K. INDERMARK

Reduction semantics for rational schemes

RAIRO – Informatique théorique, tome 18, nº 3 (1984), p. 209-223.
<http://www.numdam.org/item?id=ITA_1984__18_3_209_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 18, n° 3, 1984, p. 209 à 223)

REDUCTION SEMANTICS FOR RATIONAL SCHEMES (*)

by K. INDERMARK (1)
Communicated by J.-E. PIN

Abstract. — *For rational schemes, built up inductively by combinators for projection, composition, and resolution, we introduce reduction semantics and give an algebraic proof of their equivalence to denotational semantics.*

Résumé. — *Pour des schémas rationnels, construits par récurrence par combinaison de projections, compositions, et résolutions, nous introduisons la sémantique par réduction et nous donnons une preuve algébrique de son équivalence avec la sémantique dénotationnelle.*

INTRODUCTION

Rational schemes form a very simple class of abstract recursive definitions insofar as the recursion involved can already be represented by regular equations with parameters. Following [8], these schemes are built up inductively from a set Ω of operation symbols together with combinators $\mathbb{P}_i$ for projections, $\mathbb{C}$ for composition, and $\mathbb{R}_i$ for resolution of regular equations. Interpreting Ω by a complete Ω-algebra $\mathscr{A}$, a rational scheme defines a rational operation on $\mathscr{A}$.

The usefulness of such a rational calculus has already been recognized in [2, 12, 13, 14], because it describes the control structure of many recursively defined objects in computer science. E. g., Kleene's theorem for finite automata and Engeler's block-normal-form result for flowcharts can be understood as instances of a general normal-form theorem for rational schemes, [7]. For this and many other applications it is essential to allow the explicit use of composition and nested multiple recursion.

(*) Received in December 1982, revised in April 1983.

(1) Lehrstuhl für Informatik II, RWTH Aachen, Büchel 29-31, 5100 Aachen, W-Germany.

However, application of rational schemes is not restricted to "regular recursion". "Context-free recursion" [11] and even recursion on higher functional domains [3, 4] can be treated appropriately taking derived algebras as interpretations. In fact, we have shown in [9] how to obtain Damm's recursion hierarchy theorem using this technique.

The last observation demonstrates that the rational calculus can well be viewed as an alternative to the typed λ-calculus with fixed-point operators. First, our calculus is based on typed combinators without using variables, second, and in contrast to [6], higher type recursion is split up into regular recursion and derivation.

The purpose of this paper is to give a proper operational semantics to rational schemes and prove its equivalence to denotational semantics. We prefer the notion of reduction semantics because the order of reduction steps will not be specified. It might be an interesting consequence of this to consider the rational calculus in the context of functional programming. Our reduction semantics would directly lead to implementation on reduction machines.

The paper is organized as follows: We start by recalling abstract syntax and denotational semantics of rational schemes as introduced in [8]. In particular, we are using algebras without rank, i. e., an operation has an arbitrary number of arguments. Thereby, we avoid many-sorted algebras in the treatment of higher-type recursion. Next, we introduce reduction rules on computation terms and define a corresponding reduction relation by means of structural induction. This purely algebraic definition of the reduction relation turns out to be very useful because parallel reductions can be performed within one step. As a consequence, we get an easy proof of a standardization theorem. Then, we apply this result to show that the values reducible from a computation term form a directed set. This leads in a natural way to reduction semantics.

1. DENOTATIONAL SEMANTICS OF RATIONAL SCHEMES

In this section we briefly recall abstract syntax and denotational semantics of rational schemes. They were introduced in [8] where one can also find more motivation for our choice of rank-free algebras.

Let A be a set.

Then $\underline{\text{Ops}}(A) := \{f \mid f : A^* \to A\}$ is the set of *operations on A*.

Let Ω be a set of *operations symbols* (without arities).

Then $\varphi : \Omega \to \underline{\text{Ops}}(A)$ determines an *Ω-algebra:*

$$\mathcal{A} := \langle A; \varphi \rangle \in \underline{\text{Alg}}_{\Omega}.$$

For any set X there exists:

$$\mathcal{F}_\Omega(X) \in \underline{\mathrm{Alg}}_\Omega\ , \quad \text{freely generated by } X.$$

Hence, any assignment $\alpha: X \to A$ with $\mathcal{A} = \langle A;\ \varphi \rangle \in \underline{\mathrm{Alg}}_\Omega$ extends uniquely to a homomorphism:

$$\bar{\alpha}: \quad \mathcal{F}_\Omega(X) \to \mathcal{A}.$$

If $X = \emptyset$, we simply write $\mathcal{F}_\Omega$ instead of $\mathcal{F}_\Omega(\emptyset)$ and denote the unique homomorphism by $h_{\mathcal{A}}$.

The same situation arises when starting from a complete partially ordered set A—complete with respect to directed subsets—and continuous operations on A. The corresponding objects are denoted by:

$$\underline{\mathrm{Alg}}^c_\Omega, \qquad \mathcal{F}^c_\Omega(X) \quad \text{and} \quad h^c_{\mathcal{A}}.$$

For an arbitrary complete algebra $\mathcal{A} \in \underline{\mathrm{Alg}}^c_\Omega$ there exists a natural class of operations, so-called *rational operations*, which can be obtained uniformly from the class of projections by means of left-composition with base operations of $\mathcal{A}$, composition and resolution. The essential construction is that of resolution: it corresponds to the least solution of a system of regular equations with parameters.

Syntax of rational Ω-schemes

We define the set $R(\Omega)$ of *rational symbols over* Ω by

$$R(\Omega) := \{F' \mid F \in \Omega\} \cup \{\mathbb{P}_i \mid i > 0\} \cup \{\mathbb{C}\} \cup \{\mathbb{R}_i \mid i > 0\}$$

and the algebra Rat_Ω of *rational Ω-schemes* by

$$\mathrm{Rat}_\Omega := \mathcal{F}_{R(\Omega)}.$$

In contrast to our previous treatment we distinguish F' from F because F' will denote left-composition with the base operation denoted by F. Moreover, instead of only one resolution symbol $\mathbb{R}$ we take $\mathbb{R}_i$ in order to consider the i-th equation as the defining equation. This modification becomes necessary for a proper reduction semantics as we shall see below.

Without choosing a particular representation of the initial algebra $\mathcal{F}_{R(\Omega)}$, we get an inductive description of its carrier $\underline{\mathrm{Rat}}_\Omega$: the *set* $\underline{\mathrm{Rat}}_\Omega$ of rational

Ω-schemes is the least K such that:

(1) $$G \in R(\Omega) \Rightarrow G(e) \in K,$$

(2) $$S_1, \ldots, S_n \in K, \quad n \geqq 1, \quad G \in R(\Omega) \Rightarrow G(S_1 \ldots S_n) \in K.$$

Here, $G(S_1 \ldots S_n)$ is meant to be the free application of $\varphi_{\mathscr{F}}(G)$ to the arguments $S_1, \ldots, S_n$ in the initial algebra $\mathscr{F}_{R(\Omega)}$, and the special case $G(e) \in \underline{\mathrm{Rat}}_{\Omega}$ is short for $\varphi_{\mathscr{F}}(G)(e)$ the free application to the empty argument list. Remember that an operation can have an arbitrary number of arguments. *See* [8] for more details.

Semantics of rational Ω-schemes

Let Ω be interpreted by $\mathscr{A} = \langle A; \varphi \rangle \in \underline{\mathrm{Alg}}^c_{\Omega}$.

For the semantics of rational Ω-schemes in $\mathscr{A}$ it suffices to construct an $R(\Omega)$-algebra because of the initiality of Rat_{Ω} in $\underline{\mathrm{Alg}}_{R(\Omega)}$. Therefore, we define the *rational algebra:*

$$R(\mathscr{A}) := \langle \underline{\mathrm{Ops}}(A); \bar{\varphi} \rangle$$

of $\mathscr{A}$ by

$$\bar{\varphi}(F')(f_1 \ldots f_r)(a) := \varphi(F)(f_1(a) \ldots f_r(a)),$$
$$\bar{\varphi}(\mathbb{P}_i)(f_1 \ldots f_r)(a) := \underline{\mathrm{proj}}_i(a),$$
$$\bar{\varphi}(\mathbb{C})(e)(a) := \perp_A,$$
$$\bar{\varphi}(\mathbb{C})(f_0 \ldots f_r)(a) := f_0(f_1(a) \ldots f_r(a)),$$
$$\bar{\varphi}(\mathbb{R}_i)(f_1 \ldots f_r)(a) := \underline{\mathrm{proj}}_i(\underline{\mathrm{fix}}(f)),$$
$$\text{where } f: A^r \to A^r$$
$$b \mapsto f_1(ba) \ldots f_r(ba).$$

Here, $f_i: A^* \to A$, $a \in A^*$, $r \in \mathbb{N}$, $\perp_A$ is the least element of A, $\underline{\mathrm{proj}}_i: A^* \to A$ is defined by $\underline{\mathrm{proj}}_i(a_1 \ldots a_r) := \underline{\mathrm{if}}\ 1 \leqq i \leqq r\ \underline{\mathrm{then}}\ a_i\ \underline{\mathrm{else}}\ \perp_A$ and $\underline{\mathrm{fix}}$ takes the least fixed-point of a continuous transformation.

This definition shows that F' is taken as left-composition with $\varphi(F)$, $\mathbb{P}_i$ as the constant functional that yields $\underline{\mathrm{proj}}_i$, $\mathbb{C}$ as composition and $\mathbb{R}_i$ as the i-th component of the solution of the following system of regular equations with parameters:

$$\begin{cases} x_1 = f_1(x_1 \ldots x_r\, a), \\ \ldots\ldots\ldots\ldots\ldots \\ x_r = f_r(x_1 \ldots x_r\, a). \end{cases}$$

Note that according to our convention $\underline{\mathrm{Ops}}(A)$ contains only continuous operations because we started from *a* cpo A, and that continuity is preserved under each functional $\bar{\varphi}(G)$ with $G \in R(\Omega)$.

The semantics of rational Ω-schemes is now given by the initial $R(\Omega)$-homomorphism:

$$h_{R(\mathscr{A})}: \quad \mathrm{Rat}_\Omega \to R(\mathscr{A}),$$

and we denote the semantics of a rational scheme S in $\mathscr{A}$ by

$$[\![S]\!]_{\mathscr{A}} := h_{R(\mathscr{A})}(S) : A^* \to A.$$

2. COMPUTATION TERMS AND REDUCTIONS

We just explained the meaning of a rational scheme $S \in \underline{\mathrm{Rat}}_\Omega$ interpreted by $\mathscr{A} \in \underline{\mathrm{Alg}}^c_\Omega$ as a certain operation of $\mathscr{A}$, also called *rational operation.* Its application to a value $a \in A^*$ produces generally an *infinite* object:

$$[\![S]\!]_{\mathscr{A}}(a) \in A,$$

insofar as fixed-points are involved. More precisely, $[\![S]\!]_{\mathscr{A}}(a)$ can be understood as the homomorphic image of an infinite rational tree, [8].

Here, we shall present a method for computing the value $[\![S]\!]_{\mathscr{A}}(a)$ by means of finite approximations. For this purpose, we generalize the reduction semantics given in [10, 11], also *see* [5], for rational and recursive schemes in equational normal form using schematic grammars.

Given an interpretation $\mathscr{A} \in \underline{\mathrm{Alg}}^c_\Omega$ of Ω we define the set of *computation terms* of $\mathscr{A}$ as:

$$\underline{\mathrm{Comp}_\Omega}(\mathscr{A}) := F_{\Omega \cup \underline{\mathrm{Rat}_\Omega}}(A),$$

i. e., as the carrier of the $(\Omega \cup \underline{\mathrm{Rat}}_\Omega)$-algebra freely generated by A.

Though we speak of terms, again we do not use any particular representation of $\underline{\mathrm{Comp}}_\Omega(\mathcal{A})$, only its unique generation from A by free application of $\varphi_{\mathcal{F}}(F)$ for $F \in \Omega$ and of $\varphi_{\mathcal{F}}(S)$ for $S \in \underline{\mathrm{Rat}}_\Omega$.

Hence, a computation term $E \in \mathcal{C} := \underline{\mathrm{Comp}}_\Omega(\mathcal{A})$ is:

either *atomic*: $E \in A$,
or an *Ω-term*: $E = F(E_1 \ldots E_r)$,
or a *$\underline{\mathrm{Rat}}_\Omega$-term*: $E = S(E_1 \ldots E_r)$,
where $F \in \Omega$, $S \in \underline{\mathrm{Rat}}_\Omega$, $E_i \in \mathcal{C}$ and $r \in \mathbb{N}$.

Now, the denotational semantics of rational schemes extends canonically to computation terms. For $E \in \mathcal{C}$ we define:

$$[\![E]\!]_{\mathcal{A}} \in A$$

by induction on the algebraic structure of $\mathcal{C}$:

$$\begin{aligned}
[\![a]\!]_{\mathcal{A}} &:= a,\\
[\![F(E_1 \ldots E_r)]\!]_{\mathcal{A}} &:= \varphi(F)([\![E_1]\!]_{\mathcal{A}} \ldots [\![E_r]\!]_{\mathcal{A}}),\\
[\![S(E_1 \ldots E_r)]\!]_{\mathcal{A}} &:= [\![S]\!]_{\mathcal{A}}([\![E_1]\!]_{\mathcal{A}} \ldots [\![E_r]\!]_{\mathcal{A}}).
\end{aligned}$$

For their reduction semantics we first introduce a set of *reduction rules* which allow the elimination of rational symbols.

Let $F \in \Omega$, $S = S_1 \ldots S_s \in \underline{\mathrm{Rat}}_\Omega^*$, $E = E_1 \ldots E_r \in \mathcal{C}^*$ and $a \in A^*$. Then we take the following rules:

$$\begin{aligned}
&F(a) \to \varphi(F)(a),\\
&F'(S_1 \ldots S_s)(E) \to F(S_1(E) \ldots S_s(E)),\\
&\mathbb{P}_i(S)(E) \to \underline{\mathrm{if}}\ i \leqq r\ \underline{\mathrm{then}}\ E_i\ \underline{\mathrm{else}}\ \bot_A,\\
&\mathbb{C}(e)(E) \to \bot_A,\\
&\mathbb{C}(S_1 \ldots S_s)(E) \to S_1(S_2(E) \ldots S_s(E)) \qquad (s \geqq 1),\\
&\mathbb{R}_i(S)(E) \to \bot_A,\\
&\mathbb{R}_i(S)(E) \to S_i(\mathbb{R}_1(S)(E) \ldots \mathbb{R}_s(S)(E)\,E) \qquad (i \leqq s).
\end{aligned}$$

The essential rules are of course the last two because they will describe the fixed-point approximations according to Kleene's fixed-point Theorem.

These reduction rules determine a *reduction relation*:

$$\vdash\ \subseteq \mathcal{C}^2,$$

namely the least precongruence containing these rules and the identity relation on $\mathscr{C}$, i. e.:

(1) If $E_1 \rightarrow E_2$ is a reduction rule, then $E_1 \vdash E_2$.

(2) For each $E \in \mathscr{C}$ we have $E \vdash E$.

(3) If $E_i \vdash E'_i$, $1 \leqq i \leqq r$, $F \in \Omega$, $S \in \underline{\mathrm{Rat}}_\Omega$,

then $F(E_1 \ldots E_r) \vdash F(E'_1 \ldots E'_r)$,
and $S(E_1 \ldots E_r) \vdash S(E'_1 \ldots E'_r)$.

(4) There are no other reduction steps.

Note that we gave a purely algebraic definition of the reduction relation, not a syntactic one using "contexts". We shall *see* that this kind of reduction relation facilitates the proof of the standardization theorem as it allows parallel reductions in one step. This technique has already been used by Tait and Martin-Löf for proving the Church-Rosser theorem of the λ-calculus, *see* [1].

3. STANDARDIZATION

Before we proceed to define reduction semantics, we simplify our task insofar as we restrict ourselves to left-reductions.

A a special case of $\vdash$ we get the *left-reduction relation* $\underset{l}{\vdash} \subseteq \mathscr{C}^2$:

(1) If $E_1 \rightarrow E_2$ is a reduction rule, then $E_1 \underset{l}{\vdash} E_2$.

(2) If $F(a E_1 \mathrm{E}) \in \mathscr{C}$ with $F \in \Omega$, $a \in A^*$, $E \in \mathscr{C}^*$, and $E_1 \underset{l}{\vdash} E_2$, then $F(a E_1 E)$ $\underset{l}{\vdash} F(a E_2 E)$.

Standardization theorem

If $E \in \underline{\mathrm{Comp}}_\Omega(A)$ and $a \in A$, then $E \overset{*}{\vdash} a$ implies $E \overset{*}{\underset{l}{\vdash}} a$.

Proof by induction on the reduction length n.

(1) $n=0$: $E \overset{0}{\vdash} a$ implies $E=a$ and $E \overset{0}{\underset{l}{\vdash}} a$.

(2) $n \rightsquigarrow n+1$: By induction hypothesis, $E \overset{n}{\vdash} a$ implies $E \overset{*}{\underset{l}{\vdash}} a$ for all $E \in \mathscr{C}$ and $a \in A$.

Now, let $E \overset{n+1}{\vdash} a$. We proceed by case analysis on $E \in \mathscr{C}$.

(a) $E = a' \overset{n+1}{\vdash} a$: Necessarily, $a' = a$ and $E \overset{o}{\underset{l}{\vdash}} a$.

(b) $E = F(E_1 \ldots E_r) \overset{n+1}{\vdash} a$: There must be $a_1, \ldots, a_r \in A$ such that $F(E_1 \ldots E_r) \overset{n}{\vdash} F(a_1 \ldots a_r) \underset{l}{\vdash} a$. Therefore we have $E_i \overset{n}{\vdash} a_i$ and by induction $E_i \overset{*}{\underset{l}{\vdash}} a_i$. We compose these reductions to a left-reduction sequence:

$$F(E_1 \ldots E_r) \overset{*}{\underset{l}{\vdash}} F(a_1 E_2 \ldots E_r) \overset{*}{\underset{l}{\vdash}} F(a_1 a_2 E_3 \ldots E_r) \ldots \overset{*}{\underset{l}{\vdash}} F(a_1 \ldots a_r) \underset{l}{\vdash} a.$$

(c 1) $E = F'(S_1 \ldots S_s)(E_1 \ldots E_r) \overset{n+1}{\vdash} a$:
This reduction sequence must be of the form:

$$F'(S_1 \ldots S_s)(E_1 \ldots E_r) \overset{p}{\vdash} F'(S_1 \ldots S_s)(E'_1 \ldots E'_r)$$
$$\underset{l}{\vdash} F(S_1(E'_1 \ldots E'_r) \ldots S_s(E'_1 \ldots E'_r)) \overset{q}{\vdash} a.$$

It follows that $E_i \overset{p}{\vdash} E'_i$. Now, we can change the order of reduction *without increasing the number of steps*! This is a consequence of the algebraic definition of $\vdash$ that allows parallel substitutions:

$$F'(S_1 \ldots S_s)(E_1 \ldots E_r) \underset{l}{\vdash} F(S_1(E_1 \ldots E_r) \ldots S_s(E_1 \ldots E_r))$$
$$\overset{p}{\vdash} F(S_1(E'_1 \ldots E'_r) \ldots S_s(E'_1 \ldots E'_r)) \overset{q}{\vdash} a.$$

The assertion then follows from the induction hypothesis because $p + q = n$.

(c 2) $E = \mathbb{P}_i(S_1 \ldots S_s)(E_1 \ldots E_r) \overset{n+1}{\vdash} a$:
Case 1: $i > r$.
Then we have $a = \perp_A$ and $\mathbb{P}_i(S_1 \ldots S_s)(E_1 \ldots E_r) \underset{l}{\vdash} \perp_A$.

Case 2: $1 \leqq i \leqq r$,

$$\mathbb{P}_i(S_1 \ldots S_s)(E_1 \ldots E_r) \overset{p}{\vdash} \mathbb{P}_i(S_1 \ldots S_s)(E'_1 \ldots E'_r)$$
$$\underset{l}{\vdash} E'_i \overset{q}{\vdash} a.$$

The assertion follows as in case (c 1).

(c 3) $E=\mathbb{C}(S_1\ldots S_s)(E_1\ldots E_r)\overset{n+1}{\vdash} a$:

Case 1: $s=0$ is analogous to (c 2) case 1.

Case 2: $s \geqq 1$.

$$\mathbb{C}(S_1\ldots S_s)(E_1\ldots E_r)\overset{p}{\vdash}\mathbb{C}(S_1\ldots S_s)(E'_1\ldots E'_r)$$
$$\underset{l}{\vdash} S_1(S_2(E'_1\ldots E'_r)\ldots S_s(E'_1\ldots E'_r))\overset{q}{\vdash} a.$$

Again, we may change the order of reduction steps without increasing the reduction length and conclude the result from the induction hypothesis.

(c 4) $E=\mathbb{R}_i(S_1\ldots S_s)(E_1\ldots E_r)\overset{n+1}{\vdash} a$:

Case 1: $E\overset{p}{\vdash}\mathbb{R}_i(S_1\ldots S_s)(E'_1\ldots E'_r)\underset{l}{\vdash}\perp_A$ as in (c 2) case 1.

Case 2: $E\overset{p}{\vdash}\mathbb{R}_i(S_1\ldots S_s)(E'_1\ldots E'_r)$

$$\underset{l}{\vdash} S_i(\mathbb{R}_1(S_1\ldots S_s)(E'_1\ldots E'_r)\ldots\mathbb{R}_s(S_1\ldots S_s)(E'_1\ldots E'_r)\,E'_1\ldots E'_r)\overset{q}{\vdash} a.$$

Since all E'_i occur in parallel we can apply the same technique as above.

4. THE DIRECTED VALUE SET OF A COMPUTATION TERM

Generally, there are many ways to reduce a computation term $E\in\mathscr{C}$. So, we are led to define the *value set* of E by:

$$\underline{\text{val}}(E):=\{a\in A \mid E\overset{*}{\vdash} a\}.$$

It was shown in [10, 11] that schematic languages associated with rational and recursive program schemes in equational normal form are directed sets. Here, we prove an analogous result.

THÉORÈME : *For each $E\in\mathscr{C}$ $\underline{\text{val}}(E)$ is a directed subset of A.*

Proof: From the standardization theorem we know that:

$$\underline{\text{val}}(E)=\{a\in A \mid E\overset{*}{\underset{l}{\vdash}} a\}.$$

Hence, we may prove the assertion by induction on the number of left-reduction steps:

$$\forall n \in \mathbb{N} \left\{ \begin{array}{ccc} \forall E \in \mathscr{C}, & \forall n_1, n_2 \in \mathbb{N}, & \forall a_1, a_2 \in A: \\ n_i \leqq n, & E \overset{n_i}{\underset{l}{\vdash}} a_i, & i=1, 2 \\ & \Rightarrow & \\ \exists a \in A: E \overset{*}{\vdash} a & \text{and} & a_1, a_2 \leqq a. \end{array} \right\} \qquad (*)$$

Proof of $(*)$ by induction on n:

(1) $n=0$: $E \overset{0}{\underset{l}{\vdash}} a_i$ implies $a_1 = a_2 =: a$.

(2) $n \rightsquigarrow n+1$: We assume that $(*)$ holds for n.

Let $E \overset{n_1}{\underset{l}{\vdash}} a_1$ and $E \overset{n_2}{\underset{l}{\vdash}} a_2$ such that $\max\{n_1, n_2\} = n+1$. Now, we proceed by case analysis on E.

(a) $E = a' \in A$: This is impossible since $a' \overset{n_i}{\underset{l}{\vdash}} a_i$ implies $n_i = 0$ in contrast to the assumption that $\max\{n_1, n_2\} = n+1$.

(b) $E = F(E_1 \ldots E_r)$:

$F(E_1 \ldots E_r) \overset{n_1}{\underset{l}{\vdash}} a_1$ implies $E_1 \overset{p_1}{\underset{l}{\vdash}} b_1, \ldots, E_r \overset{p_r}{\underset{l}{\vdash}} b_r$ and $F(b_1 \ldots b_r) \underset{l}{\vdash} a_1$ with $p_i \leqq n$.

Analogously, we get $E_1 \overset{q_1}{\underset{l}{\vdash}} c_1, \ldots, E_r \overset{q_r}{\underset{l}{\vdash}} c_r$ and $F(c_1 \ldots c_r) \underset{l}{\vdash} a_2$ with $q_i \leqq n$.

By induction hypothesis, there are $d_i \in \mathbb{N}$ such that:

$$E_i \overset{*}{\vdash} d_i \quad \text{and} \quad b_i, c_i \leqq d_i \quad \text{for} \quad 1 \leqq i \leqq r.$$

Since $a_1 = \varphi(F)(b_1 \ldots b_r)$, $a_2 = \varphi(F)(c_1 \ldots c_r)$ and $\varphi(F)$ monotonic, the assertion follows with:

$$a := \varphi(F)(d_1 \ldots d_r).$$

(c) $E = S(E_1 \ldots E_r)$:

In all cases of S, except one, the first left-reduction step is uniquely determined so that the induction argument applies directly. The exception is given by $S=\mathbb{R}_i(S_1 \ldots S_s)$ with $1 \leqq i \leqq s$ because in that case we have two possible left-reduction steps. But, since one of them derives $\perp_A$, the assertion trivially holds.

5. REDUCTION SEMANTICS OF RATIONAL SCHEMES

The previous result enables us to define the reduction semantics of a computation term $E \in \mathscr{C}$ by

$$[\![E]\!]_{\mathscr{A}}^{\text{red}} := \bigsqcup \underline{\text{val}}\,(E).$$

Remember that A is complete with respect to directed subsets. From the definition we derive reduction semantics for a rational scheme $S \in \underline{\text{Rat}}_\Omega$ interpreted by $\mathscr{A} \in \underline{\text{Alg}}_\Omega$ as follows:

$$[\![S]\!]_{\mathscr{A}}^{\text{red}} : A^* \to A,$$
$$a \mapsto [\![S\,(a)]\!]_{\mathscr{A}}^{\text{red}}.$$

Our main result states that this operational type of semantics coincides with denotational semantics.

THEOREM: *For each $E \in \mathscr{C}$ we have $[\![E]\!]_{\mathscr{A}}^{\text{red}} = [\![E]\!]_{\mathscr{A}}$.*

Proof:

(1) $$[\![E]\!]_{\mathscr{A}}^{\text{red}} \leqq [\![E]\!]_{\mathscr{A}}.$$

Since each reduction rule $E_1 \to E_2$ satisfies $[\![E_1]\!]_{\mathscr{A}} \geqq [\![E_2]\!]_{\mathscr{A}}$, it follows from the monotonicity of $\varphi(F)$ and of the rational operations that $E_1 \vdash E_2$ implies $[\![E_1]\!]_{\mathscr{A}} \geqq [\![E_2]\!]_{\mathscr{A}}$ for all computation terms E_1, E_2. Hence,

$$E \overset{*}{\vdash} a \text{ implies } a \leqq [\![E]\!]_{\mathscr{A}}.$$

(2) $$[\![E]\!]_{\mathscr{A}} \leqq [\![E]\!]_{\mathscr{A}}^{\text{red}}$$

This will be shown by induction on the structure of E.

(*a*) $$\mathrm{E} = a \in A,$$
$[\![a]\!]_{\mathscr{A}} = a$ and $[\![a]\!]_{\mathscr{A}}^{\text{red}} = \bigsqcup \underline{\text{val}}\,(a) = \bigsqcup \{a\} = a$

(*b*) $E=F(E_1 \ldots E_r)$ and $[\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]_{\mathscr{A}}^{\text{red}}$ for $1 \leqq i \leqq r$:

$$[\![E]\!]_{\mathscr{A}} = \varphi\,(F)\,([\![E_1]\!]_{\mathscr{A}} \ldots [\![E_r]\!]_{\mathscr{A}})$$
$$\leqq \varphi\,(F)\,(\bigsqcup \underline{\text{val}}\,(E_1) \ldots \bigsqcup \underline{\text{val}}\,(E_r))$$

by induction hypothesis

$$= \bigsqcup \{\varphi(F)(a_1 \ldots a_r) \mid a_i \in \underline{\text{val}}(E_i)\}$$

by continuity of $\varphi(F)$

$$= \bigsqcup \underline{val}(F(E_1 \ldots E_r))$$

$$= [\![E]\!]^{\text{red}}_{\mathscr{A}}.$$

(*c*) $E = S(E_1 \ldots E_r)$ and $[\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]^{\text{red}}_{\mathscr{A}}$ for $1 \leqq i \leqq r$.

By induction on the structure of S we prove $(*)$:

$$\left\{ \begin{array}{ll} [\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]^{\text{red}}_{\mathscr{A}} & \text{for} \quad 1 \leqq i \leqq r \Rightarrow \\ [\![S(E_1 \ldots E_r)]\!]_{\mathscr{A}} \leqq [\![S(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}}. & \end{array} \right\} \qquad (*)$$

(*c* 1) $S \in \{F'(e), \mathbb{P}_j(e), \mathbb{C}(e), \mathbb{R}_j(e) \mid F \in \Omega, j \geqq 1\}$

These four cases are easily checked:

$[\![F'(e)(E_1 \ldots E_r)]\!]_{\mathscr{A}} = \varphi(F)(e)$,

$[\![F'(e)(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}} = \bigsqcup \underline{\text{val}}(F'(e)(E_1 \ldots E_r)) = \bigsqcup \{\varphi(F)(e)\}$,

$[\![\mathbb{P}_j(e)(E_1 \ldots E_r)]\!]_{\mathscr{A}} = \underline{\text{if}}\ 1 \leqq j \leqq r\ \underline{\text{then}}\ [\![E_j]\!]_{\mathscr{A}}\ \underline{\text{else}}\ \bot_A$,

$[\![\mathbb{P}_j(e)(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}} = \underline{\text{if}}\ 1 \leqq j \leqq r\ \underline{\text{then}} \bigsqcup \underline{\text{val}}(E_j)\ \underline{\text{else}}\ \bot_A$,

and we get $(*)$ by induction.

$[\![\mathbb{C}(e)(E_1 \ldots E_r)]\!]_{\mathscr{A}} = \bot_A = \bigsqcup \underline{\text{val}}(\mathbb{C}(e)(E_1 \ldots E_r))$

The same conclusion holds for $S = \mathbb{R}_j(e)$.

(*c* 2) $S = F'(S^1 \ldots S_s)$ and $(*)$ holds for all S_i.

Now, let $[\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]^{\text{red}}_{\mathscr{A}}$ for $1 \leqq i \leqq r$.

$[\![F'(S_1 \ldots S_s)(E_1 \ldots E_r)]\!]_{\mathscr{A}} = \varphi(F)([\![S_1(E_1 \ldots E_r)]\!]_{\mathscr{A}} \ldots [\![S_s(E_1 \ldots E_r)]\!]_{\mathscr{A}})$,

by induction $\leqq \varphi(F)([\![S_1(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}} \ldots [\![S_s(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}})$,

as in case $(b) = [\![F(S_1(E_1 \ldots E_r) \ldots S_s(E_1 \ldots E_r))]\!]^{\text{red}}_{\mathscr{A}}$,

by standardization $= [\![F'(S_1 \ldots S_s)(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}}$.

(*c* 3) $S = \mathbb{P}_j(S_1 \ldots S_s)$ has the same proof as $S = \mathbb{P}_j(e)$.

(*c* 4) $S = \mathbb{C}(S_1 \ldots S_s)$, $s \geqq 1$, and $(*)$ holds for all S_i.

Again, let $[\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]^{\text{red}}_{\mathscr{A}}$ for $1 \leqq i \leqq r$:

$$[\![\mathbb{C}(S_1 \ldots S_s)(E_1 \ldots E_r)]\!]_{\mathscr{A}} = [\![S_1(S_2(E_1 \ldots E_r) \ldots S_s(E_1 \ldots E_r))]\!]_{\mathscr{A}}.$$

By induction we get:

$$[\![S_i(E_1 \ldots E_r)]\!]_{\mathscr{A}} \leqq [\![S_i(E_1 \ldots E_r)]\!]^{\text{red}}_{\mathscr{A}} \qquad \text{where} \quad 2 \leqq i \leqq s,$$

and anew by induction:

$$[\![S_1 (S_2 (E_1 \ldots E_r) \ldots S_s (E_1 \ldots E_r))]\!]_{\mathscr{A}} \leqq [\![S_1 (S_2 (E_1 \ldots E_r) \ldots S_s (E_1 \ldots E_r))]\!]_{\mathscr{A}}^{\text{red}}$$

by standardization $= [\![\mathbb{C} (S_1 \ldots S_s) (E_1 \ldots E_r)]\!]_{\mathscr{A}}^{\text{red}}$

(c 5) $S = \mathbb{R}_j (S_1 \ldots S_s)$, $s \geqq 1$, and ($*$) holds for all S_i.

If $j > s$, the proof goes as in the case $s = 0$.

So, let $1 \leqq j \leqq s$.

Moreover, let $[\![E_i]\!]_{\mathscr{A}} \leqq [\![E_i]\!]_{\mathscr{A}}^{\text{red}}$ for $1 \leqq i \leqq r$.

$$\begin{aligned}[\![\mathbb{R}_j (S_1 \ldots S_s) (E_1 \ldots E_r)]\!]_{\mathscr{A}} &= [\![\mathbb{R}_j (S_1 \ldots S_s)]\!]_{\mathscr{A}} ([\![E_1]\!]_{\mathscr{A}} \ldots [\![E_r]\!]_{\mathscr{A}}) \\ &= \underline{\text{proj}}_j (\underline{\text{fix}} (f))\end{aligned}$$

where:

$$\begin{aligned} f:\ & A^s \to A^s \\ & b \mapsto [\![S_1]\!]_{\mathscr{A}} (b\, [\![E_1]\!]_{\mathscr{A}} \ldots [\![E_r]\!]_{\mathscr{A}}) \ldots [\![S_s]\!]_{\mathscr{A}} (b\, [\![E_1]\!]_{\mathscr{A}} \ldots [\![E_r]\!]_{\mathscr{A}}),\end{aligned}$$

Here, we have reached the essential point of this equivalence proof because Tarski's fixed-point theorem allows an operational characterization of $\underline{\text{fix}}(f)$:

$$\underline{\text{fix}} (f) = \bigsqcup \{ f^n (\bot_A^s) \mid n \in \mathbb{N} \}.$$

In order to describe the approximations $f^n(\bot_A^s)$ we consider the Kleene-sequences of computation terms for $S_1, \ldots, S_s$ and $E_1, \ldots, E_r$. For each $i \in \{1, \ldots, s\}$ we define $K_i^0, K_i^1, \ldots, \in \mathscr{C}$ by:

$$\begin{aligned} K_i^0 &:= \bot_A, \\ K_i^{n+1} &:= S_i (K_1^n \ldots K_s^n\, E_1 \ldots E_r). \end{aligned}$$

It is now easy to check that:

$$f^n (\bot_A^s) = [\![K_1^n]\!]_{\mathscr{A}} \ldots [\![K_s^n]\!]_{\mathscr{A}} \quad \text{for all } n \in \mathbb{N}.$$

By successive applications of the induction assumption if follows that:

$$[\![K_i^n]\!]_{\mathscr{A}} \leqq [\![K_i^n]\!]_{\mathscr{A}}^{\text{red}} \quad \text{for all } n \in \mathbb{N} \quad \text{and} \quad 1 \leqq i \leqq s$$

and therefore:

$$\begin{aligned}
&[\![\mathbb{R}_j(S_1\dots S_s)(E_1\dots E_r)]\!]_{\mathscr{A}}\\
&\qquad=\underline{\mathrm{proj}}_j(\bigsqcup\{f^n(\perp_A^s)\mid n\in\mathbb{N}\})\\
&\qquad=\bigsqcup\{[\![K_j^n]\!]_{\mathscr{A}}\mid n\in\mathbb{N}\}\\
&\qquad\leqq\bigsqcup\{[\![K_j^n]\!]_{\mathscr{A}}^{\mathrm{red}}\mid n\in\mathbb{N}\}\\
&\qquad=\bigsqcup\{\bigsqcup\underline{\mathrm{val}}(K_j^n)\mid n\in\mathbb{N}\}\\
&\qquad\leqq\bigsqcup\underline{\mathrm{val}}(\mathbb{R}_j(S_1\dots S_s)(E_1\dots E_r)).
\end{aligned}$$

The last inequality follows from the fact that:

$$\mathbb{R}_j(S_1\dots S_s)(E_1\dots E_r)\overset{*}{\vdash}K_j^n\quad\text{for all } n\in\mathbb{N}.$$

Now, since $(*)$ implies (c), this completes the proof of the equivalence theorem.

As an immediate consequence this result also holds for rational schemes because:

$$[\![S]\!]_{\mathscr{A}}=[\![S]\!]_{\mathscr{A}}^{\mathrm{red}}\Leftrightarrow\forall a\in A^*:\ [\![S]\!]_{\mathscr{A}}(a)=[\![S]\!]_{\mathscr{A}}^{\mathrm{red}}(a)$$

and

$$[\![S]\!]_{\mathscr{A}}(a)=[\![S(a)]\!]_{\mathscr{A}}.$$

ACKNOWLEDGEMENT

I am grateful to Elfriede Fehr who carefully checked the proofs. She also pointed out to me that W. Tait and P. Martin-Löf had successfully applied the algebraic treatment of reductions in proving the Church-Rosser theorem of the λ-calculus.

BIBLIOGRAPHIE

1. H. P. BARENDREGT, *The Lambda-Calculus*, Its Syntax and Semantics, North-Holland P.C. 1981.
2. H. BEKIĆ, *Definable Operations in General Algebras, and the Theory of Automata and Flowcharts*, Research Report, I.B.M. Laboratory, Vienna, 1969.
3. W. DAMM, E. FEHR, and K. INDERMARK, *Higher type recursion and self-application as control structures*, in: E. NEUHOLD, Ed., Formal Description of Programming Concepts, North-Holland, Amsterdam, 1978, pp. 461-487.
4. W. DAMM, *The IO- and OI-hierarchies*, Theoret. Comput. Sc., 20, 1982, 2, pp. 95-208.

5. I. GUESSARIAN, *Algebraic Semantics*, Lecture Notes in Computer Science, 99, 1981.

6. K. INDERMARK, *Schemes with recursion on higher types*, Proc. M.F.C.S.-76, Lecture Notes in Computer Science, 45, 1976, pp. 352-358.

7. K. INDERMARK, *Análisis algebráico de estructuras de control*, Proc. Seminario sobre relaciones entre la lógica matemática y la informática teórica, Universidad Complutense de Madrid, 1981.

8. K. INDERMARK, *On rational definitions in complete algebras without rank*, Theoret. Comput. Sc., 21, 1982, pp. 281-313.

9. K. INDERMARK, *Complexity of infinite trees*, I.C.A.L.P. 83, Barcelona, 1983.

10. M. NIVAT, *Languages algébriques sur le magma libre et sémantique des schémas de programme*, in: Automata, Languages, and Programming, M. NIVAT, Ed., North-Holland P.C., 1972, pp. 293-308.

11. M. NIVAT, *On the Interpretation of Recursive Polyadic Program Schemes*, Symposia Matematica, 15, Rome, 1975, pp. 255-281.

12. E. G. WAGNER, *An Algebraic Theory of Recursive Definitions and Recursive Languages*, Proc. 3rd A.C.M. Symp. Theory of Computing, 1971, pp. 12-23.

13. E. G. WAGNER, *Languages for Defining Sets in Arbitrary Algebras*, Proc. I.E.E.E. Conf. SWAT, 12, 1971, pp. 192-201.

14. M. WAND, *A concrete approach to abstract recursive definitions in:* Automata, Languages, and Programming, M. NIVAT, Ed., North-Holland P.C., 1972, pp. 331-344.