INFORMATIQUE THÉORIQUE ET APPLICATIONS

ANNE PRELLER

P. DUROUX

Normalisation of the theory T of Cartesian closed categories and conservativity of extensions
mathbfT[x] **of** *mathbfT*

Informatique théorique et applications, tome 33, nº 3 (1999), p. 227-257

<http://www.numdam.org/item?id=ITA_1999__33_3_227_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 227–257

NORMALISATION OF THE THEORY T OF CARTESIAN CLOSED CATEGORIES AND CONSERVATIVITY OF EXTENSIONS T[x] OF T

ANNE PRELLER[1] AND P. DUROUX[1]

Abstract. Using an inductive definition of normal terms of the theory of Cartesian Closed Categories with a given graph of distinguished morphisms, we give a reduction free proof of the decidability of this theory. This inductive definition enables us to show via functional completeness that extensions of such a theory by new constants ("indeterminates") are conservative.

AMS Subject Classification. 03F05, 03B25, 18D15.

1. INTRODUCTION

The decidability of the theory of Cartesian Closed Categories is well known, see for example [7] where this property is established by passing to the internal language of a CCC and showing that the equality of this simply typed lambda calculus can be captured by a rewrite system which is Church-Rosser. A good overview of such methods in a rather general setting can be found in [5]. Obtułowicz [8] has proved the decidability of theories of Cartesian Closed Categories by algebraic techniques which avoid the requirement of finding a noetherian and confluent rewrite system. However, he imposes equalities on objects which in general are isomorphic, but not equal. We propose a proof by syntactic means only within the theory **T** of Cartesian Closed Categories over a graph of generators. The key notion is an inductive definition of the normal terms of the theory **T**. This definition of normal terms of **T** follows closely the definition given in "Categorical reconstruction of a reduction free normalisation proof of lambda-calculus" by Altenkirch *et al.* in [1]. Our normalisation of the terms of the theory of CCC also is reduction free, however we avoid the semantic sets, *i.e.* the category of presheaves introduced in [1] and also in [4], both building on [2]. The authors of [1] already indicate in a footnote that only the properties of cartesian closedness of their

[1] LIRMM, 161 rue ADA, 34392 Montpellier, France; e-mail: `preller@lirmm.fr` and e-mail: `duroux@lirmm.fr`

category of presheaves are relevant. We follow up this hint, using a Martin-Löf style axiomatisation of Cartesian Closed Categories over a graph of generators. In such an axiomatisation, not only the usual equalities, but also denotation statements as "f is an arrow of domain A and codomain B" are part of the formal theory, making the implementation of the normalisation algorithm more elegant. Having carried out reduction free normalisation in the syntax of CCC, it may be used to decide equality of simply typed lambda-calculus with surjective pairing via the usual interpretation.

Due to the presence of an arbitrary generating graph, the simply typed lambda-calculus associated to $\mathbf{T}$ may have non inhabited types. This makes the question whether extensions $\mathbf{T}[x : 1 \to A]$ of $\mathbf{T}$ obtained by adding an "indeterminate" x are conservative over $\mathbf{T}$, less easy to answer. Indeed, functional completeness reduces this question to the problem whether the second projections from $A \times B$ to B in the freely generated CCC are epimorphisms. If there is an arrow $a : 1 \to A$ in $\mathbf{T}$, then the second projection from $A \times B$ to B is a split epimorphism (compose to the right with $\langle a \circ ter(B), id(B)\rangle$) and so $\mathbf{T}[x : 1 \to A]$ is conservative over $\mathbf{T}$. Recently Cubrič has proved in [3] that extensions are conservative in the general case, again by a detour to lambda calculus. We obtain conservativity of extensions directly in the language of Cartesian Closed Categories as an easy consequence of functional completeness and inductive normalisation.

Section 2 presents the formal theory $\mathbf{T}$ of CCC's over a graph of generators, Section 3 contains the main results. In Subsection 3.1, the normal terms together with the related neutral and cut- free terms of $\mathbf{T}$ are defined. Subsection 3.2 the operator $+$ is introduced and its properties are studied, among them the basic one: for normal g and f, the term $g + f$ is normal and provably equal to $g \circ f$. Subsection 3.3 gives the algorithm which calculates the normal form of a term and contains the technical results. Finally, Subsection 3.4 recalls briefly the functional completeness with the properties needed here and gives the theorem of conservativity of extensions of $\mathbf{T}$.

We wish to thank Martin Hofmann for his precious help at the early stages of this work during his stay at the LIRMM. His inductive definition of the normal forms of arrow-terms in $\mathbf{T}$ set us on the right track.

2. Martin-Löf style axiomatisation of the theory of Cartesian Closed Categories with a distinguished graph of objects and morphisms

2.1. The language

We allow for arbitrary object- and arrow-constants in the theory of Cartesian Closed Categories. We assume that there is an object-constant 1 for the terminal object and an arbitrary number of other object-constants. From these we define the object-terms as usual. We use upper case letters for object-terms, lower case letters for arrow-terms. The arrow-constants are given by a set G, called the *generating graph* of the theory. It consists of all the object-constants and of triples (c, A, B) where c is called an *arrow-constant* and A and B are object-terms[1]. We assume that for every arrow-constant c there is exactly one triple (c, A, B) in G. Then there is a list of variables denoted by the letters x, y, z etc. We will distinguish between the theory of Cartesian Closed Categories $\mathbf{T}[G]$ axiomatised with terms constructed from the constants in G alone and those where we add variables and context information. For example, $\mathbf{T}[G][x : D \to A]$ is the extension of $\mathbf{T}[G]$ obtained by adding the variable x and the formulae $x : D \to A$ and $x = x : D \to A$. As G is fixed, we will write $\mathbf{T}$ for $\mathbf{T}[G]$, $\mathbf{T}[x : D \to A]$ for $\mathbf{T}[G][x : D \to A]$ etc. Due to the absence of variable binding functional symbols in the theory, the distinction between variables and constants is quite artificial; it is useful in the formulation of functional completeness simulating a variable binding mechanism.

There are two kinds of formulae: *equalities* and the *denotations.* Both are divided into two sorts: object-equalities and arrow-equalities, object-denotations and arrow-denotations.

A *denotation* is of the form

1) A *object,*
2) $f : A \to B$,

judgements which are of the form 1) or 2) have the meaning "the term A denotes" respectively that "the term f denotes and has domain A and codomain B", the latter will be read as "f *is an arrow* (or *a morphism*) *of domain* A *and codomain* B".

An *equality* is of the form

3) $A = B$, meaning that A and B are equal objects
4) $f = g : A \to B$, meaning that both f and g have domain A and codomain B and are equal.

As there are no dependent objects in the theory of Cartesian Closed Categories, every object-term denotes and the derivable object-equalities are exactly the identities $A = A$, for all object-terms. The situation is somewhat less trivial if we attempt a BNF style definition of arrow-terms:

$$t \equiv c/fst/snd/ev/ter/t \circ t/\langle t, t\rangle/t^*.$$

These expressions do not always correspond to a denoting arrow-term. However, the question whether for a given BNF expression f there are object-terms A and B such that $f : A \to B$ holds in $\mathbf{T}$, is decidable. This follows easily from the axiomatisation below, as the derivation of a denotation uses only denotations of subterms.

[1]The word "triple" here has nothing to do with monads, we use it in the naive sense: a sequence of three things.

2.2. The theories $\mathbf{T}$, $\mathbf{T}[x : A \to B]$

The rules of the theory are divided into two groups, the logical and the non-logical ones. Among the logical rules there are the general ones such as symmetry and transitivity of arrow-equality, assumption, and the specific ones, associated to the various functional symbols and called the construction rules. One kind of construction rules introduces a functional symbol, the other one expresses the compatibility of the symbol with arrow-equality. The basic properties of a Cartesian closed category are expressed in the non-logical rules under the form of equalities of arrows.

We omit symmetry and transitivity of arrow-equality and object-equality which the reader will easily provide, if needed.

$$\frac{A \textit{ object} \quad B \textit{ object}}{x : A \to B} \qquad \frac{A \textit{ object} \quad B \textit{ object}}{x = x : A \to B}$$

where x is a variable.

Either of the two rules above is referred to as an *x-assumption.*

General reflexivity of arrows will be derivable under the assumptions of reflexivity of the variables occurring in the arrows. As will be readily verified, the only derivable equalities between objects are of the from $A = A$.

Below comes the full list of rules specific to the theory of CCC's. The construction rules which present the functional symbols come in two groups, the introduction rules (on the left) and the compatibility rules (on the right).

$$\frac{}{C \textit{ object}} \qquad \frac{}{C = C}$$

for every object-constant C,

$$\frac{}{f : C \to D} \qquad \frac{}{f = f : C \to D}$$

for every arrow-constant f where (f, C, D) is the unique triple in the generating graph G starting with f,

$$\frac{S\ object}{idS : S \to S} \qquad \frac{S = S'}{idS = idS' : S \to S}$$

$$\frac{s : S \to T \quad t : T \to R}{t \circ s : S \to R} \qquad \frac{s = s' : S \to T \quad t = t' : T \to R}{t \circ s = t' \circ s' : S \to R}$$

$$\frac{t : T \to R}{idR \circ t = t : T \to R'} \qquad \frac{t : T \to R}{t \circ idT = t : T \to R}$$

$$\frac{s : S \to T \quad t : T \to R \quad u : R \to V}{u \circ (t \circ s) = (u \circ t) \circ s : S \to V}$$

$$\frac{A\ object \quad B\ object}{A \times B\ object} \qquad \frac{A = A' \qquad B = B'}{A \times B = A' \times B'}$$

$$\frac{A\ object \quad B\ object}{fst(A, B) : A \times B \to A} \qquad \frac{A = A' \qquad B = B'}{fst(A, B) = fst(A', B') : A \times B \to A}$$

$$\frac{f : U \to A \quad g : U \to B}{\langle f, g \rangle : U \to A \times B} \qquad \frac{f = f' : U \to A \quad g = g' : U \to B}{\langle f, g \rangle = \langle f', g' \rangle : U \to A \times B}$$

$$\frac{f : U \to A \quad g : U \to B}{fst(A, B) \circ \langle f, g \rangle = f : U \to A'} \qquad \frac{f : U \to A \quad g : U \to B}{snd(A, B) \circ \langle f, g \rangle = g : U \to B'}$$

$$\frac{h : U \to A \times B}{\langle fst(A, B) \circ h, snd(A, B) \circ h \rangle = h : U \to A \times B}$$

$$\frac{A\ object}{ter(A) : A \to 1} \qquad \frac{A = A'}{ter(A) = ter(A') : A \to 1}$$

$$\frac{f : A \to 1}{f = ter(A) : A \to 1},$$

$$\frac{A\ object \quad B\ object}{B^A\ object} \qquad \frac{A = A' \quad B = B'}{B^A = B'^{A'}}$$

$$\frac{A\ object \quad B\ object}{ev(A, B) : B^A \times A \to B} \qquad \frac{A = A' \quad B = B'}{ev(A, B) = ev(A', B') : B^A \times A \to B}$$

$$\frac{f : C \times A \to B}{f^* : C \to B^A} \qquad \frac{f = f' : C \times A \to B}{f^* = f'^* : C \to B^A}$$

$$\frac{f : C \times A \to B}{ev(A, B) \circ \langle f^* \circ fst(C, A),\ snd(C, A) \rangle = f : C \times A \to B}$$

$$\frac{h : C \to B^A}{(ev(A, B) \circ \langle h \circ fst(C, A),\ snd(C, A) \rangle)^* = h : C \to B^A}.$$

Notations:
We refer to the term $\langle f, g \rangle : U \to A \times B$ as a *pair*. Similarly, $fst(A, B) : A \times B \to A$ and $snd(A, B) : A \times B \to B$ are the *first* and *second projection*, $ev(A, B) : B^A \times$

$A \to B$ is called *evaluation* and $f^* : C \to B^A$ the *abstraction* of $f : C \times A \to B$. Finally, $ter(A)$ goes under the name of *terminal arrow*.

Whenever the context permits, the arguments A, B are omitted so that we just use fst, snd, ev, id, ter. It is always understood that the missing arguments are such that the term denotes. The so called type checking algorithm does just that: it invents names for arguments such that the term denotes and rejects the term, if this is not possible.

By analogy to λ-calculus, the term $(ev \circ \langle h \circ fst,\ snd \rangle)^*$ is called the *η-expansion* of h, and $\langle fst \circ f,\ snd \circ f \rangle$ the *η-expansion* of f.

Definition 1. A derivation is a tree labelled by formulae such that each label other than that of a leaf is obtained from its immediate predecessors by one of the rules above. Moreover, if several x-assumptions occur in the tree, they all have the same domain and codomain. A derivation is *closed*, if all its leaves have an empty label. The *Theory* $\mathbf{T}[G]$ or $\mathbf{T}$, for short, consists of all formulae which are the root of a closed derivation without occurrences of variables and with all constants in G.

The *Theory* $\mathbf{T}[x : A \to B]$ consists of all formulae which are at the root of a closed derivation having no other occurrences of variable-assumptions than $x : A \to B$ or $x = x : A \to B$. Clearly, x is the only variable which could occur in a formula belonging to $\mathbf{T}[x : A \to B]$. We also say that a formula holds or is *derivable in* $\mathbf{T}[x : A \to B]$, if it is the label of the root of a derivation. We shall say that s equals t in $\mathbf{T}[x : A \to B]$, if $s = t$ is derivable in $\mathbf{T}[x : A \to B]$. It is easy to see that a derivation has no occurrences of x in any of its labels if and only if no x-assumption has been applied.

N.B. A closed derivation may have occurrences of x-assumptions. In opposition to type theory, there is no primitive mechanism for discharging assumptions or binding variables. Discharging is simulated with the help the κ-construction of functional completeness.

As an example of an equality derivable in $\mathbf{T}$ we mention the frequently used equality.

$$f^* \circ t = (f \circ \langle t \circ fst, snd \rangle)^*.$$

3. Normalisation

3.1. Neutral, normal and cut-free terms

We begin with the joint inductive definition of neutral and normal terms of $\mathbf{T}$. Cut-free terms are an extension of both neutral and normal terms. They play the same role as the elements of the semantic sets of [1], *i.e.* they are needed in the search of the normal form of an arbitrary arrow-term. Suppose that the generating graph has no arrows, hence is reduced to a set of object-constants O. Then $\mathbf{T}[O]$ together with the congruence relation of derivable equality is the free $P - CCC$ on O, a notion defined in [4]. Consider the Yoneda functor Y from $\mathbf{T}[O]$ to the category of presheaves over $\mathbf{T}[O]$ and the functor $[\ ]$ generated by the injection J from O to the category of presheaves which sends A in O to $Y(A)$. There is an isomorphism q from $[\ ]$ to Y. For all objects X and A, the inverse $u_{A,X}$ of $q_{A,X}$ maps a term $g : X \to A$, element of $Y(A)$, to an element of the semantic set $[A]_X$. To find the normal form of $f : B \to A$, we need only those elements of $[B]_A$ which can be written as $[f]u_{A,A}(idA)$. As terms of $\mathbf{T}[O]$ are simpler concepts than elements of the set $[B]_A$, we prefer to define the "action" $[f]_X : [A]_X \to [B]_X$ for $f : A \to B$ directly on the cut-free terms which are the syntactical variant to these elements, avoiding thus set theory and lambda- calculus. The price to pay for avoiding these powerful theories are tedious proofs in the theory of CCC's.

We treat the notions of neutral, normal, and cut-free terms as syntactical and therefore belonging to the meta-theory. These notions can be easily formalised in category theory itself: add new atomic formulae to the language of $\mathbf{T}$, namely $f : D \underset{No}{\longrightarrow} C$ meaning that f is a normal arrow-term of domain D and codomain C. Similarly, $f : D \underset{Ne}{\longrightarrow} C$ expresses that f is a neutral term of domain D and codomain C and $f : D \underset{Cf}{\longrightarrow} C$ says that f is a cut-free arrow-term of domain D and codomain C.

In the following, the letter O will stand for an arbitrary object-constant different from the object- constant 1.

Definition 2. A derivation of $f : D \underset{No}{\longrightarrow} C$ (respectively $f : D \underset{Ne}{\longrightarrow} C$) is given by the following rules:

(identity rule)

$$\frac{}{id(A) : A \underset{Ne}{\longrightarrow} A}$$

(terminal arrow rule)

$$\frac{}{ter(D) : D \underset{No}{\longrightarrow} 1} \quad \frac{}{ter(D) : D \underset{Ne}{\longrightarrow} 1}$$

(swivel-rule)

$$\frac{f : D \underset{Ne}{\longrightarrow} O}{f : D \underset{No}{\longrightarrow} O}$$

O an object-constant different from 1

(pair rule)

$$\frac{f : D \underset{No}{\longrightarrow} A \quad g : D \underset{No}{\longrightarrow} B}{\langle f, g\rangle : D \underset{No}{\longrightarrow} A \times B}$$

(projection rules)

$$\frac{f : D \underset{Ne}{\longrightarrow} A \times B}{fst(A,B) \circ f : D \underset{Ne}{\longrightarrow} A} \qquad \frac{f : D \underset{Ne}{\longrightarrow} A \times B}{snd(A,B) \circ f : D \underset{Ne}{\longrightarrow} B}$$

(abstraction rule) (evaluation rule)

$$\frac{f : D \times A \underset{No}{\longrightarrow} B}{f^* : D \underset{No}{\longrightarrow} B^A} \qquad \frac{f : D \underset{Ne}{\longrightarrow} B^A}{e\nu(A,B) \circ \langle f, a\rangle : D \underset{Ne}{\longrightarrow} B}$$

(constant-rule)

$$\frac{a : D \underset{No}{\longrightarrow} B \quad c : B \to C}{c \circ a : D \underset{Ne}{\longrightarrow} C}$$

where c is an arrow constant in the generating graph.

Definition 3. A *cut-free* term f of domain D and codomain C is derived by the following rules:

$$\frac{f : D \underset{Ne}{\longrightarrow} C}{f : D \underset{Cf}{\longrightarrow} C}$$

$$\frac{f : D \underset{Cf}{\longrightarrow} A \quad g : D \underset{Cf}{\longrightarrow} B}{\langle f, g\rangle : D \underset{Cf}{\longrightarrow} A \times B}$$

$$\frac{f : D \times A \underset{Cf}{\longrightarrow} B}{f^* : D \underset{Cf}{\longrightarrow} B^A}$$

Notice that $id : 1 \to 1$ and $ter : 1 \to 1$ are both neutrals, but only the latter is normal. It follows immediately from the definition that a normal term of codomain B^A is necessarily an abstraction, *i.e.* of the form f^*; and that a normal term which has a product as codomain is a pair. On the other hand, a neutral term is neither an abstraction nor a pair. The dominant symbol of a complex neutral is the composition $\circ$. In fact, a neutral other than id or ter is of the form $v \circ g$ where v is one of fst, snd, ev or c. The last rule applied in the derivation of a neutral, normal or cut-free term is uniquely determined by the syntactical form of that term. Hence the derivation of $f : D \underset{No}{\longrightarrow} C$, $f : D \underset{Ne}{\longrightarrow} C$ and $f : D \underset{Cf}{\longrightarrow} C$ is unique.

Cut-free terms can be interpreted as natural deductions, if one changes the discharge mechanism of the introduction rule of implication slightly: if H is a natural deduction of B such that all leaves are labelled $C \wedge D$, then $\frac{H^{\not{C}}}{C \supset B}$ is a natural deduction of $C \supset B$ where $H^{\not{C}}$ is obtained from H by replacing the label $C \wedge D$ by $\not{C} \wedge D$ for every leaf (only the leftmost occurrence of C is discharged). Ignoring the discharged factor C, consider D the label of every leaf of the new deduction $\frac{H^{\not{C}}}{C \supset B}$. Now interpret the rules defining neutral and normal terms the obvious way. For example the identity rule corresponds to the logical axioms, the projection rules to the elimination rule and the pair rule to the introduction rule of conjunction etc. Then all cut-free terms are interpreted as natural deductions without cuts, the normal terms correspond to deductions where the change from elimination to introduction rules along a branch occurs at atomic nodes. The neutral ones make no use of introduction rules along any major branch.

In order to express "action" in the syntax of **T**, we first define an operator + which when given two composable cut-free terms, will calculate a cut-free term provably equal to the composition of the two given terms. This operator is first defined for the cut-free terms which have no occurrences of ev nor $*$, *i.e.* the cut-free terms of cartesian categories. Then the definition is extended to the case where the left argument is arbitrary and finally to the general case.

Definition 4.

1) A *simple projection* is a neutral term which is derivable using the identity and projection rules only.
2) Every simple projection and terminal arrow $ter(D)$ is a *generalised projection*. If $f : D \to A$ and $g : D \to B$ are generalised projections, then $\langle f, g\rangle.D \to A \times B$ is a *generalised projection.*
3) A *generalised neutral.* is either a neutral or a pair of generalised neutrals.

Hence, a simple projection is either the identity or obtained from it by composing it with the projections fst and snd repeatedly. Generalised projections are derivable using the terminal arrow rule, the identity rule and the projection rules first and after that pair rules only. In general, they are not neutrals, but generalised neutrals. In any case, they are cut-free terms.

We now are ready for the operator + which associates to a cut-free term $f : D \to C$ and a generalised projection $t : X \to D$ a new cut-free term $f + t : X \to C$ [2]. Later, in Subsection 3.2, this operator will be extended to arbitrary cut-free terms in both arguments.

Definition 5. Let $f : D \to C$ be a cut-free term and $t : X \to D$ a generalised projection.
I) Suppose first that f is a generalised projection. Define by induction on the derivation of f a new term $f + t$:

0) $id + t \equiv t$
1) $ter(D) + t \equiv ter(X)$
2) $(fst \circ f) + t \equiv fst \circ (f + t)$, if $f + t$ is not a pair,
3) $(fst \circ f) + t \equiv f_1$, if $f + t \equiv \langle f_1, f_2\rangle$

[2] $f + t$ is a meta-notation, a name for the term given in the definition. The letter X stands for an arbitrary object term.

4) $(snd \circ f) + t \equiv snd \circ (f + t)$, if $f + t$ is not a pair,
5) $(snd \circ f) + t \equiv f_2$, if $f + t \equiv \langle f_1, f_2 \rangle$
6) $(\langle f_1, f_2 \rangle) + t \equiv \langle f_1 + t, f_2 + t \rangle$.

Hence, $f + t$ is defined for all generalised projections f and t and is again a generalised projection.
II) Now suppose that f is an arbitrary cut-free term. Then define $f + t$ again by induction on the derivation of f by the equalities 1) through 6) above and by

7) $(c \circ a) + t \equiv c \circ (a + t)$
8) $f^* + t \equiv (f + \langle t + (fst \circ id),\ snd \circ id \rangle)^*$
9) $(ev \ \circ \langle f, a \rangle) + t \equiv ev \ \circ \langle f + t, a + t \rangle$

Lemma 1. (i) *if $f : D \to C$ is a generalised neutral, then $f + t : X \to C$ is a generalised neutral. If moreover C is a power B^A (respectively a constant O), then $f + t : X \to B^A$ (respectively $f + t : X \to O$) is neutral;*
(ii) *if $f : D \to C$ is normal, then $f + t : X \to C$ is normal;*
(iii) *if $f : D \to C$ is neutral and t a simple projection, then $f + t : X \to C$ is neutral;*
(iv) *if $f : D \to C$ is a cut-free term and $t : X \to D$ a generalised projection, then $f + t : X \to C$ is a cut-free term;*
(v) *if $f : D \to C$ is a generalised projection, then $f + id \equiv f$;*
(vi) *The equality $f + t = f \circ t$ is provable in* $\mathbf{T}$.

We omit the proof of Lemma 1 which is straightforward.
Notation: A useful abbreviation is f^{+fst} for $f + (fst \circ id)$.

3.2. The operator +

We extend the operator + which until now is defined on the right for generalised projections t only to arbitrary cut-free terms t. This operator provides us with a "composition" of cut-free terms which, when restricted to normal terms, yields a normal term.

In the definition of the operator +, we use the notion of the *degree d(f)* of a cut-free term f:

The *complexity c(D)* of an object term D is defined by induction on the construction of the term: The complexity of a constant O or of 1 is 0. The complexity of a product is the maximum of the complexities of its factors. The complexity of a power B^A is

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

Then we define by induction on the derivation of a cut-free term its *degree d(f)*:

$$\begin{aligned} d(f) &= 0 \text{ for neutral } f, \\ d(\langle f_1, f_2 \rangle) &= \max\{d(f_1), d(f_2)\}, \\ d(f^*) &= c(B^A) \text{ for } f : D \to B^A \end{aligned}$$

N.B. For an arbitrary cut-free term $f : D \to C$, we can only say that $d(f) \leq c(C)$. Only for a normal term f we have the equality $d(f) = c(C)$. Notice also that if $d(f) = 0$ if and only if f is a generalised neutral. In particular, if the codomain of f is not a product and $d(f) = 0$, then f is neutral.

Definition 6. Let $f : D \to C$ and $t : X \to D$ be arbitrary cut-free terms. Define a new cut-free term $f + t : X \to C$ such that $d(f + t) \leq \max\{d(f), d(t)\}$ by induction on pairs of natural numbers (n, m) ordered in alphabetical order where $n = d(t)$ and m is the maximal length of the branches of the derivation tree of $f : D \underset{Cf}{\longrightarrow} C$:

(0) $id + t \equiv t$
(1) $ter(D) + t \equiv ter(X)$
(2) $(fst \circ f) + t \equiv fst \circ (f + t)$, if $f + t$ is not a pair,
(3) $(fst \circ f) + t \equiv f_1$, if $f + t \equiv \langle f_1, f_2 \rangle$
(4) $(snd \circ f) + t \equiv snd \circ (f + t)$, if $f + t$ is not a pair,
(5) $(snd \circ f) + t \equiv f_2$, if $f + t \equiv \langle f_1, f_2 \rangle$
(6) $(\langle f_1, f_2 \rangle) + t \equiv \langle f_1 + t, f_2 + t \rangle$
(7) $(c \circ a) + t \equiv c \circ (a + t)$
(8) $f^* + t \equiv (f + \langle t + (fst \circ id),\ snd \circ id \rangle)^*$
(9) $(ev\ \circ \langle f, a \rangle) + t \equiv ev\ \circ \langle f + t, a + t \rangle$, if $f + t$ is not an abstraction
(9′) $(ev\ \circ \langle f, a \rangle) + t \equiv h\ + \langle id, a + t \rangle$, if $f + t \equiv h^*$.

In fact, this operator extends the operator given in Definition 5 of which it reproduces the ten defining equalities 0) to 9). Only now an alternative has been added for the evaluation rule, namely 9′): if f is neutral and $f + t$ is defined for an arbitrary cut-free term t, then it does no longer follow that $f + t$ is neutral as it did in the case where t was a generalised projection. So, at the next step, when we want to define $(ev\ \circ \langle f, a \rangle) + t$, we must take care of the alternative where $f + t$ is not a neutral, but an abstraction. As the term $f + t$ may well have a longer derivation than f, we need a second induction index besides the length of the derivation of f. It is provided by the degree n of t.

Lemma 2. *For arbitrary cut-free $f : D \to C$ and $t : X \to D$, the term $f+t : X \to C$ is well defined, cut-free and satisfies $d(f + t) \leq \max\{d(f), d(t)\}$. Moreover, if f is normal, then so is $f + t$. If f and t are neutral, then $f + t$ is a generalised neutral. In particular, if C is not a product, and f and t are neutral, so is $f + t$.*

Proof. First show for an arbitrary cut-free f that the term $f + t$ is well defined and has the desired properties for all cut-free t of degree 0. Proceed by induction on m, *i.e.* by induction on the derivation of f. For example, consider the case where the derivation terminates with the abstraction rule. Hence $f \equiv h^*$ where the lemma folds for h and all s of degree 0. Then equality 8) is used to define $h\ + t$. Definition 5 and Lemma 1 make sure that $t + (fst \circ id)$ is already defined and that $d(\langle t + (fst \circ id),\ snd \circ id \rangle) = 0$. Hence the induction hypothesis applies to h and $\langle t + (fst \circ id),\ snd \circ id \rangle$. In the case where the derivation terminates by the evaluation rule, then one of the defining equalities 9) or 9′) must be used. Let

$f \equiv ev \circ \langle g, a \rangle : D \underset{Ne}{\longrightarrow} B^A$ with neutral $g : D \to B^A$ and normal $a : D \to A$. By induction hypothesis, $d(g + t) \leq \max\{d(g), d(t)\} = 0$. Hence $g + t$ is neutral and thus it is 9) which is applied. As $a + t$ is normal by induction hypothesis, the resulting term $f + t$ is neutral and has degree 0. Notice by the way that the last rule in the derivation of $f + t$ is the evaluation rule, the same as in the derivation of f.

Finally, assume $n > 0$ and that $h + s$ is defined and has the desired properties for all cut-free s of degree less than n and for arbitrary cut-free h. Now show by induction on the derivation of the cut-free $f : D \to C$ that $f + t$ is well defined and $d(f + t) \leq \max\{d(f), d(t)\}$ for all cut-free t of degree n. According to the last rule in the derivation of f, one of the defining equalities above is used. For example, consider the case of equality 8). As $d(\langle t + (fst \circ id),\ snd \circ id \rangle) = d(t) = n$ we reason as above. Suppose now that the derivation terminates with the evaluation rule, *i.e.* $f \equiv ev \circ \langle g, a \rangle : D \underset{Ne}{\longrightarrow} B^A$ with neutral $g : D \to B^A$ and normal $a : D \to A$. If $g + t$ is neutral, the argument is simular to the one given in the correpsonding case above. If $g + t \equiv h^*$, the equality 9') is used. To show that $h + \langle id, a + t \rangle$ is defined it suffices to remark that the degree of $\langle id, a + t \rangle$ is less than n. Indeed,

$$\begin{aligned} c(B^A) &= d(h^*) \\ &= d(g + t) \\ &\leq \max\{d(g), d(t)\} \text{ by induction hypothesis on } g, \\ &= d(t) \text{ by neutrality of } g. \end{aligned}$$

Hence, $c(A)$ and $c(B)$ are strictly less than $n = d(t)$. By induction hypothesis, $a + t$ is a normal term. Its degree is no greater than the complexity $c(A)$ of its codomain A. Therefore, $d(\langle id, a + t \rangle) \leq d(a + t) \leq c(A) < n$. By induction hypothesis, $h + \langle id, a + t \rangle$ is defined and a cut-free term of codomain B. Hence, $n \rangle c(B) \geq d(h + \langle id, a + t \rangle)$. Finally,

$$d(ev \circ \langle g, a \rangle + t) = d(h + \langle id, a + t \rangle) < n = d(t) = \max\{d(ev \circ \langle g, a \rangle), d(t)\},$$

as $ev \circ \langle g, a \rangle$ is neutral.

Warning: Before going on with the properties of the operator $+$, remark that it is not commutative. For example

$$id^* + id \equiv (id + \langle fst \circ id, snd \circ id \rangle)^* \equiv \langle fst \circ id, snd \circ id \rangle^*$$

whereas

$$id + id^* \equiv id^*.$$

The somewhat misleading notation has been adopted for historical reasons. Our $+$ generalises the $(.)^{+w}$ defined in [1] for weakening morphisms w.

If one interprets cut-free terms as natural deductions, then the operator $+$ can be seen as "grafting" the deduction tree corresponding to t onto the tip of every branch of the tree corresponding to f, simultaneously eliminating the cuts created by such grafting. The proof of Lemma 2 follows closely the usual proof of cut-elimination by conversion, given for natural deductions. In category language, the operator $+$ is the cut-free version of composition: □

Lemma 3. *For arbitrary cut-free f and t, the equality $f + t = f \circ t$ holds in* $\mathbf{T}$.

Proof. By induction on the index (n, m). Assume that the property holds for all cut-free s of degree less than $n = d(t)$ and all cut-free h. In order to show that it also holds for n, proceed by induction on the derivation of f. Consider for example the case of equality (9′), the other cases are straightforward. Assume that $f+t \equiv h^*$ and that $f+t = f \circ t$ and $a+t = a \circ t$ in $\mathbf{T}$ by induction hypothesis. Then

$ev \ \circ \langle f, a\rangle + t = h \ + \langle id, a + t\rangle$	property of $\langle id, a+t\rangle$, as $d(\langle id, a+t\rangle) < n$
$= h \circ \langle id, a + t\rangle,$	
$= h \circ \langle id, a \circ t\rangle$	property of a
$= ev \ \circ \langle h^* \circ fst, \ snd\rangle \circ \langle id, a \circ t\rangle,$	
$= ev \ \circ \langle h^*, a \circ t\rangle,$	
$= ev \ \circ \langle f + t \quad a \circ t\rangle$	definition of h
$= ev \ \circ \langle f \circ t, a \circ t\rangle$	property of f
$= ev \ \circ \langle f, a\rangle \circ t.$	

□

3.3. REWRITING, ACTIONS AND NORMAL FORMS

We define a normal term $rewrite(f) : D \to C$ for any cut-free term $f : D \to C$ such that the equality $rewrite(f) = f : D \to C$ holds in $\mathbf{T}$. In fact, the operator *rewrite* replaces a term by its maximal η-expansion.

Definition 7. The term *rewrite(f)* is defined by induction on the codomain of f.

$$
\begin{array}{lll}
rewrite_O(f) & \equiv f, & \text{if the codomain of } f \text{ is the constant } O \\
rewrite_1(f) & \equiv ter(D), & \text{if the codomain of } f \text{ is } 1 \\
rewrite_{A\times B}(\langle g,h\rangle) & \equiv \langle rewrite_A(g),\ rewrite_B(h)\rangle & \\
rewrite_{A\times B}(f) & \equiv \langle rewrite_A(fst\circ f),\ rewrite_B(snd\circ f)\rangle & \text{if } f \text{ is not a pair.} \\
rewrite_{B^A}(h^*) & \equiv (rewrite_B(h))^* & \\
rewrite_{B^A}(f) & \equiv (rewrite_B(ev\ \circ\langle f^{+fst}, & \\
 & \quad rewrite_A(snd\circ id)\rangle))^*, & \text{if } f \text{ is not an abstraction.}
\end{array}
$$

Whenever the context permits, we omit the subscript in *rewrite*.

It follows immediately from the definitions that

Remark 0) $rewrite(f)$ is normal for all cut-free f.

Remark 1) The equality $rewrite(h) = h : D \to C$ is derivable in **T**.

Remark 2) If a is normal, then $rewrite(a) \equiv a$.

The latter is false for neutrals. However, if we apply *rewrite* to two neutrals and get the same result, then they were already identical beforehand, unless the codomain has "to many" factors 1 as we shall explained below.

The *flattened object* associated to B, in symbols $Fl(B)$, is obtained by deleting all exponents in B :

$$
\begin{array}{ll}
Fl(O) & \equiv O \\
Fl(1) & \equiv 1 \\
Fl(A\times B) & \equiv Fl(A)\times Fl(B) \\
Fl(B^A) & \equiv Fl(B).
\end{array}
$$

The object-term C is said to be *ambiguous*, if 1 is the only constant occurring in $Fl(C)$.

Remark 3) Assume that C is not ambiguous and let $h : X \to C$ and $h' : X \to C$ be neutrals such that $rewrite(h) \equiv rewrite(h')$. Then $h \equiv h'$.

Remark 4) Let $h : X \to C$ and $h' : X \to C$ be any cut-free terms. If C is ambiguous, then $rewrite(h) \equiv rewrite(h')$.

These Remarks can be easily verified. Remarks 3) and 4) are seen using induction on the complexity of the codomain C.

Accepting the intuition that cut-free terms are the syntactical version of elements of the form $[f]u_{A,A}(idA)$ in the sets $[B]_A$ as defined in [1], *rewrite* can be compared with q, as it transforms a cut-free term into a normal one. It remains to define the syntactical version of the functor [.] of [1] respectively [4]. This is the next (and last) concept we need before defining normalisation. We call $[h]$ the "action" induced by an arbitrary term h on cut-free terms. Action also is a sort of composition, but now the term on the left is arbitrary, only the term on the right is cut-free.

Definition 8. Let $h : D \to C$ be any denoting term and $f : X \to D$ a cut-free term. The cut-free term $[h](f) : X \to C$, called the result of the *action by h on f* is defined by induction on the complexity of h as follows:

$$
\begin{array}{ll}
[id](f) & \equiv f \\
[ter(D)](f) & \equiv ter(X) \\
[c](f) & = c \circ rewrite(f) \text{ where } c \text{ is an arrow-constant} \\
[fst](\langle f_1, f_2 \rangle) & \equiv f_1, \\
[fst](f) & \equiv fst \circ f, \text{ if } f \text{ is not a pair} \\
[snd](\langle f_1, f_2 \rangle) & \equiv f_2, \\
[snd](f) & \equiv snd \circ f, \text{ if } f \text{ is not a pair} \\
[ev](\langle g^*, a \rangle) & \equiv g \ + \langle id, a \rangle \\
[ev](\langle f, a \rangle) & \equiv ev \ \circ \langle f, rewrite(a) \rangle, \text{ if } f \text{ is not an abstraction,} \\
[ev](f) & \equiv ev \ \circ \langle fst \circ f, rewrite(snd \circ f) \rangle, \text{ if } f \text{ is not a pair} \\
[h_1 \circ h_2](f) & \equiv [h_1]([h_2](f)) \\
[\langle h_1, h_2 \rangle](f) & \equiv \langle [h_1](f), [h_2](f) \rangle \\
[h^*](f) & \equiv ([h](\langle f^{+fst}, \ snd \circ id \rangle))^*.
\end{array}
$$

Lemma 4. *For all denoting $h : D \to C$ and cut-free $f : X \to D$ the equality*

$$[h](f) = h \circ f : X \to C \text{ holds in } \mathbf{T}.$$

In particular, the equality

$$rewrite([h](id)) = h : X \to C \text{ is derivable in } \mathbf{T}.$$

The proof is straight forward and we will skip it.

Definition 9 (Normal Form)**.** The normal form $nf(h) : D \to C$ of an arbitrary term $h : D \to C$ is defined by

$$nf(h) \equiv rewrite([h](id)).$$

N.B. it follows immediately from Lemma 4 that the equality $nf(h) = h : D \to C$ is derivable in $\mathbf{T}$.

Theorem 1. *For every $h : D \to C$ of $\mathbf{T}$, $nf(h)$ is normal and the equality $h = nf(h) : D \to C$ holds in $\mathbf{T}$. If $h : D \underset{No}{\longrightarrow} C$, then $nf(h) \equiv h$. The equality $h = g : D \to C$ is derivable in $\mathbf{T}$, if and only if $nf(h) \equiv nf(g)$.*

Proof. The first two assertions follow immediately from the preceding lemmas, namely that for every $h : D \to C$ of $\mathbf{T}$,

- $nf(h) \equiv rewrite([h](id))$ is normal,

- the equality $h = nf(h) : D \to C$ holds in $\mathbf{T}$,
  The last two assertions correspond to the three properties,
- $nf(h) \equiv h$ for normal h,
- if $nf(h) \equiv nf(g)$, then the equality $h = g : D \to C$ is derivable in $\mathbf{T}$
- if $h = g : D \to C$ is derivable in $\mathbf{T}$, then $nf(h) \equiv nf(g)$. □

Only the last property, namely uniqueness of normal form, needs still proving, the two preceding ones are easy consequences of the preceding lemmas. To establish uniqueness, we need a few properties of the operator $+$. They are given by the following Lemmas 5 to 9:

Lemma 5 (Associativity). *Let be given $f : D \underset{Cf}{\longrightarrow} C, t : X \underset{Cf}{\longrightarrow} D$ and $t' : Y \underset{Cf}{\longrightarrow} X$. Then $(f + t) + t' \equiv f + (t + t')$, provided one of the following holds:*

SPECIAL CASE I: f is a generalised projection, t and t' are arbitrary.
SPECIAL CASE II: t is a generalised projection, f and t' are arbitrary
SPECIAL CASE III: t' is a generalised projection, f and t are arbitrary

Proof. Show each special case separately, using induction on the derivation of f. Considering the abstraction and the evaluation rule (the others are routine), assume $(f + s) + s' \equiv f + (s + s')$ for all s, s' of the right kind. Then

Abstraction rule,

$$\begin{aligned}
(f^* + t) + t' &\equiv ((f + \langle t^{+fst}, snd \circ id\rangle) + \langle t'^{+fst}, snd \circ id\rangle)^* \\
&\equiv (f + (\langle t^{+fst}, snd \circ id\rangle + \langle t'^{+fst}, snd \circ id\rangle))^* \quad \text{property of } f \\
&\equiv (f + (\langle (t^{+fst}) + \langle t'^{+fst}, snd \circ id\rangle, \\
&\qquad snd \circ id + \langle t'^{+fst}, snd \circ id\rangle))^* \\
&\equiv (f + (\langle t + (fst \circ id\ + \langle t'^{+fst}, snd \circ id\rangle), \\
&\qquad snd \circ id\rangle))^* \qquad \text{(Eq. 1)} \\
&\equiv (f + (\langle t + (t'^{+fst}), snd \circ id\rangle))^* \\
&\equiv (f + (\langle (t + t')^{+fst},\ snd \circ id\rangle))^* \qquad \text{(Eq. 2)} \\
&f^* + (t + t').
\end{aligned}$$

where the equalities below still need justification

(Eq. 1) $(t + fst \circ id)\ + \langle t'^{+fst}, snd \circ id\rangle \equiv t + (fst \circ id\ + \langle t'^{+fst}, snd \circ id\rangle)$
(Eq. 2) $t + (t'^{+fst}) \equiv (t + t')^{+fst}$

Evaluation Rule: Suppose the property holds for neutral f and normal a, show it for $ev\ \circ \langle f, a\rangle$.

Three cases have to be distinguished:

Case 1: $(f + t) + t'$ is neutral:

Then,

$$\begin{aligned}(ev \circ \langle f,a\rangle + t)+t' &\equiv ev \circ \langle (f+t)+t', (a+t)+t'\rangle \\ &\equiv ev \circ \langle f+(t+t'), a+(t+t')\rangle \quad \text{property of } f \text{ and } a \\ &\equiv (ev \circ \langle f,a\rangle) + (t+t')\end{aligned}$$

Case 2: $(f+t)$ is neutral, $(f+t)+t' \equiv g^*$:
Then, $f+(t+t') \equiv (f+t)+t' \equiv g^*$, property of f.
Therefore

$$\begin{aligned}(ev \circ \langle f,a\rangle + t)+t' &\equiv g + \langle id, (a+t)+t'\rangle \\ &\equiv g + \langle id, (a+(t+t')\rangle \quad \text{property of } a \\ &\equiv ev \circ \langle f,a\rangle + (t+t') \quad \text{definition of } +\end{aligned}$$

Case 3: $(f+t) \equiv h^* : X \to B^A$
Notice that $d(\langle id, a+t\rangle) \leq c(A) < c(B^A) = d(h^*) = d(f+t) = d(t)$.
Then $(f+t)+t' \equiv (h + \langle t'^{+fst}, snd \circ id\rangle)^* \equiv f+(t+t')$, property of f.
and

$$\begin{aligned}(ev \circ \langle f,a\rangle + t)+t' &\equiv (h + \langle id, a+t\rangle) + t' \\ &\equiv h + (\langle id, a+t\rangle + t') \quad \text{(Eq. 3)} \\ &\equiv h + \langle t', (a+t)+t'\rangle \\ &\equiv h + \langle t', a+(t+t')\rangle \quad \text{property of } a \\ &\equiv h + \langle t'+id, a+(t+t')\rangle \quad \text{(Eq. 4)} \\ &\equiv h + \langle t' + (fst \circ id + \langle id, a+(t+t')\rangle), \\ &\qquad snd \circ id + \langle id, a+(t+t')\rangle\rangle \\ &\equiv h + \langle (t' + fst \circ id) + \langle id, a+(t+t')\rangle, \\ &\qquad snd \circ id + \langle id, a+(t+t')\rangle\rangle \quad \text{Special Case II} \\ &\equiv h + (\langle t'^{+fst}, snd \circ id\rangle \\ &\qquad + \langle id, a+(t+t')\rangle) \\ &\equiv (h + \langle t'^{+fst}, snd \circ id\rangle) \\ &\qquad + \langle id, a+(t+t')\rangle \quad \text{(Eq. 5)} \\ &\equiv (ev \circ \langle f,a\rangle) + (t+t') \quad \text{definition of } +\end{aligned}$$

where the three mentioned equalities still need to be proved:
(Eq. 3) $(h + \langle id, a+t\rangle) + t' \equiv h + (\langle id, a+t\rangle + t')$,
(Eq. 4) $t' + id \equiv t'$

(Eq. 5) $h \ + (\langle t'^{+fst}, snd \circ id\rangle + \langle id, a + (t + t')\rangle) \equiv (h \ + \langle t'^{+fst}, snd \circ id\rangle) + \langle id, a + (t + t')\rangle$

Now finish the proof by showing the property

1) first in the SPECIAL CASE I by induction on f. In this case, the equalities (Eq. 1) – (Eq. 5) are not needed;
2) next, in the SPECIAL CASE II, again by induction on f ; now (Eq. 1) and (Eq. 2) are justified by SPECIAL CASE I. The equalities (Eq. 3 – Eq. 5) are not needed as Case 3 of the Evaluation Rule does not arise, because t has degree 0.
3) in the SPECIAL CASE III, using induction on the pair (n, m) where $n = d(t)$ and m is the length of the derivation of the cut-free term f. In this case (Eq. 1) and (Eq. 2) hold by SPECIAL CASE II, (Eq. 4) is established in Lemma 1, (Eq. 3) is true by induction hypothesis (remember: $d(\langle id, a+t\rangle) \leq d(a + t) \leq c(A) < c(B^A) = d(h^*) = d(f + t) \leq \max\{d(f), d(t)\} = d(t)$ as f is neutral).

Finally (Eq. 5) holds by SPECIAL CASE II. □

This associativity of the operator + can be extended to arbitrary cut-free terms f, t, t' provided the last one, t', satisfies $t' + id \equiv t'$. We shall see later that this is true for all normal or neutral t'.

The next three Lemmas jointly make sure that the binary operator + is compatible with the unary operator *rewrite, i.e.* from $rewrite(f) \equiv rewrite(f')$ and $rewrite(t) \equiv rewrite(t')$ follows $rewrite(f+t) \equiv rewrite(f'+t')$ for arbitrary cut-free f, f', t and t'.

Lemma 6. *Suppose* $f : D \underset{Cf}{\longrightarrow} C, f' : D \underset{Cf}{\longrightarrow} C, t : X \underset{Cf}{\longrightarrow} D$ *and* $t' : X \underset{Cf}{\longrightarrow} D$, *then*

(1) $rewrite(f + t) \equiv rewrite(f) + t$

(1') $rewrite(f) \equiv rewrite(f')$ *implies* $rewrite(f + t) \equiv rewrite(f' + t)$.

(2) $rewrite(t) \equiv rewrite(t')$ *implies* $rewrite(f + t) \equiv rewrite(f + t')$

(3) $rewrite(f + id) \equiv rewrite(f)$

Corollary. *If f is normal or a generalised neutral, then* $f + id \equiv f$.

Proof of Corollary. If f is normal, then so is $f + id$. As *rewrite* leaves normal terms invariant, the identity $f + id \equiv f$ follows at once from Lemma 6, (3). If f is a generalised neutral, use induction on the derivation of f. □

Proof of Lemma 6. Notice that (1) implies (1'). We prove the properties (1), (2) and (3) simultaneously by induction on $n = \max\{d(t), d(t')\}$.

Let n be given, assume that (1), (2) and (3) hold for all cut-free g and all cut-free s and s' such that $d(s) < n$ and $d(s') < n$.

(1) is proved by induction on the codomain C of f:

For atomic C the property is trivial. If $C \equiv A \times B$, the cases where f is a pair or where f and $f+t$ are both neutral, are straightforward. Remains the case where f is neutral, but $f+t$ is a pair $\langle g_1, g_2 \rangle$. From this follows immediately

$$(fst \circ f) + t \equiv g_1$$
$$(snd \circ f) + t \equiv g_2.$$

Therefore,

$$\begin{aligned}
rewrite_{A\times B}(f+t) &\equiv \langle\ rewrite_A(g_1),\ rewrite_B(g_2)\rangle \\
&\equiv \langle\ rewrite_A((fst \circ f)+t), \\
&\qquad rewrite_B((snd \circ f)+t)\rangle \\
&\equiv \langle (rewrite_A(fst \circ f)) + t, \\
&\qquad rewrite_B(snd \circ f)) + t\rangle && \text{property of } A \text{ and } B \\
&\equiv \langle\ rewrite(fst \circ f), \\
&\qquad rewrite_B(snd \circ f)\rangle + t && \text{definition of } + \\
&\equiv rewrite_{A\times B}(f) + t
\end{aligned}$$

Finally, suppose $C \equiv B^A$.

If f is an abstraction, say $f \equiv g^*$ where $g : D \times A \to B$, then

$$\begin{aligned}
rewrite_{B^A}(f+t) &\equiv rewrite_{B^A}((g\ + \langle t^{+fst}, snd \circ id\rangle)^*) \\
&\equiv (rewrite_B(g\ + \langle t^{+fst}, snd \circ id\rangle))^* && \text{definition of } rewrite \\
&\equiv (rewrite_B(g) + \langle t^{+fst},\ snd \circ id\rangle)^* && \text{property of } B \\
&\equiv (rewrite_B(g))^* + t && \text{definition of } + \\
&\equiv rewrite_{B^A}(f) + t && \text{definition of } rewrite
\end{aligned}$$

If f is not an abstraction, hence a neutral, then

$$\begin{aligned}
rewrite_{B^A}(f) + t &\equiv (rewrite_B(ev\ \circ \langle f^{+fst}, rewrite_A(snd \circ id)\rangle))^* + t \\
&\equiv (rewrite_B(ev\ \circ \langle f^{+fst}, rewrite_A(snd \circ id)\rangle) \\
&\qquad + \langle t + fst \circ id,\ snd \circ id\rangle)^* && \text{definition of } + \\
&\equiv (rewrite_B(ev\ \circ \langle f^{+fst}, rewrite_A(snd \circ id)\rangle \\
&\qquad + \langle t + fst \circ id,\ snd \circ id\rangle))^* && \text{property of } B
\end{aligned}$$

By the definition of $+$, this last term is computed in different ways, according to whether $f^{+fst} + \langle t + fst \circ id,\ snd \circ id\rangle$ is an abstraction or not. But

$$
\begin{aligned}
f^{+fst} + \langle t + fst \circ id,\ snd \circ id\rangle &\equiv f + (fst \circ id \\
&\quad + \langle t + fst \circ id,\ snd \circ id\rangle) && \text{associativity} \\
&\equiv f + (t + fst \circ id) && \text{definition of } + \\
&\equiv (f + t) + fst \circ id && \text{associativity}
\end{aligned}
$$

Hence the left-hand term above is an abstraction iff $(f + t)^{+fst}$ is an abstraction iff $f + t$ is an abstraction.

Suppose that $f + t$ is not an abstraction. Then $(f + t)^{+fst}$ is neutral and therefore

$$
\begin{aligned}
& rewrite_{BA}(f) + t \\
&\equiv (rewrite_B(ev \ \circ \langle f^{+fst} + \langle t + fst \circ id,\ snd \circ id\rangle, \\
&\qquad rewrite_A((snd \circ id) \\
&+ \langle t + fst \circ id,\ snd \circ id\rangle)\rangle))^* \\
&\equiv (rewrite_B(ev \ \circ \langle (f + t)^{+fst}, \\
&\qquad rewrite_A(snd \circ id + \langle t + fst \circ id,\ snd \circ id\rangle)\rangle)))^* \quad \text{property of } A \\
&\equiv (rewrite_B(ev \ \circ \langle (f + t)^{+fst},\ snd \circ id\rangle)))^* \\
&\equiv rewrite_{BA}(f + t).
\end{aligned}
$$

Suppose now that $f + t \equiv g^*$. Then,

$$
f^{+fst} + \langle t + fst \circ id,\ snd \circ id\rangle \equiv g^* + fst \circ id \equiv (g \ + \langle fst \circ id^{+fst},\ snd \circ id\rangle)^*
$$

Hence

$$\begin{array}{ll}
rewrite^A_B(f) + t & \\
\quad \equiv (rewrite_B(ev \ \circ \langle f^{+fst},\ rewrite_A(snd \circ id)\rangle & \\
\qquad +\langle t + fst \circ id,\ snd \circ id\rangle))^* & \\
\quad \equiv (rewrite_B((g \ + \langle fst \circ id^{+fst},\ snd \circ id\rangle) & \\
\qquad +\langle id, rewrite_A(snd \circ id)\rangle))^* & \text{definition of } + \\
\quad \equiv (rewrite_B(g \ + \langle fst \circ id + id,\ rewrite_A(snd \circ id)\rangle))^* & \text{associat., def. of } + \\
\quad \equiv (rewrite_B(g \ + \langle fst \circ id, rewrite_A(snd \circ id)\rangle))^* & \text{definition of } + \\
\quad \equiv (rewrite_B(g \ + id))^* & \text{property (2)} \\
\quad \equiv (rewrite_B(g))^* & \text{property (3)} \\
\quad \equiv (rewrite_{B^A}(g*) & \text{definition of } rewrite \\
\quad \equiv (rewrite_{B^A}(f + t) & \text{choice of } g.
\end{array}$$

Indeed, the cut-free term $s \equiv \langle fst \circ id,\ rewrite_A(snd \circ id)\rangle$ has degree $d(s) \leq c(A) < c(B^A) = d(f + t) \leq n$, hence, by induction hypothesis, properties (2) and (3) hold for s and id. Hence (1) holds for all f and all t such that $d(t) = n$. In particular we have shown that (1) holds for $n = 0$. In that case we do not call on properties (2) and (3), as $f + t$ is neutral, if f is neutral.

(2) is established by induction on the derivation of f: Suppose $rewrite(t) \equiv rewrite(t')$. From this follows that $rewrite(\langle t^{+fst},\ snd{\circ}id\rangle) \equiv rewrite(\langle t'^{+fst},\ snd \circ id\rangle)$ by definition of *rewrite* and (1′).

Now for example, consider the case where

$$f \equiv g^*.$$

Then,

$$\begin{array}{lll}
rewrite(g^* + t) & \equiv (rewrite(g \ + \langle t^{+fst},\ snd \circ id\rangle))^* & \text{definition of } + \\
& \equiv (rewrite(g \ + \langle t'^{+fst},\ snd \circ id\rangle))^* & \text{property of } g \\
& \equiv rewrite(g^* + t') & \text{def. of } rewrite\text{, def. of } +.
\end{array}$$

As another example, consider the step where

$$f \equiv ev \ \circ \langle g, a\rangle$$

with g neutral and a normal. Suppose that the property holds for g and a. While computing $f + t$, three cases must be treated separately:

- $g \ + t$ and $g \ + t'$ are both neutral. By Lemma 2, this is always true, if $n = 0$. First, treat the case where the codomain of g is not ambiguous. As $rewrite(g \ + t) \equiv rewrite(g \ + t')$ by property of g, it follows by Remark 3) that $g \ + t \equiv g \ + t'$.

Hence,

$$
\begin{aligned}
&rewrite\ (ev\ \circ \langle g, a\rangle + t) \\
&\qquad \equiv rewrite(ev\ \circ \langle g\ + t, a + t\rangle) && \text{definition of } + \\
&\qquad \equiv rewrite(ev\ \circ \langle g\ + t, rewrite(a + t)\rangle) && \text{normality of } a + t \\
&\qquad \equiv rewrite(ev\ \circ \langle g\ + t, rewrite(a + t')\rangle) && \text{property of } a \\
&\qquad \equiv rewrite(ev\ \circ \langle g\ + t', rewrite(a + t')\rangle) && \text{Remark 3)} \\
&\qquad \equiv rewrite(ev\ \circ \langle g\ + t', a + t'\rangle) && \text{normality of } a + t' \\
&\qquad \equiv rewrite(ev\ \circ \langle g, a\rangle + t'). && \text{definition of } +.
\end{aligned}
$$

On the other hand, if the codomain B^A of g is ambiguous, then B, the common codomain of $ev\ \circ \langle g, a\rangle + t$ and $ev\ \circ \langle g, a\rangle + t'$, is also ambiguous. Use Remark 4) to conclude that

$$rewrite(ev\ \circ \langle g, a\rangle + t) \equiv rewrite\ (ev\ \circ \langle g, a\rangle + t').$$

This shows already that (2) holds whenever $n = 0$.

- Both $g\ + t$ and $g\ + t'$ are abstractions, say $g\ + t \equiv h^*$ and $g\ + t' \equiv h'^*$. This case can only happen, if $n > 0$. By property of g, $rewrite(g\ + t) \equiv rewrite(g\ + t')$, hence $rewrite(h) \equiv rewrite(h')$. Recall from Lemma 2 that

$$d(\langle id, a + t\rangle) < n \text{ and } d(\langle id, a + t'\rangle) < n.$$

From the normality of a and (1) also follows

$$(*)\ a + t \equiv rewrite(a + t) \equiv rewrite(a + t') \equiv a + t'$$

Then,

$$
\begin{aligned}
rewrite(f + t) &\equiv rewrite(ev\ \circ \langle g, a\rangle + t) \\
&\equiv rewrite(h\ + \langle id, a + t\rangle) && \text{definition of } + \\
&\equiv rewrite(h' + \langle id, a + t\rangle) && (1') \\
&\equiv rewrite(h' + \langle id, a + t'\rangle) && (*) \\
&\equiv rewrite(f + t').
\end{aligned}
$$

- One is neutral, the other one is an abstraction, say $g\ + t$ is neutral and $g\ + t' \equiv h'^*$. In this case too, n must be strictly greater than 0. Again we have $d(\langle id, a + t'\rangle) < n$. Properties (1), (2) and (3) hold for $\langle id, a + t'\rangle$ by induction hypothesis. Then

$$(**)\ rewrite\ (snd \circ id) + \langle id, a + t'\rangle \equiv a + t.$$

Indeed,

$$\begin{aligned} rewrite(snd \circ id) + \langle id, a + t' \rangle &\equiv rewrite(snd \circ id + \langle id, a + t' \rangle) && (1) \\ &\equiv rewrite(a + t') && \text{definition of } + \\ &\equiv a + t && (*). \end{aligned}$$

Therefore,

$$\begin{aligned} rewrite(f + t) &\equiv rewrite(ev \ \circ \langle g \ + t, a + t \rangle) \\ &\equiv rewrite(ev \ \circ \langle g \ + t, a + t \rangle + id) && (3) \\ &\equiv rewrite(ev \ \circ \langle g \ + t) + id, (a + t) + id \rangle) && \text{definition of } + \\ &\equiv rewrite(ev \ \circ \langle (g \ + t) + id, a + t \rangle && (3), \text{ Corollary} \\ &\equiv rewrite(ev \ \circ \langle (g \ + t)^{+fst}, rewrite(snd \circ id) \rangle \\ &\quad + \langle id, a + t' \rangle && \text{def. of } +, (**) \end{aligned}$$

whereas,

$$f + t' \qquad \equiv h' + \langle id, a + t' \rangle \qquad \text{choice of } h'.$$

By property of g, we have $rewrite(g \ + t) \equiv rewrite(g \ + t')$, hence

$$(rewrite(ev \ \circ \langle (g \ + t)^{+fst}, rewrite(snd \circ id) \rangle))^* \equiv rewrite(h'^*).$$

As, $rewrite(h'^*) \equiv (rewrite(h'))^*$, this implies

$$rewrite(ev \ \circ \langle (g \ + t)^{+fst}, rewrite(snd \circ id) \rangle) \equiv rewrite(h').$$

Then,

$$\begin{aligned} rewrite(f + t) &\equiv rewrite \ (ev \ \circ \langle (g \ + t)^{+fst}, \\ &\qquad rewrite \ (snd \circ id) \rangle + \langle id, a + t' \rangle) \\ &\equiv rewrite(h' + \langle id, a + t' \rangle) && \text{prop. of } \langle id, a + t' \rangle \\ &\equiv rewrite(f + t'). \end{aligned}$$

This terminates the last case of (2). Hence, property (2) holds for all f, t and t' such that max $(d(t), d(t')) \leq n$.

(3) Let $f : D \underset{Cf}{\longrightarrow} C$. To show that $rewrite(f + id) \equiv rewrite(f)$ use induction on f.

For example, suppose that f is of the form g^* where the property holds for g. First, notice that

$$rewrite(id) \equiv rewrite(\langle fst \circ id, \ snd \circ id \rangle)$$

and both id and $\langle fst \circ id,\ snd \circ id \rangle$ have degree 0. Hence (2) holds for them.
Therefore

$$\begin{array}{lll} rewrite(g^* + id) & \equiv rewrite((g \ + \langle id + fst \circ id,\ snd \circ id \rangle)^*) & \text{definition of } + \\ & \equiv (rewrite(g \ + \langle id + fst \circ id,\ snd \circ id \rangle))^* & \text{definition of } rewrite \\ & \equiv (rewrite(g \ + \langle fst \circ id,\ snd \circ id \rangle))^* & \text{definition of } + \\ & \equiv (rewrite(g \ + id))^* & (2) \\ & \equiv (rewrite(g))^* & \text{property of } g \\ & \equiv rewrite(g^*) & \text{definition of } rewrite. \end{array}$$

This terminates the proof of Lemma 6. N.B. The operator *rewrite* is essential here: $id^* + id$ is not the same as id^*! □

Lemma 7. *For all cut-free* $f : D \underset{Cf}{\longrightarrow} C$

$$[fst](f) \equiv (fst \circ id) + f$$

$$[snd](f) \equiv (snd \circ id) + f.$$

Lemma 8. *Let* $f : D \to C$ *and* $t : X \to D$ *be cut-free. Then* $[f](t) \equiv f + t$.

Proof by induction on (n, m) *as in Lemma 2. For given* $n = d(t)$*, use introduction on* f. For example, suppose that $f \equiv fst \circ g$ with g neutral, and that the property holds for g. Then

$$\begin{array}{lll} [fst \circ g](t) & \equiv [fst]([g](t)) & \\ & \equiv [fst](g \ + t) & \text{property of } g \\ & \equiv fst \circ id + (g \ + t) & \text{Lemma 7} \\ & \equiv (fst \circ id + g) + t & \text{associativity} \\ & \equiv fst \circ g \ + t & \text{definition of } +. \end{array}$$

The case where $f \equiv ev \circ < g, a >$ with g neutral, a normal, is as straightforward as the preceding one:
Suppose that g and a have the property. If $g \ + t$ is neutral, then

$$\begin{array}{lll} [ev \ \circ \langle g, a \rangle](t) & \equiv [ev](\langle [g](t), [a](t) \rangle) & \\ & \equiv [ev](\langle g \ + t, a + t \rangle) & \text{property of } g \text{ and } a \\ & \equiv ev \ \circ \langle g \ + t, rewrite(a + t) \rangle & \text{definition of } [\ ] \\ & \equiv ev \ \circ \langle g \ + t, a + t \rangle & \text{normality of } a \\ & \equiv ev \ \circ \langle g, a \rangle + t & \text{definition of } +. \end{array}$$

If $g + t \equiv h^*$, then

$$
\begin{aligned}
[ev \circ \langle g, a\rangle](t) &\equiv [h](\langle id, a+t\rangle) \\
&\equiv h + \langle id, a+t\rangle \quad \text{as } d(\langle id, a+t\rangle) < n \\
&\equiv f + t
\end{aligned}
$$

□

Corollary. *If f is normal, then*

1. $nf(f) \equiv rewrite([f](id)) \equiv f$.
2. *If* $nf(f) \equiv nf(g)$, *then* $f = g : D \to C$ *in* **T**.

Lemma 9. *Let $h : D \to C$ be any denoting term, $f : X \underset{Cf}{\longrightarrow} D$ be cut-free and $t : Y \to X$ a generalised projection. Then*

$$[h](f+t) \equiv ([h](f)) + t.$$

Proof by induction on h. In the cases where $h \equiv fst$ or $h \equiv snd$, use Lemma 7 and associativity. If $h \equiv id$ or ter the property is immediate.

Suppose $h \equiv ev$. Three subcases arise.

Case 1: $f \equiv \langle g^*, a\rangle$. Then $f + t \equiv \langle (g + \langle t + fst \circ id,\ snd \circ id\rangle)^*, a + t\rangle$.

Therefore

$$
\begin{aligned}
[ev](f+t) &\equiv (g + \langle t + fst \circ id,\ snd \circ id\rangle) + \langle id, a+t\rangle && \text{definition of } [\,] \\
&\equiv g + (\langle t + fst \circ id,\ snd \circ id\rangle + \langle id, a+t\rangle) && \text{associativity} \\
&\equiv g + \langle (t + fst \circ id) + \langle id, a+t\rangle, \\
&\qquad snd \circ id + \langle id, a+t\rangle\rangle && \text{definition of } + \\
&\equiv g + \langle t + id, a+t\rangle && \text{associat., def.of } + \\
&\equiv g + \langle t, a+t\rangle && \text{Lemma 1, } (v) \\
&\equiv g + (\langle id, a\rangle + t) && \text{definition of } + \\
&\equiv (g + \langle id, a\rangle) + t && \text{associativity} \\
&\equiv ([ev](f)) + t && \text{definition of } +.
\end{aligned}
$$

Case 2: $f \equiv \langle g, a\rangle$ with g neutral. Then $g + t$ is also neutral by Lemma 1, i).

Whence,

$$
\begin{aligned}
[ev](f+t) &\equiv [ev](\langle g, a\rangle + t) \\
&\equiv [ev](\langle g + t, a+t\rangle) && \text{definition of } + \\
&\equiv ev \circ \langle g + t, rewrite(a+t)\rangle && \text{definition of } [\,] \\
&\equiv ev \circ \langle g + t, rewrite(a) + t\rangle && \text{Lemma 6} \\
&\equiv ev \circ \langle g, rewrite(a)\rangle + t && \text{definition of } + \\
&\equiv [ev](f) + t.
\end{aligned}
$$

Case 3: f is neutral. Then so is $f + t$.

Therefore,

$$\begin{aligned}
[ev](f+t) &\equiv ev \circ \langle fst \circ (f+t), rewrite(snd \circ (f+t))\rangle && \text{definition of } [\,] \\
&\equiv ev \circ \langle (fst \circ f) + t, rewrite((snd \circ f) + t)\rangle && \text{definition of } + \\
&\equiv ev \circ \langle (fst \circ f) + t, rewrite(snd \circ f) + t\rangle && \text{Lemma 6} \\
&\equiv ev \circ \langle fst \circ f, rewrite(snd \circ f)\rangle + t && \text{definition of } + \\
&\equiv [ev](f) + t.
\end{aligned}$$

The other steps are straightforward and are left to the reader. □

Corollary.

(i) *from* $rewrite(t) \equiv rewrite(t')$ *follows* $rewrite([h](f+t)) \equiv rewrite([h](f+t'))$.

(ii) *In particular,* $rewrite([h](id)) \equiv rewrite([h](\langle fst \circ id,\ snd \circ id\rangle)$

(iii) $[h](id) + id \equiv [h](id)$ *for arbitrary* h.

Indeed, (i) follows from Lemma 9 by Lemma 6, (1). (ii) is obtained as follows:

$$\begin{aligned}
rewrite([h](id)) &\equiv rewrite([h](id + id)) && \textit{definition of } + \\
&\equiv rewrite([h](id + \langle fst \circ id,\ snd \circ id\rangle)) && \textit{(i)} \\
&\equiv rewrite([h](\langle fst \circ id,\ snd \circ id\rangle)) && \textit{definition of } +.
\end{aligned}$$

□

End of Proof of Theorem 1. Remains the last of the five properties constituting Theorem 1: provably equal terms have identical normal forms, *i.e.* whenever the equality $h = g$ holds in **T**, then $nf(h) \equiv nf(g)$.

Use induction on the derivation of the equality: the logical rules are easily checked. For example, when checking the compatibility of composition with equality, we must prove that from $nf(g) \equiv nf(g')$ and $nf(f) \equiv nf(f')$ follows $nf(g \circ f) \equiv nf(g' \circ f')$. Assume $nf(g) \equiv nf(g')$ and $nf(f) \equiv nf(f')$ and let $h \equiv [f](id), h' \equiv [f'](id)$.

Then $rewrite(h) \equiv rewrite(h')$ by induction hypothesis.

Hence,

$$\begin{aligned}
nf(g \circ f) &\equiv rewrite([g](h)) \\
&\equiv rewrite([g](id + h)) && \text{definition of } + \\
&\equiv rewrite([g](id + h')) && \text{Cor. of Lemma 9} \\
&\equiv rewrite([g](id)) + h' && \text{Lemma 6, (1)} \\
&\equiv rewrite([g'](id)) + h' && \text{ind. hypothesis} \\
&\equiv rewrite([g'](h')) && \text{Lemma 9} \\
&\equiv nf(g' \circ f').
\end{aligned}$$

The non-logical rules are also straight forward. Consider for example the equality

$$ev \circ \langle h^* \circ fst,\ snd\rangle = h : D \times A \to B.$$

Then, $[ev \circ \langle h^* \circ fst,\ snd\rangle](id)$

$$\begin{aligned}
&\equiv [ev](\langle [h^*](fst \circ id),\ snd \circ id\rangle)\\
&\equiv [ev](\langle ([h](\langle (fst \circ id)^{+fst},\ snd \circ id\rangle))^*, snd \circ id\rangle)\\
&\equiv [h](\langle (fst \circ id)^{+fst},\ snd \circ id\rangle + \langle id,\ snd \circ id\rangle) \qquad \text{Lemma 9}\\
&\equiv [h](\langle fst \circ id,\ snd \circ id\rangle)
\end{aligned}$$

Hence, $rewrite([ev \circ \langle h^* \circ fst,\ snd\rangle](id))$

$$\begin{aligned}
&\equiv rewrite([h](\langle fst \circ id,\ snd \circ id\rangle))\\
&\equiv rewrite([h](id)) \qquad \text{Cor. of Lemma 9}
\end{aligned}$$

Somewhat more involved is the equality

$$(ev \circ \langle h \circ fst,\ snd\rangle)^* = h : D \to B^A$$

We have

$$[(ev \circ \langle h \circ fst,\ snd\rangle)^*](id)$$

$$\begin{aligned}
&\equiv ([ev](\langle [h][fst](\langle id^{+fst},\ snd \circ id\rangle), [snd](\langle id^{+fst},\ snd \circ id\rangle)\rangle))^*\\
&\equiv ([ev](\langle [h](id^{+fst}),\ snd \circ id\rangle))^*\\
&\equiv ([ev](\langle ([h](id))^{+fst},\ snd \circ id\rangle))^* \qquad \text{Lemma 9}
\end{aligned}$$

Let $f \equiv [h](id)$

<u>Case 1</u>: f is neutral. Then, by the above,

$$rewrite([ev \circ \langle h \circ fst,\ snd\rangle)^*](id)) \equiv rewrite(([ev](\langle f^{+fst},\ snd \circ id\rangle))^*)$$

$$\begin{aligned}
&\equiv (rewrite([ev](\langle f^{+fst},\ snd \circ id\rangle)))^*\\
&\equiv (rewrite(ev \circ \langle f^{+fst}, rewrite(snd \circ id)\rangle))^* \qquad \text{neutrality of } f^{+fst}\\
&\equiv rewrite(f)\\
&\equiv rewrite([h](id)).
\end{aligned}$$

<u>Case 2</u>: $f \equiv g'^*$. Then $f^{+fst} \equiv (g + \langle (fst \circ id)^{+fst},\ snd \circ id\rangle)^*$ and therefore

$$rewrite([(ev \circ \langle h \circ fst,\ snd\rangle)^*](id)) \equiv rewrite(([ev](\langle f^{+fst},\ snd \circ id\rangle))^*)$$

$\equiv rewrite(([ev](\langle(g\ +\langle(fst\circ id)^{+fst},\ snd\circ id\rangle)^*,\ snd\circ id\langle\rangle)^*)$

$\equiv rewrite(((g\ +\langle(fst\circ id)^{+fst},\ snd\circ id\rangle)+\langle id,\ snd\circ id\rangle)^*)$

$\equiv rewrite((g\ +\langle fst\circ id,\ snd\circ id\rangle)^*)$ associativity, definition of +

$\equiv (rewrite(g\ +id))^*$ Lemma 6

$\equiv (rewrite(g))^*$ Lemma 6

$\equiv rewrite(g^*)$

$\equiv rewrite([h](id))$.

□

3.4. Functional completeness and conservativity of extensions

Functional completeness has been formulated and established by Lambek for extensions obtained by adding variables of domain 1. It says the following: let $x : 1 \to A$ be a variable. Then for every denoting arrow-term $t : B \to C$ with eventual occurrences of x, there is a denoting arrow-term $\kappa_{x\in A}t : A \times B \to C$ without occurrences of x such that

$$t = (\kappa_{x\in A}t)\circ < x\circ terB, idB > \text{ holds in } \mathbf{T}[x : 1 \to A].$$

The definition of $\kappa_{x\in A}t : A \times B \to C$ and the corresponding proofs from [7] are immediately transferable into our setting:

Fact 1: Let $s : D \to C$, in $\mathbf{T}[x : 1 \to A]$ and suppose that t has no occurrences of x. Then

$$\kappa_{x\in A}t = t\circ snd(A, B) : A \times B \to C \text{ in } \mathbf{T}$$

$$\kappa_{x\in A}(t\circ s) = t\circ \kappa_{x\in A}(s) \text{ in } \mathbf{T}[x : 1 \to A].$$

Fact 2: From $f = g : B \to C$ in $\mathbf{T}[x : 1 \to A]$ follows $\kappa_{x\in A}f = \kappa_{x\in A}g : A\times B \to C$ in $\mathbf{T}$.

Proposition. $\mathbf{T}[x : 1 \to A]$ *is a conservative extension of* $\mathbf{T}$ *iff*

$$f\circ snd(A, B) = g\circ snd(A, B) : A \times B \to C \text{ in } \mathbf{T} \text{ implies } f = g : B \to C \text{ in } \mathbf{T}$$

(the second projections are epimorphisms in the associated free category).

Proof. Suppose that the second projection $snd(A, B)$ is an epimorphism. Let $f = g : B \to C$ be an equality derivable in $\mathbf{T}[x : 1 \to A]$ where x has no occurrences in this equality. Using functional completeness we get

$$\kappa_{x\in A}f = \kappa_{x\in A}g : A \times B \to C \text{ in } \mathbf{T}.$$

Therefore

$$f \circ snd(A, B) = g \circ snd(A, B) : A \times B \to C \text{ in } \mathbf{T},$$

by the Fact 1 cited above. As we can cancel $snd(A, B)$ on the right, it follows that $f = g : B \to C$ in $\mathbf{T}$.

For the converse, compose $f \circ snd(A, B) = g \circ snd(A, B) : A \times B \to C$ with $< x \circ terB, idB >$ on the right to obtain $f = g : B \to C$ first in $\mathbf{T}[x : 1 \to A]$ and then, by conservativity, in $\mathbf{T}$. □

If for every object constant C of $\mathbf{T}$ there is an arrow-constant $a : 1 \to C$ in $\mathbf{T}$, then for every object- term A of $\mathbf{T}$, the theory $\mathbf{T}[x : 1 \to A]$ is a conservative extension of $\mathbf{T}$. Indeed, every object-term A of $\mathbf{T}$ is inhabited, *i.e.* there is an arrow-term $a : 1 \to A$. Now replace x by a in the argument above to show that the projections are epimorphisms. In general however, if there are non inhabited objects, a more involved proof seems necessary. Cubric proposes one *via* the λ-calculus in [3], we use the operator $+$ on normal terms to show that the first and second projections are epic. In fact, this property can be obtained for a somewhat bigger set of morphisms, the so-called weakening morphisms which we define next.

Definition 10. $fst \circ id$ and $snd \circ id$ are *timid weakening morphisms*, if w is a *timid weakening morphism*, then so is $\langle w^{+fst},\ snd \circ id\rangle$. Every timid weakening morphism is a *weakening morphism.* If $\langle w_1, w_2\rangle$ is a *weakening morphism*, then so are w_1 and w_2.

Notice that the domain of a weakening morphism w is a repeated product where parentheses are grouped to the left. Moreover, the angle brackets of w are also grouped to the left. Timid weakening morphisms forget exactly one of the two leftmost factors of their domain. For example

$$w \equiv \langle fst \circ (fst \circ id),\ snd \circ id\rangle : (A \times B) \times C \to A \times C,$$

$$w' \equiv \langle w^{+fst},\ snd \circ id\rangle$$

$$\equiv \langle\langle fst \circ (fst \circ (fst \circ id)),\ snd \circ (fst \circ id)\rangle,\ snd \circ id\rangle : ((A \times B) \times C) \times D \to (A \times C) \times D, \text{ etc.}$$

By Lemma 1, we know that $f + w$ is normal, if f is normal and that it is a generalised neutral, if f is neutral. However, in general from "f neutral" does not follow "$f + w$ neutral": take $f \equiv \mathrm{id}(A \times C)$ and $w \equiv \langle fst \circ (fst \circ id), snd \circ id\rangle : (A \times B) \times C \to A \times C$.

Lemma 10. *Let $f : D \to C$ and $g : D \to C$ be both neutral or both normal, and $w : U \to D$ a weakening morphism. Then from $f + w \equiv g + w$ follows $f \equiv g$.*

Proof. Consider first the special case where f and g are simple projections: Let

$$(p_m \circ \ldots \circ p_1 \circ id) + w \equiv (q_n \circ \ldots \circ q_1 \circ id) + w$$

where $m \geq 0$ and $n \geq 0$, and show that $m = n$ and $p_i = q_i$ for $1 \leq i \leq m$. To see this, it suffices to remark that w is either of the form

$$\langle ... \langle b_1, b_2 \rangle, ... b_r \rangle$$

or is reduced to b_i where for $1 \leq i \leq r$, each b_i is a simple projection of the form

$$b_i \equiv v_i \circ (fst \circ ... \circ (fst \circ id)...).$$

In this expression, fst is repeated $k + r - i$ times for a fixed $k \geq 0$ which depends only on w, v_1 is the first or the second projection and v_i is the second projection for $i \geq 2$. Therefore $f + \langle ... \langle b_1, b_2 \rangle, ... b_r \rangle$ is either one of the subterms $\langle ... \langle b_1, b_2 \rangle, ... b_i \rangle$, for $1 \leq i \leq r$, or is of the form $p_m \circ ... \circ p_l \circ b_i$. Moreover, this result determines the simple projection f uniquely.

To prove the general case, remark that if f is a neutral which is not a simple projection, then $f + w$ must also be a neutral with a derivation terminating by the same rule as that of f. Now use induction on the derivation of $f : D \xrightarrow[Ne]{} C$ (respectively. $f : D \xrightarrow[No]{} C$) to show for all w that $f + w \equiv g + w$ implies $f \equiv g$. The induction steps corresponding to the evaluation, terminal, constant, pair or abstraction rule are straight forward. For example, if $f \equiv f'^*$, then $g + w \equiv f + w \equiv (f' + \langle w^{+fst}, snd \circ id \rangle)^*$. As g is normal, it is necessarily an abstraction, say $g \equiv (g')^*$. Hence, $g + w \equiv (g' + \langle w^{+fst}, snd \circ id \rangle)^*$ and we conclude by induction hypothesis on f'.

If f is obtained by the identity rule, then $g + w \equiv f + w \equiv id + w \equiv w$. As w has no occurrences of ev, ter or c, the derivation of g does not use an evaluation, terminal or constant rule. Thus g also is a simple projection. Hence, $f \equiv g$ by the special case. If f is obtained by a projection rule, say $f \equiv fst \circ f'$, then the argument is different according to whether $f' + w$ is a pair or whether it is neutral. In the former case, f' must be a simple projection. Then f is also a simple projection. It follows that $g + w \equiv f + w$ has no occurrences of ev, c or ter, therefore g also must be a simple projection and we are in the special case. If $f' + w$ is a neutral, then $g + w \equiv fst \circ (f' + w)$. So, g must be of the form $v \circ g'$ where v is not c, ev or ter. Hence it must be a first or second projection. If $g' + w$ is a pair, we are back to the special case, if not, we use the induction hypothesis. □

Theorem 2. *Every extension* $\mathbf{T}[x_1 : 1 \to A_1, ..., x_n : 1 \to A_n]$ *of* $\mathbf{T}$ *is conservative over* $\mathbf{T}$.

Proof. It suffices to prove this for $n = 1$. Indeed, we can consider previously added variables as constants of the theory by integrating them into the generating graph.

Assume that $f \circ snd = g \circ snd$ holds in $\mathbf{T}$. By Theorem 1, we also have $nf(f) \circ snd = nf(g) \circ snd$ in $\mathbf{T}$. From Lemma 1 follows that $nf(f) + snd \circ id = nf(g) + snd \circ id$ is provable in $\mathbf{T}$. As both terms in the above equality are normal, we have syntactical equality:

$$nf(f) + (snd \circ id) \equiv nf(g) + (snd \circ id)$$

Then, by Lemma 10,

$$nf(f) \equiv nf(g)$$

and so, again by Theorem 1, the equality $f = g$ is derivable in **T**. The conclusion follows now from the proposition. □

Conclusion

The results above include as particular case a normalisation procedure and the proof of conservativity of extensions of the theory of Cartesian Categories over a given graph. In this case however, there is a much shorter proof where the "action" of an arbitrary term is directly defined on normal terms.

Our syntactical description of normal terms should be useful in the search for a category of sets and maps dual to the free CCC, extending results concerning Cartesian Categories of Došen and Petric' in [6]. A definition of normal forms in the theory of bicartesian closed categories, extending the one given here, would also be interesting. We also can turn the usual techniques the other way round: looking for a reduction-free normalisation of simply typed lambda calculus, we translate this calculus into its associated theory **T** of CCC's and normalise terms in **T**.

References

[1] T. Altenkirch, M. Hofmann and T. Streicher, Categorical reconstruction of a reduction free normalisation proof, preliminary version, P. Dybjer and R. Pollacks, Eds., Proceedings CTCS '95, Springer, *Lecture Notes in Computer Science* **953** (1995) 182-199.

[2] U. Berger and H. Schwichtenberg, An inverse to the evaluation functional for typed λ-calculus, in *Proc. of the 6th Annual IEEE Symposium of Logic in Computer Science* (1991) 203-211.

[3] D. Cubric', Embedding of a free Cartesian Closed Category in the Category of Sets. *J. Pure and Applied Algebra* (to appear).

[4] D. Cubric', P. Dybjer and P. Scott, Normalization and the Yoneda Embedding, *MSCS* **8** (1998) 153-192.

[5] R. Di Cosmo, Isomorphisms of Types, Birkhaeuser (1995).

[6] K. Došen and Z. Petric', The maximality of Cartesian Categories, Rapport Institut de Recherche de Toulouse, CNRS, 97-42-R (1997).

[7] J. Lambek and P.J. Scott, Introduction to Higher Order Categorical Logic, Cambridge University Press (1989).

[8] A. Obtułowicz, Algebra of constructions I. The word problem for partial algebras. *Inform. and Comput.* **73** (1987) 129-173.

Communicated by G. Longo.
Received December, 1997. Accepted March, 1999.