Retracts in Simple Types

Vincent Padovani

Equipe Preuves, Programmes et Systèmes
CNRS - Université Paris 7
padovani@pps.jussieu.fr

Abstract. In this paper we prove the decidability of the existence of a definable retraction between two given simple types. Instead of defining some extension of a former type system from which these retractions could be inferred, we obtain this result as a corollary of the decidability of the minimal model of simply typed λ-calculus.

Although definably isomorphic simple types are fully characterized in [BL85] as a sequel to a classical result by Dezani for untyped $\lambda\beta\eta$-calculus [Dez76], the more general problem of definable retractions has been left open: given two simple types A and B, this problem is to decide whether there exists $u : A \to B$ and $v : B \to A$ such that $(v \circ u) =_{\beta\eta} I$.

In [BL85], Bruce and Longo exhibit a type system from which one can infer all retractions that are definable without the η-rule. In the same paper, they prove the soundess and the incompleteness, with respect to the general case, of a proper extension of this system. In [LPS92], de'Liguoro, Piperno and Statman show how to extend the latter system to derive all retractions that are definable by linear λ-terms.

This paper attacks the retraction problem from a much more syntactical angle. Instead of trying to give a complete extension of a former system, we use the decision algorithm for the minimal model of simply-typed λ-calculus (see [Pad95]) to build an new algorithm that decides whether a simple type is a retract of another.

For the sake of simplicity, we will focus on the special case where the calculus contains a single ground type. However, this restriction is not crucial: as shown in [Pad95], a minimal model built with more than one ground type is still decidable, and the result can be easily extended to the general case. Also, we will add to the calculus a constant of ground type - again, this addition is harmless, and is only intended to simplify the proof. Both restrictions are discussed in the conclusion.

The paper is divided in two main parts. In section 1.2, we introduce two sets of terms $\mathcal{C}$ and $\mathcal{D}$, the sets of *coders* and *decoders*. In section 1.3, we show that the definability of a retraction by some pair in $\mathcal{C} \times \mathcal{D}$, is a decidable property. In section 2, we prove that if a retraction is definable, then it is definable by a pair in $\mathcal{C} \times \mathcal{D}$. For that purpose we define in sections 2.1 and 2.2 an algorithm which transforms any retraction pair into a pair in $\mathcal{C} \times \mathcal{D}$. The faithfulness of the conversion is proved at section 2.3.

S. Abramsky (Ed.): TLCA 2001, LNCS 2044, pp. 376–384, 2001.

1 Coders and Decoders

We consider the simply-typed λ-calculus with a single ground type $\circ$, a single constant $\perp$ of ground type, with a typing *à la Church*. In the sequel, by terms and types, we understand simply-typed terms and simple types.

The notation $M : A$ indicates that M is a term of type A. Types will be frequently ommited whenever they are irrelevant, or implicit from the context. We will often make use of the following notations:

- $A_1 \dots A_n \to \circ$ denotes the type $(A_1 \to (A_2 \to \dots (A_n \to \circ) \dots))$.
- $\lambda x_1 \dots x_n.(MN_1 \dots N_m)$ denotes $\lambda x_1 \dots \lambda x_n(\dots(MN_1)\dots N_m)$. When there is no amibugity, this term will be also denoted by $\lambda\overline{x}.(M\overline{N})$.
- $\lambda\overline{d}.M$ denotes a term of the form $\lambda x_1 \dots x_n.M$ with no x_i free in M.

1.1 Retracts and Products

We say that the *product of* $A^1, \dots, A^n$ *is a retract of* B if and only if the following property holds:

> There exists a term $M : B$, distinct variables $f^1 : A^1, \dots, f^n : A^n$ free in M, and closed terms $N^1 : B \to A^1, \dots, N^n : B \to A^n$, such that for all $i \in \{1, \dots, n\}$ we have $(N^i\, M) =_{\beta\eta} f^i$.

We denote this property by $(A^1 \times \dots \times A^n) \lhd B$, and say that the $n+1$-uplet $(M, N^1, \dots, N^n)$ is a *witness for* $(A^1 \times \dots \times A^n) \lhd B$. In the special case where $n = 1$ with $A^1 = A$, we simply write $A \lhd B$, and say that A is a retract of B.

Remark 1. In this definition, we do not require $f^1, \dots, f^n$ to be the only free variables of M. Thus, for all subset $\{i_1, \dots, i_p\}$ of $\{1, \dots, n\}$, $(M, N^{i_1}, \dots, N^{i_p})$ is also a witness for $(A^{i_1} \times \dots \times A^{i_p}) \lhd B$. In particular, if 0 denotes the empty product, every term of type B is a witness for $0 \lhd B$.

Our first aim will be to build two sets of terms, $\mathcal{C}$ and $\mathcal{D}$, and to show that the existence of a witness in $\mathcal{C} \times \mathcal{D}^n$ for a relation of the form $(A^1 \times \dots \times A^n) \lhd B$, is decidable.

1.2 Projections, Coders, and Decoders

- we let $(\circ^0 \to \circ) = \circ$, and $(\circ^{n+1} \to \circ) = (\circ \to (\circ^n \to \circ))$.
- we call *projection* every term of the form
  $\Pi_i^n = \lambda x_1 \dots x_n.x_i : (\circ^n \to \circ)$ $(i \in \{1, \dots, n\})$.
- we call *selector* every closed term whose type is of the form
  $B_1 \dots B_m \to (\circ^n \to \circ)$.
- we let $\mathcal{C}$, $\mathcal{D}$ be the least sets of terms satisfying the following:
  - $f : \circ \in \mathcal{C}$, for all variable f of ground type.
  - $\lambda x.x : \circ \to \circ \in \mathcal{D}$.
  - Suppose the types $A^1, \dots, A^n, B$ are of the following form:

$$A^i = A^i_1 \dots A^i_{p_i} \to \circ \quad (i \in \{1, \dots, n\}),$$
$$B = B_1 \dots B_m \to \circ.$$

Let $f^1 : A^1, \dots, f^n : A^n$, $z_1 : B_1, \dots, z_m : B_m$ be distinct variables.
Let Σ be a selector of type $B_1 \dots B_m \to (\circ^n \to \circ)$.
For each $i \in \{1, \dots, n\}$:
Let ϕ_i be an arbitrary function from $\{1, \dots, p_i\}$ to $\{1, \dots, m\}$.
For each $j \in \{1, \dots, p_i\}$, let $D^i_j : B_{\phi_i(j)} \to A^i_j \in \mathcal{D}$.

Then, the following term $C : B$ belongs to $\mathcal{C}$:

$$\begin{aligned} C = \lambda z_1 \dots z_m.(\Sigma\, z_1 \dots z_m)\ & (f^1\, (D^1_1\, z_{\phi_1(1)}) \dots (D^1_{p_1}\, z_{\phi_1(p_1)})) \\ & \vdots \\ & (f^n\, (D^n_1\, z_{\phi_n(1)}) \dots (D^n_{p_n}\, z_{\phi_n(p_n)})) \end{aligned}$$

The term Σ will be called the *main selector* of C.

– Let $y_1 : A_1, \dots, y_p : A_p$ be distinct variables. Let $C_1 : B_1, \dots, C_m : B_m$ be terms in $\mathcal{C}$ whose free variables are all amongst $y_1, \dots, y_p$. Let g be a fresh variable of type $B = B_1 \dots B_m \to \circ$. Then:

$$D = \lambda g\, y_1 \dots y_p.(g\, C_1 \dots C_m) : B \to (A_1 \dots A_p \to \circ) \in \mathcal{D}$$

We call *coder* every element of $\mathcal{C}$, and *decoder* every element of $\mathcal{D}$.

1.3 Observational Equivalence

We define the equivalence relation $\equiv$ on the set of selectors with:

let Σ, Σ' be selectors of same type $B_1 \dots B_m \to (\circ^n \to \circ)$. Then $\Sigma \equiv \Sigma'$ if and only if for all projection $\Pi : (\circ^n \to \circ)$ and for all term F, $(F\, \Sigma) =_\beta \Pi \Leftrightarrow (F\, \Sigma') =_\beta \Pi$.

Proposition 1. *The number of selector classes of any given type is finite. Moreover, there exists a computable function which, given an arbitrary selector type, returns a set of selectors of this type which is complete, up to $\equiv$.*

Proof. This follows immediately from the fact that, on one hand, the relation $\equiv$ on selectors coincide with observational equivalence in the minimal model of simply typed λ-calculus, on the other hand, this model is decidable. See [Pad95] for details, or [Sch98] for an alternate proof of decidability, or [Loa97] for a brief presentation of Schmidt-Schauß' algorithm.

We extend $\equiv$ to coders and decoders with the following definition:

– let C, C' be coders of same type, same free variables $f^1 : A^1, \dots, f^n : A^n$, of the following forms:
  – $C = \lambda \overline{z}.(\Sigma\, \overline{z})(f^1\, \overline{u}^1) \dots (f^n\, \overline{u}^n)$,

  - $C' = \lambda\overline{z}.(\Sigma'\,\overline{z})(f^1\,\overline{v}^1)\dots(f^n\,\overline{v}^n)$.

  Then $C \equiv C'$ if and only if:
  - $\Sigma \equiv \Sigma'$, *i.e.* the main selectors of C, C' are equivalent,
  - for all i, j, if $u^i_j = (D\,z)$ and $v^i_j = (D'\,z')$, then $z = z'$ and $D \equiv D'$.
- let D, D' be decoders of same type, of the following forms:
  - $D = \lambda g\,\overline{y}.(g\,C_1\dots C_m)$,
  - $D' = \lambda g\,\overline{y}.(g\,C'_1\dots C'_m)$.

  Then $D \equiv D'$ if and only if for all $k \in \{1,\dots,m\}$ we have $C_k \equiv C'_k$.

Lemma 1. *Let C, C' be coders. Suppose $C \equiv C'$. Then for all term F and all projection Π, we have $F\,C =_\beta \Pi$ if and only if $F\,C' =_\beta \Pi$.*

Proof. Assuming $C \equiv C'$, the terms C, C' are of the following forms:

- $C = \lambda\overline{z}.(\Sigma\,\overline{z})(f^1\,\overline{u}^1)\dots(f^n\,\overline{u}^n)$,
- $C' = \lambda\overline{z}.(\Sigma'\,\overline{z})(f^1\,\overline{v}^1)\dots(f^n\,\overline{v}^n)$,

with $\Sigma \equiv \Sigma'$. Suppose for instance $(F\,C) =_\beta \Pi$. Then $f_1,\dots,f_n$ are not free in the normal form of $(F\,C)$, therefore:

$$\begin{aligned}\Pi &=_\beta (F\,C)\\ &=_\beta (F\,C[\lambda\overline{d}.\bot/f_1,\dots,\lambda\overline{d}.\bot/f_n])\\ &=_\beta F\,(\lambda\overline{z}.(\Sigma\,\overline{z}\,\overline{\bot}))\end{aligned}$$

where $\overline{\bot}$ denotes a sequence of $\bot$ of length n. Now, the main selectors Σ, Σ' of C, C' are equivalent, therefore:

$$\begin{aligned}\Pi &=_\beta F\,(\lambda\overline{z}.(\Sigma'\,\overline{z}\,\overline{\bot}))\\ &=_\beta (F\,C'[\lambda\overline{d}.\bot/f_1,\dots,\lambda\overline{d}.\bot/f_n])\\ &=_\beta (F\,C')\end{aligned}$$

Lemma 2. *Let C be a coder, D a decoder. If (C, D) is a witness for $A \lhd B$, and if $C \equiv C'$ and $D \equiv D'$, then (C', D') is a witness for $A \lhd B$.*

Proof. Let f be the variable such that $(D\,C) =_{\beta\eta} f$. We use induction on the type B of C and C' to show that $(D'\,C') =_{\beta\eta} f$. According to the definition of equivalence, the considered terms are of the following forms:

- $C = \lambda z_1\dots z_m.(\Sigma\,\overline{z})(f^1\,\overline{u}^1)\dots(f^n\,\overline{u}^n)$,
- $C' = \lambda z_1\dots z_m.(\Sigma'\,\overline{z})(f^1\,\overline{v}^1)\dots(f^n\,\overline{v}^n)$,
- $D = \lambda g\,y_1\dots y_p.(g\,C_1\dots C_m) = \lambda g\,\overline{y}.(g\,\overline{C})$,
- $D' = \lambda g\,y_1\dots y_p.(g\,C'_1\dots C'_m) = \lambda g\,\overline{y}.(g\,\overline{C}')$,

with $\Sigma \equiv \Sigma'$ and $C_k \equiv C'_k$ for all $k \in \{1,\dots,m\}$. Suppose $f = f^i$. In that case, the sequences $\overline{u}^i$, $\overline{v}^i$ are of the forms $(u^i_1,\dots,u^i_p)$, $(v^i_1,\dots,v^i_p)$, and:

- $(\Sigma\,\overline{C}) =_\beta \Pi^n_i$,
- $(D\,C) =_\beta \lambda\overline{y}.(f^i\,\overline{u}^i)[\overline{C}/\overline{z}] =_{\beta\eta} \lambda\overline{y}.(f\,\overline{y})$.

Since $\Sigma \equiv \Sigma'$, we have $(\Sigma\,\overline{C}') =_\beta \Pi_i^n$. Since $C_k \equiv C'_k$ for all $k \in \{1, \ldots, m\}$, by lemma 1 we have $(\Sigma'\,\overline{C}') =_\beta \Pi_i^n$. Therefore $(D'\,C') =_\beta \lambda\overline{y}.(f\,\overline{v}^i)[\overline{C}'/\overline{z}]$. Now, for all j, for k and d such that $u_j^i = (d\,z_k)$, we have:

- v_j^i is of the form $(d'\,z_k)$ with $d \equiv d'$,
- $u_j^i[\overline{C}/\overline{z}] = (d\,C_k) =_{\beta\eta} y_j$,
- by induction hypothesis, $v_j^i[\overline{C}'/\overline{z}] = (d'\,C'_k) =_{\beta\eta} y_j$.

We conclude that $(D'\,C') =_\beta \lambda\overline{y}.(f^i\,\overline{v}^i)[\overline{C}'/\overline{z}] =_{\beta\eta} \lambda\overline{y}.(f^i\,\overline{y}) =_{\beta\eta} f^i$.

Theorem 1. *Let $f^1 : A^1, \ldots, f^n : A^n$ be typed variables. Let B be an arbitrary type. Then:*

- *the number of classes of coders of type B and free variables $f^1, \ldots, f^n$, is finite,*
- *for all i, the number of classes of decoders of type $B \to A^i$ is finite.*

Moreover, there exists a computable function which takes as an input the typed variables $f^1 : A^1, \ldots, f^n : A^n$ and the type B, and returns a representative of each class.

Proof. The proof of finiteness and the simultaneous construction of complete sets of representatives is straightforward, using induction on B, proposition 1 and lemmas 1 and 2.

Corollary 1. *Given the types $A^1, \ldots, A^n$ and the type B, the existence of a witness in $\mathcal{C} \times \mathcal{D}^n$ for $(A^1 \times \ldots \times A^n) \lhd B$ is decidable.*

Our next aim will be to prove that if $(A^1 \times \ldots \times A^n) \lhd B$, then there is indeed a witness in $\mathcal{C} \times \mathcal{D}^n$ for this relation. As an immediate consequence, we will get the decidability of $\lhd$.

2 Witness Conversion

This section presents an algorithm, *Convert*, which takes as an input an arbitrary witness $(M, N^1, \ldots, N^n)$ for $(A^1 \times \ldots \times A^n) \lhd B$, and returns a witness $(C, D^1, \ldots, D^n)$ in $\mathcal{C} \times \mathcal{D}^n$ for this relation.

2.1 Linearization

Note that if (M, N) is a witness for $A \lhd B$, then the β-normal, η-long form of N is necessarily of the form $\lambda g \lambda\overline{y}.(g \ldots)$ with g of type B. The following function is intended to transform N into a term with a single (head) occurrence of g.

Function Linearize $(M : B, f : A, N : B \to A)$

assuming N closed and $(N\,M) =_{\beta\eta} f$,

0. Redefine N as the β-normal, η-long form of N.
   Let $M_\perp = M[\lambda\overline{d}.\perp/f^1, \ldots, \lambda\overline{d}.\perp/f^n]$, where $f^1, \ldots, f^n$ are all free variables of M.
1. Let $\lambda g\lambda\overline{y}.(g\, u_1 \ldots u_m) = N$.
2. If $\lambda\overline{y}.(M_\perp\, u_1[M/g] \ldots u_m[M/g]) =_{\beta\eta} f$,
   redefine N as the normal form of $\lambda g\lambda\overline{y}.(M_\perp\, u_1 \ldots u_m)$, and goto 1.
3. If $\lambda\overline{y}.(M\, u_1[M_\perp/g] \ldots u_m[M_\perp/g]) =_{\beta\eta} f$,
   redefine N as the normal form of $\lambda\overline{y}.(g\; u_1[M_\perp/g] \ldots u_m[M_\perp/g])$, and return N.

Remark 2. At step 1, $\lambda\overline{y}.(M\, u_1[M/g] \ldots u_m[M/g]) =_{\beta\eta} f$, where f is free in M and not free in $u_1, \ldots, u_m$. Therefore, one and only one of the conditions at steps 2 or 3 is satisfied.

Note also that all reductions of $(N\, M_\perp)$ are of finite length, therefore this algorithm terminates. It is not hard to check that $N' = \mathit{Linearize}\,(M, f, N)$ is closed, and that we still have $(N'\, M) =_{\beta\eta} f$.

2.2 Conversion

Function Convert$(M : B, f^1 : A^1, \ldots, f^n : A^n, N^1 : B \to A^1, \ldots, N^n : B \to A^n)$

assuming $B = B_1 \ldots B_m \to \circ$, $f^1, \ldots, f^n$ *distinct, and for all* $i \in \{1, \ldots, n\}$*:* $A^i = A^i_1 \ldots A^i_{p_i} \to \circ$*,* N^i *closed,* $(N^i\, M) =_{\beta\eta} f^i$*,*

0. If $n = 0$, return $\lambda\overline{d}.\perp : B$ and exit.
1. For each $f \notin \{f^1, \ldots, f^n\}$ free in M, redefine M as $M[\lambda\overline{d}.\perp/f]$.
   For each $i \in \{1, \ldots, n\}$:
   let $\lambda g y^i_1 \ldots y^i_{p_i}.(g\, u^i_1 \ldots u^i_m) = \mathit{Linearize}(M, f^i, N^i)$.
2. Define $\Sigma : B_1 \ldots B_m \to (\circ^n \to \circ)$ as the normal form of
   $\lambda\overline{z}\lambda x_1 \ldots x_n.(M[\lambda\overline{d}.x_1/f^1, \ldots, \lambda\overline{d}.x_n/f^n]\ \overline{z})$.
3. For each $i \in \{1, \ldots, n\}$, for each $j \in \{1, \ldots, p_i\}$,
   - let σ be the least substitution satisfying:
     - $\sigma(f^i) = \lambda\overline{y}^i.(y^i_j\, \overline{X}^i_j) : A^i$ where the $\overline{X}^i_j$ are fresh variables,
     - $\sigma(f^k) = \lambda\overline{d}.\perp : A^k$ for all $k \in \{1, \ldots, n\}$, $k \neq i$,
   - let $v^i_j : A^i_j$ be the normal form of $\lambda\overline{X}^i_j.(\sigma(M)\ \overline{z})$.
4. for each $i \in \{1, \ldots, n\}$,
   - for each $k \in \{1, \ldots, m\}$, let $b^i_k = u^i_k[\lambda\overline{d}.\perp/y^i_1, \ldots, \lambda\overline{d}.\perp/y^i_{p_i}]$.
   - for each $j \in \{1, \ldots, p_i\}$, define $\phi_i(j)$ as the unique $k \in \{1, \ldots, m\}$ such that the term
     $w^i_j = v^i_j[b^i_1/z_1, \ldots, b^i_{k-1}/z_{k-1}, b^i_{k+1}/z_{k+1}, \ldots, b^i_m/z_m]$
     satisfies $w^i_j[u^i_k/z_k] =_{\beta\eta} y^i_j$.
5. For each $i \in \{1, \ldots, n\}$, for each $k \in \{1, \ldots, m\}$,

– let $\{j_1, \ldots, j_q\} = \phi_i^{-1}(k)$,
– let $(C_k^i : B_k, D_{j_1}^i : B_k \to A_{j_1}^i, \ldots, D_{j_q}^i : B_k \to A_{j_q}^i)$
$= \mathit{Convert}(u_k^i, y_{j_1}^i, \ldots, y_{j_q}^i, \lambda z_k\, w_{j_1}^i, \ldots, \lambda z_k\, w_{j_q}^i)$.

6. Define $C : B$ as $\lambda\overline{z}.(\Sigma\,\overline{z})\ (f^1\,(D_1^1\, z_{\phi_1(1)}) \ldots (D_{p_1}^1\, z_{\phi_1(p_1)}))$

$$\vdots$$

$$(f^n\,(D_1^n\, z_{\phi_n(1)}) \ldots (D_{p_n}^n\, z_{\phi_n(p_n)}))$$

For each $i \in \{1, \ldots, n\}$, define $D^i : B \to A^i$ as
$\lambda g\, y_1^i \ldots y_{p_i}^i.(g\, C_1^i \ldots C_m^i)$.

7. Return $(C, D^1, \ldots, D^n)$.

2.3 Faithfullness

Lemma 3. *Suppose* $(C, \ldots) = \mathit{Convert}(M, \ldots, \ldots)$. *Then, for all term* F *and all projection* Π, *we have* $(F\,M) =_\beta \Pi$ *iff* $(F\,C) =_\beta \Pi$.

Proof. Suppose $(F\,M) =_\beta \Pi$ or $(F\,C) =_\beta \Pi$. Then $f_1, \ldots, f_m$ are not free in the normal form of $(F\,M)$, or not free in the normal form of $(F\,C)$. According to steps 2 and 6, we have:

$$\begin{aligned} & (F\,M) \\ =_\beta\ & (F\,M[\lambda\overline{d}.\bot/f^1, \ldots, \lambda\overline{d}.\bot/f^n]) \\ =_\beta\ & F\,(\lambda\overline{z}.(\Sigma\,\overline{z}\,\overline{\bot})) \\ =_\beta\ & (F\,C[\lambda\overline{d}.\bot/f^1, \ldots, \lambda\overline{d}.\bot/f^n]) \\ =_\beta\ & (F\,C) \end{aligned}$$

where $\overline{\bot}$ denotes a sequence of $\bot$ of length n.

Lemma 4. *Let* $(M, N^1, \ldots, N^n)$ *be a witness for* $(A_1 \times \ldots \times A^n) \lhd B$, *such that* $(N^i\, M) =_{\beta\eta} f^i$ *for all* i. *Then* $(C, D^1, \ldots, D^n) = \mathrm{Convert}(M, \overline{f}, \overline{N})$ *belongs to* $\mathcal{C} \times \mathcal{D}^n$, *and is still a witness for this relation.*

Proof. Assuming $n > 0$, we use induction on the type B of M and C. Let us trace the construction of $C, D^1, \ldots D^n$ in *Convert*.

At step 2, $(N^i\, M) =_\beta \lambda\overline{y}^i.(M\,\overline{u}^i) =_{\beta\eta} f^i$ implies $(\Sigma\,\overline{u}^i) =_\beta \Pi_i^n$.

At step 3, $(M\,\overline{u}^i) =_{\beta\eta} (f^i\,\overline{y}^i)$ implies $v_j^i[\overline{u}^i/\overline{z}] =_{\beta\eta} \lambda\overline{X}_j^i.(y_j^i\,\overline{X}_j^i) =_{\beta\eta} y_j^i$. Note that all free variables of v_j^i belong to $\{z_1, \ldots, z_m\}$.

At step 4, $v_j^i[\overline{u}^i/\overline{z}] =_{\beta\eta} y_j^i$ implies the existence of a unique $k \in \{1, \ldots, m\}$ such that the free occurence of y_j^i in the $\beta\eta$-normal form of $v_j^i[\overline{u}^i/\overline{z}]$ is a residual of a free occurence of y_j^i in u_k^i. Thus, $\phi_i(j)$ is well defined. Note that $z_{\phi_i(j)}$ is the only free variable of w_j^i.

At step 5, for all $i \in \{1, \ldots, n\}$ and all $j \in \{1, \ldots, p_i\}$, we have $((\lambda z_{\phi_i(j)} w^i_j)\, u^i_{\phi_i(j)}) =_{\beta\eta} y^i_j$. Therefore, for all i and all $k \in \{1, \ldots, m\}$, $(u^i_k, \lambda z_k w^i_{j_1}, \ldots, \lambda z_k w^i_{j_q})$ is a witness for $(A^i_{j_1} \times \ldots \times A^i_{j_q}) \lhd B_k$. By induction hypothesis, C^i_k is a coder, $D^i_{j_1}, \ldots, D^i_{j_q}$ are decoders, and $(C^i_k, D^i_{j_1}, \ldots, D^i_{j_q})$ is a witness for $(A^i_{j_1} \times \ldots \times A^i_{j_q}) \lhd B_k$. In other words, for all i, j we have $(D^i_j\, C^i_{\phi_i(j)}) =_{\beta\eta} y^i_j$, therefore we have $((\lambda \overline{z}.f^i\,(D^i_1\, z_{\phi_i(1)}) \ldots (D^i_{p_i}\, z_{\phi_i(p_i)}))\, \overline{C}^i) =_{\beta\eta} (f^i\, y^i_1 \ldots y^i_{p_i})$.

At step 6, obviously C is a coder and the $D^1, \ldots, D^n$ are decoders. According to the analysis of step 5, it remains to check that for all i we have $(\Sigma\, \overline{C}^i) =_\beta \Pi^n_i$. We already know that $(\Sigma\, \overline{u}^i) =_\beta \Pi^n_i$, and that each C^i_k was computed by feeding *Convert* with u^i_k as first argument. By lemma 3, for all F, $(F\, u^i_k) =_\beta \Pi^n_k$ implies $(F\, C^i_k) =_\beta \Pi^n_i$, so we are done.

The faithfullness of the conversion immediately implies:

Lemma 5. *If* $(A^1 \times \ldots \times A^n) \lhd B$ *then there is a witness in* $\mathcal{C} \times \mathcal{D}^n$ *for this relation.*

At last, we obtain our expected main result:

Theorem 2. *The relation* $\lhd$ *is decidable.*

3 Conclusion

So far, we have proved that the existence of a definable retraction between two simple types is decidable, in the special case of one ground type $\circ$, with a constant $\perp$ of type $\circ$. The generalization to many ground types, with one constant of each ground type, requires only minor modifications of our proof, thanks to the following lemma:

Lemma 6. *If* $A^1 \times \ldots \times A^n \lhd B$ *then* $A^1, \ldots, A^n, B$ *are of same rightmost ground type.*

Proof. Suppose $A \lhd B$, with $A = A_1 \ldots A_p \to \circ$ and $B = B_1 \ldots B_m \to \circ'$. Take $M : B$ and a closed $N = \lambda g\, y_1 \ldots y_p.(g\, \overline{u}) : B \to A$ such that $(N M) =_{\beta\eta} f$. Then the $\beta\eta$-normal form of $(M\, \overline{u}[M/g]) : \circ'$ is equal to $(f\, \overline{y}) : \circ$, hence $\circ = \circ'$.

As a consequence, if we consider many ground types, the linearization and conversion algorithms still work as they are,[1] and all we need to do is to abstract $\circ$ at the right places in the definitions of selectors, coders, decoders. The decidability still holds, due to the fact that a minimal model built with finitely

[1] The assumptions made in *Convert* ensure that $A^1, \ldots, A^n, B$ are of same rightmost ground type $\circ$, so the selector built at step 2 is indeed of type $B_1 \ldots B_m \to (\circ^n \to \circ)$. At step 1, $\perp$ becomes implicitely a constant of adequate ground type. *Linearize* can be leaved unchanged, because it is always called with $f_1, \ldots, f_n$ of same rightmost ground type: a single constant $\perp$ is required to compute $M_\perp$.

many ground types is decidable, and that we do not need any other ground types than the ones appearing in $A^1, \dots A^n, B$ in order to build in $\mathcal{C} \times \mathcal{D}^n$ a witness for $A^1 \times \dots \times A^n \lhd B$. Furthermore, we can easily get rid of all constants by using instead fresh variables, distinct from every other variable mentioned in our proofs and definitions, and by modifying accordingly the notion of "closed" term.

In conclusion, let us remark that this proof leaves open the existence of a type system from which all definable retractions could be inferred, whithout appealing to some syntactical criterion - this question probably requires a further understanding of the minimal model itself. Still, it is possible to extract from the proof of lemma 4 an alternate proof of the following theorem, which is an immediate corollary of the Witness Theorem in [LPS92], and which provides an incomplete characterization of definable retractions:

Theorem 3. *Let $A = A_1 \dots A_p \to \circ$, $B = B_1 \dots B_m \to \circ$ be simple types. If a definable retraction exists between A and B then, for all j, there exists a k such that a definable retraction exists between A_j and B_k.*

Indeed, in lemma 4, as we reconstruct a witness for $(A^1 \times \dots \times A^n) \lhd B$ we prove, for each i and for each $j \in \{1, \dots, p_i\}$, the existence of an $k = \phi_i(j)$ such that $A^i_j \lhd B_k$.

Acknowledgements. Thanks to Laurent Regnier and Pawel Urzyczyn, for submitting the right problem at the right time.Special thanks to Paul-André Mellies, Vincent Danos and François Maurel, for their patience and wisdom.

References

[BL85] Bruce, K., Longo, G. (1985) Provable isomorphims and domain equations in models of typed languages. *A.C.M. Symposium on Theory of Computing (STOC 85).*

[Dez76] Dezani-Ciancaglini, M. (1976) Characterization of normal forms possessing inverse in the $\lambda\beta\eta$-calculus. *TCS* **2**.

[LPS92] de'Liguoro, U., Piperno, A., Statman, R. (1992) Retracts in simply typed $\lambda\beta\eta$-calculus. *Proceeding of the 7th annual IEEE symposium on Logic in Computer Science (LICS92)*, IEEE Computer Society Press.

[Loa97] Loader, R. (1997) An Algorithm for the Minimal Model. *Manuscript.* Available at `http://www.dcs.ed.ac.uk/home/loader/`.

[Pad95] Padovani, V. (1995) Decidability of all minimal models. *Proceedings of the annual meeting Types for Proof and Programs - Torino 1995, Lecture Notes in Computer Science* 1158, Springer-Verlag.

[Sch98] Schmidt-Schauß, M. (1998) Decidability of Behavioral Equivalence in Unary PCF. *Theorical Computer Science* 216.