language $L_{\mathscr{C}}$ (*internal language*) as follows:

- The objects become the set of basic types. When we want to be precise, the basic type corresponding to an object A we will denote by X_A (this is required when we want to make difference between types such as $X_A \times X_B$ and $X_{A\times B}$).
- The arrows from the specified terminal object 1 become the basic constants – but in several different ways! More precisely: the basic constants of type $\mathscr{T}(X_{A_1},\dots,X_{A_n})$ (a type built from the basic types $X_{A_1},\dots,X_{A_n}$) are $\hom_{\mathscr{C}}(1,\mathscr{T}(A_1,\dots,A_n))$. (Thus, we have (at least) two different constants $c_f : X_{A_1\times A_2}$ and $c_f : X_{A_1}\times X_{A_2}$ corresponding to the same $(1 \xrightarrow{f} A_1 \times A_2) \in \mathscr{C}$.)

The *standard interpretation* M is the interpretation which to every symbol of the internal language assigns the intended meaning: $X_A \mapsto A$ and $c_f : \mathscr{T}(X_{A_1},\dots X_{A_n}) \mapsto f : 1 \to \mathscr{T}(A_1,\dots A_n)$.

The corresponding *λ-theory* $T_{\mathscr{C}}$ contains all equations satisfied by the standard interpretation: $t^A =_X s^A \in T_{\mathscr{C}}$ iff $M(X \rhd t) = M(X \rhd s)$.

(We could have included "term constructors" (unary functions) – every arrow $A \xrightarrow{f} B$ becomes a term constructor: if $t:A$ is a term then $f(t):B$ is a new term. However, it would not give anything new since among the equations of the theory we would have to include $f(t) = (\hat{f}`t)$ where $\hat{f}$ is the name of the constant corresponding to the transpose of f, i.e. $\hat{f} = (f\pi'_{1,A})^*$.)

Proposition 2.8 (Completeness). *For a given λ-theory T there exists a canonical model $M : T \to \mathscr{C}_T$ such that $M(X \rhd u) = M(X \rhd v)$ only if $T \vdash u =_X v$.*

Proof. This is a standard construction and it is given as follows.

Objects: Objects are types.

Arrows: They are classes of equivalent terms with contexts. To compare $(x_1 : A_1,\dots,x_n : A_n \rhd f^D(x_1^{A_1},\dots,x_n^{A_n}))$ with $(y_1 : B_1,\dots,y_m : B_m \rhd g^D(y_1^{B_1},\dots,y_m^{B_m}))$ we first have to have $(\cdots(A_1\times A_2)\times\cdots)\times A_n \equiv (\dots(B_1\times B_2)\times\cdots)\times B_m$, call it C. (So, assuming $m \le n$ it says that $B_m \equiv A_n,\dots,B_2 \equiv A_{n-m+2}$ and $B_1 \equiv (\cdots(A_1\times A_2)\times\cdots)\times A_{n-m+1}$.) Then we say that they are equivalent iff

$$T \vdash f(\pi_1(z),\dots,\pi_n(z)) =_z g(\pi_1(z),\dots,\pi_m(z)).$$

The class above gives an arrow $C \to D$.

Composition: $(\vec{y}^B \rhd g)(\vec{x}^A \rhd f) = (\vec{x}^A \rhd g(\pi_1(f)/y_1,\dots,\pi_m(f)/y_m))$. Here f is of the type $\vec{B}$.

Units: $1_A = (x:A \rhd x)$.

Cartesian structure: This is going to be defined on the representatives of arrows which have one free variable:

- $0_A = (x:A \rhd *)$.
- $\pi_{A,B} = (x:A\times B \rhd \pi(x))$.
- $\langle (x:A \rhd f(x)),(y:A \rhd g(y))\rangle = (x:A \rhd \langle f(x),g(x)\rangle)$. (Sic!)

Closed structure:

- $\varepsilon_{A,B} = (x:A^B \times B \rhd (\pi_1(x)`\pi_2(x)))$.
- $(x:A \times B \rhd f(x))^* = (x_1:A \rhd \lambda x_2.f(\langle x_1,x_2 \rangle))$.

The equivalence classes which correspond to $(\rhd c)$, where c is a constant from the language, we will denote also by c.

As usual, the first thing to check is independence on representatives. But this is true because of the substitution rule (Sub) for typed λ-calculus. Second, it would be easy to check that this is a ccc. The canonical interpretation which assigns types to the same-name-objects, constants to the same-name-arrows is obviously a model of T. Whole construction is such that "by definition" completeness follows. □

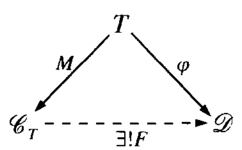
Corollary 2.9. *The canonical model $M : T \to \mathscr{C}_T$ classifies all models of T in the following way: the map $\mathscr{CCC}(\mathscr{C}_T, \mathscr{D}) \xrightarrow{-\circ M} Mod_T\mathscr{D}$ is bijection. ($\mathscr{CCC}(\mathscr{C}, \mathscr{D})$ denotes the set of all structure-preserving functors (cc-functors) between these two cartesian closed categories, $F \circ M$ was defined in Definition 2.3 and $Mod_T\mathscr{D}$ are all models of T in $\mathscr{D}$).*

Proof. Let us just prove surjectivity of the above map. Take a model $N : T \to \mathscr{D}$; we have to find cc-functor $F : \mathscr{C}_T \to \mathscr{D}$ such that $N = F \circ M$. F on $Ob(\mathscr{C}_T)$ is easily defined since $Ob(\mathscr{C}_T)$ are types of T so $F(A) = N(A)$. Since the arrows of $\mathscr{C}_T$ are classes of equivalent terms with contexts we are going to define $F = N$ on arrows (also) (recall the definition of interpretation). Now we have to show that F does not depend on the choice of representatives and that F is indeed a cc-functor. The first part follows from the completeness, and the second from the definition of the cc-structure on $\mathscr{C}_T$. □

3. On Friedman completeness for typed lambda calculus

Definition 3.1 (*Free CCC*). Let L be a λ-language and T be the theory on this language with no additional axioms (empty theory). To this T one can associate the cartesian closed category $\mathscr{C}_T$ as in Proposition 2.8. $\mathscr{C}_T$ is then called *free* cartesian closed category.

Its universal property is given in Corollary 2.9. Pictorially:



where M is the canonical model and $\varphi \in Mod_T\mathscr{D}$ but since the theory T does not have additional equations we can say that φ is just an interpretation of symbols.

This is a generalization of the notion 'category generated by graph' since the free arrows can be of arbitrary type. This is required if we want to consider these categories as categories of proofs, and also if we want to avoid identification of types in the definition of an internal language. We do not have to define a more general notion where the free arrows have arbitrary domain – this is included by the definition of exponents. If we were to analyze the above diagram we would obtain exactly the definition of a free (bi^-)cartesian closed category given in [11, 6]. Let us now recall the notion of a "free arrow".

Proposition 3.2 (Free arrow). *For every cartesian closed category $\mathscr{C}$ and for every object C in $\mathscr{C}$ there exists a cartesian closed category $\mathscr{C}[1 \xrightarrow{\xi} I(C)]$ and a cc-functor $I : \mathscr{C} \to \mathscr{C}[1 \xrightarrow{\xi} I(C)]$, such that for every cc-functor $F : \mathscr{C} \to \mathscr{D}$ and every arrow $F(1) \xrightarrow{a} F(C)$ there exists a unique cc-functor $\mathscr{C}[1 \xrightarrow{\xi} I(C)] \xrightarrow{G} \mathscr{D}$ such that $G \circ I = F$ and $G(\xi) = a$. This ξ is called the free arrow.*

Proof. First form the slice category $\mathscr{C}/C$ (in general it does not have to be cartesian closed) and consider the canonical functor $\mathscr{C} \xrightarrow{I} \mathscr{C}/C$ which maps an object A to $A \times C \xrightarrow{\pi_2} C$ and an arrow $(A \xrightarrow{f} B)$ to $(A \times C \xrightarrow{\langle f\pi_1, \pi_2 \rangle} B \times C)$. Now, form the full subcategory of $\mathscr{C}/C$ spanned by the objects from the image of I, i.e. all the objects are of the form $A \times C \xrightarrow{\pi_2} C$. Denote this full subcategory by $\mathscr{C}/\!/C$. It is easy to see that $\mathscr{C}/\!/C$ is a cartesian closed category, that the functor I is a cc-functor and that the whole construct $\mathscr{C} \xrightarrow{I} \mathscr{C}/\!/C$ satisfies "up to equivalence" the universal property from the proposition – the role of the free arrow $1 \xrightarrow{\xi} I(C)$ is played by the arrow $1 \times C \xrightarrow{\langle \pi_2, \pi_2 \rangle} C \times C$. However, to obtain the category which will satisfy the above property as stated we take for its objects objects of $\mathscr{C}$ and for its arrows between A and B we take $\mathscr{C}/\!/C(I(A), I(B))$. This category we also denote by $\mathscr{C}/\!/C$. This construction is described in more detail in [11] (for an equivalent construction see [10]).

In the special case when the category $\mathscr{C}$ is a free cartesian closed category the construction can be equivalently described as follows. The category $\mathscr{C}$ is obtained from a "free λ-theory" T as in Definition 3.1, i.e. $\mathscr{C} = \mathscr{C}_T$. Now add to the language of T a new constant $\xi : C$. Then the new theory, which we denote simply by $T \cup \{\xi\}$, has no additional axioms. Now form the category $\mathscr{C}_{T \cup \{\xi\}}$. I is the obvious functor which maps things to the same name things. □

Remark 3.3. In [10] the interpretation of terms of a λ-calculus in a cartesian closed category $\mathscr{C}$ uses the previous notion of free arrow. Similarly, the notions of the internal language $L_{\mathscr{C}}$ and the theory $T_{\mathscr{C}}$ associated to a cartesian closed category $\mathscr{C}$ use not only the notion of free arrow but also identifications of types in the theory which we avoid. It is not hard to show that this theory is essentially the same as ours – the categories associated to them are equivalent. Let us just add that the interpretation of λ-terms in [13] is the same as ours, but for definition of the theory associated to a cartesian closed category they quote [10].

Now, we want to enrich the free cartesian closed category with a lot of free arrows so that in this new category 1 generates. But first, we have to show that these new arrows do not spoil anything. However obvious it may look, one has to be careful, bearing in mind that in a nonfree case it does not have to be true (e.g. adding a free arrow from 1 to the empty set in the category of sets "spoils the thing": the canonical functor from *Set* to the new category is not faithful; moreover, the new category is equivalent to a point). In a sense this is the only case when something like that may happen as the following easy lemma describes – "nonempty can be inhabited".

Lemma 3.4. *The following statements are equivalent for any cartesian closed category $\mathscr{C}$:*

- *The canonical functor $I:\mathscr{C} \to \mathscr{C}[\xi]$ is faithful (where $\mathscr{C}[\xi]$ denotes the category $\mathscr{C}$ with the freely added arrow $\xi : 1 \to C$).*
- *The terminal arrow $0_C : C \to 1$ is epi in $\mathscr{C}$.*

Proof. Assume that I is faithful. If $f0_C = g0_C$ in $\mathscr{C}[\xi]$ then multiply by ξ and use faithfulness. The other direction is also easy. It is enough to prove faithfulness of I on arrows from 1. Take two such arrows $f, g \in \mathscr{C}$. Suppose $I(f) = I(g)$ in $\mathscr{C}[\xi]$ that is $1 \times C \overset{\langle f\pi_1, \pi_2\rangle}{\longrightarrow} C \times C = 1 \times C \overset{\langle g\pi_1, \pi_2\rangle}{\longrightarrow} C \times C$ in $\mathscr{C}$ (see Proposition 3.2). This is the same as $f\pi_1 = g\pi_1$. Multiplying from the right by $\langle 0_C, 1_C\rangle$ we get $f0_C = g0_C$. Since we assumed that 0_C was epi, we have $f = g$. □

The point which we want to make is that in a free ccc adding of a free arrow is safe. For that we use the following proposition which will be proved in Section 4.

Proposition 3.5 (Free types are nonempty). *If $f =_x g$ in a free λ-calculus and x does not occur as a free variable in either f or g then we also have $f = g$.*

Now, we can establish the following.

Proposition 3.6 (Key proposition). *In a free cartesian closed category $\mathscr{C}$ every 0_C is epi.*

Proof. Let f and g be two arrows in $\mathscr{C}$ such that $f0_C = g0_C$. In the corresponding free λ-calculus it gives $f =_{x^C} g$ (see Lemma 2.4.2); here x^C does not appear in f, g. By the previous proposition it means $f = g$ in the λ-calculus. Therefore $f = g$ in $\mathscr{C}$ (by soundness). □

Corollary 3.7. *Let $\mathscr{C}$ be a free cartesian closed category and let $\mathscr{D}$ be a free cartesian closed category obtained from $\mathscr{C}$ by adding infinitely many free arrows $1 \overset{c_{ij}}{\longrightarrow} C_j$ for every object $C_j \in \mathscr{C}$. Then the canonical functor $I : \mathscr{C} \to \mathscr{D}$ is a faithful cc-functor.*

Proof. Adding one free arrow is faithful by Lemma 3.4 and Proposition 3.6. Adding finitely many follows by induction. To add infinitely many free arrows consider the constructions of a free cartesian closed category: let $\mathscr{C} = \mathscr{C}_T$ for a free λ-theory (as in the definition of free cartesian closed category). Then $\mathscr{D}$ can be constructed as $\mathscr{C}_{T'}$ where $T' = T \cup \{\xi_{ij}^{C_j} \mid C_j \in \mathscr{C}, i \in I\}$ (see the end of the proof of Proposition 3.2). The functor I is the unique cc-functor which classifies the model $T \xrightarrow{M'|_T} \mathscr{D}$ where M' is the canonical model $T' \xrightarrow{M'} \mathscr{D}$ and $M'|_T$ is the reduct of it on T; so we have $I \circ M = M'|_T$. If I were not faithful it would mean that there are two closed terms t and s in T such that $T \not\vdash t = s$ and yet $T' \vdash t = s$. Since every proof uses finitely many symbols we would have $T'' = T \cup \{\xi_1^{C_1}, \ldots, \xi_n^{C_n}\}$ – a finite extension of T such that $T'' \vdash t = s$. Since T'' is a finite extension we know (by the above induction) that it has to be faithful and therefore $T \vdash t = s$, contrary to the assumption.

Alternatively, to add infinitely many free arrows we could form the filtered colimit of all the finite extensions. Then use that two arrows are equal in the colimit if they were already equal in a finite extension. □

To continue the proof of Theorem 1.1 we need the following result which is a corollary of the variant of Friedman completeness – this corollary will also be proved in the next section.

Corollary 3.8. *Let $\mathscr{D}$ be a free cartesian closed category which has infinitely many free arrows for every object. Then there exists a faithful, structure preserving functor $\mathscr{D} \xrightarrow{F} Set$.*

We can recapitulate as follows.

Proof of our main result – Theorem 1.1. Take a free cartesian closed category $\mathscr{C}$, add infinitely many free arrows to every object in $\mathscr{C}$. Call the new category $\mathscr{D}$. The canonical functor $I : \mathscr{C} \to \mathscr{D}$ is cc and faithful by corollary 3.7. Also the previous functor $F : \mathscr{D} \to Set$ is cc and faithful. So $F \circ I : \mathscr{C} \to Set$ is the faithful cc-functor. □

Let us prove a corollary, observed by Michael Barr, which emphasizes the usefulness of free ccc *with* free arrows and at the same time its statement does not require this notion.

Corollary 3.9. *For every projective (with respect to cc-functors which are surjective on arrows) cartesian closed category there exists a faithful cc-functor into the category of Sets.*

Proof. Let $\mathscr{D}$ be such a category. Let $\mathscr{C}$ be a free ccc generated by objects and arrows from $\mathscr{D}$. Obviously, $\mathscr{D}$ is a retract of $\mathscr{C}$ and since $\mathscr{C}$ "embeds" in Sets we are done. □

As we have seen above, the only things which remain to be proved are Proposition 3.5 and Corollary 3.8. Let us first mention the following obvious fact about λ-calculus (without additional equalities).

Lemma 3.10. *Let* $(\varphi_1{}^{B^C}`\xi^C) = (\varphi_2{}^{B^C}`\xi^C)$ *and assume* ξ_C *is a constant which does not appear in* φ_1 *and* φ_2*. Then* $\varphi_1 = \varphi_2$.

Proof. In the proof of $(\varphi_1{}^{B^C}`\xi^C) = (\varphi_2{}^{B^C}`\xi^C)$ replace all the occurrences of ξ^C with a brand new variable x^C and then use (η). □

Theorem 3.11 (Essentially Friedman [4]). *Let* L *be a free typed* λ*-calculus which has infinitely many basic constants for every type. Then there exists a model* $L \xrightarrow{N} Set$ *such that* $N(X \rhd t_1) = N(X \rhd t_2)$ *implies* $L \vdash t_1 =_X t_2$.

Proof. It is enough to specify N on the basic (free) types and the constants. To do that we introduce an auxiliary map – premodel $\Gamma : L \to Set$ which maps a type A to $\{[(\ \rhd t)] : t : A\}$, $[-]$ denotes an equivalence class (under provable equality), also notice that since the context is empty the terms have to be closed. To simplify notation a bit we will denote a term with a context only by the name of the term if it does not cause confusion. Now if X is a free type (or 1) then $N(X) \stackrel{\text{def}}{=} \Gamma(X)$. To give N on the arrows we need a family of partially defined surjective maps $s_A : N(A) \to \Gamma(A)$, $A \in Types(L)$.

Claim 1. *Let the family of partial maps* $s = \{s_D : D \in Types(L)\}$ *be defined as follows:*
- $s_X = 1_{\Gamma(X)}$, *X is a free object or 1.*
- $s_{A\times B}(a,b) = [\langle t_1, t_2\rangle]$, *where* $t_1 \in s_A(a)$ *and* $t_2 \in s_B(b)$.
- *Let* $f \in N(C)^{N(B)}$*. Then* $s_{C^B}(f)$ *is defined and equal to* $[\varphi] \in \Gamma(C^B)$ *if for every* $b \in Dom(s_B)$ $f(b) \in Dom(s_C)$ *and*

$$s_C(f(b)) = [(\varphi`r)],\ \ r \in s_B(b). \tag{1}$$

Then the family is well defined and all components are surjective.

Proof of Claim 1. The proof is by the induction on the complexity of types. Obviously, for the free types and 1 the statement is true. Also for the product types. For the exponent type C^B, Lemma 3.10 insures that there is only one such $[\varphi]$ if any. (Assume that there are two: $[\varphi_1]$ and $[\varphi_2]$, by induction hypothesis s_B is surjective so $[\xi] \in \mathrm{Im}(s_B)$ where ξ is not in φ_1, φ_2. Then from (1) follows $(\varphi_1`\xi) = (\varphi_2`\xi)$ and therefore $\varphi_1 = \varphi_2$.) To show that s_{C^B} is surjective take an arbitrary $[\varphi] \in \Gamma(C^B)$ then the witness $f \in N(C)^{N(B)}$ is chosen so that $f(b) \in s_C^{-1}([\varphi`r])$ if $b \in Doms_B$ (take any $r \in s_B(b)$) and arbitrarily otherwise. □

Now we can define $N(\xi)$ for ξ^D a basic constant. $N(\xi) = d$ such that $s_D(d) = [\xi]$ (if there are several such $d \in N(D)$ we choose one of them).

Claim 2. *For every* $(x_1^{A_1}, \dots, x_n^{A_n} \rhd f^B)$ *in* L *and every* $a_i \in Dom(s_{A_i})$

$$N(f)(a_1, \dots, a_n) \in Dom(s_B)$$

and

$$s_B(N(f)(a_1, \dots, a_n)) = [f(t_1/x_1, \dots, t_n/x_n)], \quad t_i \in s_{A_i}(a_i). \tag{2}$$

Proof of Claim 2. This is by induction on the complexity of f. If $f \equiv \xi^D$ then by the definition of $N(\xi)$ we have $s_D(N(\xi)) = [\xi]$ and this is indeed (2) since $s_1(1_1) = 1_1$. Let us check only one case more: $f^{C^B} \equiv \lambda y^B.h^C$. Take $a_i \in Dom(s_{A_i})$. We must show that $N(\lambda y.h)(a_1, \dots, a_n) \in Dom(s_{C^B})$ and $s_{C^B}(N(\lambda y.h)(a_1, \dots, a_n) = [\lambda y.h(t_1/x_1, \dots, t_n/x_n)]$, $(t_i \in s_{A_i}(a_i))$. It is enough to show that $\lambda y.h(t_1/x_1, \dots, t_n/x_n)$ satisfies (1) in place of φ, i.e. for every $b \in Dom(s_B)$ it holds that

$$s_C(N(\lambda y.h)(a_1, \dots, a_n)(b)) = [(\lambda y.h(t_1/x_1, \dots, t_n/x_n)`r)]$$

$r \in s_B(b)$, because by the uniqueness of $[\varphi]$ it will follow that

$$s_{C^B}(N(\lambda y.h)(a_1, \dots, a_n)) = [\lambda y.h(t_1/x_1, \dots, t_n/x_n)].$$

But first we have to check that $(N(\lambda y.h)(a_1, \dots, a_n))(b) \in Dom(s_C)$; this is so by the induction hypothesis since $N(\lambda y.h)(a_1, \dots, a_n)(b) = N(h)(a_1, \dots, a_n, b)$ (and $a_i \in Dom(s_{A_i}), b \in Dom(s_B)$). Again by the induction hypothesis $s_C(N(h)(a_1, \dots, a_n, b)) = [h(t_1/x_1, \dots, t_n/x_n, r/y)]$, so indeed

$$s_C(N(\lambda y.h)(a_1, \dots, a_n)(b)) = [h(t_1/x_1, \dots, t_n/x_n, r/y)] = [(\lambda y.h(t_1/x_1, \dots, t_n/x_n)`r)]$$

(recall $t_i \in s_{A_i}(a_i)$ and $r \in s_B(b)$). $\square$

Now it is clear that N reflects equality: let $N(x^A \rhd f^B) = N(x^A \rhd g^B)$, then for every $a \in Dom(s_A)$, $s_B(N(f)a) = s_B(N(g)a)$ and so by (2) we have $f(\xi/x) = g(\xi/x)$ (take $a \in s_A^{-1}(\xi)$, $\xi \notin f, g$). By Lemma 3.10 we have $f =_x g$. $\square$

Remark 3.12. The typed λ-calculus for which Friedman proved the theorem did not have product types nor the terminal type nor additional ("functional") constants. Also the equations did not have contexts. In his case $\Gamma(A) = \{[t] : t : A\}$ (t not necessarily closed). Obviously, for $A \equiv 1$ it would not work in our case. So we had to take only closed terms and therefore we had to introduce "many" constants. Let us also add that the above theorem was proved independently (and later) by Kennison [9].

And finally, we can give:

Proof of Corollary 3.8. Let L be the free λ-calculus such that $\mathscr{C}_L = \mathscr{D}$. Then by the previous theorem there is a model N of L in *Set* which reflects equality. Then, by Corollary 2.9 there exists a cc-functor $\mathscr{D} \xrightarrow{F} Set$ such that $N = F \circ M$ (M is the canonical model $M : L \to \mathscr{C}_L$). F is faithful by the construction of $\mathscr{C}_L$ and faithfulness of N. □

Remark 3.13. It is easy to show that we cannot get fullness (and even some weaker properties) in the above theorem. Also, not every ccc can be faithfully mapped in *Set*, as a matter of fact not even in a Boolean topos as observed in [15] ($U^U \cong U$ implies $U \cong 1$ in a Boolean topos). However, every small ccc can be mapped in a De Morgan topos by a full and faithful cc-functor – we will elaborate on that elsewhere.

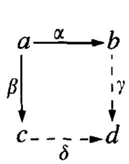
4. Mints' reductions

To finish off the proof of Theorem 1.1 we need to prove Proposition 3.5. For that we need a confluent system of reductions for (a free) typed λ-calculus as given above, which does not introduce new variables. So not only products but also the terminal object are included and not all types are inhabited. There are only two references (that we are aware of) where such a system is given: [12, 3]. We prefer the system given by Mints and we are going to use that one. The main reason for our choice is that these reductions are closer to Prawitz' reductions for natural deduction and they are simpler than the ones in [3].

The reductions in [3] are $\mathscr{R}_1$ (see below) but in the opposite direction (and no restrictions), $\mathscr{R}_2$ and in addition infinitely many reductions which are introduced to take care of "Obtulowitz' pairs" e.g. $x^{1\times A} \xleftarrow{SP^{-1}} \langle \pi_1(x^{1\times A}), \pi_2(x^{1\times A})\rangle \xrightarrow{T} \langle *, \pi_2(x^{1\times A})\rangle$. Because of these pairs they have to add new reductions and by a kind of Knuth–Bendix procedure they add infinitely many reductions but neatly classified in four groups. The above pair they "connect" by an SP_{top} reduction: $\langle *, \pi_2(x^{1\times A})\rangle \to x^{1\times A}$.

Let us briefly introduce some terminology related to the notion of reduction. A binary relation $\mathscr{R}$ on a set of terms is called a reduction; traditionally $(t,s) \in \mathscr{R}$ is denoted $t \xrightarrow{\mathscr{R}} s$. A term t is $\mathscr{R}$-normal if there is no term s such that $t \xrightarrow{\mathscr{R}} s$. A term t is weakly normalizing if there is a finite sequence $t \equiv t_0 \xrightarrow{\mathscr{R}} \cdots \xrightarrow{\mathscr{R}} t_n$ such that t_n is $\mathscr{R}$-normal. A term t is strongly normalizing if every sequence $t \equiv t_0 \xrightarrow{\mathscr{R}} \cdots \xrightarrow{\mathscr{R}} t_n \xrightarrow{\mathscr{R}} \cdots$ is finite. We say that $\mathscr{R}$ is weakly (strongly) normalizing if every term t is weakly (strongly) normalizing. The transitive and reflexive closure of $\mathscr{R}$ we will denote by $\mathscr{R}^*$.

A diagram such as



is actually a statement which says: if $a \xrightarrow{\alpha} b$ and $a \xrightarrow{\beta} c$ then there exists d such that $b \xrightarrow{\gamma} d$ and $c \xrightarrow{\delta} d$, where α, β, γ and δ are possibly different reductions.

We say that $\mathscr{R}$ is locally confluent/locally Church–Rosser if

$$\begin{array}{ccc} a & \xrightarrow{\mathscr{R}} & b \\ {\scriptstyle \mathscr{R}}\downarrow & & \downarrow{\scriptstyle \mathscr{R}^*} \\ c & \underset{\mathscr{R}^*}{\dashrightarrow} & d \end{array}$$

also we say that $\mathscr{R}$ is confluent/Church–Rosser if

$$\begin{array}{ccc} a & \xrightarrow{\mathscr{R}^*} & b \\ {\scriptstyle \mathscr{R}^*}\downarrow & & \downarrow{\scriptstyle \mathscr{R}^*} \\ c & \underset{\mathscr{R}^*}{\dashrightarrow} & d \end{array}$$

Notation: We will write $t[x]$ when we refer to a particular occurrence of the variable x (free or bound – but of course not in $\lambda x.$ position); $t[s/x]$ denotes a term equal to t except that instead of x is written s (so it means that we do not care about clashes of variables here). Example: let $t[x] \equiv \lambda x.\langle x,x\rangle$ where we are pointing to the left occurrence of x in $\langle x,x\rangle$. Then $t[f(x)/x] \equiv \lambda x.\langle f(x),x\rangle$. We can see that also $t[f(x)/x] \equiv t[f(x)/y]$ where $t[y] \equiv \lambda x.\langle y,x\rangle$. The same thing is true in general, namely writing $t[s/x]$ we can always assume that the variable x occurred only once in $t[x]$ (again not counting the occurrences in $\lambda x.$). We will try to use just $t[s]$ instead of $t[s/x]$ as often as convenient. (We just defined the notion of "context", but since we used this word earlier for a different thing, here we will not give a particular name to it.)

Mints' system of reductions $\mathscr{R}$ is the following:

$$\mathscr{R}_1 \begin{cases} C[t^{B^A}] \xrightarrow{\eta} C[\lambda x.(t`x)] \;\; x \notin FV(t) & \text{provided neither } t \equiv \lambda y.s \\ & \text{nor } C[t] \equiv D[(t`s)] \\ C[t^{A\times B}] \xrightarrow{SP} C[\langle \pi_1(t), \pi_2(t)\rangle] & \text{provided neither } t \equiv \langle s_1, s_2\rangle \\ & \text{nor } C[t] \equiv D[\pi_i(t)], \end{cases}$$

$$\mathscr{R}_2 \begin{cases} C[t^1] \xrightarrow{T} C[*] & \text{if } t^1 \not\equiv * \\ C[(\lambda x.t`s)] \xrightarrow{\beta} C[t(s/x)] & \\ C[\pi_i(\langle t_1, t_2\rangle)] \xrightarrow{Pr_i} C[t_i] & i = 1,2. \end{cases}$$

To be more precise, we should have said that $C[z]$ has exactly one occurrence of the variable z and then the above reductions would have looked, e.g. as follows:

$$C[t^{B^A}/z^{B^A}] \xrightarrow{\eta} C[(\lambda x.(t`x))/z] \;\; x \notin FV(t)$$

provided neither $t \equiv \lambda y.s$ nor $C[z] \equiv D[(z`s)/w]$ for any two terms $D[w]$, s.

The terms in the brackets on the left we call *redexes*. The positions above which are excluded we call *restricted positions*. If t is a redex of a reduction γ (γ-redex) and if $t \xrightarrow{\gamma} s$ is a γ-reduction on t then $\gamma(t)$ will denote the term s. We also write $t \xrightarrow{\mathscr{R}} s$ if there is a reduction $\gamma \in \mathscr{R}$ such that $t \xrightarrow{\gamma} s$ or $t \equiv s$. (So again we are abusing notation a bit: $\mathscr{R}$ denotes (at the same time) its reflexive closure.) The smallest equivalence relation containing $\mathscr{R}$ we will denote $\cong^{\mathscr{R}}$, so $t \cong^{\mathscr{R}} s$ if and only if there exists a sequence of terms $t \equiv t_0, t_1, \ldots, t_n \equiv s$ such that for every $0 \le i < n$ $t_i \xrightarrow{\mathscr{R}} t_{i+1}$ or $t_{i+1} \xrightarrow{\mathscr{R}} t_i$. Often, we want to be precise and to write $t \cong^{\mathscr{R}}_X s$ if there is a sequence as above so that $X = FV(t_0, \ldots, t_n)$. The system of reductions in which the restrictions (on the position as well as on the shape of terms) are omitted, we call *unrestricted reductions* and we denote it by $\mathscr{R}^u$.

The restrictions in the above system are the obvious ones to prevent nontermination – it is interesting that this is "the right" choice, i.e. with these restrictions the system is strongly normalizing and also sufficient for the λ-calculus in the following sense:

Proposition 4.1. *For every set of variables X, $\vdash t =_X s$ iff $t \cong^{\mathscr{R}}_X s$.*

Proof. To prove that we need a very simple fact which is going to be used once more:

Lemma 4.2. *For every two terms t and s, $t \cong^{\mathscr{R}}_X s$ iff $t \cong^{\mathscr{R}^u}_X s$.*

Proof. In both directions, the proof is by induction on the length of the chain which witnesses the appropriate relation. The only thing which has to be checked is the base of induction in the proof from right to left, and the only four cases worth checking are the applications of unrestricted reductions when the subterm on which we act is in the restricted position or of restricted shape (or both). Let us check just two cases: suppose that a term $\langle t_1, t_2\rangle$ appears as a subterm of a term r, we can write this as $r[\langle t_1, t_2\rangle]$, and suppose that the unrestricted SP was applied on t, i.e.,

$$r[\langle t_1, t_2\rangle] \xrightarrow{SP^u} r[\langle \pi_1(\langle t_1, t_2\rangle), \pi_2(\langle t_1, t_2\rangle)\rangle].$$

In the restricted case these two terms can be connected as follows:

$$r[\langle t_1, t_2\rangle] \xleftarrow{Pr_1^*} r[\langle \pi_1(\langle t_1, t_2\rangle), \pi_2(\langle t_1, t_2\rangle)\rangle].$$

(Notice that we do not have to separate the case when the term $\langle t_1, t_2\rangle$ appears in the restricted position.) For the second case we choose the following: suppose

$$r[(t^{A^B} {}^{\backprime} s^B)] \xrightarrow{\eta^u} r[(\lambda x^B.(t^{A^B} {}^{\backprime} x^B) {}^{\backprime} s^B)].$$

These two terms van be connected in the restricted case as follows:

$$r[(t^{A^B} {}^{\backprime} s^B)] \xleftarrow{\beta} r[(\lambda x^B.(t^{A^B} {}^{\backprime} x^B) {}^{\backprime} s^B)].$$

(Again we need not have to separate the case when $t \equiv \lambda y.u$.) The other cases are equally easy. □

To prove the above proposition we just have to prove that $\vdash t =_X s$ iff $t \cong_X^{\mathscr{R}^u} s$ but this is standard; for a simpler situation see, for example, Proposition 3.2.1 in [2]. □

The key observation is that $\mathscr{R}_1^*$ and $\mathscr{R}_2^*$ commute. More precisely we have the following.

Proposition 4.3.

$$\begin{array}{ccc} a & \xrightarrow{\mathscr{R}_2^*} & b \\ {\scriptstyle \mathscr{R}_1^*}\downarrow & & \downarrow{\scriptstyle \mathscr{R}_1^*} \\ c & \dashrightarrow_{\mathscr{R}_2^*} & d \end{array}$$

From this proposition, using some more or less obvious properties of the above system of reductions, we can establish several interesting corollaries e.g. confluence, strong normalization (giving also a particular, nice normalization strategy) and also confluence of the system same as the above one but without restrictions.

The proof will be divided in several lemmas, but before that we need to introduce some notation and some definitions.

The following notion makes sense in general: if $t[s/x] \xrightarrow{\rho} t'$ then the ρ-residual of s is whatever remains in t' of s. We are going to use that notion only when ρ is one of the $\mathscr{R}_1$-reductions and s is not the redex on which we apply ρ. Let us just add that the notion of residual as well as the concept of minimal development are standard in the literature, see for example [7].

Definition 4.4 (*$\mathscr{R}_1$-residual*). Let γ be one of the $\mathscr{R}_1$-reductions and let $t[s/x] \xrightarrow{\gamma} t'$ be on a γ-redex R such that $R \not\equiv s$. The γ-residual of s is defined as follows: first, if R is disjoint from s, i.e. $t[s/x] \equiv T[s/x, R/y]$ for some term $T[x,y]$ then $t' \equiv T[s/x, \gamma(R)/y]$ and in this case s is the residual of s. Second, if R is a proper subterm of s, i.e. $s \equiv S[R/y]$ for some term $S[y] \not\equiv y$, then $t' \equiv t[S[\gamma(R)/y]/x]$ and the residual of s is $S[\gamma(R)/y]$. Third, if s is a proper subterm of R, i.e. $R \equiv r[s/x]$ for some term $r[x] \not\equiv x$ and $t[x] \equiv T[r[x]/y]$ for some term $T[y]$. Then we have two cases depending on γ: if $\gamma = \eta$ then $t' \equiv T[(\lambda z.r[s/x]`z)/y]$ and this s is the residual of s; if $\gamma = SP$ then $t' \equiv T[\langle \pi(r[s/x]), \pi'(r[s/x])\rangle/y]$ and these two occurrences of s are the residuals of s. The residual of a residual of some term s we will call again the residual of s.

Notice that every residual of a redex remains a redex. Also that residuals of disjoint terms remain disjoint. The only case when a term t can have more then one residual is when we perform an SP reduction on a term that contains t.

Definition 4.5 (*$\mathscr{R}_1$-minimal development*). Let $R_1, \ldots, R_n$ be a set of γ-redexes in a term t ($\gamma \in \mathscr{R}_1$ or $\gamma = \mathscr{R}_1$). Then $t \xrightarrow{\gamma^*} s$ is a minimal development (denoted γ^{m}) on

$R_1, \ldots, R_n$ if in each step we reduce a redex which is a residual of one of $R_1, \ldots, R_n$ (one of them at the first step) and minimal among them (with respect to the subterm relation). When we write a set of redexes for a minimal development as above we assume that if $i < j$ then $R_j \not\preceq R_i$ (R_j is not a subterm of R_i).

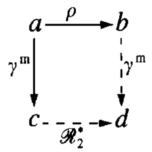
Two minimal developments performed one after another do not have to make a minimal development, but if the redexes of the second one do not contain any of the redexes of the first then they do make one minimal development on the union of the two sets of redexes. Although we are not going to use it we can notice that the above remarks on residuals tell that every minimal development on $R_1, \ldots, R_n$ ends in n steps (since we never apply SP-reduction on a redex containing a redex from the prescribed list).

Lemma 4.6. *A set of redexes determines the result of minimal development in the following sense: if* $t \xrightarrow{\gamma'^{m}} s'$ *and* $t \xrightarrow{\gamma''^{m}} s''$ *on the same set of redexes then* $s' \equiv s''$.

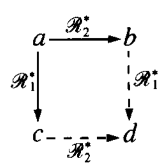
Proof. Induction on the number of redexes. Zero redexes do not make a problem. Neither does one. Since the order of reductions for the disjoint redexes is irrelevant we can assume that all the maximal redexes are reduced at the end. Suppose now that we omit all the maximal redexes. By the induction hypothesis without them both minimal developments give the same result (new minimal developments are "initial segments" of the old ones). Moreover (again by the induction hypothesis), the residuals of the maximal redexes are the same in both cases and, as observed earlier, they are disjoint (and they do not multiply). Reducing them in whatever order gives the same result. □

The main use of minimal development is in the following lemma.

Lemma 4.7. *If every reduction* $\rho \in \mathscr{R}_2$, *and every* $\gamma \in \mathscr{R}_1$, *satisfy the following condition*:



then $\mathscr{R}_2^*$ *and* $\mathscr{R}_1^*$ *commute, i.e.*,



Proof. Induction on the length of $\mathscr{R}_1^*$. When the length is 1 notice that every one step reduction γ is a minimal development and use the assumption plus induction on the length of $\mathscr{R}_2^*$. The above argument is also used when passing from "n-1" to "n". □

Notation: Let $\gamma \in \mathscr{R}_1$. Then $\gamma^{\mathrm{op}} = \beta^*$ if $\gamma = \eta$ and $\gamma^{\mathrm{op}} = Pr^*$ if $\gamma = SP$ (notice that $\gamma^{\mathrm{op}*} = \gamma^{\mathrm{op}}$). Also

$$\gamma^u(t) = \begin{cases} t & \text{if } u = 0, \\ \gamma(t) & \text{if } u = 1, \end{cases}$$

for example,

$$\eta^u(t) = \begin{cases} t & \text{if } u = 0, \\ \lambda z.(t`z) & \text{if } u = 1. \end{cases}$$

(of course $z \notin FV(t)$).

From now on we will write just $t[a]$ instead of $t[a/x]$ whenever possible.

Lemma 4.8. *Let $a[b] \xrightarrow{\gamma^m} c$ be a minimal development on redexes $R_1, \ldots, R_i, \ldots, R_{i+j}$, $R_{i+j+1}[b], \ldots, R_{i+j+k}[b]$, where the redexes $R_1, \ldots, R_i$ are proper subterms of b and the term b appears exactly where shown. Then $c \equiv a'[\gamma^u(b'), \ldots, \gamma^u(b')]$ so that $a[x] \xrightarrow{\gamma^m} a'[x, \ldots, x]$ on the redexes $R_{i+1}, \ldots, R_{i+j}$, $R_{i+j+1+u}[x], \ldots, R_{i+j+k}[x]$ and $b \xrightarrow{\gamma^m} b'$ on $R_1, \ldots, R_i$; here $u = 0$ if $R_{i+j+1}[b] \not\equiv b$ and $u = 1$ if $R_{i+j+1}[b] \equiv b$.*

Our assumption on the order of writing of redexes for a minimal development gives $R_{i+j+1}[b] \prec \cdots \prec R_{i+j+k}[b]$ (the relation $\prec$ stands for "proper subterm").

(Sometimes we will use the following form of the lemma: let $a[b] \xrightarrow{\gamma^m} c$ be a minimal development on redexes $R_1, \ldots, R_i, \ldots, R_{i+j}, R_{i+j+1}[b], \ldots, R_{i+j+k}[b]$, where the redexes $R_1, \ldots, R_i$ are subterms of b and the term b appears exactly where shown as a proper subterm and maybe $R_i \equiv b$. Then: $c \equiv a'[\gamma^u(b'), \ldots, \gamma^u(b')]$ so that $a[x] \xrightarrow{\gamma^m} a'[x, \ldots, x]$ on the redexes $R_{i+1}, \ldots, R_{i+j}, R_{i+j+1}[x], \ldots, R_{i+j+k}[x]$ and $b \xrightarrow{\gamma^m} b'$ on $R_1, \ldots, R_{i-u}$; here $u = 0$ if $R_i \not\equiv b$ and $u = 1$ if $R_i \equiv b$).

Proof. If all the redexes are disjoint from b then the statement is almost a tautology. There are two other cases – the first one is when there is a maximal redex properly contained in b. By Lemma 4.6 we can assume that we first do the reductions in b, i.e. $a[b] \xrightarrow{\gamma^m} a[b']$ and then the reductions on the redexes disjoint from b'. Since the reduction $a[b'] \xrightarrow{\gamma^m} d$ on a redex R_d disjoint from b' satisfies the statement, i.e. $d \equiv a'[b']$ so that $a[x] \xrightarrow{\gamma^m} a'[x]$ on R_d (it can be proved by induction on the complexity of $a[x]$), we have proved the lemma in this case. The proof now continues by induction on the index k; the previous part is just the base of induction $k = 0$, i.e. there is no redex containing b. So let R denote the maximal redex containing b (it can be b itself), i.e. in the notation above $R = R_{i+j+k+1}[b]$. Our minimal development is $a[b] \xrightarrow{\gamma^m} c$

on the set of redexes as in the statement of the lemma plus R. By Lemma 4.6 we can assume that R is the last one reduced. Consider now the minimal development without the last step. Since $a[b] \equiv A[R/y]$ (for an appropriate term A) we can apply the induction hypothesis and conclude that

$$A[R/y] \xrightarrow{\gamma^{m}} A'[R'/y] \tag{3}$$

on the redexes without R so that

$$A[y] \xrightarrow{\gamma^{m}} A'[y] \tag{4}$$

on the redexes outside of R – these are some of $R_{i+1},\dots,R_{i+j}$ and

$$R \xrightarrow{\gamma^{m}} R' \tag{5}$$

on the rest of the redexes – they are $R_1,\dots,R_i$, the redexes from $R_{i+1},\dots,R_{i+j}$ which are in R and $R_{i+j+1}[b],\dots,R_{i+j+k}[b]$. Applying the induction hypothesis to (5) (actually just the base of induction) we have $R' \equiv R'[\gamma^{u}(b'),\dots,\gamma^{u}(b')]$ so that

$$R[x] \xrightarrow{\gamma^{m}} R'[x,\dots,x] \tag{6}$$

on the redexes from $R_{i+1},\dots,R_{i+j}$ which are in R and $R_{i+j+1+u}[x],\dots,R_{i+j+k}[x]$, and also

$$b \xrightarrow{\gamma^{m}} b' \tag{7}$$

on $R_1,\dots,R_i$. Taking (4) and (6) we get

$$A[R[x]/y] \xrightarrow{\gamma^{m}} A'[R'[x,\dots,x]/y] \tag{8}$$

on $R_{i+1},\dots,R_{i+j},R_{i+j+1+u}[x],\dots,R_{i+j+k}[x]$. Now, if we reduce $R'[x,\dots,x]$ (which is indeed a redex) we have

$$A[R[x]/y] \xrightarrow{\gamma^{m}} A'[\gamma(R'[x,\dots,x])/y] \tag{9}$$

on $R_{i+1},\dots,R_{i+j},R_{i+j+1+1}[x],\dots,R_{i+j+k+1}[x]$. Since $A[R[x]/y] \equiv a[x]$ we can use $a'[x,\dots,x]$ to denote $A'[\gamma(R'[x,\dots,x])/y]$. This together with (7) finishes the proof. □

Lemma 4.9. *With the above notation the following hold*:

1. $\gamma^{u}(\gamma^{v}(t)) \xrightarrow{\gamma^{op}} \gamma^{u\vee v}(t)$,
2. $a[\gamma(b)] \xrightarrow{\gamma^{op}} a[b]$, *providing b is of the "forbidden shape" (i.e. $b \equiv \langle b_1,b_2\rangle$ or $b \equiv \lambda x.b_1$) or in a restricted position or both.*

Lemma 4.10. *The conditions of Lemma* 4.7 *are satisfied when* ρ *is any* $\mathscr{R}_2$ *reduction.*

Proof. *Case* 1. $\rho \equiv Pr$ (and $\gamma \equiv \eta$ or $\gamma \equiv SP$). So suppose we have

$$\begin{array}{ccc} a[\pi(\langle b_1, b_2\rangle)] & \xrightarrow{\quad Pr \quad} & a[b_1] \\ {\scriptstyle \gamma^{\mathrm{m}}}\big\downarrow & & \\ a'[\gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))/x, \ldots, \gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))/x], & & \end{array}$$

where the minimal development γ^{m} is done on the redexes $R_1, \ldots, R_i, \ldots, R_{i+j}$, $R_{i+j+1}[\pi(\langle b_1, b_2\rangle)], \ldots,$ $R_{i+j+k}[\pi(\langle b_1, b_2\rangle)]$, where the redexes $R_1, \ldots, R_i$ are subterms of b_1, redexes which are in b_2 are not even shown and the term $\pi(\langle b_1, b_2\rangle)$ is exactly where shown. By Lemma 4.8 the result of the minimal development has to be as above (since we cannot apply γ on $\langle b_1, b_2\rangle$ – either the types do not match or the shape is forbidden) where $a[x] \xrightarrow{\gamma^{\mathrm{m}}} a'[x, \ldots, x]$ on $R_{i+1}, \ldots, R_{i+j}, R_{i+j+1+u}[x], \ldots,$ $R_{i+j+k}[x]$ and $b_1 \xrightarrow{\gamma^{\mathrm{m}}} b'_1$ on $R_1, \ldots, R_{i-v}$.

For the sake of simplicity, we will write just $a'[x]$ instead of $a'[x, \ldots, x]$ and similarly $a'[\gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))]$ for $a'[\gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))/x, \ldots, \gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))/x]$ and so on. But we do not write Pr instead of Pr^* (e.g. the following diagram). Applying Pr^* we have

$$\begin{array}{ccc} a[\pi(\langle b_1, b_2\rangle)] & \xrightarrow{\quad Pr \quad} & a[b_1] \\ {\scriptstyle \gamma^{\mathrm{m}}}\big\downarrow & & \\ a'[\gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))] & \xrightarrow[Pr^*]{} & a'[\gamma^u(\gamma^v(b'_1))]. \end{array}$$

By Lemma 4.9 we can add one more arrow:

$$\begin{array}{cccc} a[\pi(\langle b_1, b_2\rangle)] & \xrightarrow{\quad Pr \quad} & a[b_1] & \\ {\scriptstyle \gamma^{\mathrm{m}}}\big\downarrow & & & a'[\gamma^{u\vee v}(b'_1)] \\ & & \nearrow {\scriptstyle \gamma^{\mathrm{op}}} & \\ a'[\gamma^u(\pi(\langle\gamma^v(b'_1), b'_2\rangle))] & \xrightarrow{Pr^*} & a'[\gamma^u(\gamma^v(b'_1))]. & \end{array}$$

Now, if $u \vee v = 0$ it is obvious that $a[b_1] \xrightarrow{\gamma^{\mathrm{m}}} a'[b'_1]$ on $R_1, \ldots, R_i, \ldots, R_{i+j}$, $R_{i+j+1}[b_1], \ldots,$ $R_{i+j+k}[b_1]$ finishes the proof. Therefore, suppose $u \vee v = 1$. If b_1 is of “forbidden shape” or in a restricted position (it could not be both because we would have $u \vee v = 0$) then by Lemma 4.9 $a'[\gamma(b'_1)] \xrightarrow{\gamma^{\mathrm{op}}} a'[b'_1]$ and again the added γ^{m} is performed on all the redexes except the one which caused $u \vee v = 1$, i.e., $R_1, \ldots, R_{i-v}$,

$R_{i+1},\dots,R_{i+j},R_{i+j+1+u}[b_1],\dots,\ R_{i+j+k}[b_1]$:

$$\begin{array}{ccccc}
a[\pi(\langle b_1,b_2\rangle)] & \xrightarrow{Pr} & a[b_1] & & \\
 & & & \searrow^{\gamma^{\mathrm{m}}} & \\
\downarrow{\scriptstyle \gamma^{\mathrm{m}}} & & & & a'[b_1'] \\
 & & & \nearrow_{\gamma^{\mathrm{op}}} & \\
a'[\gamma^u(\pi(\langle\gamma^v(b_1'),b_2'\rangle))] & \xrightarrow[Pr^*]{} & a'[\gamma^u(\gamma^v(b_1'))]. & &
\end{array}$$

And finally, if b_1 is not of "forbidden shape" nor in a restricted position (and still $u \vee v = 1$) then $a[b_1'] \xrightarrow{\gamma} a[\gamma(b_1')]$ can be performed so we have

$$\begin{array}{ccccc}
a[\pi(\langle b_1,b_2\rangle)] & \xrightarrow{P_r} & a[b_1] & & \\
 & & & \searrow^{\gamma^{\mathrm{m}}} & \\
\downarrow{\scriptstyle \gamma^{\mathrm{m}}} & & & & a'[\gamma(b_1')] \\
 & & & \nearrow_{\gamma^{\mathrm{op}}} & \\
a'[\gamma^u(\pi(\langle\gamma^v(b_1'),b_2'\rangle))] & \xrightarrow[Pr^*]{} & a'[\gamma^u(\gamma^v(b_1'))], & &
\end{array}$$

where the new γ^m is performed on $R_1,\dots,R_{i-(u\wedge v)}$, $R_{i+1},\dots,R_{i+j},R_{i+j+1}[b_1],\dots,$ $R_{i+j+k}[b_1]$. This finishes the proof of the first case.

Case 2: $\rho \equiv \beta$ (and $\gamma \equiv \eta$ or $\gamma \equiv SP$). So assume we have

$$\begin{array}{ccc}
a[(\lambda y.b`c)] & \xrightarrow{\beta} & a[b(c/y)] \\
\downarrow{\scriptstyle \gamma^{\mathrm{m}}} & & \\
a'[\gamma^u(\lambda y.\gamma^v(b')`\gamma^w(c'))], & &
\end{array}$$

where γ^{m} is done on the redexes $R_1,\dots,R_l,\dots,R_i,\dots,R_{i+j}$, $R_{i+j+1}[(\lambda y.b`c)],\dots,$ $R_{i+j+k}[(\lambda y.b`c)]$, where the redexes $R_1,\dots,R_l$ are subterms of b, redexes $R_{l+1},\dots,R_i$ are in c and the term $(\lambda y.b`c)$ is exactly where shown. (See the simplification in the notation mentioned in the first case.) Again by Lemma 4.8 the result of the minimal development has to be as above (since we cannot apply γ on $\lambda y.b$) where $a[x] \xrightarrow{\gamma^{\mathrm{m}}} a'[x]$ on the redexes $R_{i+1},\dots,R_{i+j},R_{i+j+1+u}[x],\dots,R_{i+j+k}[x]$, $b \xrightarrow{\gamma^{\mathrm{m}}} b'$ on $R_1,\dots,R_{l-v}$ and $c \xrightarrow{\gamma^{\mathrm{m}}} c'$ on $R_{l+1},\dots,R_{i-w}$. Without loss of generality, we assume that b has at most three occurrences of y in it – so b looks like $b(y,y,y)$ where only the leftmost y is among the redexes (e.g. $R_1 \equiv y$) and only the rightmost y is in the restricted position for this γ. Applying β^* we have

$$a'[\gamma^u(\lambda y.\gamma^v(b'(\gamma(y),y,y))`\gamma^w(c'))] \xrightarrow{\beta^*} a'[\gamma^u(\gamma^v(b'(\gamma(\gamma^w(c')),\gamma^w(c'),\gamma^w(c'))))].$$

The rightmost occurrence of $\gamma^w(c')$ is in a restricted position (by the assumption) so applying γ^{op} we get c' at this position (we used Lemma 4.9). Also $\gamma(\gamma^w(c')) \xrightarrow{\gamma^{\mathrm{op}}} \gamma(c')$ and $\gamma^u(\gamma^v(b')) \xrightarrow{\gamma^{\mathrm{op}}} \gamma^{u\vee v}(b')$. So, we can add one more "arrow" to the diagram above and now we have

$$\begin{array}{ccc} a[(\lambda y.b(y,y,y)`c)] & \xrightarrow{\quad\beta\quad} & a[b(c/y,c/y,c/y))] \\ \Big\downarrow{\scriptstyle \gamma^{\mathrm{m}}} & & \\ a'[\gamma^u(\lambda y.\gamma^v(b'),\gamma(y),y,y))`\gamma^w(c'))] & \xrightarrow[\mathcal{R}_2^*]{} & a'[\gamma^{u\vee v}(b'(\gamma(c'),\gamma^w(c'),c'))]\,. \end{array}$$

It is easy to see the redexes for the following minimal development: $b(c,c,c) \xrightarrow{\gamma^{\mathrm{m}}} b(c',c',c')$. Now we have two cases: c' of forbidden shape or not (let us just mention that the first case is possible exactly when c is of forbidden shape). In the first case $\gamma(c') \xrightarrow{\gamma^{\mathrm{op}}} c'$ and $\gamma^w(c') \xrightarrow{\gamma^{\mathrm{op}}} c'$ (in fact, $w = 0$ in this case). In the second case $c' \xrightarrow{\gamma} \gamma(c')$ and $c' \xrightarrow{\gamma} \gamma^w(c')$ (recall that the first two positions of c' are not restricted in b). In any case the two "branches" of the above diagram are little closer, and we have

$$\begin{array}{ccccc} a[(\lambda y.b(y,y,y)`c)] & \xrightarrow{\quad\beta\quad} & a[b(c,c,c)] & \xrightarrow{\quad\gamma^{\mathrm{m}}\quad} & a[b_0] \\ \Big\downarrow{\scriptstyle \gamma^{\mathrm{m}}} & & & & \\ a'[\gamma^u(\lambda y.\gamma^v(b'(\gamma(y),y,y))`\gamma^w(c')) & \xrightarrow[\mathcal{R}_2^*]{} & a'[\gamma^{u\vee v}(b'(\gamma(c'),\gamma^w(c'),c'))] & \xrightarrow[\gamma^{\mathrm{op}}]{} & a'[\gamma^{u\vee v}(b_0)], \end{array}$$

where

$$b_0 = \begin{cases} b'(c',c',c') & \text{if } c \text{ has forbidden shape,} \\ b'(\gamma(c'),\gamma^w(c'),c') & \text{if } c \text{ has allowed shape} \end{cases}$$

(passage to b' also does not create a problem now). Now if the $u \vee v = 0$ solution is obvious, so assume $u \vee v = 1$. So the situation is exactly as in the first case – in any case to the above diagram we can add

$$\begin{array}{ccc} a[b_0] & & \\ & \searrow{\scriptstyle \gamma^{\mathrm{m}}} & \\ & & a'[\gamma^r(b_0)] \\ & \nearrow{\scriptstyle \gamma^{\mathrm{op}}} & \\ a'[\gamma^{u\vee v}(b_0)] & & \end{array}$$

where $r = 0$ if b_0 is of "forbidden shape" or in a restricted position and $r = 1$ otherwise. (Although γ^{m} is not in general a transitive relation here we took care of that by reducing "from inside" so that these consecutive γ^{m}'s give a minimal development.)

Case 3: $\rho \equiv T$ (and $\gamma \equiv \eta$ or $\gamma \equiv SP$). So suppose we have

$$\begin{array}{ccc} a[t^1] & \xrightarrow{T} & a[*] \\ {\scriptstyle\gamma^{\mathrm{m}}}\downarrow & & \\ a'[t'/x] & & \end{array}$$

(see again the simplified notation from case 1). The minimal development was done on the redexes $R_1,\dots,R_i,\dots,R_{i+j}$, $R_{i+j+1}[t]$, …, $R_{i+j+k}[t]$, where the redexes $R_1,\dots,R_i$ are subterms of t and the term t is exactly where shown. By Lemma 4.8 the result of the minimal development has to be as above (since we cannot apply γ on t – the types do not match) where $a[x] \xrightarrow{\gamma^{\mathrm{m}}} a'[x]$ on $R_{i+1},\dots,R_{i+j},R_{i+j+1}[x],\dots,\ R_{i+j+k}[x]$ and $t \xrightarrow{\gamma^{\mathrm{m}}} t'$ on $R_1,\dots,R_i$. Since t' still has type 1 it is obvious that the following holds:

$$\begin{array}{ccc} a[t^1] & \xrightarrow{T} & a[*] \\ {\scriptstyle\gamma^{\mathrm{m}}}\downarrow & & \downarrow{\scriptstyle\gamma^{\mathrm{m}}} \\ a'[t'/x] & \xrightarrow[T^*]{} & a'[*] \end{array}$$

where the new γ^{m} is done on $R_{i+1},\dots,R_{i+j},R_{i+j+1}[*],\dots,\ R_{i+j+k}[*]$. □

Proof of Proposition 4.3. Just apply the previous lemma and Lemma 4.7. □

Lemma 4.11. $\mathscr{R}_1$ *is canonical* (*i.e. confluent and strongly normalizing*).

Proof. For details we refer to [12] – this part is correct. Let us just say that by Newman's lemma it is enough to show local confluence and strong normalization. Local confluence is easy here. Strong normalization is proved by assigning to each term t a natural number $\#t$ so that

$$t \xrightarrow{\mathscr{R}_1} t' \text{ implies } \#t > \#t'.$$

For that we first define the rank of a type as the number of the type forming operations in it, i.e. $\#(A\times B) = \#(A^B) = \#(A)+\#(B)+1$ and the rank of atomic types and terminal type is zero. Second, we define the degree of a redex

$$d(R) = 2^{\sum \#(A_i)},$$

where $A_1,\dots,A_n$ are types of redexes in t which contain R. Finally,

$$\#t = \sum d(R_j),$$

where R_j are all (occurrences of) redexes in t. □

Lemma 4.12. *The reduction* $\mathscr{R}_2$ *is canonical.*

Proof. A well-known result is that in the typed case Pr_i and β are canonical (see, for example, [5]). Adding T-contraction will not change much. Local confluence is simple to check, and we get strong normalization by showing that all T-reductions can be postponed after β, Pr-reductions. Let us just show that this is so in case of β. Suppose that before $a[(\lambda x.b`c)] \xrightarrow{\beta} a[(b(c))]$ there was a $T$ reduction. There are only two interesting cases: $C[t^1] \xrightarrow{T} C[*] \equiv c$ and $B[t^1/y] \xrightarrow{T} B[*/y] \equiv b$. In the first case the old reduction looked like $a[(\lambda x.b`C[t^1])] \xrightarrow{T} a[(\lambda x.b`c)] \xrightarrow{\beta} a[(b(c))/y]$, we transform it to $a[(\lambda x.b`C[t^1])] \xrightarrow{\beta} a[(b(C[t^1]))] \xrightarrow{T^*} a[b(c)]$. (By α-congruence we insure that there are no clashes of variables.) In the second case the old reduction looked as $a[(\lambda x.B[t^1/y]`c)] \xrightarrow{T} a[\lambda x.B[*/y]`c] \xrightarrow{\beta} a[b(c)]$. We transform it to $a[(\lambda x.B[t^1/y]`c)] \xrightarrow{\beta} a[(B[t^1/y])(c)] \equiv a[B(c)[(t^1(c))/y]] \xrightarrow{T^*} a[B(c)[*/y]] \equiv a[b(c)]$. (Here we assumed that y was not a free variable in c – it was anyway denoting just a position.) Even simpler is the proof with Pr instead of β. Notice that in those transformations the number of Pr, β reductions remains the same and they "go up". So there is no infinite $\mathscr{R}_2$-reduction, if there were it would have to have infinitely many Pr, β-reductions (no terms have infinitely many consecutive T-reductions); transforming such a reduction we would get arbitrarily long reduction of consecutive Pr, β steps which would contradict strong normalizability of this fragment. □

This (and even less) is enough to show that Mints' reductions are confluent. That is also all what we need to finish the proof of the main theorem. For the record:

Corollary 4.13. *Mints' reductions are confluent.*

Proof. Suppose we have

$$\begin{array}{ccc} a & \xrightarrow{\mathscr{R}^*} & b \\ {\scriptstyle \mathscr{R}^*}\downarrow & & \\ c & & \end{array}$$

(Recall $\mathscr{R} = \mathscr{R}_1 \cup \mathscr{R}_2$.) Then just apply the induction on the number of changes of $\mathscr{R}_1^*$ and $\mathscr{R}_2^*$ in the branches together with Lemmas 4.11, 4.12 and Proposition 4.3. (That was the pattern of the Hindley–Rossen lemma.) □

Although not needed for the main lemma we can prove that Mints' reductions are not only confluent but also weakly normalizing.

Proposition 4.14. *Terms in $\mathscr{R}_1$ normal form are closed for $\mathscr{R}_2$-reductions, so we have that Mints' reductions are weakly normalizing, the strategy being first do all $\mathscr{R}_1$-reductions then all $\mathscr{R}_2$-reductions (even more specifically $\mathscr{R}_2$ can be separated: first all Pr and β and then all T reductions).*

Proof. Just notice that application of Pr and β-reductions on the $\mathscr{R}_1$-normal term cannot introduce new $\mathscr{R}_1$ redexes. For example, if $a[\lambda x.b`c]$ is an $\mathscr{R}_1$-normal term, then $a[b(c)]$ is $\mathscr{R}_1$-normal too; all terms are in even a more restricted position than they were before the β-reduction. Also use Lemmas 4.11 and 4.12. □

Corollary 4.15 (Akama [1]). *Mints' reductions are strongly normalizing.*

Proof. First observe (examining several cases) that if a term is not in $\mathscr{R}_2$-normal form it cannot become $\mathscr{R}_2$-normal after application of $\mathscr{R}_1$-reductions. So assume that we have an infinite chain

$$t_0 \xrightarrow{R_{i_1}} t_1 \xrightarrow{R_{i_2}} \cdots \xrightarrow{R_{i_n}} t_n \xrightarrow{R_{i_{n+1}}} \cdots$$

($i_j \in \{1,2\}$). Since $\mathscr{R}_1$ is strongly normalizing as proved above, we have that in the above chain infinitely many reductions are of $\mathscr{R}_2$-type. Let $\bar{t_i}$ denote (the unique) $\mathscr{R}_1$-normal form of the term t_i. Then from the above infinite chain we can obtain (by $\mathscr{R}_1$-normalization) the following infinite chain:

$$\bar{t_0} \xrightarrow{\mathscr{R}_2^*} \bar{t_1} \xrightarrow{\mathscr{R}_2^*} \cdots \xrightarrow{\mathscr{R}_2^*} \bar{t_n} \xrightarrow{\mathscr{R}_2^*} \cdots .$$

This chain exists by the commutativity of $\mathscr{R}_1^*$ and $\mathscr{R}_2^*$ (Proposition 4.3) and the fact that $\mathscr{R}_1$ normal forms are closed for $\mathscr{R}_2$-reductions (Proposition 4.14). Also we have that the chain is infinite by the observation from the beginning of the proof. But this contradicts strong normalization of $\mathscr{R}_2$. □

It is obvious that unrestricted Mints' reductions are not normalizing (for example, x^{A^B} could be η-expanded and β-reduced infinitely many times); it is interesting, however, that they are confluent.

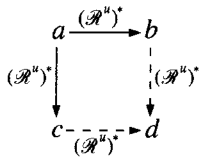
Corollary 4.16. *Mints' reductions without the restrictions are confluent.*

Proof. Suppose that

$$\begin{array}{lcl} a & \xrightarrow{(\mathscr{R}^u)^*} & b \\ {\scriptstyle (\mathscr{R}^u)^*}\big\downarrow & & \\ c & & \end{array}$$

That implies $b \cong_{\mathscr{R}^u} c$ and, by Lemma 4.2, it is the same as $b \cong_{\mathscr{R}} c$, and then from the confluence of $\mathscr{R}$ we have that there exists a term d such that $b \xrightarrow{\mathscr{R}^*} d$ and $c \xrightarrow{\mathscr{R}^*} d$.

Since $\mathscr{R} \subset \mathscr{R}^u$ we have



□

Remark 4.17. Mints' reductions were given in [12]. Unfortunately, Lemma 7.1(vi) and Theorem 7.3 are not correct. The theorem states that the normalizing strategy is first $\mathscr{R}_2$ then $\mathscr{R}_1$. Applying that on $x^{1\times A}$ we get $\langle \pi(x), \pi'(x) \rangle$. But applying the strategy on $\langle \pi(x), \pi'(x) \rangle$ gives $\langle *, \pi'(x) \rangle$. So two equal terms x and $\langle \pi(x), \pi'(x) \rangle$ do not have the same normal form. If the calculus were without the terminal object (and the appropriate rule) then first $\mathscr{R}_2$ and then $\mathscr{R}_1$ would be a normalizing strategy; this was suggested already in [14, 3.5.2 Normalization theorem] (notice however that the uniqueness of the normal form (there called expanded normal form) was not stated, cf. 3.5.3 Strong normalization theorem loc. cit.), but also recall that Prawitz considers all first-order logical connectives (even *absurdity*) but not the connective *true*.

Let us finally restate and prove

Proposition 3.5 [bis!]. If $f =_x g$ in a free λ-calculus and x does not occur as a free variable in either f or g then we also have $f = g$.

Proof. Since by Corollary 4.13 (free) typed λ-calculus is confluent for a set of reductions which do not introduce new variables, from $f =_x g$ we have that there is a term t such that f and g reduce to it, therefore $f = t$ and $t = g$. □

The above proof concludes the proof of our main result.

References

[1] Y. Akama, Mints reductions and ccc-calculus, in: M. Bezem and J.F. Groote, Eds., Typed Lambda Calculus and Applications 93, Lecture Notes in Computer Science, Vol. 664 (Springer, Berlin, 1993) 1–12.
[2] H.P. Barendregt, The Lambda Calculus – Its Syntax and Semantics, revised edition, Studies in Logic and Foundations of Mathematics, Vol. 103 (North-Holland, Amsterdam, 1985).
[3] P.-L. Curien and R. Di Cosmo, A confluent reduction for the λ-calculus with surjective pairing and terminal object, ICALP' 91, Madrid, 1991, pp. 291–302.
[4] H. Friedman, Equality between functionals, in: R. Parikh, Ed., Logic Colloquium' 73, Lecture Notes in Mathematics, Vol. 453 (Springer, Berlin 1975) 22–37.
[5] J.-Y. Girard, Y. Lafont and P. Taylor, Proofs and Types, Cambridge Tracts in Theoretical Computer Science, Vol. 7 (Cambridge Univ. Press, Cambridge, 1989).
[6] V. Harnik and M. Makkai, Lambek's categorical proof theory and Lauchli's concept of abstract realizability, Symbolic Logic 57 (1992) 200–230.
[7] J.R. Hindley and J.P. Seldin, Introduction to Combinators and Lambda-Calculus, London Mathematical Society Student Texts, Vol. 1 (Cambridge Univ. Press, Cambridge, 1986).

[8] C.B. Jay, Long $\beta\eta$-normal forms and confluence (revised), Tech. Report, University of Edinburgh, February 1992.
[9] J. Kennison, manuscript, 1992.
[10] J. Lambek and P.J. Scott, Introduction to Higher Order Categorical Logic (Cambridge Univ. Press, Cambridge, 1986).
[11] M. Makkai, The fibrational formulation of intuitionistic predicate logic I: completeness according to Gödel, Kripke and Läuchli, part 2, Notre Dame J. Formal Logic 34 (1993) 471–498.
[12] G. Mints, Teorija kategorij i teorija dokazateljstv, in: Aktualnie Voprosi Logiki i Metodologii Nauki, Kiev, 1980 (in Russian) pp. 252–278.
[13] J.C. Mitchell and P.J. Scott, Typed lambda models and cartesian closed categories, Contemp. Math. 92 (1988) 301–316.
[14] D. Prawitz, Ideas and results in proof theory, in: J.E. Fenstad, Ed., Proc. 2nd Scandinavian Logic Symp. (North-Holland, Amsterdam, 1971) 235–307.
[15] D.S. Scott, Relating theories of the λ-calculus, in: J.P. Seldin and J.R. Hindley, Ed., To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism (Academic Press, New York, 1980) 403–450.
[16] V.S. Soloviev, The category of finite sets and cartesian closed categories, J. Sov. Math. 22 (1983) 1387–1400.