Birkhoff's HSP-Theorem for Cumulative Logic Programs

Beatrice Amrhein

ETH Zürich Universität Tübingen
amrhein@math.ethz.ch

Abstract. Birkhoff's HSP theorem is that the models of a set of algebraic equations form a variety, i.e. a category of algebras which admits homomorphic images, subalgebras and products. We show here first, that every equational set of retract structures in combinatory logic is a variety, and second, that every set of combinators, closed under certain operations, is equational. It follows that the models of cumulative logic programs form an equational variety.

1 Motivation

The solutions of a set of equations form a structure, which is closed under the operations of building substructures, homomorphic images and products of solutions. This was first shown by G. Birkhoff [5] in 1935 for algebraic equations. We examined here the characterisation of solutions of *cumulative logic programs.* As in the case of symmetric search trees, where this knowledge can be used to cut symmetric branches, knowledge about the structure of the solutions of a cumulative logic program can speed up their computation.

2 Cumulative Logic Programs and Combinatory Models

To perform transformations on arbitrary knowledge bases, (even ones containing rule terms) and to allow more than one argument to such transformations, we have to build a cumulative hierarchy of rules. Therefore, the language of *cumulative logic programs* was introduced [6], which is based on the language of logic programs with the following extensions: Clauses are written as $\{b_1, \ldots, b_k\} \rightarrow h$, where the head h and the body $\{b_1, \ldots, b_k\}$ are either atomic formulas or again clauses.

This cumulative formation of rules to an arbitrary level has its mathematical counterpart in graph models $\mathcal{D}_A$.

Definition 1. Let A be a set of atomic formulas. $\mathcal{D}_A := \langle 2^{G(A)}\, ; \circ \rangle$ is a *graph model* over A, where $G(A)$ is defined recursively:

$$\begin{aligned} G_0(A) &:= A \\ G_{n+1}(A) &:= G_n(A) \cup \{\alpha \to a : a \in G_n(A),\ \alpha \subseteq G_n(A),\ \alpha \text{ finite}\} \\ G(A) &:= \textstyle\bigcup_n G_n(A) \end{aligned}$$

Note that within this cumulative hierarchy, pure logic programs are simply subsets of $G_1(A)$. The inclusion of implications in goal formulae, as proposed in [9] by Miller et al, leads then to subsets of $G(A)$ of the form

$$\{\alpha_1 \rightarrow (\alpha_2 \rightarrow (\cdots \rightarrow \alpha_n)\cdots), \quad \ldots\}$$

where α_i $(1 \leq i \leq n)$ are finite subsets of A.

Graph models are rich enough in structure to form models for knowledge bases of arbitrary complexity, where even transformations of functions (such as derivation or fixed point operation) are included as elements. [1]

The application operation $\circ$ is defined as:

$$V \circ W := \{a : \alpha \rightarrow a \in V, \alpha \subseteq W\}$$

The result of applying V to W depends on the elementary instructions $\rho \rightarrow r$ of V, which give output r any time the input W contains ρ. The application operation can be used to explain the execution of pure logic programs as a solution process for a corresponding fix-point equation as in the case of pure logic programs, the application operation $\circ$ corresponds to Kowalski's semantics operator T_P as defined in [8], i.e. for a combinator V and a set W of ground terms we obtain

$$V \circ W = T_{P_V}(W)$$

where P_V is the logic program derived from combinator V.

It's a well-known fact that these structures $\langle 2^{G(A)} ; \circ \rangle$ build *combinatory algebras*.

Definition 2. [3] An **applicative structure** $\mathcal{M} := \langle M; \cdot \rangle$ is an algebraic structure with carrier set M and a binary operation $\cdot$, which we call application. By **combinatory algebra** we denote an applicative structure $\mathcal{M} := \langle M; \cdot \rangle$ which is **combinatory complete**, which means that for every term $t(v_1, \ldots, v_k)$ (built up from elements of M and variables by means of the application $\cdot$ and parentheses), an element $T \in M$ exists such that $\forall m_1, \ldots, m_k \in M$:

$$T \cdot m_1 \cdots m_k = t(m_1, \ldots, m_k)$$

Lemma 3 Schönfinkel, Curry. An applicative structure $\mathcal{M}$ is combinatory complete iff there exists K and S such that

$$\forall v_1, v_2 : K v_1 v_2 = v_1$$

$$\forall v_1, v_2, v_3 : S v_1 v_2 v_3 = v_1 v_3 (v_2 v_3)$$

In order to present a readable and comprehensive treatment of the objects in a graph model D_A the following notation will be used:

Variables:	$a, b, c, \dots$	for elements of A
	$r, s, t, \dots$	for elements of $G(A)$
	$\rho, \sigma, \tau, \dots$	for finite subsets of $G(A)$
	V, W	for elements of D_A
Terms:	UVW	for $(U \circ V) \circ W$, i.e. we employ left-associativity and omit $\circ$
	$\sigma \rightarrow \rho \rightarrow r$	for $\sigma \rightarrow (\rho \rightarrow r)$; right-associativity for arrow terms corresponds to the left-associativity for the application

We define

$$K := \{(\sigma \rightarrow (\rho \rightarrow r)) \, : \, r \in \sigma\}$$

$$S := \{(\tau \rightarrow (\sigma \rightarrow (\rho \rightarrow r)) \, : \, \exists \rho_0, \dots, \rho_n \subseteq \rho, \, \exists r_1, \dots, r_n : \\ \rho_i \rightarrow r_i \in \sigma \, (1 \leq i \leq n), \; (\rho_0 \rightarrow (\{r_1, \dots, r_n\} \rightarrow r)) \in \tau\}$$

With this we obtain

Lemma 4. Every graph model is a combinatory algebra.

Definition 5. $\mathcal{M} := \langle M \, ; \cdot, K, S, L \rangle$ is a **combinatory model** if $\mathcal{M} := \langle M \, ; \cdot, K, S \rangle$ is a combinatory algebra with combinators L_k $(k \in I\!N)$ satisfying the axioms:

$$\forall u, v_1, \dots, v_k \in M : \; L_k u v_1 \cdots v_k \; = \; u v_1 \cdots v_k \tag{1}$$

$$\forall u_1, u_2 \in M : \; (\, L_k u_1 = L_k u_2 \\ \Leftrightarrow \forall v_1, \dots, v_k \in M : u_1 v_1 \cdots v_k = u_2 v_1 \cdots v_k \,) \tag{2}$$

As can easily be seen, the graph model combinators

$$L_{k+1} := \{ \, \rho_0 \rightarrow (\cdots \rightarrow (\rho_k \rightarrow r) \cdots) : \\ \exists \rho'_i \subseteq \rho_i \; (1 \leq i \leq k), \; \rho'_1 \rightarrow (\cdots \rightarrow (\rho'_k \rightarrow r) \cdots) \in \rho_0 \, \}$$

have the above mentioned properties (1) and (2), which means that $\mathcal{D}_A$ is indeed a combinatory model.

When querying a Prolog program, the resolution mechanism implicitly searches for the least fixed point of a program equation $v = F \cdot v$. By combinatory completeness, we can state a combinator Y with the property:

$$YV = V(YV)$$

which guarantees solutions for every fix-point program-equation in any graph model.

Equations of cumulative logic programs are solved explicitly using the algorithms formulated in [10] in the case of fix-point equations, or in more general cases using the semi-algorithm as sketched in [7]

A *problem* is in general a set of equations $< a, b >$ where a and b are combinators (programs). Solutions of a problem are, therefore, again elements of the graph model, which means, again cumulative logic programs. In general we get an (program-)approximation to a solution in the sense of [6], in more special cases the solutions can be found in the ground set A [10].

3 Definitions

In the original Theorem of Birkhoff, the signature of the algebras has to be fixed to define the language of the equations. In an analogous way, we first have to determine the language of the considered equations.

Definition 6. Let $\mathcal{M}$ be a combinatory model. Let $\Theta := \{\theta_j : j \in J\}$ be a set of distinguished combinators in $\mathcal{M}$. s is called *Θ-term* if $s \in \Theta$, $s = K$ or $s = S$, and if s_1, s_2 are Θ-terms, then $(s_1 s_2)$ is also a Θ-term.

A pair $< s, t >$ is called ***Θ-equation***, if s and t are Θ-terms, or if there is a $k \in I\!N$ with $s = L_k s'$ and $t = L_k t'$, where s' and t' are Θ-terms.

Because of combinatory completeness, every term t can be written in the form $t'v_1 \cdots v_k$, where $v_1, , \ldots, v_k$ are the variables contained in t. We will, therefore, restrict our attention to Θ-equations.

In the first section, retractions will play the role of the solution sets (as do algebras in the original theorem of Birkhoff).

Definition 7. Let $\mathcal{M}$ be a combinatory model. A combinator x in $\mathcal{M}$ is called *retraction*, if for all $v \in \mathcal{M}$:

$$x(xv) \quad = \quad xv$$

With *retract* of x we denote the set $\text{ret}(x) := \{v \in M \; : \; xv = v\}$.

We can now build restricted quantifier eliminators over retractions. For an arbitrary retraction x we denote by

$$L^x := BL(CBx), \qquad \text{and} \qquad L^x_k := B(BL(CBx))(BL^x_k)$$

where B and C are combinators defined as:

$$\forall v_1, v_2, v_3 : \; Bv_1v_2v_3 = v_1(v_2v_3)$$

and

$$\forall v_1, v_2, v_3 : \; Cv_1v_2v_3 = v_1v_3v_2$$

Then, we can show:

Lemma 8. For any retraction x, L^x is a normalising combinator for the retract of x, which means that for all $k \in I\!N$:

$$\forall u, v_1, \ldots, v_k : \; L^x_k uv_1 \cdots v_k = u(xv_1) \cdots (xv_k)$$

$$\forall u_1, u_2 \left(\forall v_1 \cdots v_k : \; u_1(xv_1) \cdots (xv_k) = u_2(xv_1) \cdots (xv_k) \; \Leftrightarrow \; L^x_k u_1 = L^x_k u_2 \right)$$

By these restricted quantifier eliminators, we can also eliminate variables in equations like $s(xv_1) = t(xv_1)(xv_2)$, because:

$$\begin{aligned} s(xv_1) = t(xv_1)(xv_2) \;&\Leftrightarrow\; K(s(xv_1))(xv_2) = t(xv_1)(xv_2) \\ &\Leftrightarrow\; K(s(xv_1))(xv_2) = t(xv_1)(xv_2) \\ &\Leftrightarrow\; BKs(xv_1)(xv_2) = t(xv_1)(xv_2) \\ &\Leftrightarrow\; L_2^x(BKs) = L_2^x t \end{aligned}$$

which means we can restrict our attention on equations without quantifiers.

The examined equations will be over a fixed language $\mathcal{L} := \Theta \cup \{K, S\}$ and variables, which will be eliminated by L-combinators:

Definition 9. A retraction x is called a *solution* of a Θ-equation $< L_k s, L_k t >$, if $L_k^x s = L_k^x t$ which is equivalent to:

$$\forall\, v_1, \ldots, v_k \,:\; x(s(xv_1)\cdots(xv_k)) = x(t(xv_1)\cdots(xv_k))$$

or literally, a retraction x is a solution of $< L_k s, L_k t >$ if the following holds: for all $v_1, \ldots, v_k \in \operatorname{ret}(x)$, if $sv_1 \cdots v_k \in \operatorname{ret}(x)$ and $tv_1 \cdots v_k \in \operatorname{ret}(x)$ then $sv_1 \cdots v_k = tv_1 \cdots v_k$ must hold.

Example. Let s and t be combinators of the form:
$s := \{\emptyset \rightarrow a,\ \{v\} \rightarrow (\{v\} \rightarrow v) :\ a \in A\}$
$t := \{\{\alpha\} \rightarrow \alpha :\ \alpha \subseteq G(A),\ \alpha \text{ finite }\}$
A retraction x is then a solution of $< s, t >$ if the elements of $\operatorname{ret}(x)$ are of the form $\{\{a\}, \{\{a\} \rightarrow a\}, \ldots\}$.

Note 10. To keep the formulas readable, we introduce the following abbreviation: Let x be a retraction, s a combinator and $v_1, \ldots, v_k$ any variables. Then, we write $s(\overline{xv})$ for $s(xv_1)\cdots(xv_k)$.

4 Birkhoff's Theorem for Retract Equations

Our target will be to show Birkhoff's theorem for equations with retract-solutions in a combinatory model $\mathcal{M}$. Solutions of such equations are originally algebras or (relational) structures of some special type. Our approach will be different: we examine equations whose solutions are retractions. These retractions are not usually closed under Θ-terms, which means that solutions need not be structures in the usual sense.

In [11] it is shown how to represent any algebraic or relational structure in a combinatory model using retractions. Our approach is therefore a generalisation of Birkhoff's original theorem.

Assume x is a retract solution of some set of equations Γ. First, we investigate the question of how to find new solutions of Γ from solutions x. For this we define subretractions, homomorphic images and products of retractions.

Definition 11. y is called a *Θ-subretraction* of x, if

$$\forall v: \; y(yv) = yv \tag{3}$$

$$\forall v: \; x(yv) = yv \tag{4}$$

and for all $\theta_j \in \Theta$, $\forall v_1, \ldots, v_k$:

$$x(\theta_j(\overline{yv})) \;=\; y(\theta_j(\overline{yv})) \tag{5}$$

Θ-subretractions are retractions (3), which define *subsets* of the retract of x (4) and which preserve Θ-equations (5). Hence, as can easily be observed, they satisfy the following lemma:

Lemma 12. Let x be a retraction and y a Θ-subretraction of x. If x is a solution of some Θ-equation $< L_k s, L_k t >$, then y is a solution too.

Proof. $\forall v_1, \ldots, v_k$:

$$\begin{aligned} & x(s(\overline{xv})) = x(t(\overline{xv})) \\ & \Rightarrow \; x(s(x(\overline{yv}))) = x(t(x(\overline{yv}))) \\ & \Rightarrow \; x(s(\overline{yv})) = x(t(\overline{yv})) \\ & \Rightarrow \; y(s(\overline{yv})) = y(s(\overline{yv})) \end{aligned}$$

□

In a next step we define *Θ-homomorphisms* and show that *Θ-homomorphic images* of a solution x are also solutions:

Definition 13. Let x and y be retractions in $\mathcal{M}$ and $\varphi \in M$ a combinator. φ is called a *Θ-homomorphism* from x to y if for all $\theta \in \Theta$:

$$y(\theta(\overline{\varphi(xv)})) = y(\varphi(\theta(\overline{xv}))) \tag{6}$$

φ is called a *Θ-epimorphism*, if for all $\theta_i, \theta_j \in \Theta$:

$$\begin{aligned} \forall v_1, \ldots, v_k: \; & y(\theta_i(\overline{\varphi(xv)})) = y(\theta_j(\overline{\varphi(xv)})) \\ & \Rightarrow \; y(\theta_i(\overline{yv})) = y(\theta_j(\overline{yv})) \end{aligned} \tag{7}$$

A combinator φ is a Θ-homomorphism if it is compatible with Θ-combinators in ret(y) (6). However, to be a Θ-epimorphism, φ needs not be onto ret(y).

With these definitions we can show:

Lemma 14. Let x and y be retractions and φ a Θ-epimorphism from x to y. If x is a solution of a Θ-equation $< L_k s, L_k t >$, then so is y.

Proof. Assume x is a solution of some Θ-equation $< L_k s, L_k t >$, so by definition of L holds for all $v_1, \ldots, v_k$:

$$x(s(\overline{xv})) = x(t(\overline{xv}))$$

and

$$y(\varphi(s(\overline{xv}))) = y(\varphi(s(\overline{xv})))$$

As φ is a Θ-homomorphism, we can infer

$$y(s(\overline{\varphi(xv)})) = y(s(\overline{\varphi(xv)}))$$

which is equivalent to

$$L_k^{B\varphi x}(B^k ys) = L_k^{B\varphi x}(B^k yt)$$

and by (7)

$$L_k^y(B^k ys) = L_k^y(B^k yt)$$

we conclude that y is also a solution. □

A third possibility to find new solutions is by forming *products* of given solutions.

Definition 15. Let J and $\{x(Jj) : j \in M\}$ be retractions. y is called a *Θ-product* of $\{x(Jj) : j \in M\}$ if the following equations are satisfied:
for all v:

$$y(yv) = yv \tag{8}$$

for all $\theta \in \Theta$ and all $v_1, \ldots, v_k$:

$$\begin{aligned} &y(\theta(yv_1)\cdots(yv_k))(Jj) \\ &\quad = x(Jj)(\theta(x(Jj)(v_1(Jj)))\cdots(x(Jj)(v_k(Jj)))) \end{aligned} \tag{9}$$

and

$$L^J(y(\theta(\overline{yv}))) = y(\theta(\overline{yv})) \tag{10}$$

To be a Θ-product of a given set of retractions, y itself has to be a retraction (8). The equations in (9) and (10) confirm that the Θ-terms in the *projections* $y(Jj)$ of y have the same normal form as those in $x(Jj)$. These conditions are sufficient to satisfy the following lemma.

Lemma 16. Let $\{x(Jj) : j \in M\}$ be a set of solutions of some Θ-equation $< L_k s, L_k t >$. If y is a Θ-product of $\{x(Jj) : j \in M\}$ then y is also a solution.

Proof. Let $< L_k s, L_k t >$ be a Θ-equation such that $\{x(Jj) : j \in M\}$ are solutions of it. Hence:
$\forall j \in \mathrm{ret}(J), \forall v_1, \ldots, v_k$:

$$(xj)(s((xj)(v_1 j))\cdots((xj)(v_k j))) = (xj)(t((xj)(v_1 j))\cdots((xj)(v_k j)))$$

By equation (9) we can infer for all $v_1, \ldots, v_k$:

$$\forall j : \quad y(s(\overline{yv}))(Jj) \;=\; y(t(\overline{yv}))(Jj)$$

By definition of L^J this is equivalent to:

$$L^J(y(s(\overline{yv}))) = L^J(y(t(\overline{yv})))$$

and as the Θ-terms are in normal form over J we conclude:

$$y(s(\overline{yv})) \;=\; y(s(\overline{yv}))$$

thus, the product y is also a solution of $< L_k s,\ L_k t >$. □

4.1 Proof of a HSP-Theorem for Retract-Solutions

Now, we have found operations, which preserve Θ-equations. It is much more difficult, however, to ascertain whether or not these three operations (Θ-subretraction, Θ-homomorphism, Θ-product) are sufficient to construct all solutions of a given set of equations. The following theorem provides an answer:

Theorem 17. Let $\mathcal{M} := \langle M; \cdot, K, S, L\rangle$ be a combinatory model. Given a set $\mathcal{K} := \{xi : i \in I\}$ of retractions in M, $\mathcal{K}$ is equational iff $\mathcal{K}$ is closed under Θ-subretractions, Θ-epimorphisms and Θ-products.

Proof. We will split the proof into lemmas 18 and 19. Let $\Gamma(x_i)$ be the set of Θ-equations, of which $x_i \in \mathcal{K}$ is a solution. We define $\Gamma(\mathcal{K}) := \cap_{i \in I}\Gamma(x_i)$. Let y be a solution of $\Gamma(\mathcal{K})$. M is not empty, so is $\mathrm{ret}(y)$ either, because for all v in M, yv is an element of $\mathrm{ret}(y)$.

We define Ψ to be the following set of combinators:

$$\Psi := \{\varphi : \varphi(yv) = x(\varphi(yv))\}$$

We choose a retraction χ with the property $|\mathrm{ret}(\chi)| \geq |\Psi|$. Without loss of generality, we can assume that χ is an enumeration of Ψ so $\Psi = \{\varphi_i : i \in \chi\}$. Let x^χ be a power of x. We then define the following combinator:

$$\phi(yu)(\chi i) := \varphi_i(yu)$$

In a next step we describe the image of ϕ in x^χ: Let v_0 be an element of $\mathrm{ret}(y)$.

$$zu := \begin{cases} x^\chi u & \exists v,\ u = \phi(yv) \\ x^\chi(\theta(\overline{zv})) & \exists v_1, \ldots, v_k,\ u = x^\chi(\theta(\overline{\phi(yv)})) \\ x^\chi(\phi(yv_0)) & \text{else} \end{cases}$$

Lemma 18. z is a Θ-subretraction of x^χ.

Proof. We first show that z is a retraction:

$$z(zu) = \begin{cases} z(x^\chi u) & \exists v, u = \phi(yv) \\ z(x^\chi(\theta(\overline{\phi(yv)}))) & \text{or} \\ z(x^\chi(\phi(yv_0))) & \text{else} \end{cases}$$

As $x^\chi(\phi(yv)) = \phi(yv)$, the assumption is clear.

In the second step, we show that z is a Θ-subretraction. We have to show that for all $\theta \in \Theta$ and all $v_1, \ldots, v_k$:

$$x^\chi(\theta(zv_1)\cdots(zv_k)) = z(\theta(zv_1)\cdots(zv_k))$$

$z(\theta(zv_1)\cdots(zv_k)) = z(\theta(\phi(yw_1))\cdots(\phi(yw_k)))$ for some $w_1, \ldots, w_k$.

$$\begin{aligned} z(\theta(\phi(yw_1))\cdots(\phi(yw_k))) &= z(\phi(\theta(yw_1)\cdots(yw_k))) \\ &= x^X(\phi(\theta(yw_1)\cdots(yw_k))) \\ &= x^X(\theta(\phi(yw_1))\cdots(\phi(yw_k))) \\ &= x^X((zv_1)\cdots(zv_k)) \end{aligned}$$

□

It remains to be shown that y is a homomorphic image of the retraction z. We define:

$$\psi(zu) := \begin{cases} yv \underline{\qquad\quad} & \exists v,\ u = \phi(yv) \\ y(s(\overline{\psi(zv)})) & \exists v_1, \ldots, v_k,\ \exists s \in \theta\ u = x^X(s(\overline{zv})) \end{cases}$$

Lemma 19. ψ is a Θ-epimorphism from z to y.

Proof. Let u and v be combinators in ret(y) with $u \neq v$. By definition of ϕ we can also infer that $\phi v \neq \phi u$, which means that $x^X(\phi v) \neq x^X(\phi u)$.

Assume that $y(s(\overline{yv})) \neq y(t(\overline{yv}))$ for some Θ-terms $L_k s$ and $L_k t$. As y is a solution of $\Gamma(\mathcal{K})$ we can also infer that $x^X(s(\overline{x^X v})) \neq x^X(t(\overline{x^X v}))$, hence $z(s(\overline{zv})) \neq z(t(\overline{zv}))$, from where we can infer, that ψ is well defined. Obviously, ψ is also a Θ-epimorphism.

□

This completes the proof for Theorem 17, as we have constructed a Θ-subretraction z of a Θ-product x^X and a Θ-homomorphism ψ from z to y.

5 Birkhoff's Theorem for Combinator Solutions

In a second approach, we drop the condition that solutions have to be retractions, i.e. every combinator (or every cumulative logic program) is admitted as a solution (or as an approximation).

The intended meaning of such a solution (set) is the following: Let x be any combinator, a combinator a is an *element* of a set $\tilde{x}$ if a can be written as $a = a'x$. Or to put it simply:

$$\tilde{x} := \{ax : a \in D\}$$

This gives a generalization of the retract-construction we used in the last section. Let x be any retraction, then xa is an element of the retract of x. However, xa can be written as $xa = CIax$ and because $a' = CIa$, a is shown to be an element of $\tilde{x}$.

Therefore, the equations we examine will be of the form $< s, t >$, where a combinator x is a solution of this equation, if x satisfies $sx = tx$.

As in the previous section, we first try to find operations which preserve Θ-equations. Roughly speaking, these are multiplications from the right and from the left. The strength of combinatory logic allows us to deduce some results from these two operations already.

Definition 20. Let Θ be a set of distinguished combinators and Γ any set of Θ-equations, i.e.

$$\Gamma \subseteq \{< s,t >: \; s,t \; \Theta\text{-terms}\}$$

Now, a combinator x is a *solution of* Γ if x satisfies $sx = tx$ for every pair $< s,t >$ in Γ.

This is a more general definition than the one we had before, because for every term of the form $x(s(\overline{xv}))$ we can (by combinatory completeness) find a term of the form $s'x$, which is equivalent to $x(s(\overline{xv}))$.

If $sx = tx$ then for all combinators ν and μ, $\nu(sx)\mu = \nu(tx)\mu$ holds. This leads to the definition of Θ-restrictions.

Definition 21. y is called a *Θ-restriction* of x if there are combinators ν, μ so that for all Θ-terms s:

$$sy = \nu(sx)\mu \tag{11}$$

Obviously, the following lemma is true:

Lemma 22. If a combinator x is a solution of a set of equations Γ, then every restriction of x is a solution, too.

Proof. Let $< s,t >$ be a Θ-equation and assume x to be a solution of this equation. Then:

$$\begin{aligned} sx = tx \; &\Rightarrow \; \forall \nu, \; \nu(sx) = \nu(tx) \\ &\Rightarrow \; \forall \nu, \mu \; \nu(sx)\mu = \nu(tx)\mu \\ &\Rightarrow \; sy = ty \end{aligned}$$

□

x is always a Θ-restriction of itself. We can show this by choosing $\nu = \mu = K$, because for all s, $K(sx)K = sx$.

We will call a combinator φ homomorphic on x if φ commutes on the *Θ-elements* of x.

Definition 23. A combinator φ is a *Θ-homomorphism* on x if φ satisfies for all Θ-terms u:

$$\varphi(ux) = u(\varphi x)$$

φx is then called the *Θ-homomorphic image* of φ.

Every homomorphism on x leads to a Θ-restriction of x. Because $\varphi(sx)$ is equal to $K(\varphi(sx))I$ for all Θ-terms s, and as $K(\varphi(sx))I$ is equal to $BK\varphi(sx)I$, we can choose $\nu = BK\varphi$ and $\mu = I$ and obtain $s(\varphi x) = \nu(sx)\mu$.

Therefore, we can state:

Lemma 24. If y is the homomorphic image of a combinator x (and of a Θ-homomorphism φ), then y is a Θ-restriction of x.

Lemma 25. Let x be a solution of any Θ-equation $< s,t >$ and φ a Θ-homomorphism on x. If $y = \varphi x$ is the homomorphic image of φ, then y is a solution of $< s, t >$.

Proof. Let s and t be any Θ-terms such that x is a solution of $< s,t >$.

$$\begin{aligned} sx = tx \;&\Rightarrow\; \varphi(sx) = \varphi(tx) \\ &\Rightarrow\; s(\varphi x) = t(\varphi x) \\ &\Rightarrow\; sy = ty \end{aligned}$$

□

We will call a combinator y a product of a set of combinators $\{xj : j \in J\}$, if its ***projections*** (yj), $j \in J$ behave on Θ-terms the same way as (xj), $j \in J$.

Definition 26. Let J be any retraction and $\{xj : j \in J\}$ a set of combinators in D. y is called a *Θ-product* of $\{xj : j \in J\}$ if for all Θ-terms s:

$$L^J(sy) = sy \tag{12}$$

and

$$L^J(sy) = L^J(Bsx) \tag{13}$$

These two conditions are sufficient to show the following lemma:

Lemma 27. If $\{xj : j \in J\}$ are solutions of any Θ-equation $< s,t >$ and y is a Θ-product of $\{xj : j \in J\}$, then y is also a solution of $< s,t >$.

Proof. Assume y to be a Θ-product of the set of combinators $\{xj : j \in J\}$, and s and t two Θ-terms, so that for all j in J, xj is a solution of $< s,t >$:

$$\begin{aligned} &\forall j : \; s(x(Jj)) = t(x(Jj)) \\ &\Rightarrow \forall j : \; (sy)(Jj) = (ty)(Jj) \\ &\Rightarrow L^J(sy) = L^J(ty) \\ &\Rightarrow sy = ty \end{aligned}$$

because by equation (12) sy and ty are in normal form. □

Proof of the HSP-Theorem

We have shown that every Θ-restriction, every Θ-homomorphic image, and every Θ-product of solutions of Θ-equations are new solutions. In the next theorem we will show that the converse is also true. We begin with some definitions.

Let $\tilde{\Theta}$ be the set of Θ-terms and x a combinator. We define Φ to be the following set of combinators:

$$\begin{aligned}\Phi_x &:= \{\varphi_j : \exists \nu_j, \mu_j, \forall s \in \tilde{\Theta}, \ \varphi_j s = \nu_j(sx)\mu_j\} \\ &= \{\varphi_j : \varphi_j = C(B(C\nu_j\mu_j))x, \ \nu_j, \mu_j \in D\}\end{aligned}$$

Let κ be a retraction with $|\mathrm{ret}(\kappa)| \geq |\Phi_x|$.

For $j \in \kappa$ let zj be the image of Θ under φ_j , i.e.

$$sz(\kappa j) \ := \ \ \varphi_j s \quad (= \ \nu_j(sx)\mu_j)$$

Some of the φ_j may occur more than once if $|\mathrm{ret}(\kappa)| > |\Phi_x|$.

Lemma 28. Let $\mathcal{K} := \{x_j : j \in \kappa\}$ be a set of combinators in D, Γ the set of Θ-equations which are satisfied by $\mathcal{K}$ and x the Θ-product of $\mathcal{K}$. Let Φ_x and zj be the combinators as defined above and $\bar{z}$ the Θ-product of these zj, $j \in \kappa$. Every solution y of Γ is then a Θ-homomorphic image of $\bar{z}$.

Proof. Define ψ as the following homomorphism:

$$\psi(s\bar{z}) := sy \quad (= s(\psi z) \ )$$

We have to show that ψ is well defined.

Assume $s\bar{z} = t\bar{z}$ for any Θ-terms s and t. By definition of $\bar{z}$ we find:

$$\begin{aligned}s\bar{z} = t\bar{z} \ &\Rightarrow \ \forall j : \ s\bar{z}(\kappa j) = t\bar{z}(\kappa j) \\ &\Rightarrow \ \forall j : \ (\nu_j)(sx)(\mu_j) = (\nu_j)(tx)(\mu_j)\end{aligned}$$

Now, assume $sx \neq tx$. We choose φ_i with $\nu = \mu = K$ and find $\nu(sx)\mu \neq \nu(tx)\mu$. We can then infer $s\bar{z} \neq t\bar{z}$, but this is a contradiction.

So $sx = tx$ must be true and, furthermore, $sy = ty$ also, as y is a solution of Γ. So ψ is well defined. □

Finally, we can show:

Theorem 29. Let $\mathcal{D} := \langle D; \cdot, K, S, L\rangle$ be a combinatory model and $\mathcal{K} := \{x_j : j \in J\}$ a set of combinators which is closed under Θ-restrictions, Θ-products and Θ-homomorphic images.

Let Γ be the set of Θ-equations which are satisfied by all of these combinators in $\mathcal{K}$. If y is a solution of the equations Γ, then y is an element of $\mathcal{K}$.

Proof. Let x be a Θ-product of $\{x_j : j \in J\}$. Defining $\bar{z}$ the same way as above, every $z(\kappa j)$ is a Θ-restriction of the product x. So $\bar{z}$ is a Θ-product of Θ-restrictions of x.

y is then the Θ-homomorphic image of $\bar{z}$ where the homomorphism ψ is defined as before.

References

[1] K. Aberer, *Combinatory differential fields and constructive analysis.* PhD thesis ETH No. 9357, ETH Zürich, (1991).

[2] B. Amrhein, *Universal Algebra in Combinatory Logic.* PhD thesis ETH No. 10 005, ETH Zürich, (1992).

[3] H.P. Barendregt. *The Lambda Calculus.* North Holland, Amsterdam, (1984).

[4] M. J. Beeson. *Foundations of Constructive Mathematics.* Springer, (1984).

[5] G. Birkhoff. *On the Structure of Abstract Algebras.* Proc. Cambridge Phil. Soc. 31, (1935).

[6] E. Engeler. *Cumulative Logic Programs and Modelling.* Logic Colloquium '86, North Holland, pp. 83 - 93, (1988).

[7] E. Engeler. *Sketch of a New Discipline of Modelling.* Report Math. Dept. ETH Zürich, (1988).

[8] J. W. Lloyd. *Foundations of Logic Programming.* Springer (1987).

[9] J. Hodas, D. Miller, *Logic Programming in a Fragment of Intuitionistic Linear Logic,* Logics in Computer Science, Amsterdam, (1991).

[10] G. Schwärzler *Knowledge-Based Modelling of Cooperative Processes.* PhD thesis ETH No. 9926, ETH Zürich (1992).

[11] T. Weibel, *Extension of Combinatory Logic to a Theory of Combinatory Representation,* TCS, 97: 157-173, (1992).