A New Characterization of Lambda Definability

Achim Jung[1] and Jerzy Tiuryn[⋆2]

[1] Fachbereich Mathematik, Technische Hochschule Darmstadt, Schloßgartenstraße 7, D-6100 Darmstadt, Germany, jung@mathematik.th-darmstadt.de
[2] Instytut Informatyki, Uniwersytet Warszawski, ul. Banacha 2, PL-02-097 Warszawa, Poland, tiuryn@mimuw.edu.pl

Abstract. We give a new characterization of lambda definability in Henkin models using logical relations defined over ordered sets with varying arity. The advantage of this over earlier approaches by Plotkin and Statman is its simplicity and universality. Yet, decidability of lambda definability for hereditarily finite Henkin models remains an open problem. But if the variable set allowed in terms is also restricted to be finite then our techniques lead to a decision procedure.

1 Introduction

An *applicative structure* consists of a family $(A_\sigma)_{\sigma\in\mathbb{T}}$ of sets, one for each type σ, together with a family $(app_{\sigma,\tau})_{\sigma,\tau\in\mathbb{T}}$ of application functions, where $app_{\sigma,\tau}$ maps $A_{\sigma\to\tau} \times A_\sigma$ into A_τ. For an applicative structure to be a model of the simply typed lambda calculus (in which case we call it a *Henkin model*, following [4]), one requires two more conditions to hold. It must be *extensional* which means that the elements of $A_{\sigma\to\tau}$ are uniquely determined by their behavior under $app_{\sigma,\tau}$, or, more intuitively, that $A_{\sigma\to\tau}$ can be thought of as a set of functions from A_σ to A_τ. Secondly, the applicative structure must be rich enough to interpret every lambda term. (This requirement can be formalized using either the combinatory or the environment model definition, see Sect. 2 below.) The simplest examples for Henkin models are derived if one takes a set A_ι for the base type ι (more base types could be accommodated in the same way) and then defines $A_{\sigma\to\tau}$ to be the set of all functions from A_σ to A_τ. The application functions are in this case just set-theoretic application of a function to an argument. These models are sometimes called the *full type hierarchy over* A_ι.

Simple as this construction is, there remains a nagging open question. Suppose A_ι is finite (in which case every A_σ is finite), is there an algorithm which, given an element of some A_σ, decides whether it is the denotation of a closed lambda term? We could also ask for an algorithm which works uniformly for all finite sets A_ι, but the essential difficulty seems to arise with the first question. The assumption that a positive solution exists is known under the name *lambda definability conjecture*. We shall speak of the *lambda definability problem* instead. Besides this being an intriguing question in itself, there are also connections to

⋆ Supported by Polish KBN grant No. 2 1192 91 01

other open problems, such as the *higher order matching problem* (cf. [9], and also [13]) and the *full abstraction problem for PCF* (cf. [1]).

Let us quickly review the existing literature on the question. A first attempt to characterize lambda definable elements in the full type hierarchy was made by H.Läuchli [2]. He showed that lambda definable elements are invariant under permutations of the ground set A_ι which is a not too surprising result as there are no means by which the lambda calculus could speak about particular elements of A_ι. He also observed that permutation invariance was too weak a property for full characterization at all types. This line of thought was taken up by G.Plotkin in [7] (a precursor of this is [6]). He replaced permutation invariance by invariance under logical relations and proved that for infinite ground sets this characterizes lambda definability at types of rank less than three. Using more complicated logical relations defined over a quasi-ordered set he could remove the restriction on the rank. The restriction on the size of A_ι, however, remained. In both cases the proof is by coding the theory of lambda terms into the ground set. The problem is also discussed in papers by R.Statman (cf. [9, 10, 11, 12]). In [12] a characterization is stated (without proof) which is applicable to all Henkin models and which employs logical relations on a free extension of the given model by infinitely many variables. More recently, K.Sieber [8] used logical relations in a novel fashion to tackle the full abstraction problem for PCF. His logical relations have large arity and are reminiscent of value tables. It was this paper from which we got the initial idea for the results presented here. By looking at logical relations which are defined over an ordered set (as in [7]) but which in addition increase their arity as we pass to later "worlds", we derive a characterization theorem which works for all ground sets A_ι and, in fact, every Henkin model, which again contrasts to the characterization in [7], which can not be generalized to arbitrary Henkin models. (A counterexample is given in [12].) Furthermore, our characterization theorem has a very straightforward proof. Indeed, the proof is so simple that it suggests a positive solution to the lambda definability problem. Even though we do not achieve this, at least we can make the obstacles very clear. These lie in the fact that higher order terms (even when they are in normal form) can contain arbitrarily many auxiliary variables. For a restricted set of variables one would expect a decidability result. This can be achieved as we show in Sect. 5, but the proof becomes somewhat technical.

Our definition of logical relation will still make sense if we replace the ordered set by a small category and, in fact, it reduces to a logical predicate on the presheaf model built from the initial Henkin model (for details, see [3] or [5]). A bit of this generality indeed simplifies our presentation of Kripke logical relations with varying arity in the next section. The characterization theorem in Sect. 3, however, works with a very simple fixed ordered set.

2 Kripke Logical Relations with Varying Arity

Suppose A_ι is a set and we are studying the semantics of the simply typed lambda calculus in the full type hierarchy over A_ι. (We could take an arbitrary Henkin model instead but would have to write out the application functions explicitly in every instance.) Let $\mathcal{C}$ be a small category of sets. We want to build a logical relation over each object w of $\mathcal{C}$, taking the cardinality of w as the arity of the relation at w. Thus elements of the relations are tuples indexed by elements of w. It makes no difference whether w is finite or infinite.

We start with ground relations $R^w_\iota \subseteq A^w_\iota$ which have the following compatibility property: Whenever $f: v \to w$ is a map in $\mathcal{C}$ and $(x_j)_{j\in w}$ is an element of R^w_ι then $(x_{f(i)})_{i\in v}$ is an element of R^v_ι (note the contravariance). The ground relations are extended to higher types as usual. For a function type $\sigma \to \tau$ let

$$R^w_{\sigma\to\tau} = \{(g_j)_{j\in w} \mid \forall j \in w. g_j \in A_{\sigma\to\tau} \wedge \forall f: v \to w \forall (x_i)_{i\in v} \in R^v_\sigma. \\ (g_{f(i)}(x_i))_{i\in v} \in R^v_\tau\}.$$

(A tuple of functions at w must have the defining property of logical relations at all v reachable - via a map in $\mathcal{C}$ - from w.) Relations $(R^w_\sigma)^{w\in Obj(\mathcal{C})}_{\sigma\in\mathbb{T}}$ constructed this way we shall call *Kripke logical relations with varying arity.* Ordinary logical relations are subsumed by this concept - just take a one object one morphism category $\mathcal{C}$ - as well as Plotkin's "I-relations": take a category all of whose objects have the same cardinality and all of whose morphisms are bijections such that the category is isomorphic to a quasi-ordered set.

We observe that for each type σ we have the compatibility with morphisms of $\mathcal{C}$ we required at ground level:

Lemma 1. *Let $(R^w_\sigma)^{w\in Obj(\mathcal{C})}_{\sigma\in\mathbb{T}}$ be a Kripke logical relation with varying arity. For all types σ, objects v, w of $\mathcal{C}$, morphisms $f: v \to w$, and tuples $(x_j)_{j\in w}$ in R^w_σ, the tuple $(x_{f(i)})_{i\in v}$ is in R^v_σ.*

Proof. By induction on types. For ι it is part of the definition. If $\sigma \to \tau$ is a function type we have to show that $(g_j)_{j\in w} \in R^w_{\sigma\to\tau}$ implies $(g_{f(i)})_{i\in v} \in R^v_{\sigma\to\tau}$. By definition we have to supply arguments $(x_l)_{l\in u} \in R^u_\sigma$, for $h: u \to v$ to the functions. The resulting tuple has the form $(g_{f(h(l))}(x_l))_{l\in u}$ which belongs to R^u_τ because $f \circ h: u \to w$ is also a map in $\mathcal{C}$ and was taken account of in the definition of $R^w_{\sigma\to\tau}$. □

Our logical relations have the usual "un-Currying" property.

Lemma 2. *Let $(R^w_\sigma)^{w\in Obj(\mathcal{C})}_{\sigma\in\mathbb{T}}$ be a Kripke logical relation with varying arity. For any type $\sigma = \sigma_1 \to \ldots \to \sigma_n \to \iota$ and any object w, a tuple $(g_j)_{j\in w}$ is in R^w_σ if and only if for every chain of maps $v_n \xrightarrow{f_n} \ldots \xrightarrow{f_2} v_1 \xrightarrow{f_1} w$ and tuples $(x^k_i)_{i\in v_k} \in R^{v_k}_{\sigma_k}$, $k = 1, \ldots, n$, the result of applying the functions coordinatewise to all n arguments is in $R^{v_n}_\iota$.*

Proof. Easy induction on the length of the unfolded types $\sigma_1 \to \ldots \to \sigma_n \to \iota$. □

In order to prove the "Fundamental Theorem of Logical Relations" (in the words of [12]) let us recall how the simply typed lambda calculus is interpreted over A_σ. Free variables are assigned values by environments $\rho\colon Var \to \bigcup_{\sigma\in\mathbb{T}} A_\sigma$ (where a variable x^σ of type σ is mapped to A_σ) and the denotation of a lambda term M is then defined with respect to environments as follows:

$M \equiv x^\sigma$: $[\![x^\sigma]\!]\rho = \rho(x^\sigma)$.

$M \equiv M_1M_2$: $[\![M_1M_2]\!]\rho = [\![M_1]\!]\rho([\![M_2]\!]\rho)$.

$M \equiv \lambda x^\sigma.M_1$: $[\![\lambda x^\sigma.M_1]\!]\rho$ = the map which assigns to $a \in A_\sigma$ the value $[\![M_1]\!]\rho[x^\sigma \mapsto a]$. (In a general extensional applicative structure there need not be a representative in $A_{\sigma\to\tau}$ for this map. This is the "richness" of Henkin models we referred to in the Introduction.)

Theorem 3. *For every Kripke logical relation with varying arity $(R^w_\sigma)^{w\in Obj(\mathcal{C})}_{\sigma\in\mathbb{T}}$, object w of $\mathcal{C}$, and closed term M of type σ the constant tuple $([\![M]\!])_{j\in w}$ is in R^w_σ.*

Proof. The proof is for all objects of $\mathcal{C}$ simultaneously by induction on the term structure. Hence we must also take open terms into account. For $w \in Obj(\mathcal{C})$ let $(\rho_j)_{j\in w}$ be a tuple of environments such that for every free variable x^σ of M the tuple $(\rho_j(x^\sigma))_{j\in w}$ is in R^w_σ. We show that under this condition the tuple $([\![M]\!]\rho_j)_{j\in w}$ is in R^w_σ. We check the three cases in the definition of $[\![\cdot]\!]$:

$M \equiv x^\sigma$: $([\![x^\sigma]\!]\rho_j)_{j\in w} = (\rho_j(x^\sigma))_{j\in w} \in R^w_\sigma$ by assumption.

$M \equiv M_1M_2$: $([\![M_1M_2]\!]\rho_j)_{j\in w} = ([\![M_1]\!]\rho_j([\![M_2]\!]\rho_j))_{j\in w}$. By induction hypothesis $([\![M_1]\!]\rho_j)_{j\in w}$ is in $R^w_{\sigma\to\tau}$ and $([\![M_2]\!]\rho_j)_{j\in w}$ is in R^w_σ. Because a category contains an identity morphism for every object, we get that the tuple resulting from pointwise application is in R^w_τ.

$M \equiv \lambda x^\sigma.M_1$: $([\![\lambda x^\sigma.M_1]\!]\rho_j)_{j\in w}$ is a tuple of functions from A_σ to A_τ. To check that it is in relation we to apply to it a tuple $(a_i)_{i\in v}$ of arguments from R^v_σ for an object v and a morphism $f\colon v \to w$. We get the tuple $([\![M_1]\!]\rho_{f(i)}[x^\sigma \mapsto a_i])_{i\in v}$. From Lemma 1 we know that each of the tuples $(\rho_{f(i)}(y))_{i\in v}$, y a variable, is in relation at v. Updating the environments at x^σ to $(a_i)_{i\in v}$ retains this property. So we can conclude from the induction hypothesis that $([\![M_1]\!]\rho_{f(i)}[x^\sigma \mapsto a_i])_{i\in v}$ is in R^v_τ. □

Let us emphasize again that the preceding theorem is neither a surprise nor a generalization over already established results. Our Kripke logical relation with varying arity is nothing more than a logical predicate over a particular Henkin model in the Cartesian closed functor category $\mathcal{S}et^{\mathcal{C}^{op}}$. The point is that we want to look at a complicated logical relation over a simple Henkin model in order to characterize lambda definability in the latter. We included the proof of the Fundamental Theorem in order to acquaint the reader with the technical apparatus.

3 A Characterization of Lambda Definability

We will now characterize lambda definability in the full type hierarchy over some ground set A_ι. (The proof for an arbitrary Henkin model is the same but involves more notational overhead.) From A_ι we construct a concrete category $\mathcal{A}$ as follows. Objects are finite products $A_{\sigma_1\ldots\sigma_n} = A_{\sigma_1} \times \ldots \times A_{\sigma_n}$ of our denotational domains, one for each sequence $\sigma_1 \ldots \sigma_n$ of types. The empty sequence ϵ is represented by an arbitrary one-point set A_ϵ. If $\sigma_1 \ldots \sigma_n$ is a prefix of the sequence $\sigma_1 \ldots \sigma_n \tau_1 \ldots \tau_m$ then our category contains the projection morphism from $A_{\sigma_1} \times \ldots \times A_{\sigma_n} \times A_{\tau_1} \times \ldots \times A_{\tau_m}$ to $A_{\sigma_1} \times \ldots \times A_{\sigma_n}$. So $\mathcal{A}$ is really an ordered set, namely, the dual of $\mathbb{T}^*$ with the prefix ordering. The logical relation $(T_\sigma^w)_{\sigma\in\mathbb{T}}^{w\in Obj(\mathcal{A})}$ which will give us the characterization, has arity $|A_{\sigma_1} \times \ldots \times A_{\sigma_n}|$ at the object $w = A_{\sigma_1} \times \ldots \times A_{\sigma_n}$. A tuple from T_σ^w is therefore indexed by tuples $\boldsymbol{a} = (a_1, \ldots, a_n) \in A_{\sigma_1} \times \ldots \times A_{\sigma_n}$. At ground level we take those tuples $(x_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$ into T_ι^w for which there is a closed lambda term M of type $\sigma_1 \to \ldots \to \sigma_n \to \iota$ such that for each $\boldsymbol{a} = (a_1, \ldots, a_n)$ in $A_{\sigma_1} \times \ldots \times A_{\sigma_n}$ we have $x_{\boldsymbol{a}} = [\![M]\!](a_1)\ldots(a_n)$. Intuitively, we take only those tuples which are "value tables" of lambda definable functions. This idea is taken directly from [8]. These relations have the compatibility property with morphisms in $\mathcal{A}$. Indeed, if M defines the tuple $(x_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$ at $w = A_{\sigma_1} \times \ldots \times A_{\sigma_n}$ then $\lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n} y_1^{\tau_1} \ldots y_m^{\tau_m}.M x_1^{\sigma_1} \ldots x_n^{\sigma_n}$ defines the corresponding tuple at $v = A_{\sigma_1} \times \ldots \times A_{\sigma_n} \times A_{\tau_1} \times \ldots \times A_{\tau_m}$. The following lemma asserts that the lambda definable functions can be read off at A_ϵ, the one element object in $\mathcal{A}$.

Lemma 4. *A one-element tuple (x) is in $T_\sigma^{A_\epsilon}$ if and only if x is the denotation of a closed lambda term of type σ.*

Proof. We prove by induction on types (simultaneously for all objects $w = A_{\sigma_1} \times \ldots \times A_{\sigma_n}$ of $\mathcal{A}$) that T_σ^w only contains tuples which are definable by closed lambda terms of type $\sigma_1 \to \ldots \to \sigma_n \to \sigma$. For $\sigma = \iota$ this is the definition of T_ι^w, so let us look at a function type $\sigma \to \tau$.

If M is a closed term of type $\sigma_1 \to \ldots \to \sigma_n \to \sigma \to \tau$ which defines the tuple $(f_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$ we want to assert that it is in relation at w. To this end we supply an argument tuple $(x_{\boldsymbol{b}})_{\boldsymbol{b}\in v}$ for an object $v = A_{\sigma_1} \times \ldots \times A_{\sigma_n} \times A_{\tau_1} \times \ldots \times A_{\tau_m}$. By induction hypothesis, this tuple is represented by a closed term N of type $\sigma_1 \to \ldots \to \sigma_n \to \tau_1 \to \ldots \to \tau_m \to \sigma$. The resulting tuple $(f_{\pi(\boldsymbol{b})}(x_{\boldsymbol{b}}))_{\boldsymbol{b}\in v}$ is represented by the term $\lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n} y_1^{\tau_1} \ldots y_m^{\tau_m}.(M x_1^{\sigma_1} \ldots x_n^{\sigma_n})(N x_1^{\sigma_1} \ldots x_n^{\sigma_n} y_1^{\tau_1} \ldots y_m^{\tau_m})$ and so, by induction hypothesis, is contained in T_τ^v.

Conversely, assume that the tuple $(f_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$ belongs to $T_{\sigma\to\tau}^w$. We supply it with the argument tuple over the object $v = A_{\sigma_1} \times \ldots \times A_{\sigma_n} \times A_\sigma$ which is given by the term $\lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n} x^\sigma.x^\sigma$. By induction hypothesis it is contained in T_σ^v. The resulting tuple $(f_{\boldsymbol{a}}(a))_{\boldsymbol{a}a\in v}$ is in T_τ^v and, again by induction hypothesis, there is a closed term N of type $\sigma_1 \to \ldots \to \sigma_n \to \sigma \to \tau$ representing it. We claim that N also represents $(f_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$: Using the denotation of N we get a tuple $(g_{\boldsymbol{a}})_{\boldsymbol{a}\in w}$ of functions of type $\sigma \to \tau$ where $g_{(a_1,\ldots,a_n)} = [\![N]\!](a_1)\ldots(a_n)$. In order to see that such a function is equal to the corresponding $f_{\boldsymbol{a}}$ we supply a generic argument a from A_σ. We get $f_{\boldsymbol{a}}(a) = [\![N]\!](a_1)\ldots(a_n)(a) = g_{\boldsymbol{a}}(a)$, which completes our argument. ☐

Theorem 3 and Lemma 4 together give our main result:

Theorem 5. *An element of a Henkin model is lambda definable if and only if it is invariant under all Kripke logical relations with varying arity.*

Somewhat slicker but maybe less transparent is the following description of the relations T^w_σ. We replace sequences of types by finite sets of variables. The objects of $\mathcal{A}$ remain almost the same, $\{x_1^{\sigma_1}, \dots, x_n^{\sigma_n}\}$ corresponds to the set $Env\{x_1^{\sigma_1}, \dots, x_n^{\sigma_n}\}$ of finite environments over this collection of variables. The tuples should now be labeled by our symbol for environments ρ. For $w = Env\{x_1^{\sigma_1}, \dots, x_n^{\sigma_n}\}$ we take a tuple $(x_\rho)_{\rho \in w}$ into T^w_ι if there is a lambda term M whose free variables are contained in $\{x_1^{\sigma_1}, \dots, x_n^{\sigma_n}\}$ such that for all $\rho \in w$ we have $x_\rho = [\![M]\!]\rho$. The proof of Lemma 4 can be changed accordingly.

4 The Lambda Definability Problem

We return to the problem of finding an effective characterization of lambda definability for hereditarily finite Henkin models. Indeed, studying the definability lemma 4 one gets the impression that for a particular type ϕ only a finite piece of the category $\mathcal{A}$ is used. More formally, we can precisely define the objects from $\mathcal{A}$ that occur in the proof of Lemma 4. Fix a type ϕ and define two relations $\vdash_\phi$ and $\Vdash_\phi$ between strings of types and types as follows:

(i) $\epsilon \vdash_\phi \phi$,

(ii) if $s \vdash_\phi \sigma \to \tau$ then $s\sigma \vdash_\phi \tau$ and $s\sigma \Vdash_\phi \sigma$,

(iii) if $s \Vdash_\phi \sigma_1 \to \dots \to \sigma_n \to \iota$ and if for strings $s_1 \leq \dots \leq s_n$ there are types $\xi_1, \dots, \xi_n$ such that for all $k = 1, \dots, n$, $s_k \vdash_\phi \xi_k$ then for all $k = 1, \dots, n$, $s_k \vdash_\phi \sigma_k$.

Now let $\mathcal{F}_\phi$ be the full sub-category of $\mathcal{A}$ whose objects are given by $\{A_s \in \mathcal{A} \mid \exists \xi \in \mathbb{T}.s \vdash_\phi \xi\}$. (Note that the strings occurring on the left hand side of the relation $\Vdash_\phi$ all occur on the left hand side of $\vdash_\phi$ already.) We show that the proof of Lemma 4 for a particular type ϕ can be based on $\mathcal{F}_\phi$. At ground type we start with the same logical relation $(T^w_\sigma)^{w \in Obj(\mathcal{F}_\phi)}_{\sigma \in \mathbb{T}}$ as before.

Lemma 6. *Given a type $\phi \in \mathbb{T}$ the following is true for all $\sigma \in \mathbb{T}$ and $s \in \mathbb{T}^*$:*

(i) If $s \vdash_\phi \sigma$ then every element of $T^{A_s}_\sigma$ is lambda definable.
(ii) If $s \Vdash_\phi \sigma$ then every lambda definable tuple is in $T^{A_s}_\sigma$.

Proof. By induction on σ. If σ is the ground type ι then both statements follow from the definition of T^w_ι. The proof of (i) for a function type $\sigma \to \tau$ works as in Lemma 4: Assume $s = \sigma_1 \dots \sigma_n \vdash_\phi \sigma \to \tau$ and $(f_j)_{j \in A_s} \in T^{A_s}_{\sigma \to \tau}$ (where we have identified w with A_s). We have $s\sigma \Vdash_\phi \sigma$ and so by induction hypothesis we can apply the tuple given by the term $\lambda x_1^{\sigma_1} \dots x_n^{\sigma_n} x^\sigma . x^\sigma$ to it. The result is in $T^{A_s \times A_\sigma}_\tau$ and since $s\sigma \vdash_\phi \tau$ it is given by a term N. As before we see easily that N also defines $(f_j)_{j \in A_s}$.

To prove part (ii) we have to un-Curry completely: $\sigma \to \tau = \sigma_1 \to \ldots \to \sigma_n \to \iota$ (we have re-named σ to σ_1). Assume that the tuple $(f_j)_{j \in A_s}$ is given by a term M. By Lemma 2 we have to apply the functions to argument tuples from $T_{\sigma_k}^{A_{s_k}}$, $k = 1, \ldots, n$ for strings $s \leq s_1 \leq \ldots \leq s_n$ from $\mathcal{F}_\phi$. By our rule (iii) we have for each k, $s_k \vdash_\phi \sigma_k$. Hence we can use the induction hypothesis and conclude that all argument tuples are lambda definable. The application of $(f_j)_{j \in A_s}$ to these argument tuples results in a tuple which again is lambda definable and of type ι. But at ground type lambda definable tuples are in relation and we are done. □

Theorem 7. *An element of type ϕ of a Henkin model is lambda definable if and only if it is invariant under all logical relations based on $\mathcal{F}_\phi$.*

If we are looking at a hereditarily finite Henkin model, for example the full type hierarchy over a finite ground set, and if for some type ϕ the category $\mathcal{F}_\phi$ happens to have only finitely many objects then we can effectively determine the lambda definable elements of A_ϕ by simply checking the finitely many Kripke logical relations with varying arity over $\mathcal{F}_\phi$. Unfortunately, this approach can only succeed for types of rank less than 3:

Lemma 8. *For every type ϕ of rank at least 3 the category $\mathcal{F}_\phi$ has infinitely many objects.*

Proof. We illustrate the idea for the type $\phi = ((\iota \to \iota) \to \iota) \to \iota$. The general proof is exactly the same but involves a lot of indices. Using rules (i)–(iii) above, we get

(1) $\epsilon \vdash_\phi \phi$ by (i).
(2) $(\iota \to \iota) \to \iota \vdash_\phi \iota$ and $(\iota \to \iota) \to \iota \Vdash_\phi (\iota \to \iota) \to \iota$ by (1) and (ii).
(3) $(\iota \to \iota) \to \iota \vdash_\phi \iota \to \iota$ by (2) and (iii).
(4) $\langle(\iota \to \iota) \to \iota\rangle\langle\iota\rangle \vdash_\phi \iota$ by (3) and (ii).
(5) $\langle(\iota \to \iota) \to \iota\rangle\langle\iota\rangle \vdash_\phi \iota \to \iota$ by (2), (4), and (iii).

The last two steps can be repeated forever. □

Behind this proof is the observation that from rank 3 on we can no longer bound the number of variables occurring in a normal form. What happens if we do impose a bound is the topic of the next section.

5 Lambda Definability with Fixed Sets of Variables

5.1 Two-layered logical relations

We proceed by further refining the notion of logical relation and we begin by studying this refinement for ordinary logical relations, letting the varying arity and the Kripke universe at the side for the moment.

Observe that the definition of the extension of a logical relation to a type $\sigma \to \tau$ falls naturally into two halves:

(1) If $f\colon A_\sigma \to A_\tau$ belongs to $R_{\sigma\to\tau}$ then it maps each element of R_σ into R_τ.
(2) If $f\colon A_\sigma \to A_\tau$ maps each element of R_σ into R_τ then it belongs $R_{\sigma\to\tau}$.

In the proof of the Fundamental Theorem the first condition is needed in order to show that an application remains invariant if the constituents are, and the second is needed for abstraction. Of course, the power of logical relations resides in the fact that the two properties are fulfilled simultaneously. Nevertheless, we shall study these two conditions separately and thus tie up our logical relations more closely with the structure of lambda terms. To this end we define the following two-layer type system $(\mathbb{T}_0, \mathbb{T}_1)$ (over a single ground type ι and over the set Var of variables):

- $\iota \in \mathbb{T}_0$
- $\sigma, \tau \in \mathbb{T}_0 \Longrightarrow \sigma \to \tau \in \mathbb{T}_0$
- $B \in \mathsf{Var}^*, \tau \in \mathbb{T}_0 \Longrightarrow B \to \tau \in \mathbb{T}_1$

Note that $\mathbb{T}_0$ may be viewed as a subset of $\mathbb{T}_1$ by virtue of the empty string in Var^*. We will also need the forgetful map $e\colon \mathbb{T}_1 \to \mathbb{T}_0$ which maps $x_1^{\sigma_1} \dots x_n^{\sigma_n} \to \tau$ to $\sigma_1 \to \dots \to \sigma_n \to \tau$.

Now let $R_\iota \subseteq A_\iota$ be an arbitrary relation (for simplicity we let the arity be 1). It is extended to the types of $\mathbb{T}_0$ and $\mathbb{T}_1$ as follows. For $\sigma \to \tau \in \mathbb{T}_0$ let $R_{\sigma\to\tau}$ be *any subset* of

$$\{f \in A_{\sigma\to\tau} \mid \forall \Sigma \in \mathbb{T}_1.(e(\Sigma) = \sigma \Longrightarrow \forall a \in R_\Sigma. f(a) \in R_\tau)\}$$

and for $n \geq 1, B = x_1^{\sigma_1} \dots x_n^{\sigma_n}, B \to \tau \in \mathbb{T}_1$ let $R_{B\to\tau}$ be *any superset* of

$$\{f \in A_{e(B\to\tau)} \mid \forall a_1 \in R_{\sigma_1} \dots \forall a_n \in R_{\sigma_n}. f(a_1)\dots(a_n) \in R_\tau\} \ .$$

Obviously, such two-layered relations are no longer determined by their value at ground type. But starting from some R_ι we can always construct a two-layered logical relation. The Fundamental Theorem now reads as follows:

Theorem 9. *Let $M \equiv \lambda x_1^{\sigma_1} \dots x_n^{\sigma_n}.N$ be a lambda term in normal form and of type $\sigma_1 \to \dots \to \sigma_n \to \tau$ such that N is not an abstraction and let ρ be an environment which maps each free variable y^σ of M into R_σ. Then $[\![M]\!]\rho \in R_{x_1^{\sigma_1} \dots x_n^{\sigma_n} \to \tau}$.*

Proof. We have to argue more carefully, but the proof is essentially as usual. Variables can't cause any problems. In the case that M is an application $M_1 M_2$, we employ the assumption that M is in normal form, hence the denotation of M_1 under ρ is in $R_{\sigma\to\tau}$ where $\sigma \to \tau$ is an ordinary type. The denotation of M_2 under ρ is in some R_Σ where $e(\Sigma) = \sigma$. So the composed term is in R_τ as required.

The case that M is an abstraction is characterized by the fact that $n \geq 1$. Unlike in the usual proof we have to unwind all leading lambdas at one stroke. We want $[\![M]\!]\rho$ to be in $R_{x_1^{\sigma_1} \dots x_n^{\sigma_n} \to \tau}$ and to check this we have to apply it to arguments a_i from R_{σ_i}, $i = 1, \dots, n$, and see whether the result is in R_τ.

This is indeed the case, as $[\![M]\!]\rho(a_1)\dots(a_n) = [\![N]\!]\rho[x_1 \mapsto a_1,\dots,x_n \mapsto a_n]$ and the induction hypothesis applies to N and the updated environment. (The updating must be read from left to right. This way the lemma remains true also for sequences $x_1^{\sigma_1}\dots x_n^{\sigma_n}$ which contain some variables more than once.) $\square$

5.2 Two-layered Kripke logical relations with varying arity

Let us now combine the techniques of Sect. 2 with these two-layered logical relations. We use the presentation of Kripke logical relations with varying arity via environments as briefly described at the end of Sect. 2.

So let $V \subseteq \mathsf{Var}$ be a set of variables. It is our goal to characterize all functionals which are definable by lambda terms containing variables (free or bound) only from V. Our base category $\mathcal{V}$ is the set of all subsets of V together with inclusion morphisms. There is a contravariant equivalence between $\mathcal{V}$ and the category $\mathcal{E}$ of environments $\mathrm{Env}(F)$ over sets F of variables contained in V with restriction maps. It no longer helps to think of $\mathcal{E}$ as a concrete example of a general category, as we make use of its particular structure. In other words, we have so far no abstract concept for a two-layered Kripke logical relation with varying arity.

For each object in $\mathcal{V}$, that is, for each set F of variables contained in V, we want a two-layered logical relation $(R^F_\Sigma)_\Sigma$ of arity $\left|\prod_{x^\sigma\in F} A_\sigma\right|$. (Elements from the set $\prod_{x^\sigma\in F} A_\sigma$ serve a double purpose. We use them to index elements from the relations and we use them as environments. From now on, we will always use the letter μ to denote them.) Since we have restricted the set of variables available we cannot allow arbitrary types Σ to occur, only those $\Sigma = B \to \sigma$ for which the sequence $B = x_1^{\sigma_1}\dots x_n^{\sigma_n}$ contains each variable at most once and all variables are contained in V. Let us call such sequences and types built from them *non-repeating* and let $\mathbb{T}_1(V)$ stand for the collection of all non-repeating types over V. We will also allow ourselves to treat B as a set sometimes, just to keep the complexity of our formulas within manageable range.

Definition 10. Let $(R^F_\Sigma)^{F\subseteq V}_{\Sigma\in\mathbb{T}_1(V)}$ be a family of relations such that the following conditions are satisfied:

(1) $\forall\sigma\to\tau \in \mathbb{T}_0.R^F_{\sigma\to\tau} \subseteq \{(f_\mu)_{\mu\in\mathrm{Env}(F)} \in A^{\mathrm{Env}(F)}_{\sigma\to\tau} \mid \forall\Sigma \in \mathbb{T}_1(V).(e(\Sigma) = \sigma \Longrightarrow \forall(x_\mu)_{\mu\in\mathrm{Env}(F)} \in R^F_\Sigma.(f_\mu(x_\mu))_{\mu\in\mathrm{Env}(F)} \in R^F_\tau)\}$,

(2) $\forall\Sigma \in \mathbb{T}_1(V)$ where $\Sigma = B \to \tau$ and $B = x_1^{\sigma_1}\dots x_n^{\sigma_n}, n \geq 1$ it is the case that $R^F_\Sigma \supseteq \{(f_\mu)_{\mu\in\mathrm{Env}(F)} \in A^{\mathrm{Env}(F)}_{e(\Sigma)} \mid f_\mu = f_{\mu'}$ if $\mu|_{F\setminus B} = \mu'|_{F\setminus B}$ and $\forall(a^1_\mu)_{\mu\in\mathrm{Env}(F\cup B)} \in R^{F\cup B}_{\sigma_1},\dots,\forall(a^n_\mu)_{\mu\in\mathrm{Env}(F\cup B)} \in R^{F\cup B}_{\sigma_n}.$ $(f_{\mu|_F}(a^1_\mu)\dots(a^n_\mu))_{\mu\in\mathrm{Env}(F\cup B)} \in R^{F\cup B}_\tau\}$,

(3) $\forall\sigma \in \mathbb{T}_0\forall F \subseteq F' \subseteq V.(x_\mu)_{\mu\in\mathrm{Env}(F)} \in R^F_\sigma \Longrightarrow (x_{\mu'|_F})_{\mu'\in\mathrm{Env}(F')} \in R^{F'}_\sigma$.

If these three conditions are satisfied then we call the family $(R^F_\Sigma)^{F\subseteq V}_{\Sigma\in\mathbb{T}_1(V)}$ a *two-layered Kripke logical relation with varying arity over* V.

We need to check carefully whether the Fundamental Theorem remains valid:

Theorem 11. *Let $M \equiv \lambda x_1^{\sigma_1}\dots x_n^{\sigma_n}.N$ be a lambda term in normal form and of type $\sigma_1 \to \dots \to \sigma_n \to \tau$ such that N is not an abstraction, let $F \subseteq V \subseteq$ Var be sets of variables such that all variables occurring in M are contained in V and such that all its free variables are contained in F, let $(R^F_\Sigma)^{F\subseteq V}_{\Sigma\in\mathbb{T}_1(V)}$ be a two-layered Kripke logical relation with varying arity over V and, finally, let $(\rho_\mu)_{\mu\in Env(F)}$ be a family of environments such that for all $x^\sigma \in FV(M)$, $(\rho_\mu(x^\sigma))_{\mu\in Env(F)}$ is in R^F_σ. Then the tuple $([\![M]\!]\rho_\mu)_{\mu\in Env(F)}$ is in $R^F_{x_1^{\sigma_1}\dots x_n^{\sigma_n}\to\tau}$.*

Proof. The proof is by induction on the complexity of M, simultaneously for all appropriate F, V, and $(\rho_\mu)_{\mu\in\mathrm{Env}(F)}$. The situation is trivial as usual for variables. If $M \equiv M_1M_2$ is an application then because M is in normal form, M_1 is not an abstraction. The free variables of M_1 and M_2 are also contained in F. We can therefore apply the induction hypothesis and get that $([\![M_1]\!]\rho_\mu)_{\mu\in\mathrm{Env}(F)}$ is in $R^F_{\sigma\to\tau}$ and $([\![M_2]\!]\rho_\mu)_{\mu\in\mathrm{Env}(F)}$ is in R^F_Σ where $e(\Sigma) = \sigma$. So $([\![M]\!]\rho_\mu)_{\mu\in\mathrm{Env}(F)}$ is in R^F_τ by part (1) of the definition.

Let now $M \equiv \lambda x_1^{\sigma_1}\dots x_n^{\sigma_n}.N$ be an abstraction, that is, $n \geq 1$. We want to see that $([\![M]\!]\rho_\mu)_{\mu\in\mathrm{Env}(F)}$ is in $R^F_{B\to\tau}$ where we have introduced B as an abbreviation for $x_1^{\sigma_1}\dots x_n^{\sigma_n}$. Let F' stand for $F \cup B$. By part (2) of our definition we have to supply the functions $[\![M]\!]\rho_\mu$ with arguments $(a^i_\mu)_{\mu\in\mathrm{Env}(F')}$ from $R^{F'}_{\sigma_i}$, $i = 1,\dots,n$. But since $([\![M]\!]\rho_{\mu|_F}(a^1_\mu)\dots(a^n_\mu))_{\mu\in\mathrm{Env}(F')}$ equals $([\![N]\!]\rho_{\mu|_F}[x_1 \mapsto a^1_\mu,\dots,x_n \mapsto a^n_\mu])_{\mu\in\mathrm{Env}(F')}$ we may apply the induction hypothesis to N, F', and $(\rho_{\mu|_F}[x_1 \mapsto a^1_\mu,\dots,x_n \mapsto a^n_\mu])_{\mu\in\mathrm{Env}(F')}$. That the new family of environments meets the requirements of the theorem is a consequence of the persistency part (3) of our definition. □

5.3 The term relation

As usual, a term construction will give us completeness of the characterization. So fix a set V of variables and let $(T^F_\Sigma)^{F\subseteq V}_{\Sigma\in\mathbb{T}_1(V)}$ be the family of relations for which each T^F_Σ is the collection of all $(f_\mu)_{\mu\in\mathrm{Env}(F)} \in A_\Sigma^{\mathrm{Env}(F)}$ definable by lambda terms, i.e., $(f_\mu)_{\mu\in\mathrm{Env}(F)}$ is in T^F_Σ if there exists a lambda term $M \equiv \lambda x_1^{\sigma_1}\dots x_n^{\sigma_n}.N$ (N not an abstraction) in normal form all of whose variables belong to V, all of whose free variables belong to F, for which $x_1^{\sigma_1}\dots x_n^{\sigma_n} = B$ is non-repeating and $B \to \tau = \Sigma$ such that $\forall\mu \in \mathrm{Env}(F)$ we have $f_\mu = [\![M]\!]\mu$. We have to check that we get a valid relation this way.

Lemma 12. *$(T^F_\Sigma)^{F\subseteq V}_{\Sigma\in\mathbb{T}_1(V)}$ is a two-layered Kripke logical relation with varying arity over V.*

Proof. (1) Let $(f_\mu)_{\mu\in\mathrm{Env}(F)}$ be in $T^F_{\sigma\to\tau}$. Then this tuple is given by a term M of type $\sigma\to\tau$ which is not an abstraction. Let further N be term which defines a tuple $(x_\mu)_{\mu\in\mathrm{Env}(F)}$ from T^F_Σ where $e(\Sigma)=\sigma$. Then MN is in normal form and defines $(f_\mu(x_\mu))_{\mu\in\mathrm{Env}(F)}$.

(2) Let $(f_\mu)_{\mu\in\mathrm{Env}(F)}$ be an element from the right hand side of (2) in Definition 10. We can apply it to the tuples defined by $x_1^{\sigma_1},\dots,x_n^{\sigma_n}$ and the result $(f_{\mu|_F}([\![x_1]\!]\mu)\dots([\![x_n]\!]\mu))_{\mu\in\mathrm{Env}(F')}$ will be in $T^{F'}_\tau$, hence given by a lambda term M which is not an abstraction. We claim that $(f_\mu)_{\mu\in\mathrm{Env}(F)}$ is given by $\lambda x_1^{\sigma_1}\dots x_n^{\sigma_n}.M$. Indeed, if $a_1\in A_{\sigma_1},\dots,a_n\in A_{\sigma_n}$ are arguments for the function f_μ then

$$\begin{aligned} f_\mu(a_1)\dots(a_n) &= f_{\mu'|_F}([\![x_1]\!]\mu')\dots([\![x_n]\!]\mu') \\ &= [\![M]\!]\mu' \\ &= [\![\lambda x_1^{\sigma_1}\dots x_n^{\sigma_n}.M]\!]\mu(a_1)\dots(a_n) \end{aligned}$$

where $\mu'(y) = \begin{cases} a_i & \text{if } y\equiv x_i; \\ \mu(y) & \text{otherwise.}\end{cases}$ Here we have used the fact that $f_\mu = f_{\mu'|_F}$ because μ and $\mu'|_F$ differ only at variables from $F\cap\{x_1,\dots,x_n\}$. Also the fact that $x_1^{\sigma_1}\dots x_n^{\sigma_n}$ is non-repeating is crucial here.

(3) Persistency is clear as the denotation of a term only depends on its free variables. □

Theorem 13. *A functional is definable from a fixed set V of variables if and only if it is invariant under all two-layered Kripke logical relations with varying arity over V.*

5.4 Decidability

We are now ready to harvest the fruit from our hard labor in this section. Unlike for full definability, the notion of definability from a fixed set of variables becomes decidable if we restrict to finite ground sets A_σ and finite sets V of variables. The reason for this simply is that there are only finitely many relations to check. Thus we have:

Theorem 14. *The problem whether a given functional from a hereditarily finite Henkin model is lambda definable by a term over a fixed finite set of variables is decidable.*

Although two-layered Kripke logical relations with varying arity over some set of variables may seem complicated, there is nevertheless a fairly simple underlying idea. The relations can be thought of as value tables for functionals where the new types $\Sigma\in\mathbb{T}_1$ and the restrictions to subsets F of V are just a way of keeping track of free and bound variables in defining terms. (Note that a variable may be re-used several times, that is, may occur both bound and free.) Gordon Plotkin has suggested to us that one may obtain Theorem 14 by working backwards from the given value table for a functional in search for a defining term. At each stage, one determines the *set* of value tables which, applied in the right order, give a value table in the set of sought after tables. Each branch of the search stops if either a projection (which corresponds to a variable) can satisfy the requirements or if only tables occur which we are looking for already. Since the set of variables is restricted the tables are finite objects and the search must eventually end. Bookkeeping over free and bound variables is also necessary in this approach and while we haven't formally carried through this approach, we think that it will amount to a scheme with probably the same complexity as ours.

Acknowledgement

The results reported here were obtained while the second author was holding a visiting professorship at Technische Hochschule Darmstadt. We would like to thank Allen Stoughton for directing our attention to Kurt Sieber's paper on sequential logical relations.

The results presented in Sect. 5 were obtained while the first author visited the University of Sussex at the invitation of Matthew Hennessy and Allen Stoughton.

References

1. A. Jung and A. Stoughton. Studying the Fully Abstract Model of PCF within its Continuous Function Model. In this proceedings, 1993.
2. H. Läuchli. An Abstract Notion of Realizability for which Intuitionistic Predicate Calculus is Complete. In A. Kino, J. Myhill, and R. E. Vesley, editors, *Intuitionism and Proof Theory, Proc. summer conference at Buffalo N.Y., 1968*, pages 227–234. North-Holland, 1970.
3. J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic.* Cambridge Studies in Advanced Mathematics Vol. 7. Cambridge University Press, 1986.
4. J. C. Mitchell. Type Systems for Programming Languages. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 365–458. North Holland, 1990.
5. J.C. Mitchell and E. Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51:99–124, 1991. Preliminary version in *Proc. IEEE Symp. on Logic in Computer Science,* 1987, pages 303–314.
6. G. D. Plotkin. Lambda-Definability and Logical Relations. Memorandum SAI-RM-4, University of Edinburgh, October 1973.
7. G. D. Plotkin. Lambda-Definability in the Full Type Hierarchy. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 363–373. Academic Press, London, 1980.
8. K. Sieber. Reasoning about Sequential Functions via Logical Relations. In M. P. Fourman, P. T. Johnstone, and A. M. Pitts, editors, *Proc. LMS Symposium on Applications of Categories in Computer Science, Durham 1991*, volume 177 of *LMS Lecture Note Series*, pages 258–269. Cambridge University Press, 1992.

9. R. Statman. Completeness, Invariance and λ-definability. *Journal of Symbolic Logic*, 47:17–26, 1982.
10. R. Statman. Embeddings, Homomorphisms and λ-definability. Manuscript, Rutgers University, 1982.
11. R. Statman. λ-definable Functionals and $\beta\eta$ Conversion. *Arch. Math. Logik*, 23:21–26, 1983.
12. R. Statman. Logical Relations and the Typed λ-Calculus. *Information and Control*, 65:85–97, 1985.
13. R. Statman and G. Dowek. On Statman's Finite Completeness Theorem. Technical Report CMU-CS-92-152, Carnegie Mellon University, 1992.