Accepted for publication in the *Proceedings of the Third International Workshop on Conditional Term Rewriting Systems, Pont-á-Mousson, France, 8 - 10 July 1992,* Lecture Notes in Computer Science, **656**, Springer-Verlag, 1993.

Citations of this work should refer to the Proceedings, not to this preprint.

Contents

Consistency and Semantics of Equational Definitions over Predefined Algebras

Valentin Antimirov[*]

Computer Science Department, Aarhus University, Aarhus DK-8000, Denmark
email : anti@daimi.aau.dk

Anatoli Degtyarev

Department of Cybernectics, Kiev University, 252127, Kiev, Ukraine
email : caphedra%105.icyb.kiev.ua

January 1993

Abstract

We introduce and study the notion of an equational definition over a predefined algebra (EDPA) which is a modification of the notion of an algebraic specification enrichment. We argue that the latter is not quite appropriate when dealing with partial functions (in particular, with those defined by non-terminating functional programs), and suggest EDPA as a more adequate tool for specification and verification purposes. Several results concerning consistency of enrichments and correctness of EDPA are presented. The relations between EDPA and some other approaches to algebraic specification of partial functions are discussed.

1 Introduction

1.1 Motivation

Algebraic specification and term-rewriting methods seem very convenient to use in the following wide-spread situation: given a set A of data with several

[*] On leave from the V. M. Glushkov Institute of Cybernetics, Kiev, Ukraine;

predefined functions $g_1, \ldots, g_k$ on it, one needs to define (sometimes constructively) a set of new functions on A (a "specification" or "programming" stage) and to analyse their logical properties (a "verification" stage).

A standard algebraic specification approach to this task would consist of two steps:

i) to consider the set A with $g_1, \ldots, g_k$ as an algebra $\mathcal{A}$, and to specify it as an abstract data type - an *initial model* of some basic specification $SP_{\mathcal{A}}$;

ii) to construct an *enrichment* $SP' = SP_{\mathcal{A}} + (F, R)$ of the basic specification where a set of new axioms R (together with those of $SP_{\mathcal{A}}$) are supposed to define the meaning of the new function symbols $f \in F$.

This construction was introduced long ago (cf., e.g. [GTW78]) and is known to work quite well when all the functions to be defined in this way are total. It is possible in this case to find a so-called *conservative* (i.e. *sufficiently complete* and *consistent*) enrichment SP' which has "practically the same" initial model as the predefined algebra $\mathcal{A}$, and therefore the standard interpretation of F, defined by the initial model of SP', unambiguously defines also a corresponding interpretation $f^{\mathcal{A}}$ of each $f \in F$ on $\mathcal{A}$. This allows to make use of equational logic (with induction), as well as of various automated deduction procedures based on term-rewriting technique for proving logical properties of functions $f^{\mathcal{A}}$.

In this paper we intend to generalize this approach to the case of partial functions over a predefined algebra - the case which is known to be of big importance for practical applications of algebraic specifications. To illustrate that the task is not trivial and to point out some subtle problems one should expect on this way, let us consider an example.

1.2 Instructive Example: Is Induction Safe for Functional Program Verification?

Consider the following functional program for integer division of natural numbers:

$$\begin{array}{rcl} & & \mathtt{fun}\quad \mathtt{div} : \mathtt{Nat}, \mathtt{Nat}- > \mathtt{Nat}; \\ \mathtt{div}(\mathtt{x}, \mathtt{y}) & = & \mathtt{if}\ \mathtt{x} < \mathtt{y}\ \mathtt{then}\ 0\ \mathtt{else}\ 1 + \mathtt{div}(\mathtt{x} - \mathtt{y}, \mathtt{y}), \end{array} \qquad (1)$$

(here "$-$" denotes a natural minus, i.e. $m - n = 0$ for all $m < n$).

The program is not always terminating, still it can be used safely in a context where it is supposed to be called with positive second argument. Let's try to define *algebraic* (not denotation!) semantics of (1) within the "initial algebra approach" sketched above. For this purpose first we need to specify the predefined algebra (of natural numbers) with all operations involved in the program; let's take the following specification: [1]

```
spec NAT is
sorts   Bool Nat
   ops  true  fa1se :  -> Bool .
   ops  0 1 : -> Nat .
   op   suc_  : Nat  ->  Nat .
   ops  (_+_),  (_-_)   :    Nat  Nat   ->   Nat .
   op   _<_  :  Nat  Nat  ->  Bool .}
   op   if_then_else_  : Bool  Nat  Nat  ->  Nat .
vars    x,  y  :   Nat
eqs
  [e1]    1 =  suc  0  .
  [e2]    x + 0 = x .
  [e3]    x + suc y = suc(x + y) .
  [e4]    x < 0 = false .
  [e5]    0 < suc x = true .
  [e6]    (suc x) < (suc y) = x < y .
  [e7]    x - 0 = x .
  [e8]    0 - x = 0 .
  [e9]    (suc x) - (suc y) = x - y .
  [e10]   x - x = 0 .
  [e11]   if true  then x else y = x .
  [e12]   if fa1se then x else y = y .
end
```

It is not difficult to check that NAT is indeed a correct specification of natural numbers.

Now the enrichment DIV = NAT $+(\{\texttt{div}\}, (1))$ is expected to define semantics of `div`. Obviously, the enrichment is not sufficiently complete (e.g., $\texttt{div}(1, 0)$ isn't equal to any natural number), still it is consistent. [2]

[1] We use 0BJ-like syntax (cf. [GW88])

[2] The fact which is not so easy to prove! We shall return to it in Sect. 3.

For all pairs of natural numbers $\mathtt{m}, \mathtt{n}$ (represented as canonical NAT-terms $\mathtt{0}, \mathtt{suc0}, \mathtt{suc}(\mathtt{suc}, \mathtt{0}), \ldots$), where $\mathtt{n}$ is not equal to 0, the ground term `div(m,n)` can be reduced in DIV to some (uniquely defined) natural number. This defines `div(m,n)` as a partial function on `Nat` (i.e., on the carrier of this sort in the initial algebra of NAT).

So far so good, but a problem arises if we try to use the enrichment for verification purposes. Suppose we have to prove the following correctness condition for div:

$$\mathtt{0} < \mathtt{y} \quad \mathtt{=>} \quad \mathtt{div}(\mathtt{x} * \mathtt{y},\ \mathtt{y}) = \mathtt{x}. \tag{2}$$

for all $\mathtt{x}, \mathtt{y} : \mathtt{Nat}$, where $\_ * \_$ denotes the multiplication operation which should also be specified. Let's add to NAT the following usual axioms for multiplication:

```
op (_*_) : Nat Nat -> Nat .
eqs [e13] 0 * y = 0 .      [el4]  (suc x)  *  y = x * y + y .
```

(note, the enriched NAT is still correct) and try to prove (2) as a theorem of DIV = NAT $+(\{\mathtt{div}\}, (1))$ using induction on the variable x. The basic case, when $x = 0$, does not cause any difficulty, for

$$\mathtt{div(0\ *\ y,y)\ =\ div(0,y)\ =\ if\ 0\ <y\ then\ 0\ else}\ \ldots = \mathtt{0}$$

whenever the premise of (2) holds. Then, assuming (2) for $\mathtt{x} = \mathtt{x0}$, we need to prove

$$\mathtt{0} < \mathtt{y} \quad \mathtt{=>} \quad \mathtt{div}((\mathtt{suc\ x0}) * \mathtt{y},\ \mathtt{y}) = \mathtt{suc\ x0}.$$

Simplifying the term $\mathtt{div}((\mathtt{sucx0}) * \mathtt{y}, \mathtt{y})$ in DIV we obtain the expression $\mathtt{div}((\mathtt{x0} * \mathtt{y} + \mathtt{y}) - \mathtt{y}, \mathtt{y}) + \mathtt{1}$; this could be simplified further to $\mathtt{div}(\mathtt{x0} * \mathtt{y}, \mathtt{y})$ $+1 = \mathtt{x0} + \mathtt{1} = \mathtt{suc\ x0}$ (that would complete the proof) if we had the following lemma:

$$(\forall\ \ \mathtt{x}, \mathtt{y} : \mathtt{Nat})\ (\mathtt{x} + \mathtt{y}) - \mathtt{y} = \mathtt{x} \tag{3}$$

Well, the lemma can be easily proved in NAT by induction on the variable y, so we are done. Or are we?

Once the lemma (3) has been obtained, one can use it for proving the following "powerful" theorem:

$$(\forall\ \ \mathtt{x}, \mathtt{y} : \mathtt{Nat})\ \mathtt{x} = \mathtt{y} \tag{4}$$

First, using the axioms `el0`, (1) and the lemma (3) one derives

$$0 = \mathtt{div}(1,0) - \mathtt{div}(1,0) = (1 + \mathtt{div}(l,0)) - \mathtt{div}(1,0) = 1 = \mathtt{suc}(0);$$

then, using this with the basic axioms `e4, e5`, one gets `true` = `false` and then proves (4) using `e11, e12`.

Now it is not a problem to prove (2) - as well as *any other* conditional equation in DIV – but who would accept this as a *verification* of the program (1)?

Of course, the point is that the enrichment of DIV by the lemma (3) is *inconsistent*, in spite of the fact that (3) is an inductive theorem of NAT. So one probably should prohibit to use it for proving theorems in DIV. But then we may not use it for proving (2) too; so our first proof was not correct?

Putting off the answer to the end of the paper, let us here just note that the example illustrates one of the main problem we are going to deal with: how to formalise the algebraic semantics of equational definitions of (possibly partial) functions over a predefined algebra within (some natural extension of) the "initial algebra" approach so that one could use safely inductive theorems of the predefined model for verification purposes.

1.3 Overview of the Paper

The example considered above gives rise to some general questions:

i) How to check consistency of (incomplete) enrichments?

ii) How to define an appropriate algebraic semantics of incomplete equational definitions like (1) ?

In our paper [AD92] we have already addressed the first question (in a slightly more general framework of Horn-equational logic), and suggested a model-theoretic technique for proving consistency of enrichments, as well as some sufficient conditions of consistency.

In the present paper we are going to extend and improve those results, as well as to suggest a very general approach to the problem formulated in the second question above. Specifically, after brief overview of basic notions and notations in Sect. 2, we shall introduce the notion of *equational definition over a predefined algebra* (EDPA), and study its semantics in Sect. 3. Using some model-theoretic technique, we shall demonstrate that a standard "free-extension" construction does not give appropriate semantics of EDPA. Our Instructive Example will be used again to illustrate this point:

we shall prove that DIV is indeed a consistent enrichment of NAT, but the equation (3) is not valid in its initial algebra. In Sect. 4 we shall develop an approach that will allow to define a kind of "safe" semantics of EDPA. The approach is based on the idea of "restrictions on substitutions" which has also been exploited in several papers devoted to algebraic specifications with partial functions [GDLE84, SNGM89] and term-rewriting systems over built-in algebras [AB92]; in Sect. 5 we consider relations of our results with these papers.

2 Basic Notions and Notations

In this section we briefly recall some standard notions and notations of algebraic specification and term-rewriting theory [EM85], [Wir90], [DJ90].

Given a set of sorts S, a *many-sorted*, or *S-sorted signature* Σ is a disjoint union of sets $\Sigma_{w,s}$ of function symbols (or f-symbols, for short) of type $w \to s$ where $w \in S^*, s \in S$; constants are nullary functions of type $\to s$.

The notions of Σ-algebra A, Σ-subalgebra $B \subset A$, Σ-congruence on A, Σ-homomorphisms are supposed to be defined as usual [EM85, Wir90]; we shall use S-indexed notations for denoting carriers A_s of sort s and Cartesian products $A_w = A_{s_1} \times \ldots \times A_{s_n}$ (where $w = s_1 \ldots s_n$).

Given an S-sorted set of variables $X = \cup_{s \in S} X_s$ (a disjoint union), $T_\Sigma(X)$ denotes the *absolutely free Σ-algebra over X* whose elements are Σ-terms; then T_Σ denotes the set (and the algebra) of *ground* Σ-terms (the latter is an initial object in the category $\mathbf{Alg}_\Sigma$ of all Σ-algebras). To avoid the "empty sorts" problem, we consider in this paper only those signatures Σ for which $(T_\Sigma)_s$, is non-empty for each $s \in S$.

An *(algebraic) specification SP* over a signature Σ is a pair (Σ, E) where E is a set of *axioms* of SP which are universal quasi-equations (often called conditional equations). Sometimes we shall restrict E to be a set of atomic equations; then SP will be called as *purely equational* specification.

Given a specification $SP = (\Sigma, E)$, $\mathbf{Alg}(SP)$ denotes the category of all *SP-algebras* (also called *models of SP*), i.e. Σ-algebras satisfying all the axioms in E. This category has the initial object $I(SP)$; it can be represented (up to isomorphism) as the quotient $T_{SP} = T_\Sigma/_{\equiv_E}$ of the ground-term algebra T_Σ by the least Σ-congruence $\equiv_E$ generated by E. The uniquely defined homomorphism from $I(SP)$ to a given SP-algebra is called *initial.*

Given two S-sorted signatures Σ, Σ', a specification $SP' = (\Sigma', E')$ is called an *enrichment* of $SP = (\Sigma, E)$ if $\Sigma' \subseteq \Sigma$ (we also say that Σ' is an *enrichment* of Σ) and $E \subseteq E'$; SP' can be presented in the form $SP+(F,R)$ or $(\Sigma+F, E+R)$ where $F = \Sigma' \setminus \Sigma$ is the set of *new* f-symbols, and $R = E' \setminus E$ is a set of *new* axioms (then f-symbols and axioms of SP will be referred to as "old"). This enrichment is called

- *consistent* (wrt. SP) if it satisfies the "no-confusion" condition, i.e. if the restriction of the congruence $\equiv_{E+R}$ to the set T_Σ coincides with $\equiv_E$;
- *complete* (wrt. SP) if it satisfies the "no-junk" condition, i.e., if each E'-equivalence class $[t']_{E'} \in T_{SP'}$ contains some ground Σ-term;
- *conservative* (wrt. SP) if it is both consistent and complete (wrt. SP).

We say that $SP + (F,R)$ is a *functional* enrichment (or f-enrichment, for short), if for some F-indexed family of $(\Sigma + F)$-terms r_f the set R consists of (oriented) equations of the form $f(\mathbf{x}) = r_f$ where $\mathbf{x}$ is a list of distinct variables including all those occurring in r_f; then the pair (F,R) (and R itself) will be called a *functional definition*.

A *forgetful functor* from $\mathbf{Alg}(\Sigma+F)$ to $\mathbf{Alg}(\Sigma)$ maps each $(\Sigma+F)$-algebra A' to its Σ*-reduct* $A = A' \mid_\Sigma$ which gets its carriers A_s, and interpretations of function symbols f^A (for all $f \in \Sigma$) from A'; then A' is called an *enrichment of A by $F^{A'}$*. The forgetful functor is known to map any SP'-algebra to some SP-algebra for any enrichment $SP' = SP + (F,R)$; moreover, it has a *left adjoint* functor (also called a *free functor*) which maps each SP-algebra A to its *free SP'-enrichment*.

A Σ-reduct $I(SP') \mid_\Sigma$ of the initial algebra $I(SP')$ will be also denoted as $I_\Sigma(SP')$.

An algebraic specification $SP = (\Sigma, E)$ can be considered as a *termrewriting system* (t.r.s. for short) [DJ90] through orienting of equations in E from left to right. This t.r.s. defines the *rewrite relation* $\rightarrow_E$ on Σ-terms; its symmetric (reflexive, transitive, reflexive transitive, symmetric reflexive transitive) closure is denoted by $\leftrightarrow_E$ (correspondingly by $\rightarrow_E^=, \rightarrow_E^+, \rightarrow_E^*, \leftrightarrow_E^*$; the latter is known to coincide with $\equiv_E$).

The t.r.s. is called *normalising* on some set of terms T if each term $t \in T$ has at least one *normal form*; it is called *confluent* if the composition $\leftarrow_E^* \circ \rightarrow_E^*$ is included into $\rightarrow_E^* \circ \leftarrow_E^*$ (where $\leftarrow_E^*$ denotes the converse to $\rightarrow_E^*$).

We shall also use standard notations for a subterm $t|_\pi$ and a context $t[\ ]_\pi$ of a term t where π is some *position* in t (cf. [DJ90]); $\mathcal{V}(\varepsilon)$ denotes the set of variables occurring in a syntactic object (a term, a formula, a set of those, etc.) ε.

3 Equational Definitions over Predefined Algebras as Enrichments

Given an S_0-sorted signature Σ_0, a Σ_0-algebra $\mathcal{A}$ and a set of f-symbols F such that $\Sigma_0 + F$ is an enrichment of Σ_0 a triple $(\mathcal{A}, F, R)$ (denoted also $(F, R)_{\mathcal{A}}$) where R is a set of oriented $\Sigma_0 + F$-equations, is called an *equational definition over* $\mathcal{A}$ if the main f-symbol of the left-hand side of each $e \in R$ belong to F.

This gives only syntax; to define semantics of EDPA means to set a correspondence between triples $(\mathcal{A}, F, R)$ and sets of partial functions

$$F^{\mathcal{A}} = \{f^{\mathcal{A}} : \mathcal{A}_w \overset{\sim}{\rightarrow} \mathcal{A}_s \mid f \in F_{w,s}\}$$

Suppose $SP = (\Sigma, E)$ is an algebraic specification of $\mathcal{A}$ in the sense that Σ is a finite extension of Σ_0 and the Σ_0-reduct of $I(SP)$ is isomorphic to $\mathcal{A}$. Then we say that the enrichment $SP + (F, R)$ is an *algebraic presentation* of EDPA $(F, R)_{\mathcal{A}}$.

We are going to define semantics of EDPA using their algebraic presentations. For the sake of simplicity, we shall identify a predefined algebra $\mathcal{A}$ with the initial algebra of its basic specification SP (forgetting about the possible difference of their signatures).

3.1 Algebraic Semantics of Enrichments

Consider initial algebras $I(SP) \cong T_{\Sigma/\equiv_E}$ and $I(SP') \cong T_{\Sigma+F/\equiv_{E+R}}$ of a given specification $SP = (\Sigma, E)$ and its enrichment $SP' = SP + (F, R)$. They are known to relate in the following way: there is a (unique) homomorphism h from $I(SP)$ to $I_\Sigma(SP')$ which maps an equivalence class $[t]_E \in T_{\Sigma/\equiv_E}$ to a corresponding equivalence class $[t]_{E+R} \in T_{\Sigma+F/\equiv_{E+R}}$ for each Σ-term t, i.e., $h([t]_E) = [t]_{E+R}$. This homomorphism is known to be injective (surjective) iff the enrichment SP' is consistent (complete) wrt. SP.

The interpretation $f^{I(SP')}$ of a new f-symbol $f \in F$ on $I(SP')$ satisfies the following equation:

$$f^{I(SP')}([t'_1]_{E+R}, \dots [t'_n]_{E+R}) = [f(t'_1, \dots t_n)]_{E+R} \tag{5}$$

for all tuples of $(\Sigma + F)$-terms t'_i, of appropriate sorts; the same equation defines the interpretation of $f$on $I_\Sigma(SP')$ that gives a free SP'-enrichment of $I(SP)$

Now we need to define some *basic interpretation* $F^{\mathcal{A}}$ of F on $\mathcal{A}$, i.e. on $I(SP)$, such that the corresponding enrichment $\mathcal{A}' = \mathcal{A} + F^{\mathcal{A}}$ would be in "good relations" with the set of axioms $E + R$ (since we are going to use them for reasoning about $F^{\mathcal{A}}$).

Whenever SP' is a conservative enrichment, the basic interpretation is uniquely defined by (5) and gives a set of *total* functions $F^{\mathcal{A}}$. In order to capture the case of partially defined functions over $\mathcal{A}$, we should, at least, drop the "no-junk" condition and consider incomplete enrichments.

Still it seems quite reasonable to impose the "no-confusion" requirement on algebraic presentations, for in this case they would "preserve" the structure of $\mathcal{A}$ in the initial algebra $I(SP')$: the homomorphism $h : I(SP) \rightarrow I_\Sigma(SP')$ would be injective and its image would be a Σ-subalgebra of $I_\Sigma(SP')$ isomorphic to $\mathcal{A}$. The following proposition shows how to obtain from this a *partial* $(\Sigma + F)$-subalgebra of $I(SP')$.

Proposition 1. *Given a consistent enrichment $SP' = SP + (F, R)$ of $SP = (\Sigma, E)$, there exists a set of partial functions*

$$F^{I(SP)} = \{f^{I(SP)} : I(SP)_w \tilde{\rightarrow} I(SP)_s \mid f \in F_{w,s}\}$$

defined as follows:

$$f^{I(SP)}([t_1]_E, \ldots [t_m]_E) = [f(t_1, \ldots t_m)]_{E+R} \cap T_\Sigma \qquad (6)$$

for all tuples $t_1, \ldots t_m$ of Σ-terms of appropriate sorts provided the right-hand side is not the empty set, otherwise $f^{I(SP)}([t_1]_E, \ldots [t_m]_E)$ is undefined.

Moreover, the enrichment of $I(SP)$ with $F^{I(SP)}$ will be a partial $(\Sigma + F)$-subalgebra of $I(SP')$.

Proof. The correctness of (6), as well as the statement in whole follow from the consistency condition: the initial homomorphism h is injective in this case, so $I(SP)$ is (isomorphic to) a Σ-subalgebra of $I_\Sigma(SP')$, and the equation (6) just defines the restriction of $f^{I(SP)}$ to $I(SP)$ considered as a subset of $I(SP')$. □

As a matter of fact, this construction is quite similar to that in [Kre87] which was intended to provide an approach to formalise partial functions within "the simpler framework of total algebras and conventional specifications". It does seem natural to take (6) as the definition of semantics of $(F, R)_{\mathcal{A}}$. To explain why this would not be quite satisfactory, we need first to address the problem of how to check the conditions when this definition can be used, i.e. how to prove consistency of (possibly incomplete) enrichments.

3.2 Consistency of Enrichments

A general model-theoretic method for proving consistency of enrichments, which does not impose any requirements on specifications involved, was introduced in [GTW78] (cf. also [EM85]). It is based on the following sufficient condition.

Fact 1. (a sufficient condition of consistency of enrichments)
*An enrichment $SP' = SP + (F, R)$ is consistent wrt. an algebraic specification $SP = (\Sigma, E)$ if there exists an algebra $A \in$ **Alg**(SP') such that its Σ-reduct $A|_\Sigma$ is isomorphic to I(SP).*

Thus, to prove consistency of $SP' = SP + (F, R)$ it suffices to find some interpretation of new function symbols $f \in F$ on $I(SP)$ satisfying (together with the known interpretation of old symbols from SP) all axioms in R.

This technique can always be applied to complete enrichments, for in this case the condition gets necessary. Occasionally, it can also be applied to some incomplete enrichments, but not to all of them – e.g., this does not work for the enrichment DIV considered in Sect. 1.2.

To overcome this disadvantage, we have obtained the following criterion of consistency[3]:

Theorem 2. *Given an algebraic specification $SP = (\Sigma, E)$, an enrichment $SP' = SP + (F, R)$ is consistent iff there exists an algebra $A \in$ **Alg**(SP') such that its Σ-reduct $A|_\Sigma$ contains a subalgebra isomorphic to I(SP).*

Proof. If the enrichment is consistent, then the initial homomorphism $h : I(SP) \rightarrow I_\Sigma(SP')$ is an injection and its image gives a subalgebra isomorphic to $I(SP)$. So we can take $A = I(SP')$ in this case.

[3] Peter Padowitz pointed us out recently that he had obtained a similar criterion. However, his formulation and proof (cf. [Pad90, page 35, Corol. 3.15]) are based on so-called *canonical term structures* and, in our opinion, are more complicated than ours.

For the converse, let the enrichment be inconsistent (so that h isn't injective). Then for any $A \in \mathbf{Alg}(SP')$ the initial homomorphism $k : I(SP) \to A \mid_\Sigma$ is not injective since it can be (uniquely) factored into the composition $h' \circ h$ where h' is the Σ-reduct of the initial homomorphism from $I(SP')$ to A. Thus $A \mid_\Sigma$ doesn't contain a subalgebra isomorphic to $I(SP)$ (since k is the only homomorphism from $I(SP)$ to A). □

The corresponding technique for proving consistency of an enrichment $SP' = SP + (F, R)$ consists of the following steps:

1.) To construct an extension $T^C_{SP} \in \mathbf{Alg}(SP)$ of the initial algebra $T_{SP} \simeq I(SP)$ by a set C of new "non-standard" elements (i.e., to extend the interpretation of all basic operations $g \in \Sigma$ to the carrier $T_{SP} \cup C$);

2.) To construct some interpretation of new function symbols F on $T_{SP} \cup C$ such that the enrichment of T^C_{SP} with this interpretation would satisfy all the axioms in R.

Theorem 2 guarantees that these steps can always be fulfilled whenever the enrichment $SP + (F, R)$ is consistent.

Let us apply this technique to confirm consistency of the enrichment DIV from Sect. 1.2.

3.3 Proving Consistency of the Instructive Example

The initial algebra of the basic specification NAT from Sect. 1.2 is isomorphic to a two-sorted algebra with carriers $\mathbf{N}$ of `Nat` (the set of natural numbers), $\mathbf{B} = \{true, false\}$ of `Bool` and usual interpretation of all the operations. To prove consistency of DIV=NAT+($\{$`div`$\}, (1)$), let's construct the following extension A of I(NAT) by one "non-standard" natural number $\mathbf{c}$: `Nat`, i.e. $A_{\texttt{Bool}} = \mathbf{B}$, $A_{\texttt{Nat}} = \mathbf{N} \cup \{\mathbf{c}\}$. The extensions of all operations to $A_{\texttt{Nat}}$ are defined by the following equations:

$$
\begin{array}{l}
\texttt{suc}^A(\mathbf{c}) = \mathbf{c};\\
+^A(n,\mathbf{c}) = +^A(\mathbf{c}, n) = +^A(\mathbf{c},\mathbf{c}) = \mathbf{c};\\
{}^A(n',\mathbf{c}) \ \ = *^A(\mathbf{c}, n') = *^A(\mathbf{c},\mathbf{c}) = \mathbf{c}; *^A(0,\mathbf{c}) = *^A(\mathbf{c}, 0) = 0;\\
-^A(n,\mathbf{c}) = -^A(\mathbf{c},\mathbf{c}) = 0; -^A(\mathbf{c}, n) = \mathbf{c}\\
<^A\ (n,\mathbf{c}){=}\ true; <^A\ (\mathbf{c}, n) = <^A\ (\mathbf{c},\mathbf{c}) = false;\\
if^A(true,\mathbf{c}, n) \ \ = \mathbf{c}; \quad if^A(true, n,\mathbf{c}) = n;\\
if^A(false,\mathbf{c}, n) \ = n; \quad if^A(false, n,\mathbf{c}) {=}\mathbf{c}
\end{array}
$$

for all $n, n' \in \mathbf{N}$, $n' > 0$. It is easy to check by direct calculations that the extensions of operations satisfy all the axioms of NAT, i.e. $A \in \mathbf{Alg}(\mathrm{NAT})$. In order to check the fact that $I(\mathrm{NAT}) \subset A$, one can observe that the enrichment of NAT by a constant **c** : `Nat` and these equations forms a *terminating* rewrite system - this just makes it possible to prove its consistency wrt. NAT by methods suggested in [JK89, Kir92].

To complete the proof, we suggest the following interpretation of `div` on A that satisfies (1):

$$\mathtt{div}^A(n,0) = \mathtt{div}^A(\mathbf{c},n) = \mathbf{c};$$
$$\mathtt{div}^A(n,\mathbf{c}) = 0; \mathtt{div}^A(\mathbf{c},\mathbf{c}) = 1;\ \mathtt{div}^A(n,n') = k$$

for all $n, n' \in \mathbf{N}$, $n' > 0$ where $k \in \mathbf{N}$ is the quotient of integer division n on n'.

Note that (3) is not valid in A (consider $x = y = c$), therefore it is also not valid in $I(\mathrm{DIV})$, since A is its surjective image. Thus we have proved that a (rather ordinary) functional program can be consistent wrt. some basic specification of a predefined model and inconsistent wrt. some of its inductive consequences. That is why one couldn't use induction (over predefined model) for verification of functional programs (considered as EDPA) if (6) was taken as the definition of semantics of EDPA. Let's consider an approach to overcome this problem.

4 "Safe" Semantics of EDPA

One can guess that the basic reason of inconsistency of some f-enrichment SP' is the opportunity to substitute terms with new function symbols into old axioms: eventually such a term can denote a "junk" (a non-reachable value of $I_\Sigma(SP')$) that extends the range of interpretation of variables in the axioms and in inductive theorems of the basic specification $SP = (\Sigma, E)$.

A natural idea, then, is to prohibit those substitutions. However, the restriction would be too strong, because in this case all old operations would get *strict* with respect to new terms: for instance, one couldn't simplify the term $0 + f(1)$ to $f(1)$ using an old axiom $0 + x = x$. The situation with conditionals (if-then-else) would be even worse: e.g., the equation (1) would define `div` as an empty function provided `if-then-else` was strict.

In [AD92] we have already shown how to solve the problem with conditionals. Here we are going to suggest a more general solution that will allow most of old functions (axiomatized in some "safe" way) to be non-strict. The benefit of this approach is that it provides a wider class of possible operational semantics of predefined operations in EDPA (not only call-by-value).

4.1 Restricted Rewriting and Equality

In this section we consider purely equational specifications SP (of a predefined Σ-algebra $\mathcal{A}$) equipped with the following additional information: the set X of all variables (used in axioms of SP) contains a distinguished subset X^+ of *safe* variables (then variables in $X \setminus X^+$ will be called *unsafe*). When we need to reflect this information in the terminology and definitions, we shall be using the notation $\Sigma(X^+)$ for a signature Σ.[4]

A $\Sigma(X^+)$-equation $e \in E$ will be called *safe* if it contains only safe variables (i.e., $\mathcal{V}(e) \subset X^+$). A substitution θ on $T_{\Sigma+F}(X)$ will be called *safe* if it maps safe variables into Σ-terms (i.e., $\theta(X^+) \subset T_\Sigma(X^+)$).

Now we introduce the following relations that will serve for term rewriting and equational derivations with some restrictions on substitutions.

Definition 3. Given a basic specification $SP = (\Sigma, E)$ and its enrichment $SP' = SP + (F, R)$, let $\rightarrow_{E:}$ denote the following relation on $T_{\Sigma+F}(Y)$: $t \rightarrow_{E:} t'$ holds if there exists an equation $l = r \in E$, a safe substitution $\theta : X \rightarrow T_{\Sigma+F}(Y)$ and a position π such that $t|_\pi \doteq \theta(l)$ and $t' \doteq t[\theta(r)]_\pi$. Then a *restricted rewrite relation* $\rightarrow_{E:R}$ and a *restricted equality* $=_{E:R}$ (both on $T_{\Sigma+F}(Y)$) are defined as follows:

$$\rightarrow_{E:R} \quad \rightleftharpoons \quad \rightarrow_{E:} \cup \rightarrow_R; \quad t =_{E:R} t' \quad \rightleftharpoons \quad t \leftrightarrow^*_{E:R} t'$$

It follows from the definition that the restricted equality $=_{E:R}$ is a congruence on $T_{\Sigma+F}(Y)$ (included into that $=_{E+R}$) This enables us to define a quotient $T_{\Sigma+F}/_{=_{E:R}}$ and to use its Σ-reduct (rather than $I_\Sigma(SP')$) in the definition of semantics of EDPA. To provide correctness of this new construction, an enrichment has to satisfy the "no-confusion" condition wrt. $=_{E:R}$. This is the matter of the next definition and proposition.

Definition 4. Given a specification $SP = (\Sigma(X^+), E)$, its enrichment $SP + (F, R)$ is said to be *safe-consistent* (wrt. SP) if $t_1 =_{E:R} t_2$ implies

[4] This construction, as well as the terminology, is inspired by the approach to partial functions suggested in [GDLE84] (cf. also a survey [Mos92]).

$t_1 =_E t_2$ for any pair of ground Σ-terms t_1, t_2.

Proposition 5. *Given a safe-consistent enrichment $SP' = SP + (F, R)$ of $SP = (\Sigma(X^+), E)$, there exists a set of partial functions*

$$F^{I(SP)} = \{f^{I(SP)} : I(SP)_w \stackrel{\sim}{\to} I(SP)_s \mid f \in F_{w,s}\}$$

defined as follows:

$$f^{I(SP)}([t_1]_E, \dots [t_m]_E) = [f(t_1, \dots t_m)]_{E+R} \cap T_\Sigma \qquad (7)$$

for all tuples $t_1 \dots t_m$ of ground Σ-terms of appropriate sorts provided the right-hand side is not the empty set, otherwise $f^{I(SP)}([t_1]_E, \dots [t_m]_E)$ is undefined. Moreover, the enrichment of $I(SP)$ with $F^{I(SP)}$ will be a partial subalgebra of $T_{\Sigma+F}/_{=_{E:R}}$ □

We take (7) as the "generic" definition of semantics of an EDPA $(F, R)_{\mathcal{A}}$ presented by a safe-consistent enrichment $(\Sigma(X^+), E) + (F, R)$; the set X^+ of safe variables is a parameter of this definition. As far as safe-consistency is in general weaker than consistency, this gives an opportunity to get a wider class of correct EDPA which will include *all* functional definitions. The following technical details are just steps toward this goal.

In what follows, we consider a specification $SP = (\Sigma(X^+), E)$ and its enrichment $SP' = SP + (F, R)$, where R is a rewrite system such that all its left-hand sides contain some $f \in F$ (this is a bit more general class of enrichments than algebraic presentations of EDPA).

Definition 6. We say that the t.r.s. R *respects* the set of equations E if the following inclusion holds:

$$\leftrightarrow_{E:} \circ \to_R \subseteq \to_R^{=} \circ \leftrightarrow_{E:}^{*} .$$

Given a set of terms $\mathcal{T} \subset T_{\Sigma+F}(X)$, we say that R *respects E on $\mathcal{T}$* if the same inclusion holds for restrictions of the relations involved on $\mathcal{T}$.

Proposition 7. *If the system R respects the set of equations E on $\mathcal{T}$, then for all $t_1, t_2, t_3 \in \mathcal{T}$ there exists $t_4 \in \mathcal{T}$ such that*

i) *if* $t_1 \leftrightarrow_{E:}^{*} t_2 \to_R t_3$ *then* $t_1 \to_R^{=} t_4 \leftrightarrow_{E:}^{*} t_3$;
ii) *if* $t_1 \leftrightarrow_{E:}^{*} t_2 \to_R^{*} t_3$ *then* $t_1 \to_R^{*} t_4 \leftrightarrow_{E:}^{*} t_3$.

Proof. (sketch).

i) By straightforward induction on the length of the derivation $t_1 \leftrightarrow^*_{E:} t_2$
ii) By straightforward induction on the length of the derivation $t_2 \leftrightarrow^*_R t_3$ using (i).

□

Now we formulate and prove the following fundamental property of the restricted rewriting relation $\rightarrow_R$ and the congruence $=_{E:R}$.

Lemma 8. *Suppose the system R is confluent and respects the set of equations E. Then the following inclusion holds:*

$$=_{E:R} \subseteq \rightarrow^*_R \circ =_{E:} \circ \leftarrow^*_R$$

(i.e., the relation $\rightarrow_R$ is Church-Rosser modulo $=_{E:}$.)

Proof. Let $t =_{E:R} t'$ hold for some $(\Sigma + F)$-terms t, t'. The derivation $t \leftrightarrow^*_{E:R} t'$ can be represented as a chain

$$t \doteq t_0 \sim t_1 \sim \ldots \sim t_n \doteq t', \tag{8}$$

where each occurrence of $\sim$ denotes either $\rightarrow^+_R, \leftarrow^+_R$ or $\leftrightarrow^+_{E:}$, and adjacent occurrences are different. Due to Prop. 7 and the confluence of R, the following transformation rules $\alpha_i, i = 1, 2, 3$ can be applied to the chain:

$$\begin{array}{ll} \alpha_1 : & t_1 \leftarrow^+_R t_2 \rightarrow^+_R t_3 \Rightarrow t_1 \rightarrow^*_R t_4 \leftarrow^*_R t_3; \\ \alpha_2 : & t_1 \leftarrow^+_{E:} t_2 \rightarrow^+_R t_3 \Rightarrow t_1 \rightarrow^*_R t_4 \leftarrow^*_{E:} t_3; \\ \alpha_3 : & t_1 \leftarrow^+_R t_2 \rightarrow^+_{E:} t_3 \Rightarrow t_1 \rightarrow^*_{E:} t_4 \leftarrow^*_R t_3; \end{array}$$

This system of rules is normalising on the set of chains of the form (8), because each application of α_1, α_2 (of α_1, α_3) to the rightmost occurrence of $\rightarrow^+_R$ (the leftmost occurrence of $\leftarrow^+_R$) reduces either the distance from that occurrence to the left (right) end of the chain, or the number of those in the chain. Therefore after a finite number of steps the chain (8) will get the form

$$t \rightarrow^*_R t_1 \leftrightarrow^*_{E:} t_2 \leftarrow^+_R t', \tag{9}$$

for some $(\Sigma + F)$-terms t_1, t_2. □

Corollary 9. *The enrichment $SP+(F,R)$ of $SP=(\Sigma,E)$is safe-consistent if R is confluent on $T_{\Sigma+F}$ and respects E on $T_{\Sigma+F}$.*

Proof. It suffices to observe that each derivation $t_1 \leftrightarrow^*_{E:R} t_2$, where the outermost terms t_1,t_2 belong to T_Σ, after transforming by α_i to (9) will get the form $t_1 \leftrightarrow^*_{E:} t_2$, because none rule from R can be applied to t_1,t_2. □

This lemma (with the corollary) can be used for inventing various sufficient conditions of safe-consistency (i.e., correctness of EDPA). We present the corresponding results in the next subsection.

4.2 Sufficient Conditions of Safe-Consistency

First, we show how to obtain the following result (announced in [AD92]) about correctness of a wide class of functional definitions with non-strict conditionals. To define this class, we suppose that the predefined algebra $\mathcal{A}$ and its specification SP satisfy the following requirements:

1) They contain the sort `Bool` of boolean values with constants `true` and `false` which are interpreted by two distinct values in $\mathcal{A}_{\texttt{Bool}}$. (Other *total* boolean operations may occur in $\mathcal{A}$ and SP too).

2) They contain the conditional functions `if:` $\texttt{Bool}, s, s \to s$ for each sort s with the usual axioms in SP:

$$\texttt{if}(\texttt{true},\texttt{x},\texttt{y})=\texttt{x};\ \texttt{if}(\texttt{false},\texttt{x},\texttt{y})=\texttt{y} \tag{10}$$

Let IF denote the set of equations (10) for all sorts; then the set of axioms of SP will be represented as $IF \cup E$ (where E is a set of other equations).

Theorem 10. *Any functional enrichment $SP+(F,R)$ of an algebraic specification $SP=(\Sigma(X^+), IF\cup E)$ is safe-consistent provided all axioms in E are safe.*

Proof. (sketch) The system R is obviously confluent; thus, due to the lemma, it suffices to check that R respects both E and IF on the set of ground $(\Sigma+F)$-terms.

The first fact is easy to prove by case analysis of possible overlappings of applications of $\rightarrow_R$ at a position π_1, and $\leftarrow_{E:}$ or $\rightarrow_{E:}$ at a position π_2 of some $(\Sigma + F)$-term t in the derivation $t_1 \leftrightarrow_{E:} t \rightarrow_R t_2$ (since each $e \in E$ is safe, the only possible non-trivial case is when $t|_{\pi_2}$ is a subterm of $t|_{\pi_1}$). In each case there exists a term t' such that $t_1 \rightarrow_R t' \leftrightarrow^*_{E:} t_2$.

To prove that R respects IF, we need to add to the above one additional case (since variables in IF are not supposed to be safe): a derivation $t_1 \leftrightarrow_{IF} t \rightarrow_R t_2$ where the arrow $\rightarrow_R$ is applied at a position π_1 of t, and the arrow $\leftarrow_{IF}$ (or $\rightarrow_{IF}$) is applied to a subterm $t|_{\pi_2}$ which contains the $(\Sigma + F)$-subterm $t|_{\pi_1}$. Again, one can show that in this case there exists a term t' such that $t_1 \rightarrow^{=}_R t' \leftrightarrow^*_{IF} t_2$.

Thus the inclusion $\leftrightarrow_{E:\cup IF} \circ \rightarrow_R \subseteq \rightarrow^{=}_R \circ \leftrightarrow^*_{E:\cup IF}$; holds, so the enrichment $SP + (F, R)$ is safe-consistent. □

We also announce here the following theorem, which offers even more general sufficient conditions of safe-consistency. Recall that an equation is called *left-linear* (*right-linear*) if its left-hand (right-hand) side is linear; it is called *linear* if it is both left- and right-linear.

Theorem 11. *Any f-enrichment $SP + (F, R)$ of an algebraic specification $SP = (\Sigma(X^+), E)$ is safe-consistent provided all non-linear axioms in E are safe.* □

This theorem gives the corresponding specialisation of (7) which provides correctness of any functional EDPA wrt. a wide class of basic specifications. Using this, we can suggest the following solution of the puzzle with verification given in Sect. 1.2: to make NAT "safe" for (1), as well as for any functional definition, it suffices to mark the variables in the non-linear axioms `el0, el4` as safe. The same should be done with all non-linear inductive theorems of NAT, in particular – with the lemma (3). This makes it impossible to deduce the contradiction (4), but allows to use the lemma for proving the correctness condition (2).

However, we don't know at the moment the most general (syntactical) conditions that would provide *the widest* class of basic specifications "safe" for an arbitrary functional enrichment; this is one of interesting questions for further research.

5 Relations with Other Approaches

As we have already mentioned, the idea to impose restrictions on substitutions of new terms into old axioms in order to treat partial functions in algebraic specifications properly is not new . For instance, it was used in the fifth chapter of [SNGM89] within the framework of order-sorted equational logic [GM89], as well as in [AB92]. Let's recall the corresponding construction of [SNGM89] called there *stratification.*

Suppose a basic specification $SP = (\Sigma, E)$ with an S-sorted signature Σ is to be enriched by a set F of some (possibly partial) functions. Then one should proceed as follows.

First, Σ should be extended by a set of new sorts and declarations: for each basic sort $s \in S$ its *error supersort* $s^?$ should be introduced (i.e., s is a subsort of $s^?$), and each old function symbol $g \in \Sigma$ of type $s_1, \ldots, s_n \to s$ gets an additional declaration

$$g : s_1^?, \ldots, s_n^? \to s^? \ .$$

Then, if one was going to specify a new (partial) function $f \in F$ of type $s_1, \ldots, s_n \to s$, one actually should introduce the following declaration:

$$f : s_1^?, \ldots, s_n^? \to s^? \tag{11}$$

and a set of corresponding axioms R.

As a consequence, any term of the form $f(t_1, \ldots, t_n)$ will have the sort $s^?$ even in the case when each t_i has the sort s_i. Since none of the axioms of SP contains variables of the error supersorts, it is impossible to substitute terms with new function symbols into them; so the corresponding congruence, specified by $SP + (F, R)$ in this way, is just our "restricted equality" $=_{E:R}$ constructed when *all* basic axioms are safe.[5]

However, as we have pointed out in Sect. 4, this approach is very restrictive since it makes all the old functions strict wrt. new terms. This, for instance, makes it impossible to use non-strict conditionals and functional definitions like the program (1) (e.g., `if-then-else` was modeled by an auxiliary strict function in examples of stratified specifications in [SNGM89]).

[5] to be very precise, we should also add here that all variables in new axioms $e \in R$ should be of "questioned" sorts, for otherwise the congruence will be even weaker then $=_{E:R}$; still the latter would be even "better" for consistency.

Our theorems 10 and 11 show that actually this is not necessary for consistency. To reformulate our results for order-sorted specifications, let's introduce the following construction: given an old axiom $e \in E$, let $e^?$ denote its "sort-lifted" version - the result of substitution $x : s^?$ instead of $x : s$ for each $x \in \mathcal{V}(e)$; let $E^?$ denote the set $\{e^? \mid e \in E\}$.

Proposition 12. *Let an f-enrichment* $SP + (F, R)$ of $SP = (\Sigma, E)$ *be obtained by stratification (where all new function symbols $f \in F$ are declared as in (11)), and let $E_1 \subset E$ be a subset of linear equations. Then the enrichment $SP + (F, R + E_1^?)$ is consistent.* □

In particular, one can get non-strict conditionals by adding $IF^?$ - the set of sort-lifted versions of usual (linear!) axioms IF.

Thus, following [GM87], we can add one more problem (let us call it the "functional enrichment consistency" problem) to the long list of those solved by order-sorted algebra.

Still the stratification construction is not the only possible way to represent our "safe" semantics of EDPA. Another possibility (which seems simpler and more convenient for this specific task) is to make use of *algebras with Okay predicates* - the specification framework introduced in [GDLE84] and developed further in [ANK90]. Algebraic specifications in this approach make explicit syntactical distinction between safe/unsafe variables, functions and terms; this gives a direct way to implement the restrictions on substitutions and to represent EDPA (cf. more details in [AD92]).

Acknowledgements:

We thank Natalya Soboleva for helpful technical assistance, and Michael Rusinowitch for careful reading this paper and useful remarks.

The first author is much obliged to Peter Mosses for valuable remarks on the text of the paper, as well as for permanent help and encouragement. He would like to thank Helene Kirchner for discussing some interesting examples concerning this work. He also acknowledges the financial support of the Danish Natural Sciences Research Council, grant No 11-9479.

References

[AD92] Antimirov V., Degtyarev A. Consistency of equational enrichments. In A. Voronkov, editor, *Logic Programming and Automated Reasoning. International Conference LPAR '92.* LNCS **624**, pp. 393-402, Springer-Verlag, 1992.

[ANK90] Antimirov V., Naidich D., Koval V.: Partial Functions in simulation: formal models and calculi. In *Proc. IMA CS European Simulation Meeting,* pp.143-148, Esztergom, Hungary, 1990.

[AB92] Avenhaus J., Becker K.: Conditional rewriting modulo a built-in algebra. Technical report (SEKI Report SR-92-11), 1992, 23p.

[DJ90] Dershowitz N., Jouannaud J.-P. Rewrite systems. In J.van Leeuwen, A.Meyer, M.Nivat, M.Paterson, and D.Perrin editors, *Handbook of Theoretical Computer Science,* volume B, chapter 6, Elsevier Sci. Pub, 1990.

[EM85] Ehrig H., Mahr B. *Fundamentals of algebraic specification 1: Equations and Initial Semantics.* Number 6 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1985.

[GDLE84] Gogolla M., Drosten K., Lipeck U., Ehrich H.-D. Algebraic and operational semantics of specifications allowing exceptions and errors. *Theoretical Comp. Sci.,* **34**:289-313, 1984.

[GM87] Goguen J., Meseguer J. Order-sorted algebra solves the constructor selector, multiple representation and coercion problems. In *Proc. Second Symposium on Logic in Comp. Sci.,* pp. 18-29, IEEE Comp. Society Press, 1987.

[GM89] Goguen J., Meseguer J. Order-sorted algebra 1. SRI International, Technical Report SRI-CLS-89, July 1989.

[GTW78] Goguen J., Thatcher J., Wagner E. An initial algebra approach to the specification, correctness and implementation of abstract data types. In *Current trends in programming methodology,* volume IV, pp.80-149, Prentice-Hall, 1978.

[GW88] Goguen J., Winkler T. Introducing OBJ3. Technical report SRI-CSL-89-10, Comp. Sci. Lab., SRI International, 1988.

[JK89] Jouannaud J.-P., Kounalis E. Automatic proofs by induction in theories without constructors. *Information and Computation,* **82**, 1:1-33 1989.

[Kir92] Kirchner H. Proofs in parameterized specifications. Technical report (extended version) CRIN 91-R-045.

[Kre87] Kreowski H.-J. Partial algebras flow from algebraic specifications. In *ICALP'871 Proc. Int. Coll. on Automata, Languages, and Programming,* LNCS **267**, pp. 521-530, Springer-Verlag, 1987.

[Mos92] Mosses P. The Use of Sorts in Algebraic Specifications. In Michel Bidoit and Christine Choppy, editors, *Recent Trends in Data Type Specification.* LNCS **655**, Springer-Verlag, 1992.

[Pad90] P. Padawitz. Horn logic and rewriting for functional and logic program design. Report MIP-9002, Universität Passau, 1990.

[SNGM89] Smolka J., Nutt W., Goguen J., Meseguer J. Order-sorted equational computation. In H.Aït-Kaci and M.Nivat, editors, *Resolution of Equations in Algebraic Structures,* pp. 297-367, Academic Press, New-York, 1989.

[Wir90] Wirsing M. Algebraic specification. In J. van Leeuwen, A. Meyer M. Nivat, M. Paterson, and D. Perrin editors, *Handoook of Theoretical Computer Science,* volume B, chapter 13. Elsevier Sci. Pub., 1990.