Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 325 (2016) 305–312

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

How to Think of Intersection Types as Cartesian Products

Rick Statman[1]

Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA
USA

Abstract

In their paper "Intersection types and lambda definability" [3] Bucciarelli, Piperno, and Salvo give a mapping of the strongly normalizable untyped terms into the simply typed terms via the assignment of intersection types. Here we shall both generalize their result and provide a converse. We shall do this by retracting untyped terms with surjective pairing onto untyped terms without pairing by using a special variant of Stovring's notion [8] of a symmetric term. The symmetric ones which have simple types with Cartesian products are precisely the ones which retract onto (eta expansions of) strongly normalizable untyped terms. The intersection types of the strongly normalizable terms are related to the simple types with products in that we just replace $\wedge$ by Cartesian product and vice versa.

Keywords: lambda calculus, semigroups, representation

1 Introduction

In their paper "Intersection types and lambda definability" [3] Bucciarelli, Piperno, and Salvo give a mapping of the strongly normalizable untyped terms into the simply typed terms via the assignment of intersection types. Here we shall both generalize their result and provide a converse. We shall do this by retracting untyped terms with surjective pairing onto untyped terms without pairing by using a special variant of Stovring's notion [8] of a symmetric term. The symmetric ones which have simple types with Cartesian products are precisely the ones which retract onto (eta expansions of) strongly normalizable untyped terms. The intersection types of the strongly normalizable terms are related to the simple types with products in that we just replace $\wedge$ by Cartesian product and vice versa.

[1] Email: statman@cs.cmu.edu

http://dx.doi.org/10.1016/j.entcs.2016.09.045
1571-0661/© 2016 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Definition 1.1 Atoms of the language:

- the variables $x, y, z, \ldots$ are atoms, and
- the constants P, L, R are atoms.

Definition 1.2 Terms of the language:

- atoms are terms, and
- if X, Y are terms then so are (XY) and $\lambda x X$.

We shall adopt the customary conventions:

(i) parens are deleted and restored by left association and the use of Church's infixed "dot" notation

(ii) parens are added around abstractions, and additional unary operations, for readability.

The axiom and rules of untyped lambda calculus are the following. The first 5 axioms correspond to the classical theory of untyped lambda calculus with surjective pairing SP.

(beta)	$(\lambda x X)Y$	$= [Y/x]X$	
(eta)	X	$= \lambda x.\, Xx$	x not free in X
(L/Pa)	$L(PXY)$	$= X$	
(R/Pa)	$R(PXY)$	$= Y$	
(P/Dp)	$P(LX)(RX)$	$= X$	

The next 6 axioms correspond to the extended theory of Stovring (FP) and Statman (PSP [7], in the combinator case), which enjoys the Church-Rosser property when formulated by reductions.

(P/Ap)	$PXYZ$	$= P(XZ)(YZ)$
(L/Ap)	LXY	$= L(XY)$
(R/Ap)	RXY	$= R(XY)$
(L/Ab)	$L(\lambda x X)$	$= \lambda x(LX)$
(R/Ab)	$R(\lambda x X)$	$= \lambda x(RX)$
(P/Ab)	$P(\lambda x X)(\lambda x Y)$	$= \lambda x\, PXY$

There are certain useful derived rules:

(1) (P/Dp) and $(P/Ap) \Rightarrow (L/Ap)$ and (R/Ap)

$L(XY) = L(P(LX)(RX)Y) = L(P(LXY)(RXY)) = LXY$

and similarly for R.

(2) (L/Ap) and (R/Ap) and $(P/Dp) \Rightarrow (P/Ap)$

$L(PXYZ) = L(PXY)Z = XZ$ and $R(PXYZ) = R(PXY)Z = YZ$

and therefore

$$PXYZ = P(L(PXYZ))(R(PXYZ))y = P(XZ)(YZ).$$

(3) (eta) and $(P/Ap) \Rightarrow (P/Ab)$

$$\begin{aligned} P(\lambda xX)(\lambda xY) &= \lambda y.\, P(\lambda xX)(\lambda xY)y \\ &= \lambda y.\, P((\lambda xX)y)((\lambda xY)y) \\ &= \lambda x\ PXY. \end{aligned}$$

(4) (eta) and $(L/Ap) \Rightarrow (L/Ab)$

$$L(\lambda xX) = \lambda y.\, L(\lambda xX)y = \lambda y.\, L((\lambda xX)y) = \lambda x(LX)$$

and similarly for R.

(5) (eta) $\Rightarrow LP = K$ and $RP = K^*$

$$L(PX) = \lambda x.\, L(PX)x = \lambda x.\, L(PXx) = \lambda x.\, X,$$

and thus,

$$LP = \lambda x.\, LPx = \lambda x.\, L(Px) = \lambda xy.\, x = K.$$

Similarly

$$R(PX) = \lambda x.\, R(PX)x = \lambda x.\, R(PXx) = \lambda x.\, x,$$

hence

$$RP = \lambda yx.x = K^*.$$

The result of Klop is that the Church-Rosser property fails for the following classical reductions for SP:

(beta)	$(\lambda xX)Y \to [Y/x]X$	
(eta)	$\lambda x.Xx \to X$	x not free in X
(L/Pa)	$L(PXY) \to X$	
(R/Pa)	$R(PXY) \to Y$	
(P/Dp)	$P(LX)(RX) \to X$	

Nevertheless, this theory was proved conservative over beta-eta by de Vrijer [3]. Stovring and, later, Statman (for the combinator case) introduced new reductions for the first 5 and an additional one which enjoy Church-Rosser.

Stovring Reductions for FP with eta:

(beta)	$(\lambda xX)Y \to [Y/x]X$	
(etae)	$X \to \lambda x.\, Xx$	x not free in X
(L/Pa)	$L(PXY) \to X$	
(R/Pa)	$R(PXY) \to Y$	
(P/De)	$X \to P(LX)(RX)$	
(P/Ap)	$PXYZ \to P(XZ)(YZ)$	

Now, Church-Rosser is enough to obtain de Vrijer's theorem and a co-de Vrijer theorem that beta-eta is conservative over pairing with FP. Stovring's argument

uses the notion of symmetric term defined below.

Our basic theory of intersection types is Barendregt, Dekkers, and Ststman ([1], p. 580). There is a simpler theory consisting of $\to I, \to E, \wedge I$, and $\wedge E$. We shall call the later BPS since it is easy to see that is equivalent to the theory S of intersection types defined in [3]. Each intersection type A can be converted into a simple type A' with products by replacing $(B \wedge C)$ with (B^*C). The inverse of ' is ".

Definition 1.3 Simple types with Cartesian products:

Type atoms $p, q, r, \ldots$ are types.

If A and B are types then so are $(A \to B)$ and (A^*B)

We shall employ Curry's substitution prefix both for terms and types. $[B/p]A$ is the result of substituting B for p in A. We adopt Church typing for terms as follows, but it is convenient to adopt the Curry notation $S \vdash X : A$, where S (the base) is a set of declarations $x : B$ for the Church typings . Here S will only contain declarations $x : B$ provided x is indeed a typed variable of type B.

Definition 1.4 Typed terms:

$x^A, y^B, z^C, \ldots$ are typed variables

$\vdash x^A : A, \vdash y^B : B, \vdash z^C : C, \ldots$

We abbreviate by omitting superscripts.

$\vdash X : A, \vdash Y : B \quad \Rightarrow \vdash PXY : (A^*B)$

$\vdash X : (A^*B) \quad \Rightarrow \vdash LX : A, \vdash RX : B$

$\vdash X : B \quad \Rightarrow \vdash \lambda x^A X : A \to B$

$\vdash X : A \to B, \vdash Y : A \quad \Rightarrow \vdash (XY) : B$

$\vdash X : A \to B, \vdash Y : A \quad \Rightarrow \vdash \langle XY \rangle : B$

$\vdash \langle (LX)Y \rangle : B, \vdash \langle (RX)Y \rangle : C \Rightarrow \vdash \langle XY \rangle : B^*C$

Here $\langle \Rightarrow \rangle$ simply represents the pointwise action of a pair on an element in the common domain of the coordinates. We could define $\langle \Rightarrow \rangle$ as follows:

$\vdash X : A \to B, \vdash Y : A \quad \to \langle XY \rangle := (XY)$

$\vdash \langle (LX)Y \rangle : B, \vdash \langle (RX)Y \rangle : C \to \langle XY \rangle := P\langle (LX)Y \rangle \langle (RX)Y \rangle$

but we prefer to introduce $\langle\rangle$ as a primitive with reduction rules:

$(\text{beta})(\lambda x.XY) \quad \to [Y/x]X$

$(L/Pa)\ L(PXY) \quad \to X$

$(R/Pa)\ R(PXY) \quad \to Y$

$(P/Dp)\ P(LX)(RX) \to X$

$(P/<>)\ \langle (PXY)Z \rangle \to P\langle XZ \rangle \langle YZ \rangle$

$(\text{zeta})\ \langle XY \rangle \quad \to (XY)$ if $\vdash X : A \to B. \vdash Y : A$

Facts:

(1) subject reduction

(2) weak diamond

(3) strong normalization

(4) Church-Rosser

Definition 1.5 Faces $f(X)$ of an untyped term X:

$$\begin{aligned} f(x) &= \{x\} \\ f(\lambda x.\, X) &= \{\lambda x.\, Z \mid Z : f(X)\} \\ f(PXY) &= f(X) \cup f(Y) \\ f(LX) &= f(X) \\ f(RX) &= f(X) \\ f(XY) &= \{Z'Z" \mid Z' : f(X) \text{ and } Z" : F(Y)\} \end{aligned}$$

Definition 1.6 $\sim$ regular term for a binary relation $\sim$:

X is $\sim$ regular if for any subterm PYZ of X and $U : f(Y)$, $V : f(Z)$ we have $U \sim V$

Examples of $\sim$:

Example 1.7 alpha conversion

Example 1.8 eta conversion

Example 1.9 beta-eta conversion

Stovring's notion of symmetric is the same as beta-eta regular. Here we note that the condition $U : f(X),\ V : f(X) \Rightarrow U \sim V$ is not sufficient to ensure that X is $\sim$ regular already for $\sim$= eta.

Example 1.10

$$\lambda y.\, Px((\lambda z.\, x)y)$$

has only one face $\lambda z.\, x$, modulo eta, but $P\, x((\lambda z.\, x)y)$ has two. The faces of a term typed with simple types and Cartesian products are obtained by erasing the typing, changing $\langle\rangle$ to ( ) and computing f of the result.

Lemma 1.11 *If, in BCD, $A \leq B$ then there exist an eta regular x' such that in simple types with Cartesian products we have $x : A' \vdash x' : B'$ and any face of x' eta reduces to x.*

Proof. By induction on the length of a BCD derivation of A [B.
Basis:
Case 1: (refl.) We set $x' := x$

$$x : A \vdash x : A$$

Case 2: (inc_L). We set $x' := Lx$

$$x : A^*B \vdash x : A^*B$$
$$x : A^*B \vdash Lx : A$$

Case 3: (inc_R). We set $x' := Rx$

$$x : A^*B \vdash x : A^*B$$
$$x : A^*B \vdash Rx : B$$

Case 4: ($\to \wedge$). We set $x' := \lambda y.\, \langle xy \rangle$

$$x : (A \to B)^*(A \to C) \vdash x : (A \to B)^*(A \to C)$$
$$x : (A \to B)^*(A \to C) \vdash Lx : A \to B$$
$$x : (A \to B)^*(A \to C) \vdash Rx : A \to C$$
$$y : A \vdash y : A$$
$$x : (A \to B)^*(A \to C), y : A \vdash \langle Lxy \rangle : B$$
$$x : (A \to B)^*(A \to C), y : A \vdash \langle Rxy \rangle : C$$
$$x : (A \to B)^*(A \to C), y : A \vdash \langle xy \rangle : B^*C$$
$$x : (A \to B)^*(A \to C) \vdash \lambda y.\, \langle xy \rangle : A \to (B^*C)$$

Induction step:

Case 1: (glb) By induction hypothesis we may assume that we have $x : C' \vdash x'1 : A'$ and $x : C' \vdash x'2 : B'$ in simple types with surjective pairing. Thus, we can set $x' := P(x'1)(x'2)$.

Case 2: (trans) By induction hypothesis we may assume that we have $y : B' \vdash y'1 : C'$ and $x : A' \vdash x'2 : B'$ in simple types with surjective pairing. Thus we can set $[x'2/y](y'1)$

Case 3: ($\to$) By induction hypothesis we may assume that we have $y : C' \vdash y'1 : A'$ and $z : B' \vdash z'2 : D$ in simple types with surjective pairing. Then we can set $x' := \lambda y.\, (x(y'1))'2$ and we have

$$x : A' \to B', y : C' \vdash x(y'1) : B'$$
$$x(y'1) : B' \vdash (x(y'1))'2 : D'$$
$$x : A' \to B', y : C' \vdash (x(y'1))'2 : D'$$
$$x : A' \to B' \vdash \lambda y.\, (x(y'1))'2 : C' \to D'$$

□

Theorem 1.12 *If in BCD we have $S \vdash X : A$ then there is eta regular X' such that in simple types with Cartesian products $S' \vdash X' : A'$ and for any face U of X', U eta reduces to X.*

Proof. By induction on the length of a BCD derivation of $S \vdash X : A$.

Basis: (Ax) This case is trivial.

Induction step:

Case 1: The derivation ends in the [rule. This is the content of Lemma 1.

Case 2: The derivation ends in $\wedge E, \to I$, or $\to E$. This follows directly the induction hypothesis.

Case 3: The derivation ends in $\wedge I$. So in BCD we have $S \vdash X : A$ and $S \vdash X : B$. By induction hypothesis we have $S' \vdash X'1 : A'$ and $S' \vdash X'2 : B'$. Thus $S' \vdash P(x'1)(x'2) : (A \wedge B)'$ and $P(x'1)(x'2)$ is eta regular. □

Theorem 1.13 *If X is eta regular and in simple types with Cartesian products we have $S \vdash X : A$ then for any face U of X there exists an eta reduct U" such that we have in BCD S" $\vdash U$" $: A$".*

Proof. By induction on the length of a typing derivation of X.
Basis: $X = x$. Obvious.

Case 1: $X = PYZ : A^*B$. Y and Z are eta regular, and the induction hypothesis applies to them. Thus, for any $U : f(Y)$ and $V : f(Z)$ there exist eta reducts U" and V" respectively such that in BCD, S" $\vdash U$" $: A$" and S" $\vdash V$" $: B$". Now U" $\sim V$" so by subject reduction and Church-Rosser for eta there exists W" such that U" eta reduces to W" eta expands to V" and S" $\vdash W$" $: A$" $\wedge B$" in BCD.

Case 2: $X = LY$ or RY. By induction hypothesis and $\wedge E$.

Case 3: $X = \lambda y.\, Y : A \to B$. Now the induction hypothesis applies to $S \cup \{y : A\} \vdash Y : B$ so S" $\cup \{y : A$"$\}$ $\vdash Y$" $: B$". Hence S" $\vdash \lambda y.\, Y$" $: (A \to B)$".

Case 4: $X = (YZ) : B$ with $S \vdash Y : A \to B$. As in Case 1 for any $U : f(Y)$ and $V : f(Z)$ there exist eta reducts U" and V" respectively such that in BCD, S" $\vdash U$" $: A$" $\to B$" and S" $\vdash V$" $: A$". Thus S" $\vdash U$"V" $: B$".

Case 5: $X = \langle YZ \rangle$. As in Case 1 for any $U : f(Y)$ and $V : f(Z)$ there exist eta reducts U" and V" respectively such that in BCD, S" $\vdash U$"V" $: A$, and other eta reducts U" and V" respectively such that in BCD, S" $\vdash U$"V" $: B$". By Church-Rosser for eta and subject reductions there exists W such that UV eta reduces to U"V" eta reduces to W eta expands to U"V" eta expands to UV and S" $\vdash W : A \wedge B$. □

Now Theorems 1 and 2 combine for a nice characterization of the case when X is eta normal.

Corollary 1.14 *If X is eta normal, then in BCD we have $S \vdash X : A$ if and only if there is an eta regular X' such that $S' \vdash X' : A$ and X is an eta reduct of any face of X'.*

Remark 1.15 The eta reductions in Theorem 1 and 2 could be removed by the technique of [6]. There we extend the type structure to make $\to$ "almost" surjective. For each atom p we add new atoms p_l, p_r and the definitional equality $p := p_l \to p_r$. Replacing an atom by its is referred to as "type expansion". It is obvious that two distinct type expansions of a given type have a common type expansion. This was our original formulation of the result. However, here we prefer to state the outcome for the original BCD. With type expansions BPS becomes useful. We state without proof the useful lemmas.

Lemma 1.16 *If in BPS, $S \vdash X : A$ then for any eta expansion $X\#$ of X there exists type expansions $S\#, A\#$ such that $S\# \vdash X\# : A\#$.*

Lemma 1.17 *If in BCD, $S \vdash X : A$ then there is an eta expansion $X\#$ of X and a type expansion $S\#$ of S and $A\#$ of A such that in BPS,*

$$S\# \vdash X\# : A\#.$$

References

[1] Barendregt, H., Dekkers, W., Statman, R., Lambda Calculus with Types, *CUP*, (2013).

[2] Brarendregt, H., M. Coppo and M. Dezani, A filter lambda model and the completeness of type assignment, Journal of Symbolic Logic **48** (1983), pp. 931–940.

[3] Bucciarelli, A., Piperno, A., Salvo I., Intersection types and lambda definability, *MSCS03*, **13** (1), pp. 15-53, (2003).

[4] Coppo, M., Dezani, M., A new type assignment for lambda terms, *Archiv fur math. logik*, **19** (l), pp. 139-156, (1978).

[5] de Vrijer, R., Extending the lambda calculus with surjective pairing is conservative, *LICS* **4**, pp. 204-215, (1989).

[6] Statman, R., A local translation of untyped into simply typed lambda-calculus, CMU Research Report #91-134, (1991).

[7] Statman, R., Surjective pairing revisited, in Liber Armicorum for Roel DeVrijer. Klop, van Oostrom, and van Raamsdonk, eds., University of Amsterdam, (2009).

[8] Støvring, K., Extending the extensional lambda calculus with surjective pairing is conservative, *LMCS* **2**, pp. 1-14.