# Semantics of Higher-Order Recursion Schemes

Jiří Adámek[1,⋆], Stefan Milius[1,⋆], and Jiří Velebil[2,⋆⋆]

[1] Institut für Theoretische Informatik, Technische Universität Braunschweig, Germany
[2] Faculty of Electrical Engineering, Czech Technical University of Prague, Czech Republic

**Abstract.** Higher-order recursion schemes are equations defining recursively new operations from given ones called "terminals". Every such recursion scheme is proved to have a least interpreted semantics in every Scott's model of $\lambda$-calculus in which the terminals are interpreted as continuous operations. For the uninterpreted semantics based on infinite $\lambda$-terms we follow the idea of Fiore, Plotkin and Turi and work in the category of sets in context, which are presheaves on the category of finite sets. Whereas Fiore *et al* proved that the presheaf $F_\lambda$ of $\lambda$-terms is an initial $H_\lambda$-monoid, we work with the presheaf $R_\lambda$ of rational infinite $\lambda$-terms and prove that this is an initial iterative $H_\lambda$-monoid. We conclude that every guarded higher-order recursion scheme has a unique uninterpreted solution in $R_\lambda$.

*Key Words:* Higher-order recursion schemes, infinite $\lambda$-terms, sets in context, rational tree

## 1 Introduction

Recursion is a fundamental tool for constructing new programs: given a collection $\Sigma$ of existing programs of given types (that is, a many-sorted signature of "terminals") one defines new typed programs $p_1, \ldots, p_n$ (a many-sorted signature of "nonterminals") using symbols from $\Sigma$ and $\{p_i\}$. If the recursion only concerns application, we can formalize this procedure as a collection of equations

$$p_i = f_i \qquad (i = 1, \ldots, n) \tag{1.1}$$

whose right-hand sides $f_i$ are terms in the signature of all terminals and all nonterminals. Such collections are called (first-order) *recursion schemes* and were studied in 1970's by various authors, e.g. B. Courcelle, M. Nivat and I. Guessarian (see the monograph [8] and references there) or S. J. Garland and D. C. Luckham [7]. Recently, a categorical approach to semantics of first-order recursion schemes was treated by S. Milius and L. Moss [13]. In the present paper we take a first step in an analogous approach to the semantics of *higher-order recursion schemes* in which $\lambda$-abstraction is

⋆ Supported by the German Research Foundation (DFG) under the project "Coalgebraic Specification".
⋆⋆ Supported by the grant MSM 6840770014 of the Ministry of Education of the Czech Republic.

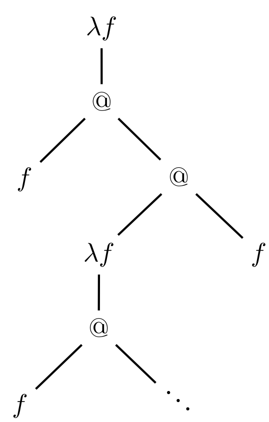
also used as one of the operations. That is, a higher-order recursion scheme, as introduced by W. Damm [4] (see also recent contributions [3] and [14]) is a collection of equations $p_i = f_i$ where $f_i$ are terms using application and $\lambda$-abstraction on symbols of $\Sigma$ and of $\{p_1, \ldots, p_n\}$. As in [13], we first study the uninterpreted semantics, where the given system, is regarded as a purely syntactic construct. At this stage the operation symbols in $\Sigma$ as well as $\lambda$-abstraction and application have no interpretation on actual data. So the semantics is provided by formal (infinite) terms. These terms can be represented by rational trees, i. e., infinite trees having finitely many subtrees. Thus the uninterpreted solution assigns to each of the recursive variables $p_i$ in (1.1) a rational tree $p_i^\dagger$ such that the formal equations become identities if we substitute $p_i^\dagger$ for $p_i$ ($i = 1, \ldots, n$). We assume $\alpha$-conversion (renaming of bound variables) but no other rules in the uninterpreted semantics. We next turn to an interpreted semantics. Here a recursion schemes is given together with an interpretation of all symbols from $\Sigma$ as well as $\lambda$-abstraction and application. We shall consider as an interpretation a Scott model of $\lambda$-calculus as a CPO, $D$. The symbols of $\Sigma$ are interpreted as continuous operations on $D$ and formal $\lambda$-abstraction and application are the actual $\lambda$-abstraction and application in the model $D$. An interpreted solution in $D$ the assigns to each $p_i$ in the context $\Gamma$ of all free variables in (1.1), an element of **CPO**$(D^\Gamma, D)$ (continuously giving to each assignment of free variables in $D^\Gamma$ an element of $D$) such that the formal equations in the recursion scheme become identities in $D$ when the right-hand sides are interpreted in $D$.

*Example 1.1.* The fixed-point operator $Y$ is specified by

$$Y = \lambda f.f(Yf)$$

and the uninterpreted semantics is the rational tree

$Y^\dagger =$



(1.2)

(The symbol @ makes application explicit.) The interpreted solution in $D$ is the least fixed point operator (considered as an element of $D$).

The above example is untyped, and indeed we are treating the untyped case only in the present paper since its uninterpreted semantics is technically simpler than the typed case; however, the basic ideas are similar. In contrast, the interpreted semantics (based on a specified model of $\lambda$-calculus with "terminal" symbols interpreted as operations) is more subtle in the untyped case.

Our main result is that every guarded higher-order recursion scheme has a unique uninterpreted solution, and a least interpreted one. This demonstrates that the methods for iteration in locally finitely presentable categories developed earlier [1] can serve not only for first-order iteration, when applied to endofunctors of **Set**, but also for higher-order iteration: it is possible to apply these methods to other categories, here the category of sets in context.

**Acknowledgments.** The authors are grateful to an anonymous referee and to Thorsten Palm for several suggestions how to improve the presentation of our paper.

## 2 Presheaves as Algebras

**Notation 2.1.** (1) Throughout the paper a given countably infinite set $\mathsf{Var}$ of variables is assumed. Finite subsets $\Gamma \subseteq \mathsf{Var}$ are called *contexts* and form a full subcategory $\mathscr{F}$ of **Set**.

When speaking about formulas in context $\Gamma$ we mean those that have all free variables in $\Gamma$. For example, $\lambda x.yx$ is a formula in context $\Gamma = \{y, y'\}$.

(2) We assume that a one-sorted signature $\Sigma$ is given.

(3) The category $\mathbf{Set}^{\mathscr{F}}$ of "covariant presheaves" on $\mathscr{F}$ is well known to be equivalent to the category of finitary endofunctors of **Set** (but the word endofunctor is later reserved for endofunctors of $\mathbf{Set}^{\mathscr{F}}$ throughout our paper). In fact, every endofunctor $X$ yields the presheaf $X/\mathscr{F}$, and conversely, every presheaf $X$ in $\mathbf{Set}^{\mathscr{F}}$ has a left $\mathrm{Kan}$ extension to a finitary endofunctor of **Set**: for every set $M$ we have

$$X(M) = \bigcup X i_\Gamma \big[X(\Gamma)\big]$$

where the union ranges over embeddings $i_\Gamma \colon \Gamma \hookrightarrow M$ of contexts $\Gamma$ into $M$, and $X i_\Gamma[X(\Gamma)]$ denotes the image of $X i_\Gamma$.

*Example 2.2.* (i) The *presheaf of variables*, $V$, is our name for the embedding $\mathscr{F} \hookrightarrow$ **Set**: $V(\Gamma) = \Gamma$.

(ii) *Free presheaf* on one generator of context $\Gamma$ is our name for the representable presheaf

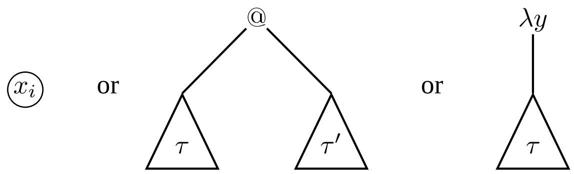
$$\mathscr{F}(\Gamma, -).$$

In fact, the Yoneda lemma states that this presheaf is freely generated by the element $\mathrm{id}_\Gamma$ of context $\Gamma$: for every presheaf $X$ and every $x \in X(\Gamma)$ there exists a unique morphism $f \colon \mathscr{F}(\Gamma, -) \to X$ with $f_\Gamma(\mathrm{id}_\Gamma) = x$.

(iii) The *presheaf* $F_\lambda$ *of (finite)* $\lambda$*-terms* is defined via a quotient since we want to treat $\lambda$-terms always modulo $\alpha$-conversion. As explained in [6], the following approach is equivalent to defining $\lambda$-terms up to $\alpha$-equivalence by de Bruijn levels: We first define the set $F'_\lambda(\Gamma)$ of all finite $\lambda$-trees $\tau$ for every context $\Gamma = \{x_1, \dots, x_n\}$ as

$$\tau ::= x_i \mid \tau @ \tau \mid \lambda y.\tau \qquad (y \in \mathsf{Var} \setminus \Gamma). \tag{2.1}$$

In the graphic form:



(2.2)

We then define the presheaf $F_\lambda$ in every context $\Gamma$ by

$$F_\lambda(\Gamma) = F'_\lambda(\Gamma)/{\sim_\alpha}$$

where $\sim_\alpha$ represents the $\alpha$-conversion: this is the least equivalence with $\lambda x_i.\tau \sim_\alpha \lambda x_j.\tau[{}^{x_j}/{}_{x_i}]$.

We call the congruence classes of finite $\lambda$-trees *finite $\lambda$-terms* (more precisely, we should say finite $\lambda$-terms modulo $\alpha$-conversion). Whereas finite $\lambda$-trees do not form a presheaf, due to possible crashes of bound and free variables, finite $\lambda$-terms do: define $F_\lambda$ on morphisms $\gamma : \Gamma \to \Gamma'$ by choosing a term $t \in F_\lambda(\Gamma)$, relabelling all bound variables so that they do not lie in $\Gamma'$, and denoting by $F\gamma(t)$ the tree obtained by relabelling every free variable $x \in \Gamma$ to $\gamma(x) \in \Gamma'$.

(iv) The *presheaf $F_{\lambda,\Sigma}$ of finite $\lambda$-$\Sigma$-terms* is defined analogously: in (2.1) we just add the term $\sigma(\tau, \dots, \tau)$ for every $\sigma \in \Sigma_n$, and in (2.2) the corresponding tree.

(v) The presheaf $T_\lambda$ of all (finite and infinite) *$\lambda$-terms* is defined analogously to $F_\lambda$. We first denote by $T'_\lambda(\Gamma)$ the set of all trees (2.2) dropping the assumption of finiteness. Then we use $\alpha$-conversion: for infinite trees $t$ and $t'$ we write $t \sim_\alpha t'$ if their (finite) cuttings at level $k$ (with label $\perp$ for all leaves at level $k$) are $\alpha$-equivalent in the above sense for all $k \in \mathbb{N}$. (Put $\Sigma = \{\perp\}$ with $\perp$ a constant symbol). The presheaf $T_\lambda$ is defined on objects $\Gamma$ by $T_\lambda(\Gamma) = T'_\lambda(\Gamma)/{\sim_\alpha}$ and on morphisms $\gamma : \Gamma \to \Gamma'$ by relabelling of variables as in (iii). Observe that since $\mathsf{Var} \setminus \Gamma$ is infinite, the relabelling of bound variables needed here causes no problem.

(v) The presheaf $R_\lambda$ of *rational $\lambda$-terms* is also defined analogously. Recall that a tree is called rational if it has up to isomorphism only finitely many subtrees. We denote by $R'_\lambda(\Gamma)$ the set of all rational trees in $T'_\lambda(\Gamma)$ and define a presheaf $R_\lambda$ by $R_\lambda(\Gamma) = R'_\lambda(\Gamma)/{\sim_\alpha}$ on objects, and by relabellings of variables (as in (iii)) on morphisms.

(vi) The presheaves $T_{\lambda,\Sigma}$ (of $\lambda$-$\Sigma$-terms) and $R_{\lambda,\Sigma}$ (of rational $\lambda$-$\Sigma$-terms) are obvious modifications of (iv) and (v): one adds to (2.2) the case $\sigma(\tau_1, \dots, \tau_n)$ for all $n$-ary symbols $\sigma \in \Sigma$ and all (rational) $\lambda$-$\Sigma$-terms $\tau_1, \dots, \tau_n$. Observe that, by definition, every rational $\lambda$-$\Sigma$-term $t$ is represented by a rational $\lambda$-$\Sigma$-tree. However, $t$ can also be represented by non-rational $\lambda$-$\Sigma$-trees—for example, if it contains infinitely many $\lambda$'s, the $\alpha$-conversion can introduce an infinite number of bound variables.

**Notation 2.3.** We denote by $\delta\colon \mathbf{Set}^{\mathscr{F}} \to \mathbf{Set}^{\mathscr{F}}$ the endofunctor defined by

$$\delta X(\Gamma) = X(\Gamma + 1).$$

Observe that $\delta$ preserves limits and colimits.

Note that an algebra for $\delta$ is a presheaf $Y$ together with an operation $Y(\Gamma + 1) \to Y(\Gamma)$ for all contexts $\Gamma$—this is precisely the form of $\lambda$-abstraction, where to a formula

$f$ in $Y(\Gamma+\{y\})$ we assign $\lambda y.f$ in $Y(\Gamma)$. The other $\lambda$-operation, application, is simply a presheaf morphism $X \times X \to X$, that is, a binary operation on $X$. We put these two together:

**Notation 2.4.** Let $H_\lambda$ denote the endofunctor of **Set**$^{\mathscr{F}}$ given by

$$H_\lambda X = X \times X + \delta X.$$

Thus, an algebra for $H_\lambda$ is a presheaf $X$ together with specified operations of application $X(\Gamma) \times X(\Gamma) \to X(\Gamma)$ and abstraction $X(\Gamma+1) \to X(\Gamma)$ for all contexts $\Gamma$; these operations are compatible with renaming of free variables.

*Example 2.5.* $F_\lambda$, $T_\lambda$ and $R_\lambda$ are algebras for $H_\lambda$ in the obvious sense.

*Remark 2.6.* (i) The slice category $V/$ **Set**$^{\mathscr{F}}$ of presheaves $X$ together with a morphism $i\colon V \to X$ is called the category of *pointed presheaves*. For example $F_\lambda$ is a pointed presheaf in a canonical sense: $i^F\colon V \to F_\lambda$ takes a variable $x$ to the term $x$. Analogously $i^T\colon V \to T_\lambda$ and $i^R\colon V \to R_\lambda$ are pointed presheaves, and so are $F_{\lambda,\Sigma}$, $R_{\lambda,\Sigma}$ and $T_{\lambda,\Sigma}$.

(ii) Recall that the category **Alg** $H_\lambda$ of algebras for $H_\lambda$ has as morphisms the usual $H_\lambda$-homomorphisms, i.e., a morphism from $a\colon H_\lambda X \to X$ to $b\colon H_\lambda Y \to Y$ is a presheaf morphism $f\colon X \to Y$ such that $f\cdot a = b\cdot H_\lambda f$. Then **Alg** $H_\lambda$ is a concrete category over **Set**$^{\mathscr{F}}$ with the forgetful functor $(H_\lambda X \to X) \mapsto X$.

**Theorem 2.7 (see [6]).** *The presheaf $F_\lambda$ of finite $\lambda$-terms is the free $H_\lambda$-algebra on $V$.*

**Definition 2.8 (see [1]).** *Given an endofunctor $H$, an algebra $a\colon HA \to A$ is called* ***completely iterative*** *if for every object $X$ (of variables) and every (flat equation) morphism $e\colon X \to HX+A$ there exists a unique* ***solution*** *which means a unique morphism $e^\dagger\colon X \to A$ such that the square below commutes*

$$\begin{array}{ccc} X & \xrightarrow{\quad e^\dagger \quad} & A \\ {\scriptstyle e}\downarrow & & \uparrow{\scriptstyle [a,\mathrm{id}]} \\ HX + A & \xrightarrow[He^\dagger + \mathrm{id}]{} & HA + A \end{array} \tag{2.3}$$

**Theorem 2.9.** *The presheaf $T_\lambda$ of infinite $\lambda$-terms is the free completely iterative $H_\lambda$-algebra on $V$.*

*Proof.* As proved in [12], Corollary 6.3, the free completely iterative algebra for $H_\lambda$ on $V$ is precisely the terminal coalgebra for $H_\lambda(-) + V$. The latter functor clearly preserves limits of $\omega^{\mathrm{op}}$-chains. Consequently, its terminal coalgebra is a limit of the chain $W$ with $W_0 = 1$ (the terminal presheaf) and $W_{n+1} = H_{\lambda,\Sigma} W_n + V$, where the connecting maps are the unique $w_0\colon W_1 \to W_0$ and $w_{n+1} = H_{\lambda,\Sigma} w_n + \mathrm{id}_V$.

For every context $\Gamma$ identify $W_0(\Gamma)$ with the set $\{\bot\}$ where $\bot \notin \mathsf{Var}$. Then $W_n(\Gamma)$ can be identified with the set of all $\lambda$-terms of depth at most $n$ having all leaves of depth $n$ labelled by $\bot$. And $w_{n+1}$ cuts away the level $n+1$ in trees of $W_{n+1}(\Gamma)$, relabelling level-$n$-leaves by $\bot$. With this identification we obtain $T_\lambda$ as a limit of $W_n$ where the limit maps $T_\lambda \to W_n$ cut trees in $T_\lambda(\Gamma)$ at level $n$ and relabel level-$n$-leaves by $\bot$. ◻

*Remark 2.10.* We are going to characterize the presheaf $R_\lambda$ as a free iterative algebra for $H_\lambda$. This concept differs from 2.8 by admitting only objects $X$ of variables that are *finitely presentable*. This means that the $\hom$-functor $\mathbf{Set}^{\mathscr{F}}(X, -)$ preserves filtered colimits. Recall from [2] that a presheaf $X$ is called *super-finitary* provided that each $X(\Gamma)$ is finite and there exists a nonempty context $\Gamma_0$ *generating* $X$ in the sense that for every nonempty context $\Gamma$ we have

$$X(\Gamma) = \bigcup_{\gamma\colon \Gamma_0 \to \Gamma} X\gamma\big[X(\Gamma_0)\big]. \tag{2.4}$$

*Example 2.11.* A signature $\Sigma$ defines the polynomial presheaf $X_\Sigma$ by $X_\Sigma(\Gamma) = \coprod_{\sigma \in \Sigma} \Gamma^{\mathrm{ar}(\sigma)}$. This is a super-finitary presheaf iff $\Sigma$ is a finite set. Other super-finitary presheaves are precisely the quotients of $X_\Sigma$ with $\Sigma$ finite.

**Theorem 2.12.** *A presheaf in* $\mathbf{Set}^{\mathscr{F}}$ *is finitely presentable iff it is super-finitary.*

*Sketch of proof.* Every finitely presentable presheaf $X$ is proved to be a directed colimit of presheaves generated by $\Gamma_0$ for all possible contexts $\Gamma_0$. From finite presentability it then follows that one $\Gamma_0$ generates all of $X$.

Conversely, for every super-finitary presheaf $X$ a finite diagram of representable presheaves is constructed with colimit $X$, this implies that $X$ is finitely presentable.

□

**Definition 2.13 (see [1]).** *An algebra $A$ for $H$ is called **iterative** if every equation morphism $e\colon X \to HX + A$ with $X$ finitely presentable has a unique solution.*

**Theorem 2.14.** *The presheaf $R_\lambda$ of rational $\lambda$-terms is the free iterative $H_\lambda$-algebra on $V$.*

*Sketch of proof.* The iterativity of $R_\lambda$ follows from that of $T_\lambda$ and the fact that if for an equation morphism $e\colon X \to H_\lambda X + R_\lambda$ the object $X$ is super-finitary, all terms one obtains as solutions in $e^\dagger$ are always rational.

The universal property of $R_\lambda$ is proved on the basis of the construction presented in [1]: all equation morphisms $e\colon X \to H_\lambda X + V$ with $X$ finitely presentable form a filtered diagram whose colimit is the free iterative algebra on $V$. We now prove that $R_\lambda$ is indeed this colimit: for every $e$ we form (using $i^R$ in 2.6(i)) the equation morphism $\tilde{e} = (\mathrm{id}_{H_\lambda X} + i^R) \cdot e\colon X \to H_\lambda X + R_\lambda$, and obtain the unique solution $\tilde{e}^\dagger\colon X \to R_\lambda$; we then verify that the morphisms $\tilde{e}^\dagger$ form the desired colimit cocone. □

*Remark 2.15.* As mentioned in the Introduction we want to combine application and abstraction with other operations. Suppose $\Sigma = (\Sigma)_{n\in\mathbb{N}}$ is a signature (of "terminals"). Then we can form the endofunctor $H_{\lambda,\Sigma}$ of $\mathbf{Set}^{\mathscr{F}}$ on objects by

$$H_{\lambda,\Sigma} X = X \times X + \delta X + \coprod_{n\in\mathbb{N}} \Sigma_n \bullet X^n \tag{2.5}$$

where $\Sigma_n \bullet X^n$ is the coproduct (that is: disjoint union in every context) of $\Sigma_n$ copies of the $n$-th Cartesian power of $X$. For this endofunctor an algebra is an $H_\lambda$-algebra $A$ together with an $n$-ary operation on $A(\Gamma)$ for every $\sigma \in \Sigma_n$ and every context $\Gamma$.

**Theorem 2.16.** *For every signature $\Sigma$*

(i) $F_{\lambda,\Sigma}$ *is the free* $H_{\lambda,\Sigma}$*-algebra on* $V$*,*
(ii) $R_{\lambda,\Sigma}$ *is the free iterative* $H_{\lambda,\Sigma}$*-algebra on* $V$*, and*
(iii) $T_{\lambda,\Sigma}$ *is the free completely iterative* $H_{\lambda,\Sigma}$*-algebra on* $V$*.*

In fact, (i) was proved in [6], and the proofs of (ii) and (iii) are completely analogous to the proofs of Theorems 2.14 and 2.9.

# 3 Presheaves as Monoids

So far we have not treated one of the basic features of $\lambda$-calculus: substitution of subterms. For the presheaf $F_{\lambda,\Sigma}$ of finite $\lambda$-$\Sigma$-terms this was elegantly performed by Fiore *et al* [6] based on the monoidal structure of the category $\mathbf{Set}^{\mathscr{F}}$. As mentioned in 2.1(2), we can work with the equivalent category $\mathrm{Fin}(\mathbf{Set},\mathbf{Set})$ of all finitary endofunctors of **Set**. Composition of functors makes this a (strict, non-symmetric) monoidal category with unit $\mathrm{Id}_{\mathbf{Set}}$. This monoidal structure, as shown in [6], corresponds to simultaneous substitution. In fact, let $X$ and $Y$ be objects of $\mathrm{Fin}(\mathbf{Set},\mathbf{Set})$. Then the "formulas of the composite presheaf $X{\cdot}Y$" in context $\Gamma$ are the elements of

$$X{\cdot}Y(\Gamma) = X\big(Y(\Gamma)\big) = \bigcup_{u:\bar{\Gamma}\hookrightarrow Y(\Gamma)} X(\bar{\Gamma}), \tag{3.1}$$

where $u:\bar{\Gamma}\hookrightarrow Y(\Gamma)$ranges over finite subobjects of $Y(\Gamma)$. In fact, $X$ preserves the filtered colimit $Y(\Gamma) = \operatorname{colim}\bar{\Gamma}$.

That is, in order to specify an $X{\cdot}Y$-formula $t$ in context $\Gamma$ we need (a) an $X$-formula $s$ in some new context $\bar{\Gamma}$ and (b) for every variable $x\in\bar{\Gamma}$ a $Y$-formula of context $\Gamma$, say, $r_x$. We can then think of $t$ as the formula $s(r_x/x)$ obtained from $s$ by simultaneous substitution.

*Remark 3.1.* (i) The monoidal structure on $\mathbf{Set}^{\mathscr{F}}$ corresponding to composition in $\mathrm{Fin}(\mathbf{Set},\mathbf{Set})$ will be denoted by $\otimes$, with the unit $V$, see Notation 2.2(i). Observe that every endofunctor $-\otimes X$ preserves colimits, e.g., $(A+B)\otimes X \cong (A\otimes X)+(B\otimes X)$.

(ii) Explicitly, the monoidal structure can be described by the coend

$$(X\otimes Y)(\Gamma) = \int^{\overline{\Gamma}} \mathbf{Set}(\overline{\Gamma}, Y(\Gamma)) \bullet X(\overline{\Gamma}). \tag{3.2}$$

(iii) Recall that monoids in the monoidal category $\mathrm{Fin}(\mathbf{Set},\mathbf{Set})$ are precisely the finitary monads on **Set**.

(iv) The presheaf $F_{\lambda,\Sigma}$ is endowed with the usual simultaneous substitution of $\lambda$-terms which defines a morphism $m^F\colon F_{\lambda,\Sigma}\otimes F_{\lambda,\Sigma}\to F_{\lambda,\Sigma}$. Together with the canonical pointing $i^F\colon V\to F_{\lambda,\Sigma}$ above this constitutes a monoid.

Analogously the simultaneous substitution of infinite terms defines a monoid $m^T\colon T_{\lambda,\Sigma}\otimes T_{\lambda,\Sigma}\to T_{\lambda,\Sigma}$. It is easy to see that given a rational term, every simultaneous substitution of rational terms for variables yields again a rational term. Thus, we have a submonoid $m^R\colon R_{\lambda,\Sigma}\otimes R_{\lambda,\Sigma}\to R_{\lambda,\Sigma}$ of $T_{\lambda,\Sigma}$.

(v) The monoidal operation of $F_{\lambda,\Sigma}$ is well connected to the structure of $H_{\lambda,\Sigma}$-algebra. This was expressed in [6] by the concept of an $H_{\lambda,\Sigma}$-*monoid*.

In order to recall this concept, we need the notion of point-strength introduced in [5] under the name $(I/\mathscr{W})$-strength; this is a weakening of the classical strength (necessary since $H_{\lambda,\Sigma}$ is unfortunately not strong). Recall that objects of the slice category $I/\mathscr{W}$ are morphisms $x\colon I \to X$ for $X \in \operatorname{obj}\mathscr{W}$.

**Definition 3.2 (see [5]).** *Let $(\mathscr{W}, \otimes, I)$ be a strict monoidal category, and $H$ an endofunctor of $\mathscr{W}$. A **point-strength** of $H$ is a collection of morphisms*

$$s_{(X,x)(Y,y)}\colon HX \otimes Y \to H(X \otimes Y)$$

*natural in $(X,x)$ and $(Y,y)$ ranging through $I/\mathscr{W}$ such that*

(i) $s_{(X,x)(I,\mathrm{id})} = \mathrm{id}_{HX}$, *and*
(ii) $s_{(X,x)(Y\otimes Z, y\otimes z)} = s_{(X\otimes Y, x\otimes y)(Z,z)} \cdot \big(s_{(X,x)(Y,y)} \otimes \mathrm{id}_Z\big)$.

*Example 3.3.* (i) The endofunctor $X \mapsto X \otimes X$ (which usually fails to be strong) has the point-strength

$$s_{(X,x)(Y,y)} = (X\otimes X)\otimes Y = (X \otimes I \otimes X)\otimes Y \xrightarrow{\mathrm{id}_X \otimes y \otimes \mathrm{id}_{X\otimes Y}} (X\otimes Y)\otimes(X\otimes Y).$$

(ii) The endofunctor $X \mapsto X^n$ of $\mathbf{Set}^{\mathscr{F}}$ is clearly (point-)strong for every $n \in \mathbb{N}$.

(iii) The functor $\delta$ in 2.3 is point-strong, as observed in [6]. The easiest way to describe its point-strength is by working in $\mathrm{Fin}(\mathbf{Set}, \mathbf{Set})$. Given pointed objects $(X,x)$, $(Y,y)$, then the point-strength $s_{(X,x)(Y,y)}\colon (\delta X)\cdot Y \to \delta(X\cdot Y)$ has components

$$X\big(Y(\Gamma)+1\big) \xrightarrow{X(\mathrm{id}+y_1)} X\big(Y(\Gamma)+Y(1)\big) \xrightarrow{X\,\mathsf{can}} X\cdot Y(\Gamma+1),$$

where $\mathsf{can}\colon Y(\Gamma)+Y(1) \to Y(\Gamma+1)$ denotes the canonical morphism.

(iv) A coproduct of point-strong functors is point-strong.

**Corollary 3.4.** *The endofunctors $H_\lambda$ and $H_{\lambda,\Sigma}$ are point-strong. Their point-strength is denoted by $s^H$.*

**Definition 3.5 (see [6]).** *Let $H$ be a point-strong endofunctor of a monoidal category. By an $H$-**monoid** is meant an $H$-algebra $(A,a)$ which is also a monoid $m\colon A\otimes A \to A$ and $i\colon I \to A$ such that the square below commutes:*

$$\begin{array}{ccccc} HA\otimes A & \xrightarrow{s_{(A,i)(A,i)}} & H(A\otimes A) & \xrightarrow{Hm} & HA \\ \big\downarrow{\scriptstyle a\otimes \mathrm{id}} & & & & \big\downarrow{\scriptstyle a} \\ A\otimes A & & \xrightarrow{\quad m \quad} & & A \end{array} \tag{3.3}$$

**Theorem 3.6 (see [6]).** *The presheaf $F_{\lambda,\Sigma}$ of finite $\lambda$-$\Sigma$-terms is the initial $H_{\lambda,\Sigma}$-monoid.*

*Example 3.7.* Although $F_\lambda$ is the free $H_\lambda$-algebra on $V$, see Theorem 2.7, it is not in general true that $F_\lambda \otimes Z$ is the free $H_\lambda$-algebra on a presheaf $Z$. For example, if $Z = V \times V$, then terms in $(F_\lambda \otimes Z)(\Gamma)$ are precisely the finite $\lambda$-terms whose free variables are substituted by pairs in $\Gamma \times \Gamma$. In contrast, the free $H_\lambda$-algebra on $V \times V$ contains in context $\Gamma$ also terms such as $\lambda x.(x, y)$ for $y \in \Gamma$, that is, in variable pairs one member can be bound and one free.

**Theorem 3.8 (see [11]).** *The presheaf $T_{\lambda,\Sigma}$ of $\lambda$-$\Sigma$-terms is an $H_{\lambda,\Sigma}$-monoid with simultaneous substitution as monoid structure.*

Although in [11], Example 13, just $T_\lambda$ is used, the methods of that paper apply to $T_{\lambda,\Sigma}$ immediately.

**Theorem 3.9.** *The presheaf $T_{\lambda,\Sigma}$ of $\lambda$-$\Sigma$-terms is the initial completely iterative $H_{\lambda,\Sigma}$-monoid. That is, for every $H_{\lambda,\Sigma}$-monoid $A$ whose $H_{\lambda,\Sigma}$-algebra structure is completely iterative there exists a unique monoid homomorphism from $T_{\lambda,\Sigma}$ to $A$ which is also an $H_{\lambda,\Sigma}$-homomorphism.*

*Sketch of proof.* Given a completely iterative algebra $a\colon H_{\lambda,\Sigma}A \to A$ which is also a monoid with $m\colon A \otimes A \to A$ and $i\colon V \to A$, and assuming that (3.3) commutes, we have, due to 2.16, a unique homomorphism $h\colon T_{\lambda,\Sigma} \to A$ of $H_{\lambda,\Sigma}$-algebras with $h \cdot i^T = i$. Thus, it remains to prove that $h$ is a monoid homomorphism, that is,

$$h \cdot m^T = m \cdot (h \otimes h)\colon T_{\lambda,\Sigma} \otimes T_{\lambda,\Sigma} \to A. \tag{3.4}$$

To see this we define an equation morphism $e$ with the object $T_{\lambda,\Sigma} \otimes T_{\lambda,\Sigma}$ of variables as follows: we first observe that $T_{\lambda,\Sigma} \cong H_{\lambda,\Sigma}T_{\lambda,\Sigma} + V$ (recall that $T_{\lambda,\Sigma}$ is the terminal coalgebra for $H_{\lambda,\Sigma}(-) + V$, see the proof of Theorem 2.9) and then derive $T_{\lambda,\Sigma} \otimes T_{\lambda,\Sigma} \cong (H_{\lambda,\Sigma}T_{\lambda,\Sigma}) \otimes T_{\lambda,\Sigma} + T_{\lambda,\Sigma}$, see 3.1(i). Put

$$e \equiv (H_{\lambda,\Sigma}T_{\lambda,\Sigma}) \otimes T_{\lambda,\Sigma} + T_{\lambda,\Sigma} \xrightarrow{s^H + h} H_{\lambda,\Sigma}(T_{\lambda,\Sigma} \otimes T_{\lambda,\Sigma}) + A$$

and recall that $e$ has a unique (!) solution $e^\dagger\colon T_{\lambda,\Sigma} \otimes T_{\lambda,\Sigma} \to A$. The verification of (3.4) is performed by proving that both sides are solutions of $e$. $\square$

*Remark 3.10.* The presheaf $R_{\lambda,\Sigma}$ is an $H_{\lambda,\Sigma}$-monoid: this is a submonoid of $T_{\lambda,\Sigma}$, and the verification of (3.3) thus follows from that for $T_{\lambda,\Sigma}$.

**Theorem 3.11.** *The presheaf $R_{\lambda,\Sigma}$ of rational $\lambda$-$\Sigma$-terms is the initial iterative $H_{\lambda,\Sigma}$-monoid.*

*Sketch of proof.* For the universal property of being the initial iterative monoid we need to work with the free iterative monad on $H_{\lambda,\Sigma}$ which is the monad $(\mathbb{R}_{\lambda,\Sigma}, \mu^R, \eta^R)$ of free iterative algebras for $H_{\lambda,\Sigma}$. In fact, since $H_{\lambda,\Sigma}$ is clearly a finitary endofunctor of the locally finitely presentable category $\mathbf{Set}^{\mathscr{F}}$, it follows from results of [1] that every presheaf $A$ generates a free iterative algebra $\mathbb{R}_{\lambda,\Sigma}(A)$. We denote its algebra structure by $\varrho_A\colon H_{\lambda,\Sigma}(\mathbb{R}_{\lambda,\Sigma}(A)) \to \mathbb{R}_{\lambda,\Sigma}(A)$ with the universal arrow $\eta^R_A\colon A \to \mathbb{R}_{\lambda,\Sigma}(A)$. In particular, by Theorem 2.14 we have

$$R_{\lambda,\Sigma} = \mathbb{R}_{\lambda,\Sigma}(V). \tag{3.5}$$

We now prove that the endofunctor $\mathbb{R}_{\lambda,\Sigma}$ of $\mathbf{Set}^{\mathscr{F}}$ has a point-strength

$$s^R_{(X,x)(Y,y)} \colon \mathbb{R}_{\lambda,\Sigma}(X) \otimes Y \to \mathbb{R}_{\lambda,\Sigma}(X \otimes Y)$$

such that the monoid structure $m^R \colon R_{\lambda,\Sigma} \otimes R_{\lambda,\Sigma} \to R_{\lambda,\Sigma}$ is simply

$$m^R = \mu^R_V \cdot s_{(V,\mathrm{id})(R_{\lambda,\Sigma}, i^R)}. \tag{3.6}$$

Moreover, we know from [1] that the morphisms $\varrho_A$ and $\eta^R_A$ form coproduct injections of

$$\mathbb{R}_{\lambda,\Sigma}(A) = H_{\lambda,\Sigma}\big(\mathbb{R}_{\lambda,\Sigma}(A)\big) + A \tag{3.7}$$

and one then proves that this is related to the strength $s^R$ by

$$s^R_{(X,x)(Y,y)} = H_{\lambda,\Sigma} s^R_{(X,x)(Y,y)} + \mathrm{id}_{X \otimes Y}\,. \tag{3.8}$$

Based on this equation, we prove that for every $H_{\lambda,\Sigma}$-monoid $(A, a, m, i)$ such that the algebra $(A, a)$ is iterative the unique homomorphism $h \colon R_{\lambda,\Sigma} \to A$ of $H_{\lambda,\Sigma}$-algebras with $h \cdot i^R = i$ (see Theorem 2.16(ii)) preserves monoid multiplication. This proof is based on the construction of $\mathbb{R}_{\lambda,\Sigma}(A)$ in [1] and is rather technical. $\square$

## 4 Higher-Order Recursion Schemes

We can reformulate and extend higher-order recursion schemes (1.1) categorically:

**Definition 4.1.** *A **higher-order recursion scheme** on a signature $\Sigma$ (of "terminals") is a presheaf morphism*

$$e \colon X \to F_{\lambda,\Sigma} \otimes (X + V) \tag{4.1}$$

*where $X$ is a finitely presentable presheaf.*

*Remark 4.2.* (i) The presheaf $F_{\lambda,\Sigma} \otimes (X + V)$ assigns to a context $\Gamma$ the set $F_{\lambda,\Sigma}(X(\Gamma) + \Gamma)$ of finite $\lambda$-terms in contexts $\overline{\Gamma} \subseteq X(\Gamma) + \Gamma$.

(ii) In the introduction we considered, for a given context

$$\Gamma_{nt} = \{p_1, \ldots, p_n\}$$

of "nonterminals", a system of equations $p_i = f_i$ where $f_i$ is a $\lambda$-$\Sigma$-term in some context $\Gamma_0 = \{x_1, \ldots, x_k\}$. Let $X$ be the free presheaf in $n$ generators $p_1, \ldots, p_n$ of context $\Gamma_0$ (a coproduct of $n$ copies of $\mathscr{F}(\Gamma_0, -)$, see Example 2.2(ii)). Then the system of equations defines the unique morphism

$$e \colon X \to F_{\lambda,\Sigma} \otimes (X + V)$$

assigning to every $p_i$ the right-hand side $f_i$ lying in

$$F_{\lambda,\Sigma}(\Gamma_{nt} + \Gamma_0) \subseteq F_{\lambda,\Sigma}(X(\Gamma_0) + \Gamma_0)$$

where we once again consider $F_{\lambda,\Sigma}$ as an object of $\mathrm{Fin}(\mathbf{Set}, \mathbf{Set})$.

(iii) Conversely, every morphism (4.1) yields a system of equations $p_i = f_i$ as follows: let $\Gamma_0$ fulfill (2.4) in Remark 2.10, and define $\Gamma_{nt} = X(\Gamma_0)$. The element $f_p = e_{\Gamma_0}(p)$ lies, for every nonterminal $p \in \Gamma_{nt}$, in $F_{\lambda,\Sigma}(\Gamma_{nt} + \Gamma_0)$. We obtain a system of equations $p = f_p$ describing the given morphism $e$.

(iv) We will use the presheaf $R_{\lambda,\Sigma}$ for our uninterpreted solutions of recursion schemes:

A solution of the system of (formal) equations $p_i = f_i$ are rational terms $p_1^\dagger, \dots, p_n^\dagger$ making those equations identities in $R_{\lambda,\Sigma}(\Gamma_0)$ when we substitute in $f_i$ the terms $p_j^\dagger$ for the nonterminals $p_j$ ($j = 1, \dots, n$). This is expressed by the Definition 4.3.

(v) The general case of "equation morphisms" as considered in [1] is (for the endofunctor $H_{\lambda,\Sigma}$) a morphism of type $e : X \to \mathbb{R}_{\lambda,\Sigma}(X + V)$. Then we see that every higher-order recursion scheme gives an equation morphism via the inclusion $F_{\lambda,\Sigma} \hookrightarrow R_{\lambda,\Sigma}$ and the strength of the monad $\mathbb{R}_{\lambda,\Sigma}$ (but not necessarily conversely). Our solution theorem below is an application of the general result of [1].

**Definition 4.3.** *A **solution** of a higher-order recursion scheme $e \colon X \to F_{\lambda,\Sigma} \otimes (X + V)$ is a morphism $e^\dagger \colon X \to R_{\lambda,\Sigma}$ such that the square below, where $j \colon F_{\lambda,\Sigma} \to R_{\lambda,\Sigma}$ denotes the embedding, commutes:*

$$
\begin{array}{ccc}
X & \xrightarrow{\quad e^\dagger \quad} & R_{\lambda,\Sigma} \\
{\scriptstyle e}\downarrow & & \uparrow{\scriptstyle m^R} \\
F_{\lambda,\Sigma} \otimes (X+V) & & \\
{\scriptstyle j\otimes(X+V)}\downarrow & & \\
R_{\lambda,\Sigma} \otimes (X+V) & \xrightarrow[R_{\lambda,\Sigma}\otimes[e^\dagger, i^R]]{} & R_{\lambda,\Sigma} \otimes R_{\lambda,\Sigma}
\end{array}
$$

*Example 4.4.* The fixed-point combinator (see Example 1.1) with $\Sigma = \emptyset$ defines $e$ whose domain is the terminal presheaf 1, that is, $e \colon 1 \to F_\lambda \otimes (1 + V)$. The solution $e^\dagger \colon 1 \to R_\lambda$ assigns to the unique element of 1 the tree (1.2).

*Remark 4.5.* Recursion schemes such as $p_1 = p_1$ make no sense—and certainly fail to have a unique solution. In general, we want to avoid right-hand sides of the form $p_i$. A recursion scheme is called *guarded* if no right-hand side lies in $\Gamma_{nt}$. (Theorem 4.7 below shows that no other restrictions are needed.) Guardedness can be formalized as follows: since

$$R_{\lambda,\Sigma} = H_{\lambda,\Sigma}(R_{\lambda,\Sigma}) + V \qquad \text{with injections } \varrho_V \text{ and } i^R$$

by (3.7), we have (see 3.1(i))

$$R_{\lambda,\Sigma} \otimes (X + V) \cong H_{\lambda,\Sigma}(R_{\lambda,\Sigma}) \otimes (X + V) + X + V$$

with coproduct injections $\varrho_V \otimes \mathrm{id}_{X+V}$ and $i^R \otimes \mathrm{id}_{X+V}$. Then $e$ is guarded if its extension $(j \otimes (X+V)) \cdot e \colon X \to R_{\lambda,\Sigma} \otimes (X + V)$ factorizes through the embedding of the first and third summand of this coproduct:

**Definition 4.6.** *A higher-order recursion scheme* $e\colon X \to F_{\lambda,\Sigma} \otimes (X+V)$ *is called* ***guarded*** *if* $(j \otimes (X+V))\cdot e$ *factorizes through*

$$\left[\varrho \otimes \mathrm{id}, (i^R \otimes \mathrm{id})\cdot \mathsf{inr}\right]\colon H_\lambda(R_{\lambda,\Sigma}) \otimes (X+V) + V \to R_{\lambda,\Sigma} \otimes (X+V).$$

**Theorem 4.7.** *Every guarded higher-order recursion scheme has a unique solution.*

*Remark.* In Definition 4.1 we restricted high-order recursion schemes to have $F_{\lambda,\Sigma}$ in their codomain. This corresponds well to the classical notion of recursion schemes as explained in Remark 4.2. Moreover, this leads to a simple presentation of the interpreted semantics in Section 5 below. However, Theorem 4.7 remains valid if we replace $F_{\lambda,\Sigma}$ by $R_{\lambda,\Sigma}$ in Definition 4.1 and define solution by $e^\dagger = m^R \cdot R_{\lambda,\Sigma} \otimes [e^\dagger, i^R] \cdot e$. This extends the notion of a higher-order recursion scheme (1.1) to allow the right-hand sides $f_i$ to be rational $\lambda$-$\Sigma$-terms. We shall prove Theorem 4.7 working with higher-order schemes of the form $e\colon X \to R_{\lambda,\Sigma} \otimes (X+V)$, $X$ finitely presentable. We call $e$ guarded if it factorizes through $[\varrho \otimes \mathrm{id}, (i^R \otimes \mathrm{id})\cdot \mathsf{inr}]$.

*Sketch of proof.* Here we work with the free iterative monad $\mathbb{R}_{\lambda,\Sigma}$ generated by the endofunctor $H_{\lambda,\Sigma}$ on $\mathbf{Set}^{\mathscr{F}}$, see 3.11. By (3.5) we have $R_{\lambda,\Sigma} = \mathbb{R}_{\lambda,\Sigma}(V)$. For every guarded recursion program scheme $e\colon X \to \mathbb{R}_{\lambda,\Sigma}(V) \otimes (X+V)$ one constructs a guarded rational equation morphism $\bar{e}\colon X \to \mathbb{R}_{\lambda,\Sigma}(X+V)$ for the monad $\mathbb{R}_{\lambda,\Sigma}$ in the sense of [1]. Since guarded rational equation morphisms have unique solutions for this monad, the proof is finished by verifying that $e$ and $\bar{e}$ have the same solutions. $\square$

## 5 Interpreted Solutions

In the uninterpreted semantics of higher-order recursion schemes $\lambda$-abstraction and application are only syntactic operations. Therefore terms such as $f$ and $\lambda x.fx$ are unrelated. This is not satisfactory, so we turn to interpreted semantics where the $\beta$ and $\eta$ conversions hold. For that we need an interpretation of the $\lambda$-calculus plus an interpretation of the terminals in $\Sigma$.

We denote by **CPO** the cartesian closed category of posets with directed joins and continuous functions. We assume that a Scott model $D$ of $\lambda$-calculus is given, i.e., a CPO with $\bot$ and with an embedding-projection pair

$$\mathsf{fold} : \mathbf{CPO}(D,D) \lhd D : \mathsf{unfold} \tag{5.1}$$

together with continuous operations

$$\sigma^D : D^n \to D \quad \text{for every } n\text{-ary } \sigma \text{ in } \Sigma.$$

We then work with the presheaf $\langle D,D\rangle$ defined by

$$\langle D,D\rangle\Gamma = \mathbf{CPO}(D^\Gamma, D)$$

as our interpretation object. Observe that elements of $\langle D,D\rangle$ can always be interpreted in $D$: the above function $\mathsf{fold} : \langle D,D\rangle 1 \to D$ yields obvious functions $\mathsf{fold}_\Gamma : \langle D,D\rangle\Gamma \to D$ for all contexts $\Gamma$ by putting

$$\mathsf{fold}_{\Gamma+1} = \mathsf{fold}_\Gamma \cdot \mathbf{CPO}(D^\Gamma, \mathsf{fold}) \cdot \mathsf{curry},$$

where $\mathsf{curry} : \mathbf{CPO}(D^\Gamma \times D, D) \to \mathbf{CPO}(D^\Gamma, D^D)$ is the currification.

*Remark 5.1.* The presheaf $\langle D, D\rangle$ is an $H_{\lambda,\Sigma}$-monoid. In fact, application and abstraction are naturally obtained from (5.1), see [6], and the pointing $\iota : V \to \langle D, D\rangle$ assigns to an element $x \in \Gamma$ the $x$-projection in $\langle D, D\rangle\Gamma = \mathbf{CPO}(D^{\Gamma}, D)$. The monoid structure

$$m : \langle D, D\rangle \otimes \langle D, D\rangle \to \langle D, D\rangle$$

can be described directly by using the coend formula (3.2) and considering the component of $m_\Gamma$ corresponding, for an element $f \in \mathbf{Set}(\overline{\Gamma}, \mathbf{CPO}(D^{\Gamma}, D))$, to the injection

$$\mathsf{in}_f : \mathbf{CPO}(D^{\overline{\Gamma}}, D) \to \int^{\overline{\Gamma}} \mathbf{Set}(\overline{\Gamma}, \mathbf{CPO}(D^{\Gamma}, D)) \bullet \mathbf{CPO}(D^{\overline{\Gamma}}, D).$$

This component $m_\Gamma \cdot \mathsf{in}_f$ takes $g : D^{\overline{\Gamma}} \to D$ to the function

$$m_\Gamma \cdot \mathsf{in}_f(g) : (x_i) \mapsto g \cdot \langle f(x_i)\rangle \quad \text{for all } (x_i) \text{ in } D^{\Gamma}. \tag{5.2}$$

There is a much more elegant way of obtaining the monoid structure of $\langle D, D\rangle$. From results of Steve Lack [10] we see that the monoidal category $(\mathbf{Set}^{\mathscr{F}}, \otimes, V)$ has the following monoidal action $*$ on **CPO**: given $X$ in $\mathbf{Set}^{\mathscr{F}}$ and $C$ in **CPO**, we put $X * C = \int^{\Gamma} X(\Gamma) \bullet C^{\Gamma}$. Moreover, extending the above notation to pairs $C$, $C'$ of CPO's and defining $\langle C, C'\rangle\Gamma = \mathbf{CPO}(C^{\Gamma}, C')$ we obtain a presheaf with a natural isomorphism

$$\mathbf{Set}^{\mathscr{F}}(X, \langle C, C'\rangle) \cong \mathbf{CPO}(X * C, C').$$

As observed by George Janelidze and Max Kelly [9] this yields an enriched category whose hom-objects are $\langle C, C'\rangle$. In particular, $\langle D, D\rangle$ receives a monoid structure. It is tedious but not difficult to prove that (a) this monoid structure is given by (5.2) above and (b) it forms an $H_{\lambda,\Sigma}$-monoid (cf. Definition 3.5).

**Notation 5.2.** We denote by

$$[\![-]\!] : F_{\lambda,\Sigma} \to \langle D, D\rangle$$

the unique $H_{\lambda,\Sigma}$-monoid homomorphism (see Theorem 3.6). For every finite term $t$ in context $\Gamma$ we thus obtain its interpretation as a continuous function $[\![t]\!]_\Gamma : D^{\Gamma} \to D$

*Remark 5.3.* What is our intuition of an interpreted solution of $e : X \to F_{\lambda,\Sigma} \otimes (X+V)$ in the presheaf $\langle D, D\rangle$? This should be an interpretation of $X$-terms in $\langle D, D\rangle$, that is a natural transformation

$$e^{\dagger} : X \to \langle D, D\rangle$$

with the following property: Given an $X$-term $x$ in context $\Gamma$ then $e_\Gamma$ assigns to it an element $e_\Gamma(x)$ of $(F_{\lambda,\Sigma} \otimes (X + V))(\Gamma)$ that is a finite term $t \in F_{\lambda,\Sigma}(\overline{\Gamma})$ for some $\overline{\Gamma} \subseteq X(\Gamma) + \Gamma$. We request that the solution assigns to $x$ the same value $e^{\dagger}_\Gamma(x) : D^{\Gamma} \to D$ that we obtain from the interpretation $[\![t]\!]$ of the given term by substituting the $\overline{\Gamma}$-variables using $[e^{\dagger}, \iota] : X + V \to \langle D, D\rangle$. This substitution is given by composing $[\![t]\!] \otimes [e^{\dagger}, \iota]$ with the monoid structure of $\langle D, D\rangle$. This leads to the following

**Definition 5.4.** *Given a higher-order recursion scheme $e : X \to F_{\lambda,\Sigma} \otimes (X + V)$ by an* ***interpreted solution*** *is meant a presheaf morphism $e^\dagger : X \to \langle D, D \rangle$ such that the square below commutes:*

$$\begin{array}{ccc} X & \xrightarrow{\quad e^\dagger \quad} & \langle D, D \rangle \\ {\scriptstyle e}\downarrow & & \uparrow{\scriptstyle m} \\ F_{\lambda,\Sigma} \otimes (X + V) & \xrightarrow[\llbracket - \rrbracket \otimes [e^\dagger, \iota]]{} & \langle D, D \rangle \otimes \langle D, D \rangle \end{array} \tag{5.3}$$

**Theorem 5.5.** *Every higher-order recursion scheme has a least interpreted solution in $\langle D, D \rangle$.*

*Sketch of proof.* Observe that $\mathbf{Set}^{\mathscr{F}}(X, \langle D, D \rangle)$ is a CPO with $\bot$ if the ordering is defined pointwise: for $s, s' : X \to \langle D, D \rangle$ we put $s \sqsubseteq s'$ if and only if for every context $\Gamma$ and every $x \in X(\Gamma)$ we have $s_\Gamma(x) \sqsubseteq s'_\Gamma(x)$ in $\mathbf{CPO}(D^\Gamma, D)$. Therefore it is sufficient to prove that the endomap of $\mathbf{Set}^{\mathscr{F}}(X, \langle D, D \rangle)$ given by

$$s \mapsto m \cdot (\llbracket - \rrbracket \otimes [s, \iota]) \cdot e$$

is continuous, then we can use Kleene Theorem. In fact, from the (obvious) continuity of $s \mapsto [s, \iota]$ it follows (less obviously, but this is not too difficult) that $s \mapsto m \cdot (\llbracket - \rrbracket \otimes [s, \iota])$ is continuous, and precomposing with $e$ then also yields a continuous function. □

## 6 Conclusions

We proved that guarded higher-order recursion schemes have a unique uninterpreted solution, i.e., a solution as a rational $\lambda$-$\Sigma$-term. And they also have the least interpreted solution for interpretations based on Scott's models of $\lambda$-calculus as CPO's with continuous operations for all "terminal" symbols of the recursion scheme.

Following M. Fiore *et al* [6] we worked in the category $\mathbf{Set}^{\mathscr{F}}$ of sets in context, that is, covariant presheaves on the category $\mathscr{F}$ of finite sets and functions. A presheaf is a set dependent on a context (a finite set of variables). For every signature $\Sigma$ of "terminal" operation symbols it was proved in [6] that the presheaf $F_{\lambda,\Sigma}$ of all finite $\lambda$-$\Sigma$-terms is the initial $H_{\lambda,\Sigma}$-monoid. This means that $F_{\lambda,\Sigma}$ has (i) the $\lambda$-operations (of abstraction and application) together with operations given by $\Sigma$ rendering an $H_{\lambda,\Sigma}$-algebra and (ii) the operation expressing simultaneous substitution rendering a monoid in the category of presheaves. And $F_{\lambda,\Sigma}$ is the initial presheaf with such structure. In [11] R. Matthes and T. Uustalu proved that the presheaf $T_{\lambda,\Sigma}$ of finite and infinite $\lambda$-$\Sigma$-terms is also an $H_{\lambda,\Sigma}$-monoid. Here we proved that this is the initial completely iterative $H_{\lambda,\Sigma}$-monoid. And its subobject $R_{\lambda,\Sigma}$ of all rational $\lambda$-$\Sigma$-terms is the initial iterative $H_{\lambda,\Sigma}$-monoid. We used that last presheaf in our uninterpreted semantics of recursion schemes.

Our approach was based on untyped $\lambda$-calculus. The ideas in the typed version are quite analogous. If $T$ is the set of all types, then we form the full subcategory $\mathscr{F}$ of $\mathbf{Set}^T$

of finite $T$-sorted sets and consider presheaves in $(\mathbf{Set}^T)^{\mathscr{F}}$—the latter category is equivalent to that of finitary endofunctors of the category $\mathbf{Set}^T$ of $T$-sorted sets. The definition of $H_{\lambda,\Sigma}$ is then completely analogous to the untyped case, and one can form the presheaves $F_{\lambda,\Sigma}$ (free algebra on $V$), $T_{\lambda,\Sigma}$ (free completely iterative algebra) and $R_{\lambda,\Sigma}$ (free iterative algebra). Each of them is a monoid, in fact, an $H_{\lambda,\Sigma}$-monoid in the sense of [6]. Moreover, every guarded higher-order recursion scheme has a unique solution in $R_{\lambda,\Sigma}$. The interpreted semantics can be built up on a CPO-enriched cartesian closed category (as our model of typed $\lambda$-calculus) with additional continuous morphisms for all terminals. The details of the typed version are more involved, and we leave them for future work.

Related results on higher-order substitution can be found e.g. in [11] and [15].

In future work we will, analogously as in [13], investigate the relation of uninterpreted and interpreted solutions.

## References


1. Adámek, J., Milius, S., Velebil, J.: Iterative algebras at work. Math. Structures Comput. Sci. **16** (2006), 1085–1131.
2. Adámek, J., Trnková, V.: Automata and algebras in a category. Kluwer Academic Publishers, Dordrecht, 1990.
3. Aehlig, K.: A finite semantics of simply-typed lambda terms for infinite runs of automata. Proc. Computer Science Logic 2006, Lect. Notes Comput. Sci. **4207**, Springer 2006, 104–118.
4. Damm, W.: Higher-order program schemes and their languages. Lect. Notes Comput. Sci. **48**, Springer 1979, 51–72.
5. Fiore M.: Second order dependently sorted abstract syntax. Proc. Logic in Computer Science 2008, IEEE Press 2008, 57–68.
6. Fiore, M., Plotkin, G., Turi, D.: Abstract syntax and variable binding. Proc. Logic in Computer Science 1999, IEEE Press 1999, 193–202.
7. Garland, S. J., Luckham, D. C.: Program schemes, recursion schemes and formal languages. J. Comput. Syst. Sci. **7** (1973), 119–160.
8. Guessarian, I.: Algebraic semantics. Lect. Notes Comput. Sci. 99, Springer, 1981.
9. Janelidze, G., Kelly, G. M.: A note on actions of a monoidal category, Theory Appl. Categ. **9** (2001), 61–91
10. Lack, S.: On the monadicity of finitary monads, J. Pure Appl. Algebra **140** (1999), 65–73.
11. Matthes, R., Uustalu, T.: Substitution in non-wellfounded syntax with variable binding. Theoret. Comput. Sci. **327** (2004), 155–174.
12. Milius, S.: Completely iterative algebras and completely iterative monads. Inform. and Comput. **196** (2005), 1–41.
13. Milius, S., Moss, L.: The category theoretic solution of recursive program schemes. Theoret. Comput. Sci. **366** (2006), 3–59, corrigendum in **403** (2008), 409–415.
14. Miranda, G.: Structures generated by higher-order grammars and the safety constraint. Ph.D. Thesis, Merton College, Oxford, 2006.
15. Power, J.: A unified category theoretical approach to variable binding. Proc. MERLIN 2003.

## Appendix: Full Proofs of Some Theorems

*Proof of Theorem 2.12*

(1) Let $X$ be a super-finitary presheaf and let $\Gamma_0$ be a context of $n$ variables generating $X$. We prove that $X$ is a finite colimit of representables. Since representables are (due to Yoneda lemma) clearly finitely presentable, this proves finite presentability of $X$.

Form the finite diagram of all objects

$$Z_a = \mathscr{F}(\Gamma, -)$$

where $\Gamma \subseteq \Gamma_0 + \Gamma_0$ is a context of at most $2n$ variables and $a \in X(\Gamma)$, whose connecting morphisms are the Yoneda transformations

$$Yf\colon Z_a \to Z_{a'} \qquad \text{for } a \in X(\Gamma) \text{ and } a' \in X(\Gamma')$$

where the function $f\colon \Gamma' \to \Gamma$ fulfils $X(f)(a') = a$. The Yoneda transformation

$$z_a\colon Z_a \to X, \quad \text{with the components defined by} \quad f \mapsto Xf(a),$$

clearly form a compatible cocone of this finite diagram. We prove that this is a colimit cocone. In other words, for every context $\bar{\Gamma}$ we must prove that the cocone of all $\bar{\Gamma}$-components $z_a^{\bar{\Gamma}}$ (sending elements $f\colon \Gamma \to \bar{\Gamma}$ of $Z_a = \mathscr{F}(\Gamma, -)$ to $Xf(a)$) is a colimit in **Set**. For that we only need to verify that in every context $\bar{\Gamma}$

1. $z_a^{\bar{\Gamma}}$ are collectively epimorphic

and

(ii) whenever two elements $f\colon \Gamma \to \bar{\Gamma}$ of $Z_a$ and $f'\colon \Gamma' \to \bar{\Gamma}$ of $Z_{a'}$ fulfil $z_a^{\bar{\Gamma}}(f) = z_{a'}^{\bar{\Gamma}}(f')$, then there exists a zig-zag connecting $f$ and $f'$ in the $\bar{\Gamma}$-component of our diagram.

The proof of (i) is trivial: given an element $a \in X(\bar{\Gamma})$, either $\bar{\Gamma} = \emptyset$ or there exists $f\colon \Gamma_0 \to \bar{\Gamma}$ and an element $b \in X(\Gamma_0)$ with $a = Xf(b)$, in other words,

$$a = z_b^{\bar{\Gamma}}(f).$$

In case $\bar{\Gamma} = \emptyset$ we have $a = z_a^{\bar{\Gamma}}(\mathrm{id}_\emptyset)$.

To prove (ii) observe that the given equation states

$$Xf(a) = Xf'(a').$$

In case $\bar{\Gamma}$ has at most $2n$ variables we can assume $\bar{\Gamma} \subseteq \Gamma_0 + \Gamma_0$ and the desired zig-zag is

$$Z_a \xleftarrow{Yf} Z_b \xrightarrow{Yf'} Z_{a'},$$

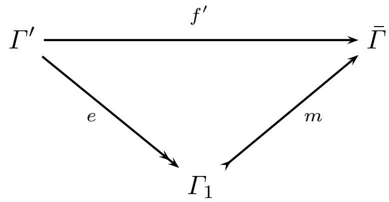
where $b = Xf(a)$. Thus, we can assume that $\bar{\Gamma}$ has more than $2n$ elements.

Case 1: $\Gamma = \emptyset = \Gamma'$. Here $f = f'$ and we have $Xf(a) = Xf(a')$. Choose a monomorphism $m\colon \Gamma_0 \to \bar{\Gamma}$ and observe that $f = m \cdot g$ for the unique $g\colon \emptyset \to \Gamma_0$.

Thus $Xm(Xg(a)) = Xm(Xg(a'))$ and since $m$ is a split monomorphism, we conclude $Xg(a) = Xg(a') = c$. The desired zig-zag is

$$Z_a \xleftarrow{Yg} Z_c \xrightarrow{Yg'} Z_{a'}.$$

Case 2: $\Gamma = \emptyset \neq \Gamma'$. Factorize $f'$ into an epimorphism $e$ followed by a split monomorphism $m$:



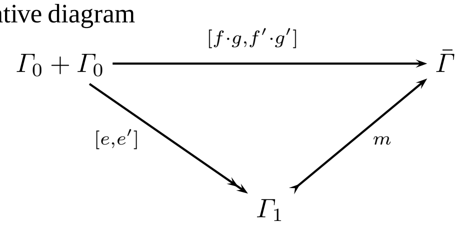

Then, since for the unique $h\colon \emptyset \to \Gamma_1$ we have $f = m{\cdot}h$, we obtain

$$Xm\big(Xe(a')\big) = Xm\big(Xh(a)\big).$$

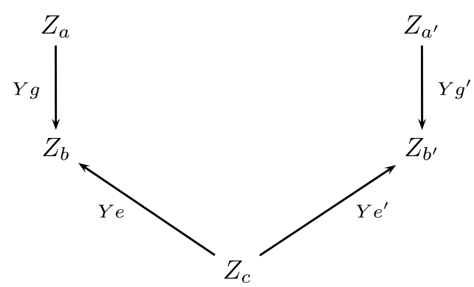
Thus, $Xe(a') = Xh(a) = c$ which yields the zig-zag

$$Z_a \xleftarrow{Yh} Z_c \xrightarrow{Ye} Z_{a'}.$$

Case 3: $\Gamma \neq \emptyset \neq \Gamma'$. Find $g\colon \Gamma_0 \to \Gamma$ with $a = Xgf(b)$ and $g'\colon \Gamma_0 \to \Gamma'$ with $a' = Xg'(b')$ for some $b, b' \in X(\Gamma_0)$. Then $X(f{\cdot}g)(b) = X(f'{\cdot}g')(b')$. Now factorize $[f{\cdot}g, f'{\cdot}g']\colon \Gamma_0 + \Gamma_0 \to \bar{\Gamma}$ as an epimorphism followed by a split monomorphism; so we obtain a commutative diagram




Since $m$ is a split monomorphism, conclude that $Xe(b) = Xe'(b') = c$. The desired zig-zag is




(2) Let $X$ be a finitely presentable object of $\mathbf{Set}^{\mathscr{F}}$. The empty maps are denoted by $t_\Gamma\colon \emptyset \to \Gamma$. For every nonempty context $\Gamma_0$ let $X_{\Gamma_0}$ be the subfunctor of $X$ generated by the elements of $X(\Gamma_0) \cup X(\emptyset)$: it assigns to every $\Gamma$ the subset of $X(\Gamma)$ given by

$$X_{\Gamma_0}(\Gamma) = Xt_\Gamma\big[X(\emptyset)\big] \cup \bigcup_{f\colon \Gamma_0 \to \Gamma} Xf\big[X(\Gamma_0)\big].$$

We obviously have a union

$$X = \bigcup_{\Gamma_0 \in \mathscr{F} - \{\emptyset\}} X_{\Gamma_0}$$

which is directed: given nonempty contexts $\Gamma_0, \Gamma_1$ then $X_{\Gamma_0} \cup X_{\Gamma_1} \subseteq X_{\Gamma_0 \cup \Gamma_1}$. Since $X$ is finitely presentable, the morphism

$$\mathrm{id}_X \colon X \to \operatorname*{colim}_{\Gamma_0 \in \mathscr{F} - \{\emptyset\}} X_{\Gamma_0}$$

factorizes through one of the colimit injections $X_{\Gamma_0} \hookrightarrow X$. In other words

$$X = X_{\Gamma_0} \qquad \text{for some } \Gamma_0 \neq \emptyset.$$

It remains to prove that the sets $X(\Gamma_0)$ and $X(\emptyset)$ are finite, then all $X(\Gamma)$ are finite.

For every finite set $M \subseteq X(\emptyset)$ we have the subfunctor $X^M$ of $X$ equal to $X$ on nonempty objects and maps, and assigning $M$ to $\emptyset$. We obviously get $X$ as a directed union of these subfunctors $X^M$, thus, as above, there exists $M$ with $X = X^M$. Then $X(\emptyset) = M$ is finite.

For every finite set $M \subseteq X(\Gamma_0)$ we have the subfunctor ${}^M X$ of $X = X_{\Gamma_0}$ generated by the elements of $M \cup X(\emptyset)$:

$${}^M X(\Gamma) = X t_\Gamma \big[X(\emptyset)\big] \cup \bigcup_{f\colon \Gamma_0 \to \Gamma} X f[M].$$

Again $X$ is a directed union of these subfunctors ${}^M X$, thus, there exists $M$ with $X = {}^M X$, proving that $X(\Gamma_0) = M$ is finite. □

*Proof of Theorem 2.14*

(I) $R_\lambda$ is an iterative algebra for $H_\lambda$. In fact, given an equation morphism

$$e \colon X \to H_\lambda X + R_\lambda$$

where (2.4) holds for $\Gamma_0$ we know that its extension

$$\bar{e} \colon X \xrightarrow{e} H_\lambda X + R_\lambda \hookrightarrow H_\lambda X + T_\lambda$$

has a unique solution $e^\dagger \colon X \to T_\lambda$, and we are going to prove that the trees $e^\dagger_{\Gamma_0}(p)$ and $e^\dagger_\emptyset(p)$ are all rational. It then follows that all the trees $e^\dagger_\Gamma(p)$ are rational for all contexts $\Gamma$, and this gives us the desired solution $X \to R_\lambda$. In fact, for each $x \in X(\Gamma)$ with $\Gamma \neq \emptyset$ we have $x = Xf(p)$ for some $f \colon \Gamma_0 \to \Gamma$ and $p \in X(\Gamma_0)$. Then $e^\dagger_\Gamma(x) = e^\dagger_\Gamma(Xf(p)) = T_\lambda f(e^\dagger_{\Gamma_0}(p))$ by the naturality of $e^\dagger$, and since $e^\dagger_{\Gamma_0}(p)$ is rational, so is $T_\lambda f(e^\dagger_{\Gamma_0}(p))$. (The action of $T_\lambda f$ is just relabelling leaves according to $f$.)

Now every element of $X(\Gamma_0) = \{p_1, \dots, p_n\}$ yields an element

$$e_{\Gamma_0}(p_i) \in X(\Gamma_0) \times X(\Gamma_0) + X\big(\Gamma_0 + \{x\}\big) + R_\lambda(\Gamma_0)$$

which is either (i) a pair $(p_j, p_k)$ or (ii) $q \in X(\Gamma + \{x\})$ or (iii) a rational tree in $R_\lambda(\Gamma_0)$. Put $t_i = e^\dagger_{\Gamma_0}(p_i)$, then in the last case (2.3) implies $e_{\Gamma_0}(p) = t_p$. From (2.3) we also obtain in cases (i) and (ii)

$$t_i = t_j \mathbin{@} t_k \qquad \text{and} \qquad t_i = \lambda x. e^\dagger_{\Gamma_0 + \{x\}}(q), \qquad \text{respectively.}$$

From (2.4) we see that in case (ii) there exists $f\colon \Gamma_0 \to \Gamma_0 + \{x\}$ with $q = Xf(p_j)$ for some $j$, then $e^\dagger_{\Gamma_0+\{x\}}(q) = T_\lambda f(e^\dagger_\Gamma(p_j)) = T_\lambda f(t_j)$. Thus we get equations telling us that for every $i$ either $t_i = t_j \mathbin{@} t_k$ or $t_i = \lambda x{\cdot}T_\lambda f(t_j)$ or $t_i$ is a rational tree. Using these equations it is now easy, for every $i = 1, \dots, n$, to prove by induction on the depth $k$ of nodes in $t_i$ that each subtrees of $t_i$ of depth $k$ is either of the form $s = T_\lambda f(e^\dagger_{\Gamma_0}(r))$ for some $r \in X(\Gamma_0)$ and some $f\colon \Gamma_0 \to \Gamma_0 + \{x\}$, or $s$ is a subtree of some rational tree $e^\dagger_{\Gamma_0}(r) = e_{\Gamma_0}(r)$ in case (iii). Since $X(\Gamma_0)$ is a finite set, it follows that every tree $t_i$ has only finitely many subtrees, whence $t_i \in R_\lambda(\Gamma_0)$.

The case $X(\emptyset) = \{p_1, \dots, p_n\}$ is analogous: for $t_i = e^\dagger_\Gamma(p_i)$ we get (i) $t_i = t_j \mathbin{@} t_k$ or (ii) $t_i = \lambda x{\cdot}e^\dagger_{\{x\}}(q)$ or (iii) $t_i = e_\emptyset(p_i) \in R_\lambda(\emptyset)$. We already know that the trees in case (ii) are rational. Thus, each subtree of $e^\dagger_\emptyset(p_i)$ is either $e^\dagger_\emptyset(r)$ or it is a subtree of some rational tree in cases (ii) or (iii).

The solution of $e$ in $R_\lambda$ is unique because every solution in $R_\lambda$ yields a solution of the extended morphism $\bar{e}$ in $T_\lambda$.

(II) Let $\mathscr{D}$ be the category of all equation morphisms

$$e\colon X \to H_\lambda X + V, \qquad X \text{ finitely presentable,}$$

whose morphisms are the coalgebra homomorphisms for $H_\lambda(-) + V$. The diagram $D\colon \mathscr{D} \to \mathbf{Set}^{\mathscr{F}}$, $D(e) = X$, is filtered and its colimit is the free iterative $H_\lambda$-algebra on $V$, see [1]. We will prove that $R_\lambda$ is a colimit of $D$. Recall that $R_\lambda$ is a pointed candidate (see Remark 2.6).

For every $e$ as above the equation morphism

$$\tilde{e} \equiv X \xrightarrow{\ e\ } H_\lambda X + V \xrightarrow{\ \mathrm{id} + i^R\ } H_\lambda X + R_\lambda$$

has a unique solution $\tilde{e}^\dagger\colon X \to R_\lambda$. It is easy to verify that these morphisms form a cocone for the diagram $D$. Since $D$ is a filtered diagram in $\mathbf{Set}^{\mathscr{F}}$ and since colimits in $\mathbf{Set}^{\mathscr{F}}$ are constructed objectwise in **Set**, in order to prove that

$$R_\lambda = \operatorname{colim} D \qquad \text{with the colimit cocone } (\tilde{e}^\dagger)$$

all we need to prove is that for every context $\Gamma$

(a) the cocone $\tilde{e}^\dagger_\Gamma$ is collectively surjective: $R_\lambda(\Gamma) = \bigcup \tilde{e}^\dagger_\Gamma[X]$

and

(b) whenever $\tilde{e}^\dagger_\Gamma$ merges $x, x' \in X(\Gamma)$ there exists a connecting morphism in $D$ merging $x$ and $x'$ too.

To prove (a), let $t \in R_\lambda(\Gamma)$ be a rational tree and let $\Gamma_0$ be the context of variables $x_s$ labelled by the finite set of all subtrees $s$ of $t$ (up to isomorphism). Let $X$ be the free presheaf on the set $\Gamma_0$ of generators of context $\bar{\Gamma} = \Gamma \cup \Gamma_0$, that is, a coproduct of $\Gamma_0$ copies of $\bar{\Gamma}_{\mathrm{fr}}$, see Example 2.2(ii). Define

$$e\colon X \to H_\lambda X + V$$

by assigning to every variable $x_s$, for a subtree $s$ of $t$, the following value: if $s = s' @ s''$ in $t$, then

$$e_\Gamma(x_s) = x_{s'} @ x_{s''} \qquad \text{in } X(\bar\Gamma) \times X(\bar\Gamma),$$

if $s = \lambda y.s'$ in $t$, then

$$e_\Gamma(x_s) = \lambda y.x_{s'} \qquad \text{in } X\big(\bar\Gamma + \{y\}\big),$$

and if $s$ is a leaf labelled by $x \in \Gamma$, then

$$e_\Gamma(x_s) = x \qquad \text{in } \Gamma = V(\Gamma).$$

This object $e$ of $\mathscr{D}$ yields two equation morphisms: $\tilde e\colon X \to H_\lambda X + R_\lambda$ above and analogously $\hat e = (\mathrm{id} + i^T)\cdot e\colon X \to H_\lambda X + T_\lambda$. The solution of the latter is the unique morphism

$$\hat e^\dagger\colon X \to T_\lambda \qquad \text{with} \qquad \hat e^\dagger_{\bar\Gamma}(x_s) = s \text{ for all } s \in \Gamma_0.$$

In fact, (2.3) is easily seen to commute for $\hat e$ and $\hat e^\dagger$. In (I) above we saw that the solution $\tilde e^\dagger\colon X \to R_\lambda$ is a codomain restriction of $\hat e^\dagger$. In particular:

$$t = \tilde e^\dagger_{\bar\Gamma}(x_t).$$

This proves (a).

To prove (b) let $\tau\colon H_\lambda T_\lambda \to T_\lambda$ denote the algebra structure of $T_\lambda$. By Theorem 2.9 and Corollary 6.3 in [12] we have that

$$[\tau, i^T]\colon H_\lambda T_\lambda + V \to T_\lambda \qquad \text{is an isomorphism.}$$

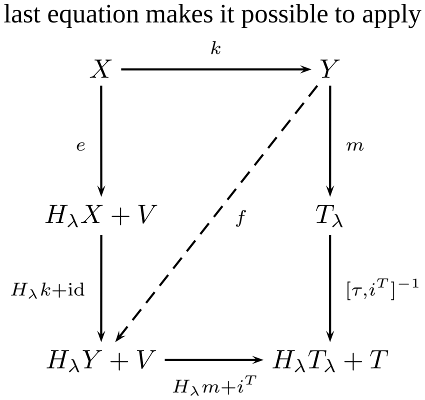
From (2.3) we get

$$\hat e^\dagger = [\tau, \mathrm{id}_{T_\lambda}]\cdot[H_\lambda \hat e^\dagger + \mathrm{id}_{T_\lambda}]\cdot(\mathrm{id}_{H_\lambda X} + i^T)\cdot e$$

which yields

$$[\tau, \mathrm{id}_{T_\lambda}]^{-1}\cdot\hat e^\dagger = (H_\lambda \hat e^\dagger + i)\cdot e.$$

Let us factorize $\hat e^\dagger$ as a strong epimorphism $k\colon X \to Y$ followed by a monomorphism $m\colon Y \to T_\lambda$. Then the last equation makes it possible to apply the diagonal fill in:



In fact, $H_\lambda = (-)^2 + \delta$ preserves connected limits (because each summand does), thus, monomorphisms; consequently, $H_\lambda m + i^T$ is a monomorphism. Since $Y$ is a quotient of $X$, it follows from Theorem 2.12 that $Y$ is finitely presentable. Thus,

$$f\colon Y \to H_\lambda Y + V$$

is an object of $\mathscr{D}$, and clearly $k$ is a connecting morphism from $e$ to $f$.

From (I) we know that $\tilde{e}^\dagger$ is the domain restriction of $\hat{e}^\dagger$, thus we see that $\tilde{e}^\dagger_\Gamma(x) = \tilde{e}^\dagger_\Gamma(x')$ implies $\hat{e}^\dagger_\Gamma(x) = \hat{e}^\dagger_\Gamma(x')$, and since $m_\Gamma$ is a monomorphism with $\hat{e}^\dagger_\Gamma = m_\Gamma \cdot k_\Gamma$, we conclude

$$k_\Gamma(x) = k_\Gamma(x')$$

as requested. □

*Proof of Theorem 3.9* We present the proof for $H_\lambda$ (that is $\Sigma = \emptyset$), the proof for $H_{\lambda,\Sigma}$ is completely analogous.

By Theorem 2.9 $T_\lambda$ is a completely iterative algebra, let $\tau\colon H_\lambda T_\lambda \to T_\lambda$ denote its algebra structure, and by Theorem 3.8 it is an $H_\lambda$-monoid.

Let $a\colon H_\lambda A \to A$ be a completely iterative algebra which is also an $H_\lambda$-monoid $(A, m, i)$, that is, the square (3.3) commutes. It is our task to prove that there exists a unique homomorphism $h\colon T_\lambda \to A$ of algebras for $H_\lambda$ satisfying

$$h \cdot i^T = i \tag{A.1}$$

and

$$h \cdot m^T = m \cdot (h \otimes h). \tag{A.2}$$

In fact, by Theorem 2.9 the condition (A.1) determines $h$ uniquely, thus, it is sufficient to prove (A.2).

We know from the proof of Theorem 2.9 that $T_\lambda$ is a terminal coalgebra for $H_\lambda(-)+V$, which by Lambek's Lemma proves that its coalgebra structure is an isomorphism—in other words:

$$T_\lambda = H_\lambda T_\lambda + V \qquad \text{with coproduct injections } \tau\colon H_\lambda T_\lambda \to T_\lambda \text{ and } i^T\colon V \to T_\lambda.$$

Since the composition $- \otimes T_\lambda$ with $T_\lambda$ preserves coproducts, we get

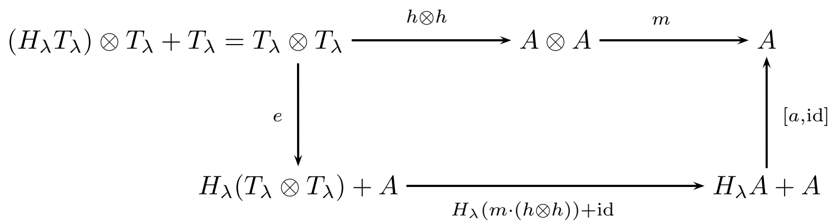
$$T_\lambda \otimes T_\lambda = (H_\lambda T_\lambda) \otimes T_\lambda + T_\lambda \qquad \text{with injections } \tau \otimes \mathrm{id} \text{ and } i^T \otimes \mathrm{id}. \tag{A.3}$$

We define a flat equation morphism $e\colon T_\lambda \otimes T_\lambda \to H_\lambda(T_\lambda \otimes T_\lambda) + A$ by the components (of the above coproduct) as follows

$$\begin{array}{ccc} H_\lambda(T_\lambda)\otimes T_\lambda & \xrightarrow{s^H} & H_\lambda(T_\lambda\otimes T_\lambda) \\ \downarrow{\scriptstyle \tau\otimes\mathrm{id}} & & \downarrow{\scriptstyle \mathsf{inl}} \\ T_\lambda\otimes T_\lambda & \xrightarrow{e} & H_\lambda(T_\lambda\otimes T_\lambda)+A \\ \uparrow{\scriptstyle i^T\otimes \mathrm{id}} & & \uparrow{\scriptstyle \mathsf{inr}} \\ T_\lambda = V\otimes T_\lambda & \xrightarrow[h]{} & A \end{array} \tag{A.4}$$

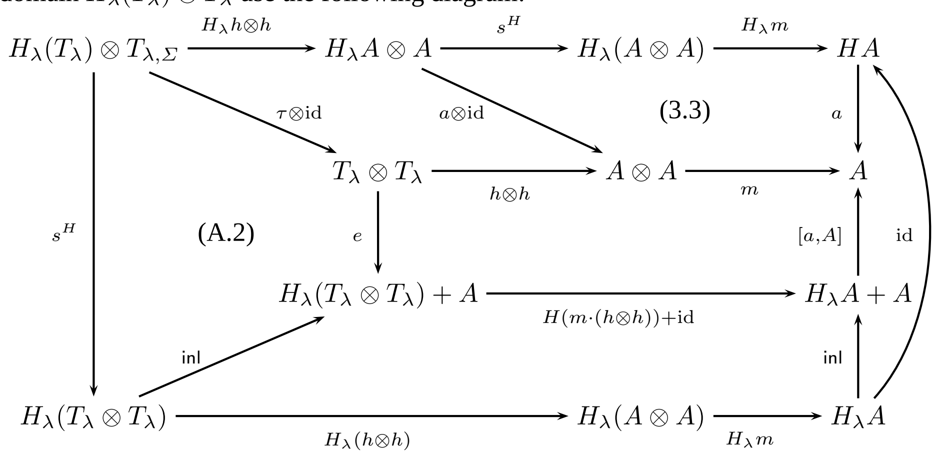
We prove (A.2) by verifying that both sides serve as the (unique!) solution of $e$.

(i) $m \cdot (h \otimes h)$ is a solution of $e$, that is, the square



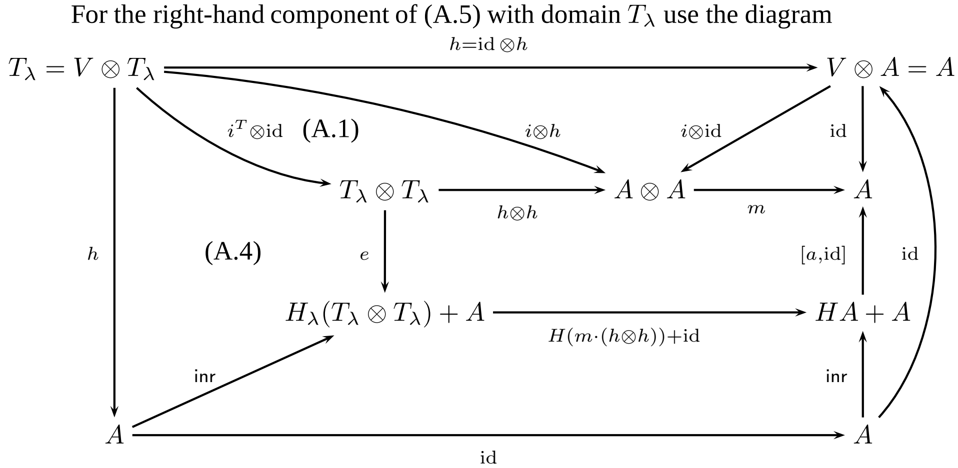

(A.5)

commutes. We verify the two components separately. For the left-hand component with domain $H_\lambda(T_\lambda) \otimes T_\lambda$ use the following diagram:



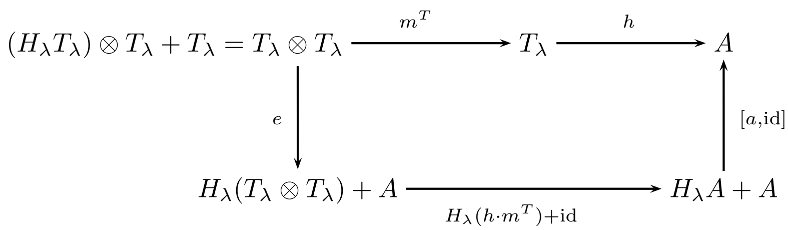

The upper left-hand square commutes since $h$ is a homomorphism from $(T_\lambda, \tau)$ to $(A, a)$, the other inward parts (except the middle square) commute as indicated or are trivial. Also the outside of the diagram commutes by naturality of $s^H$. This proves the commutativity of the desired middle square extended by $\tau \otimes \mathrm{id}$.

For the right-hand component of (A.5) with domain $T_\lambda$ use the diagram



The upper right-hand triangle commutes since $(A, m, i)$ is a monoid, all other inner parts commute as indicated or trivially—except the middle square, which then commutes when extended by $i^T \otimes \mathrm{id}$ since the outside of the diagram does.

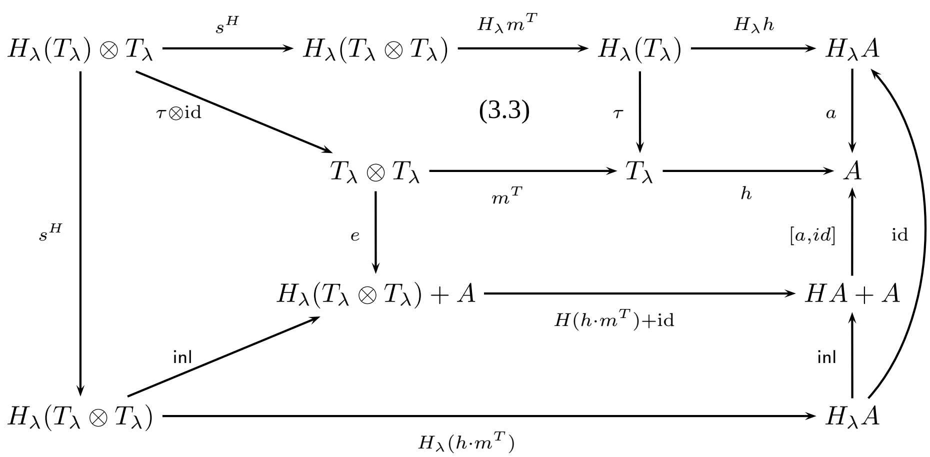
(ii) $h{\cdot}m^T$ is a solution of $e$, that is, the square




(A.6)

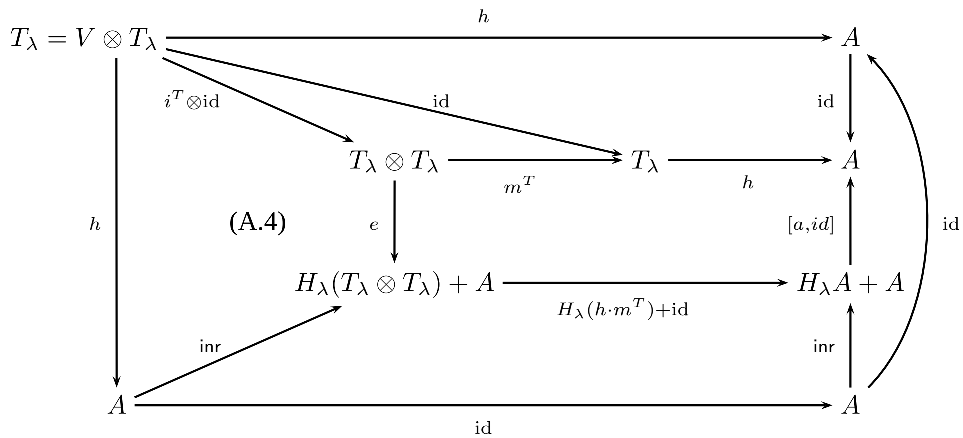
commutes. We again verify the two components of (A.6) separately.

For the left-hand component with domain $(H_\lambda T_\lambda) \otimes T_\lambda$ consider the commutative diagram




In fact, the outside of the diagram commutes trivially and so do the left-hand, right-hand and lower parts. The upper right-hand square commutes because $h$ is a homomorphism of $H_\lambda$-algebras. Finally, the left-hand upper square commutes since $h$ is homomorphism of algebras for $H_\lambda$. Thus, the inner square commutes when extended by $\tau \otimes \mathrm{id}$.

For the right-hand component of (A.6) with domain $T_\lambda$ we have the diagram

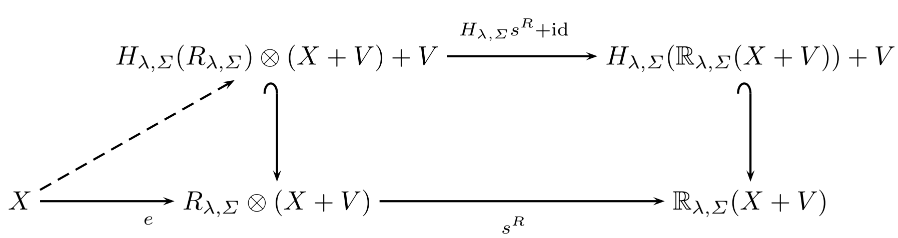



The upper triangle commutes since $(T_\lambda, m^T, i^T)$ is a monoid, everything else is trivial. This completes the proof of (A.2). □

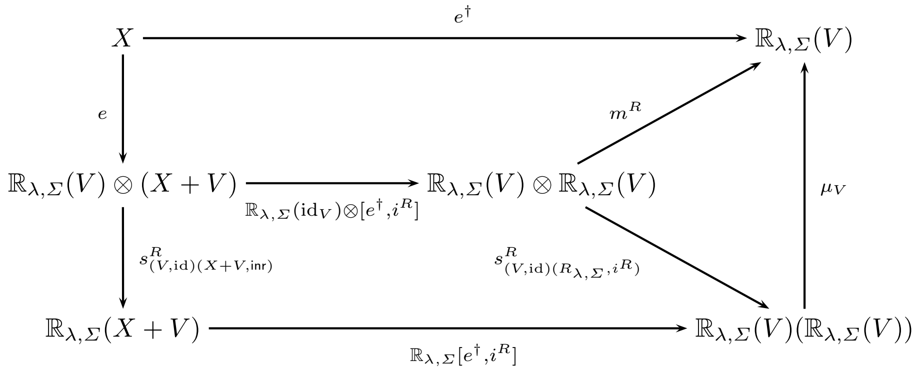
*Proof of Theorem 4.7* Let us apply the monad $\mathbb{R}_{\lambda,\Sigma}$ and its point-strength as in the proof of Theorem 3.11. We construct for every higher-order recursion scheme $e\colon X \to \mathbb{R}_{\lambda,\Sigma}(V) \otimes (X + V)$ a rational equation morphism $\bar{e}\colon X \to \mathbb{R}_{\lambda,\Sigma}(X + V)$ in the sense of [1] as follows:

$$\bar{e} \equiv X \xrightarrow{e} \mathbb{R}_{\lambda,\Sigma}(V) \otimes (X + V) \xrightarrow{s_{(V,\mathrm{id})(X+V,\mathsf{inr})}} \mathbb{R}_{\lambda,\Sigma}(X + V).$$

¿From the guardedness of $e$ we conclude that $\bar{e}$ is guarded in the sense of [1], that is, $\bar{e}$ factorizes through the summand $H_{\lambda,\Sigma}(\mathbb{R}_{\lambda,\Sigma})) + V$ of $\mathbb{R}_{\lambda,\Sigma}(X + V)$, see (3.7). In fact, this follows from (3.8) via the following diagram




Consequently, by Theorem 4.5 in [1] there exists a unique solution $\bar{e}^\dagger$ of $\bar{e}$ with respect to the monad $\mathbb{R}_{\lambda,\Sigma}$—this means a unique morphism $e^\dagger\colon X \to \mathbb{R}_{\lambda,\Sigma}(V)$ such that the outside of the diagram below




commutes. Since the right-hand triangle is (3.6) and the lower square commutes by the naturality of $s^R$, we see that $\bar{e}^\dagger$ is a solution of $e$ in the sense of Definition 4.3 iff $e^\dagger$ is a solution of $\bar{e}$ in the sense of [1]. This proves that $e$ has a unique solution. $\square$