REVISTA MATEMÁTICA de la
Universidad Complutense de Madrid
Volumen 10, número 2: 1997
http://dx.doi.org/10.5209/rev_REMA.1997.v10.n2.17446

Correctness of horizontal and vertical composition for implementation concepts based on constructors and abstractors.

Hartmut EHRIG, Hans-Jörg KREOWSKI and Fernando OREJAS

Abstract

A new implementation concept for parameterized specifications based on constructors and abstractors was recently introduced by Orejas, Navarro and Sánchez which includes most of the implementation concepts in the literature for initial as well as loose semantics. In this paper we redefine vertical and different kinds of horizontal compositions using the new concept of semi-pushout defined for a mixture of signature and specification morphisms. The main results concerning correctness of horizontal and vertical composition are based on new correctness requirements for constructors and abstractors.

1 Introduction

Inspired by various approaches in the literature a unifying implementation concept for unparameterized and parameterized specifications was recently presented by Orejas, Navarro and Sánchez [ONS 93, ONS 96]. On one hand it is based on the ideas of constructors and abstractors in the sense of Sannella and Tarlecki [ST88], on the other hand it includes explicitly extension and restriction steps as considered in various other approaches (e.g.[Ehrich 82], [EK 83], [EG 94]). Although the problems of horizontal and vertical composition of unparameterized and parameterized specifications have been discussed in [ONS 93, ONS 96],

Mathematics Subject Classification: 68Q65, 68Q60, 68N05.
Servicio Publicaciones Univ. Complutense. Madrid, 1997.

only explicit conditions for correctness of vertical composition in the unparameterized case have been given up to now. It remains to study correctness of horizontal and vertical composition for parameterized specifications in this framework.

The general case of horizontal composition of parameterized specifications allows to have a parameter passing morphism between the parameter of the second and the body of the first specification. Hence actualization is an essential part of horizontal composition, which has been studied in our paper [EKO 95]. In this paper we study the general case of horizontal and vertical composition. The main results of this paper show under which conditions for contructors and abstractors we have correctness of horizontal and vertical composition of implementations.

The constructions for horizontal and vertical composition of parameterized specifications are based on a new concept, called semi-pushouts defined for a mixture of signature and specification morphisms. The general case of horizontal composition can be obtained as a combination of direct horizontal composition, where the given parameterized specifications are directly composable, and actualized implementation, where a given implementation is actualized according to a given parameter passing morphism. The constructions for vertical and horizontal composition are given in section 2 of this paper, where we assume to have an implementation concept based on an institution in the sense of [GB 84], which associates to each specifications morphism a constructor and for each signature morphism an abstractor in the sense of [ONS 96].

In section 3 of this paper we define axiomatic properties for constructors and abstractors and show under which of these properties we have correctnes of vertical composition, direct horizontal composition, special horizontal composition used in [ONS 96], actualized implementations and of general horizontal composition of implementations. Moreover we give some remarks how corresponding results in the literature can be obtained as special cases, especially those in [EK 83] and [EG 94] for initial and final semantics and a standard case for loose semantics.

Finally in section 4 we discuss some remaining conceptual problems with the framework presented in sections 2 and 3 and several proposals how to solve these problems. Especially it is open to analyse the implementation concepts and results known from the literature in a systematic way, to find out which are the corresponding constructors and

abstractors and to analyse how far the axiomatic properties given in section 3 are satisfied in each of these cases.

For a more detailed motivation of the general implementations based on these concepts we refer to [ST 88, ONS 93, ONS 96].

2 Horizontal and vertical composition of implementations

In this section we introduce a general implementation concept based on constructors and abstractors in the sense of [ONS 93, ONS 96], which are motivated by those in [ST 88]. With respect to this general notion we define the implementation of parameterized specification as well as different kinds of horizontal and vertical composition.

2.1 Implementations

In 1972 Hoare [Hoare 72] presented the first notion of data type implementation in the literature. This was the beginning of a long series of papers dealing with the same concept (see [ONS 96]). There are several reasons for such a number of approaches: in some papers the framework studied is different ("loose" vs "initial" specifications, parameterized vs non-parameterized specifications, partial vs total data types, etc.), in other papers the aim is different (some approaches would stress only "semantic" aspects of implementation while others would focus on "syntactic" or "proof-theoretic" aspects); nevertheless, the underlying intuition is often the same.

Given specifications $SP1 = (\Sigma 1, E1)$ and $SP2 = (\Sigma 2, E2)$, where $E1$ and $E2$ are sets of formulae over any suitable institution (i.e. not necessarily equations), we may consider that implementing the data type specified by $SP1$ by the data type specified by $SP2$ consists in defining the operations (and the data sorts) in $\Sigma 1$ in terms of the operations (and data sorts) from $\Sigma 2$, in such a way that the enriched $SP2-$models "behave" like the $SP1-$models. In this sense, syntactically, an implementation would be an enrichment together with a mapping (technically, a signature morphism) relating the sorts and operations from $\Sigma 1$ with the sorts and operations from the enriched $\Sigma 2$ signature and, semantically, it would be a *construction* (associated to the enrichment) together with

some kind of *abstraction* (associated to the given signature morphism) that relates the models of $SP1$ with the enriched $SP2-$models.

For example, suppose that we want to implement (finite) sets of integers by sequences of integers. The implemented (abstract) specification is:

SP1=INTEGER+BOOLEAN+

sorts set

operations

$\emptyset :\rightarrow set$

$add : set \times int \rightarrow set$

$\_$ is-in $\_ : int \times set \rightarrow bool$

var $S : set; n, n' : int$

equations

$add(add(S,n),n) = add(S,n)$

$add(add(S,n),n') = add(add(S,n'),n)$

n is-in $\emptyset$ =false

n is-in $add(S,n') = (n \; eq \; n')$ or (n is-in S)

and the specification of the "implementing" (concrete) data type is:

SP2=INTEGER+

sorts seq

operations

$e - s :\rightarrow seq$

$app : seq \times int \rightarrow seq$

$head : seq \rightarrow int$

$tail : seq \rightarrow seq$

var $S : seq; n : int$

equations

$tail(app(S,n)) = S$

$tail(e - s) = e - s$

$head(app(S,n)) = n$

Now, if we decide to represent the sets of integers by sequences without repetitions then, the first step would consist in an enrichment, where all the set operations are defined in terms of the sequence operations in an adequate manner. For example:

IMPL= SP2+

operations

$\emptyset :\rightarrow seq$

$add : seq \times int \rightarrow seq$

$\_$ is-in $\_ : int \times seq \rightarrow bool$

var $S : seq; n : int$

equations

$\emptyset = e - s$

n is-in $e - s$ =false

n is-in $app(S, n') = (n \, eq \, n')$ or (n is-in S)

$add(S, n) = S$ **if** n is-in S =true

$add(S, n) = app(S, n)$ **if** n is-in S =false

The second step would consist in relating the sorts and operations (to be implemented) defined in $SP1$ with the sorts and operations (in the implementation) defined in IMPL. As said above, this can be done by means of a signature morphism mapping the sort set into the sort seq and the operations $\emptyset$, add and is-in from $SP1$ into the operations $\emptyset$, add and is-in from IMPL.

Semantically, the result of this enrichment would be an IMPL-algebra A not satisfying some of the axioms in $SP1$. In particular, A does not satisfy the equation

$$add(add(S, n), n') = add(add(S, n'), n)$$

The reasson is that, in general, a set may be represented by several different sequences. For instance, the set $\{1, 2\}$ is represented in this algebra by two sequences: $< 1, 2 >$ and $< 2, 1 >$. Nevertheless, A "behaves" like the algebra of sets, in the sense that the evaluation of any $\Sigma(SP1)-$term t of integer or boolean sort (and its corresponding translation through the given signature morphism) yields the same value in both algebras. $\Sigma(SP1)$ is assumed to denote the signature of the specification $SP1$.

These ideas can be easily generalized to deal with parameterized specifications. In particular, if $h : SP \rightarrow SP1$ and $h' : SP \rightarrow SP1'$ are parameterized specifications (for simplicity, let us assume that h and h' are inclusions, i.e. if $SP = (\Sigma, E)$, $SP1 = (\Sigma 1, E1)$ and $SP2 = (\Sigma 2, E2)$ then $\Sigma \subseteq \Sigma i$ and $\Sigma \subseteq Ei, i = 1, 2$), we may also consider that implementing the parameterized data type specified by h by the parameterized data

type specified by h' consists in defining the operations and the data sorts in $\Sigma 1$ (or, rather, $\Sigma 1 - \Sigma$) in terms of the operations and data sorts from $\Sigma 2$, in such a way that for every SP−model A, the enrichment applied to $K_{h'}(A)$ "behaves" like $K_h(A)$, where K_h and $K_{h'}$ are, respectively, the meaning of the parameterizations h and h'. Therefore, again, syntactically an implementation would be an enrichment together with a signature morphism relating the sorts and operations from $\Sigma 1$ with the sorts and operations from the enriched $\Sigma 2$ signature and, semantically, it would also be a *construction* (associated to the enrichment) together with some kind of *abstraction* (associated to the given signature morphism) that relates the models of $K_h(SP)$ with the enriched $K_{h'}(SP)$ models.

2.2 General assumptions

Given an institution in the sense of [GB 84] we have a category of signatures and signature morphisms as well as a category of specifications and specification morphisms and for each specifications SP a class $\mathrm{Mod}(SP)$ of models over SP.

Moreover we assume to have an implementation concept IC defined by the property that we have for each specification morphism $m : SP1 \to SP2$ a constructor $K_{IC}(m) : \mathrm{Mod}(SP1) \to \mathcal{P}(\mathrm{Mod}(SP2))$, short K_m, and for each signature morphism $f : \Sigma(SP1) \to \Sigma(SP2)$ between specifications, short $f : SP1 \multimap SP2$, an abstractor $\alpha_{IC}(f) : \mathrm{Mod}(SP1) \to \mathcal{P}(\mathrm{Mod}(SP1))$, short α_f, in the sense of [ONS 93, ONS 95].

2.3 Definition (implementation of parameterized specifications)

Given parameterized specifications $h : SP \to SP1$ and $h' : SP \to SP1'$, where h and h' are specification morphisms with same formal parameter specifications SP, an **implementation** $I = (m, f)$ **of** h **by** h' is given by a specification morphism $m : SP1' \to SP2$ and a signature morphism $f : SP1 \multimap SP2$ with same target specifications $SP2$ s.t. $f \circ h = m \circ h'$ (as signature morphisms).

$$\begin{array}{ccc} SP & \xrightarrow{h} & SP1 \\ h' \downarrow & = & \downarrow f \\ SP1' & \xrightarrow[m]{} & SP2 \end{array}$$

The implementation $I = (m, f)$ of h by h' is called **correct**, if for all $A \in \mathrm{Mod}(Sp)$ and all $A2 \in K_{h' \circ m}(A)$ there is $A1 \in K_h(A)$ s.t. $U_f(A2) \in \alpha_f(A1)$.

Remark. $U_f : \mathrm{Mod}(\Sigma(SP2)) \to \mathrm{Mod}(\Sigma(SP1))$ is the forgetful functor corresponding to the signature morphism $f : SP1 \to SP2$ and $\Sigma(SPi)$ is the signature part of $SP1$ for $i = 1, 2$.

2.4 Special cases of implementation concepts

For each choice of constructors and abstractors and by specialization of the components of an implementation we obtain a specific implementation concept. Reachability in the sense of [EK 83] or [EG 94] leads to an abstractor α defined by

$$B1' \in \alpha_r(B1) \Leftrightarrow \mathrm{REACH}_{s1}(B1') = B1$$

where $\mathrm{REACH}_{s1}(B1')$ is the intersection of all submodels B' of $B1'$ with $U_{s1}(B') = U_{s1}(B1')$. Standard abstraction α_s is defined by

$$B1' \in \alpha_s(B1) \Leftrightarrow \exists \text{ morphism } m : B1' \to B1.$$

For other notions of abstractors we refer to [ONS 93, ONS 96], where especially behavioural abstraction is discussed as an important example.

If all constructors K_s for $s : SP1 \to SP2$ are free constructions F_s : $\mathrm{Mod}(SP1) \to \mathrm{Mod}(SP2)$ and α is the reachability abstractor we obtain implementation of parameterized specifications with IR−semantics in the sense of [EK 83]. If in addition the extension e is the identity we obtain R-implementations and – without restriction – refinements of parameterized specifications in the sense of [EG 94], where restriction is given by a specification morphism. If, on the other hand, we keep general monomorphic constructors and general abstractors, but have only identical restrictions, then we obtain constructor and abstractor

implementations in the sense of Sannella and Tarlecki [ST 88]. For other special cases we refer to [ONS 93, ONS 96].

In order to define vertical and horizontal composition of implementations in an elegant way we need the notion of semi-pushouts, defined for a mixture of signature and specification morphisms.

2.5 Definition (semi-pushout)

Given a signature morphism $f : SP \multimap SP1$ and a specification morphism $g : SP \to SP2$ a specification $SP3$ together with a signature morphism $f' : SP2 \multimap SP3$ and a specification morphism $g' : SP1 \to SP3$ is called **semi-pushout** of f and g if

1. $g' \circ f = f' \circ g$ as signature morphisms.
2. For all $h1 : SP1 \to SP4$ and $h2 : SP2 \multimap SP4$, with $h1 \circ f = h2 \circ g$, there is a unique $h : SP3 \to SP4$ s.t. $h \circ g' = h1$ and $h \circ f' = h2$, where $h1$ and h are specification morphisms and $h2$ is a signature morphism.

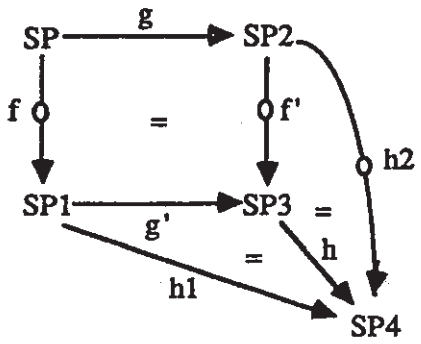

Figure 1: Semi-Pushout

2.6 Fact (semi-pushout)

1. The semi-pushout of $f : SP \multimap SP1$ and $g : SP \to SP2$ can be constructed as the pushout $\Sigma 3$ of f and g in the category of signature morphisms, where $SP3 = (\Sigma 3, E3)$ with $E3 = g'\#(E1)$, i.e. the translated axioms of $E1$ for $SP2 = (\Sigma 1, E1)$.

2. Semi-pushout objects are unique up to isomorphism in the category of specification morphisms.

3. Horizontal and vertical composition of semi-pushouts are semi-pushouts.

4. If f is a specification morphism the pushout of f and g is in general different from the semi-pushout of f and g.

Proof.

1. By definiton of $SP3$ the signature morphism g' in figure 1 becomes a specification morphism. Moreover, given $h1$ and $h2$, as in 2.5.2, there is a unique signature morphism $h : SP3 \xrightarrow{\circ} SP4$, where $h\#(E3) = h\#(g'\#(E1)) = (h \circ g')\#(E1) = h1\#(E1)$ is derivable fron $E4$ because $h1$ is a specification morphism. Hence, also h is a specification morphism.

2. Follows in the same way as uniqueness of pushouts using that h in figure 1 is a specification morphism.

3. Follows as usual from the universal properties of the construction in part 1 which is a characterization due to part 2.

4. If f is specification morphism the specification pushout is given by

$$SP3' = (\Sigma 3, g'\#(E1) \cup f'\#(E2))$$

which is in general different from the semi-pushout $SP3 = (\Sigma 3, g'\#(E1))$.

■

2.7 Definition (vertical composition of implementations)

Given implementations $I1 = (m1, f1)$ of $h1$ by $h2$ and $I2 = (m2, f2)$ of $h2$ by $h3$ for $hi : SP \to SPi (i = 1, 2, 3)$ the **vertical composition** $I3 = (m3, f3)$ of $I1$ and $I2$, written $I3 = I2^* I1$, is given by $m3 = m1' \circ m2$ and $f3 = f2' \circ f1$ where (3) in figure 2 is the semi-pushout of $m1$ and $f2$. $I3$ is an implementation of $h1$ by $h3$.

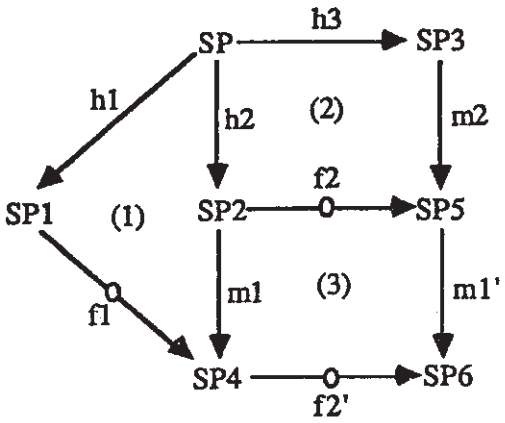

Figure 2: Vertical composition of implementations

Remark. In [ONS 96] the vertical composition is defined by a pushout $\Sigma 6$ of $m1$ and $f2$ considered as signature morphisms which causes the problem making $m3$ a specification morphism.

Before we define the general horizontal composition we consider the special cases of direct horizontal composition and actualized implementations which allow to obtain the general case by combination of both constructions.

2.8 Definition (direct horizontal composition of implementations)

Given implementations $I1 = (m1, f1)$ of $h1$ by $h1'$ and $I2 = (m2, f2)$ of $h2$ by $h2'$ as shown in figure 3, where $h1$ and $h2$ are directly composable. Then the **direct horizontal composition** $I3 = (m3, f3)$ of $I1$ and $I2$, written $I3 = I2 \circ I1$, is an implementation of $h3$ by $h3'$ given by

$h3 = h2 \circ h1$, $h3' = h4 \circ m1 \circ h1'$, $m3$, $f3 = f5 \circ f2$ as defined in figure 3 where (3) and (4) are constructed as semi-pushouts.

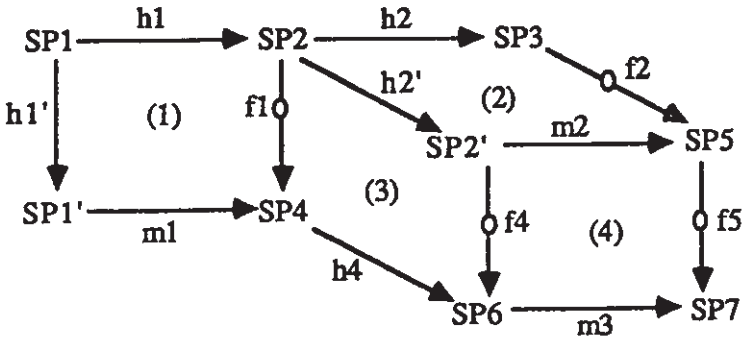

Figure 3: Direct horizontal composition of implementations

2.9 Definition (actualized implementation)

Given an implementation $I1 = (m1, f1)$ of $h1$ by $h1'$ and a specification morphism g, called **parameter passing morphism**, the **actualized implementation** $I2 = (m2, f2)$ of $I1$ via g is an implementation of $h2$ by $h2'$, written $I2 = g\#(I1)$, $h2 = g\#(h1)$, $h2' = g\#(h1')$, where $h2$, $h2'$, and $m2$ are constructed via the pushouts $SP3$, $SP3'$ and $SP4$ in the back, left and front square of figure 4 and $f2$ is the induced signature morphism.

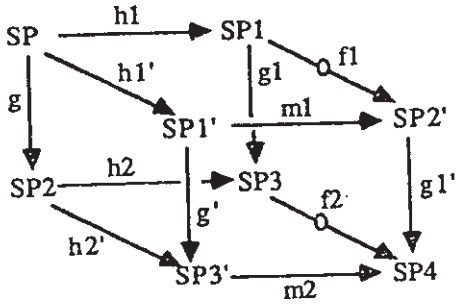

Figure 4: Actuallized implementation

2.10 Definition (horizontal composition of implementations)

Given implementations $I1 = (m1, f1)$ of $h1$ and $h1'$ and $I2^* = (m2^*, f2^*)$ of $h2^*$ by $h2'^*$ and a specification morphism g as shown in figure 5,

i.e. the actualization $g\#(h2^*) = h2$ of $h2^*$ is composable with $h1$. Then the **horizontal composition** $I3 = (m3, f3)$ of $I1$ and $I2^*$ via g, written $I3 = I2^* \circ_g I1$, is an implementation of $h3$ by $h3'$ given by $h3 = h2 \circ h1$, $h3' = h4 \circ m1 \circ h1'$, $f3 = f5 \circ f2$ as shown in figure 5 constructed as in figure 4 and then figure 3.

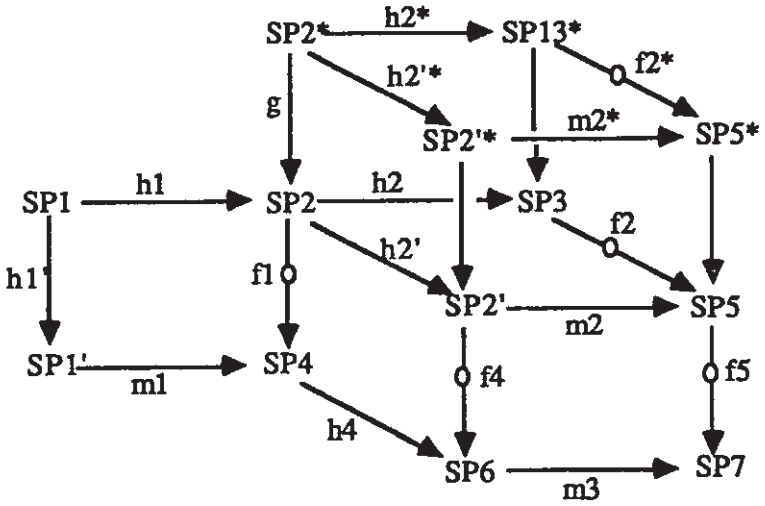

Figure 5: Horizontal composition of implementations

Remark. The horizontal composition considered in [ONS 96] is the special case of horizontal composition above where only an implementation $I1$ of $h1$ by $h1'$, $h2^*$ and g but no implementation $I2^*$ of $h2^*$ by $h2'^*$ are given. This means that in figure 4 we have $h2^* = h2'^*$, $h2 = h2'$ and $m2^*$, $m2, m3$, and $f5$ are identities.

2.11 Fact (composability of horizontal composition)

The horizontal composition $I3 = I2^* \circ_g I1$ of $I1$ and $I2^*$ via g is equal to the direct horizontal composition $I3 = I2 \circ I1$ of $I1$ with the actualized implementation $I2 = g\#(I2^*)$ of $I2^*$ via g, i.e.

$$I2^* \circ_g I1 = g\#(I2^*) \circ I1$$

3 Correctness results

In order to prove correctness of horizontal and vertical composition of implementations we have to formulate several axiomatic properties for the constructor and abstractor of the implementation concept which are defined below. In addition to the general assumptions in 2.2 we assume that the given institution has amalgamation in the sense of [EG 94], i.e. the model functor Mod transforms pushouts into pullbacks.

3.1 Definition (axiomatic properties of constructors)

1. The constructor K is **functorial**, if we have

$$K_{f2 \circ f1} = K_{f2} \circ K_{f1}$$

for all specification morphisms $f1 : SP \to SP2$ and $f2 : SP2 \to SP3$, where $K_{f2} \circ K_{f1}(A1) = \cup\{K_{f2}(A2)/A2 \in K_{f1}(A1)\} \subseteq \text{Mod}(SP3)$.

2. The constructor K is **persistent**, if for each $f1 : SP1 \to SP2$, $A1 \in \text{Mod}(SP1)$, $A2 \in K_{f1}(A1)$ we have $U_{f1}(A2) = A1$.

3. The constructor K is **compatible with amalgamation** if K is persistent and for all pushout

$$\begin{array}{ccc} SP & \xrightarrow{h1} & SP1 \\ {\scriptstyle h2}\downarrow & (PO) & \downarrow{\scriptstyle g1} \\ SP2 & \xrightarrow[g2]{} & SP3 \end{array}$$

and all $A \in \text{Mod}(SP)$, $A1 \in K_{h1}(A)$, $A2 \in \text{Mod}(SP2)$ with $U_{h2}(A2) = A$ we have

$$A1 +_A A2 \in K_{g2}(A2).$$

If moreover we have

$$K_{g2}(A2) = \{A1 +_A A2/A1 \in K_{h1}(A) \text{ and } A = U_{h2}(A2)\}$$

then K is called **strongly compatible with amalgamation.**

4. The constructor K has the **extension property** if for all pushouts as given above we have

$$K_{h1} \circ U_{h2} = U_{g1} \circ K_{g2}$$

3.2 Fact (strong compatibility of constructors)

A constructor is strongly compatible with amalgamation if and only if it has the extension property and is compatible with amalgamation.

Proof. If K is strongly compatible then it is also compatible and $A1 \in K_{h1} \circ U_{h2}(A2) = K_{h1}(A)$ implies $A1 +_A A2 \in K_{g2}(A2)$ and hence $A1 \in U_{g1} \circ K_{g2}(A2)$. Conversely, $A1 \in U_{g1} \circ K_{g2}(A2)$ implies $A3 \in K_{g2}(A2)$ with $A1 = U_{g1}(A3)$ and $U_{g2}(A3) = A2$ by persistency of K. Hence we have $A3 = A1 +_A A2 \in K_{g2}(A2)$ with $A = U_{h2}(A2)$ which implies $A1 \in K_{h1}(A) = K_{h1} \circ U_{h2}(A2)$ by strong compatibility of K. Hence we have $K_{h1} \circ U_{h2} = U_{g1} \circ K_{g2}$. Conversely, let K be a compatible with extension property. It remains to show that $A3 = A1 +_A A2 \in K_{g2}(A2)$ implies $A1 \in K_{h1}(A)$. But $A3 \in K_{g2}(A2)$ implies $A1 \in U_{g1} \circ K_{g2}(A2) = K_{h1} \circ U_{h2}(A2) = K_{h1}(A)$.

■

3.3 Definiton (axiomatic properties of abstractors)

1. An abstractor α is **compatible with amalgamation** if for all diagrams (1) and (2)

$$\begin{array}{ccccc} SP & \xrightarrow{h1} & SP1 & \xrightarrow{f1} & SP2' \\ g\downarrow & (1) & \downarrow g1 & (2) & \downarrow g1' \\ SP2 & \xrightarrow[h2]{} & SP3 & \xrightarrow[f2]{} & SP4 \end{array}$$

where (1) is a pushout and (2) a signature PO (not necessarily semi-pushout as given in figure 4) then for all $A1 \in \mathrm{Mod}(SP1)$, $A2 \in \mathrm{Mod}(SP2)$ with $U_g(A2) = A$ and all $A1' \in \alpha_{f1}(A1)$ with $U_{h1}(A1') = A = U_{h1}(A1)$ we have

$$(A1' +_A A2) \in \alpha_{f2}(A1 +_A A2)$$

Remark. Reachability abstractors and standard abstractors are compatible with amalgamation.

2. An abstractor α is **compositional** if for all signature morphisms $f1 : SP1 \multimap SP2$ and $f2 : SP2 \multimap SP3$ and all $Ai \in \mathrm{Mod}(SPi)\, i = 1,2,3$ with

$$U_{f1}(A2) \in \alpha_{f1}(A1),\ U_{f2}(A3) \in \alpha_{f2}(A2)$$

we also have $U_{f2 \circ f1}(A3) \in \alpha_{f2 \circ f1}(A1)$.

Remark. This condition is similar to $C1$ in [ONS 96], especially satisfied for α defined by reachability and free constructions K provided that $A1$ and $A3$ are freely generated.

3. An abstractor α **has amalgamation complements w.r.t. a constructor** K if for all semi-pushouts

$$\begin{array}{ccc} SP & \xrightarrow{g} & SP2 \\ h2\downarrow & (S-PO) & \downarrow f' \\ SP1 & \xrightarrow[g']{} & SP3 \end{array}$$

and all $A \in \mathrm{Mod}(SP)$, $Ai \in \mathrm{Mod}(SPi) i = 1,3$ with $A3 \in K_{g'}(A1)$ and $U_f(A1) \in \alpha_f(A)$ there is $A2 \in \mathrm{Mod}(SP2)$ with $A2 \in K_g(A)$ and $U_{f'}(A3) \in \alpha_{f'}(A2)$.

Remark. If f, f' are specification morphisms and α =equality then the condition is satisfied for translation constructors.

3.4 Theorem (correctnes of vertical composition)

Given correct implementations $I1$ and $I2$ the vertical composition $I3 = I2^* I1$ is correct provided that

1. The constructor K is functorial.
2. The abstractor α is compositional.
3. The abstractor α has amalgamation complements w.r.t. K.

Proof. Given correct $I1$ and $I2$, and $I3$ as defined in figure 2 we have to show for all $A \in \mathrm{Mod}(SP)$ and $A6 \in K_{m3 \circ h3}(A)$ the existence of

$A1 \in K_{h1}(A)$ with $U_{f3}(A6) \in \alpha_{f3}(A1)$. For $A6 \in K_{m3 \circ h3}(A)$ we have by functoriality of K some $A5 \in K_{m2 \circ h}(A)$ with $A6 \in K_{m1'}(A5)$. Correctness of $I2$ implies that there is $A2 \in K_{h2}(A)$ with $U_{f2}(A5) \in \alpha_{f2}(A2)$. Since α has amalgamation complements and (3) in figure 2 is semi-pushout we have some $A4 \in \text{Mod}(SP4)$ with $A4 \in K_{m1}(A2)$ and $U_{f2'}(A6) \in \alpha_{f2'}(A4)$. Now correctness of $I1$ implies that there is $A1 \in K_{h1}(A)$ with $U_{f1}(A4) \in \alpha_{f1}(A4) \in \alpha_{f1}(A1)$. Finally compositionality of α applied to $f3 = f2' \circ f1$ implies $U_{f3}(A6) \in \alpha_{f3}(A1)$. Since we have already $A1 \in K_{h1}(A)$ this implies correctness of $I3$. ■

Remark. The correctness of vertical composition of refinements and $R-$implementations in [EG 94] restricted to identical parameter specifications is a special case of this result.

3.5 Theorem (correctness of direct horizontal composition)

Given correct implementations $I1$ and $I2$ the direct horizontal composition $I3 = I2 \circ I1$ is correct provided that

1. The constructor K is functorial.
2. The abstractor α is compositional.
3. The abstractor α has amalgamation complements w.r.t. K.

Proof. Given correct $I1$ and $I2$, and $I3$ as defined in figure 3 we have to show for all $A1 \in \text{Mod}(SP1)$ and $A7 \in K_{m3 \circ h3'}(A1)$ the existence of $A3 \in K_{h3}(A1)$ with $U_{f3}(A7) \in \alpha_{f3}(A3)$. For $A1 \in \text{Mod}(SP1)$ and $A7 \in K_{m3 \circ h3'}(A1)$ we have by functoriality of K some $A4 \in K_{m1 \circ h1'}(A1)$ with $A7 \in K_{m3 \circ h4}(A4)$. Correctness of $I1$ implies the existence of $A2 \in K_{h2}(A1)$ with $U_{f1}(A4) \in \alpha_{f1}(A2)$. Since α has amalgamation complements and (3) $\cup$ (4) in figure 3 is a semi-pushout we have some $A5 \in \text{Mod}(SP5)$ with $A5 \in K_{m2 \circ h2'}(A2)$ and $U_{f5}(A7) \in \alpha_{f5}(A5)$. Now correctness of $I2$ implies the existence of $A3 \in K_{h2}(A2)$ with $U_{f2}(A5) \in \alpha_{f2}(A3)$, and compositionality of α applied to $f3 = f5 \circ f2$ implies $U_{f3}(A7) \in \alpha_{f3}(A3)$. Finally $A2 \in K_{h1}(A1)$ and $A3 \in K_{h2}(A2)$ imply $A3 \in K_{h3}(A1)$ by functioriality of K and $h3 = h2 \circ h1$, which implies correctness of $I3$.

■

3.6 Corollary (correctness of special horizontal composition)

The correctness of special horizontal composition considered as property $P2$ in [ONS 96] (see Remark below 2.10) holds provided that the constructor K is functorial and the abstractor α has amalgamation complements w.r.t. K.

Proof. Special case of the proof of theorem 3.5 where $h2 = h2'$ and $m2$, $m3$, $f2$ and $f5$ are identities. Hence we only have to use the existence of amalgamation complements applied to semi-pushout (3) in figure 2 but do not need compositionality of α.

■

3.7 Theorem (correctness of actualized implementations)

Given a correct implementation $I1$ of $h1$ by $h1'$ and a parameter passing morphism g the actualized implementation $I2 = g\#(I1)$ of $h2 = g\#(h1)$ by $h2' = g\#(h1')$ is correct provided that

1. The constructor K is functorial and is strongly compatible with amalgamation, i.e. persistent, compatible with amalgamation and has the extension property.

2. The abstractor α is compatible with amalgamation.

Proof. Given correct $I1$ of $h1$ by $h1'$ and the actualized implementation $I2$ of $I1$ via g as shown in figure 4 correctness of $I2$ requires to show for each $A2 \in \mathrm{Mod}(SP2)$ and $A4 \in K_{m2 \circ h2'}(A2)$ the existence of $A3 \in K_{h2}(A2)$ with $U_{f2}(A4) \in \alpha_{f2}(A3)$. For $A2$ and $A4$ as above we define $A = U_g(A2)$ and $A2' = U_{g1'}(A4)$. The extension property of K applied to the composed pushout of left and front square in figure 4 means $K_{m1 \circ h1'} \cdot U_g = U_{g1'} \cdot K_{m2 \circ h2'}$. Hence we have

$$A2' = U_{g1'}(A4) \in U_{g1'} \cdot K_{m2 \circ h2'}(A2) = K_{m1 \circ h1'} \cdot U_g(A2) = K_{m1 \circ h1'}(A).$$

Now correctness of $I1$, implies the existence of $A1 \in K_{h1}(A)$ with $U_{f1}(A2') \in \alpha_{f1}(A1)$. Let $A3 = A1 +_A A2 \in \mathrm{Mod}(SP3)$ which is well-defined because $A1 \in K_{h1}(A)$ implies $U_{h1}(A1) = A$ by persistency of K and we have already $U_g(A2) = A$. It remains to show $A3 \in K_{h2}(A2)$

and $U_{f2}(A4) \in \alpha_{f2}(A3)$. The first property follows from compatibility of K with amalgamation using $A3 = A1 +_A A2$ and $A1 \in K_{h1}(A)$. In order to show the second property we apply compatibility of α with amalgamation applied to diagrams (1) and (2) in 3.3.1, where (2) is a signature pushout as a consequence of the construction in figure 4. Let $A1' = U_{f1}(A2')$, then $A1' = U_{f1}(A2') \in \alpha_{f1}(A1)$ and we have $U_{h1}(A1') = U_{h1} \cdot U_{f1} \cdot (A2') = U_{f1 \circ h1}(A2') = U_{m1 \circ h1'}(A2') = A$ using $A2' \in K_{m1 \circ h1'}(A)$ as shown above and persistency of K. Hence we have $A1' \in \alpha_{f1}(A1), U_{h1}(A1') = A = U_{h1}(A1)$, and $U_g(A2) = A$ s.t. compatibility of α with amalgamation implies $(A1' +_A A2) \in \alpha_{f2}(A1 +_A A2) = \alpha_{f2}(A3)$ (see 3.3.1). In order to show $U_{f2}(A4) \in \alpha_{f2}(A3)$ it remains to show $U_{f2}(A4) = A1' +_A A2$ and hence by uniqueness of amalgamation we only have to show equality after application of U_{g1} and U_{h2}. In fact we have:

$$U_{g1} \cdot U_{f2}(A4) = U_{f1} \cdot U_{g1'}(A4) = U_{f1}(A2') = A1' = U_{g1}(A1' +_A A2)$$

and using persistency of K in the case $A4 \in K_{m2 \circ h2'}(A2)$ we have

$$U_{h2} \cdot U_{f2}(A4) = U_{m2 \circ h2'}(A4) = A2 = U_{h2}(A1' +_A A2).$$

■

Remarks.

1. According to 2.4 we obtain implementation of parameterized specifications with IR-semantics in the sense of [EK 83] if all constructors are persistent free constructions and α is the reachability abstractor. The assumptions of theorem 3.7 are satisfied, because persistent free constructions which are closed under extension (see e.g. [EM 85]) and reachability is compatible with amalgamation as shown in [EKO 95]. The reason is that if $\mathrm{REACH}_S(A1 +_A A2) = \mathrm{REACH}_S(A1' +_A A2)$ then $\mathrm{REACH}_S(A1) = \mathrm{REACH}_S(A1')$ since REACH commutes with the forgetful functor. Hence theorem 3.7 can be applied leading to $IR-$ correctness of actualized implementations, which is shown explicitly in [EK 83].

2. If in addition in Remark 1 the specification morphism m is the identity we obtain correctness of actualized $R-$implementations

and –without restriction– refinements of parameterized specifications in a suitable institution, which is explictly shown in [EG 94]. The approach in [EG 94] also includes final semantics.

3. If all constructors are translations and abstraction is standard abstractions (see 2.4) then again the assumptions of theorem 3.7 are satisfied. In fact, it is easy to show that translations are (persistent) constructors which are closed under extension, i.e.

$$\text{translates1}'(P') = \{P' +_p B1/B1 \in \text{translates1}(P)\}.$$

Moreover standard abstraction is compatible with amalgamation, because $B1^* \in \alpha_s(B1)$ according to 2.4 means existence of $m : B1^* \to B1$ which implies

$$m' = P' +_p m : P' +_p B1^* \to P' +_p B1$$

and hence $B1'^* \in \alpha'(B1')$. This means that theorem 3.7 can be applied in this case leading to correct actualized implementations in the framework of loose semantics.

3.8 Theorem (correctness of horizontal composition of implementations)

Horizontal composition of correct implementations is correct provided that the constructor K and the abstractor α satisfy all the properties stated in 3.1 and 3.2.

■

Proof. Direct consequence of fact 2.11, theorem 3.5 and 3.7.

Remark. It remains to check how far the properties stated in 3.1 and 3.2 are satisfied for various cases of implementations of parameterized specifications studied in the literature. Some examples have been discussed already in the remarks of 3.3, 3.4 and 3.7.

4 Open problems

Although we have shown, for some of the correctness results in section 3, how to obtain several results in the literature as special cases, it remains open to analyse all the results in the literature in a systematic way to see how far they can be obtained as special cases of the general approach in this paper. Moreover, there are still some conceptual problems which are discussed in 4.1-4.3 below.

4.1 Conceptual problem with semi-PO for composition

According to Def. 2.5 (semi-pushout) the specifications $SP6$ in the semi-PO of (3) in figure 2 includes only translated axioms from $SP5$ (relevant for $I2$) but not from $SP4$ (relevant for $I1$). Similarly $SP7$ in figure 3 lacks translated axioms from $SP5$ (relevant for $I2$). This problem can be solved by redefinition of semi-$PO's$ in Def. 2.5 requiring f' and $h2$ in figure 1 to be specification morphisms. This can be achieved by defining $E3 = g'\#(E1) \cup f'\#(E2)$. The effect of this new definition is that $f2'$ in figure 2 (vertical composition), and $f4$, $f5$ in figure 3 and 5 (direct and general horizontal composition) become specification morphisms. Moreover the translated axioms of $SP2$ in figure 2 are derivable from those of $SP6$, because $f2' \circ m1$ is specification morphism, although $f2$ is only signature-morphism.

4.2 Syntactical representation of composite implementations

Even with a redefinition of Semi-PO as above it might be too restrictive to require that the vertical composition $I2^* I1$ and the (direct) horizontal composition $I2 \circ I1$ have an explicit syntactical representation. This could be avoided by the following semantical redefinition of Def. 2.3 (in the spirit of Sannella-Tarlecki [ST 88]) which is partly included in our paper [EKO 95].

The parameterized specifications h, h' are given by the constructors $K_h : \mathrm{Mod}(SP) \to \mathcal{P}(\mathrm{Mod}(\Sigma 1))$ and $K_{h'} : \mathrm{Mod}(SP) \to \mathcal{P}(\mathrm{Mod}(\Sigma 1'))$, based on $h : \Sigma(SP) \to \Sigma 1$ and $h' : \Sigma(SP) \to \Sigma'$ and the implementation uses a constructor $K_m : \mathrm{Mod}(\Sigma') \to \mathcal{P}(\mathrm{Mod}(\Sigma 2))$ based on $m : \Sigma 1' \to \Sigma 2$.

Correctness: $\forall A \in \mathrm{Mod}(SP) \forall A2 \in K_m \circ K_{h'}(A)$

$$\exists A1 \in K_h(A) \text{ s.t. } U_f(A2) \in \alpha_f(A1) \quad (\text{ or } \alpha_h(A1))$$

In this case all constructions and results are essentially the same. But it might be too restrictive to require that K_m is defined on $\mathrm{Mod}(\Sigma 1')$ and not only on a subclass $\mathrm{Def}(K_m) \subseteq \mathrm{Mod}(\Sigma 1')$ with image $(K_{h'}) \subseteq \mathrm{Def}(K_m)$. This additional correctness condition would have to be shown for all constructions of composite implementations and may lead to additional requirements for K (e.g. closure properties w.r.t. abstractors). In both cases we have to require an extension property for constructors w.r.t. pushouts and semi-PO.

4.3 Abstractors are based on f in (m, f)

For reachability and behavioural abstractors it seems to be more suitable to consider α_h instead of α_f in Def. 2.3 of implementations. This, however, means that we have to redefine the axiomatic properties of abstractors (Def. 3.3) s.t. theorem 3.4-3.8 are still valid. Hence also reformulation of the proofs of these theorems are necessary. Moreover, it has to be checked for which kind of abstractors and constructors these conditions are satisfied. In [EKO 95] we have assumed already that abstractors can be depend on morphisms f and / or h.

Acknowledgements

The author would like to thank the anonymous referee for his/her helpful suggestions. This work has been suppported by ESPRIT W. G COMPASS II (ref. 6112) and by the Spanish CICYT project COSMOS (ref. TIC95-1016-C02-01)

References

[Ehrich 82] Ehrich, H.D., *On the theory of specification, implementation, and parameterization of abstract data types*, J. Assoc. Comp. Mach. **29** (1982), pp. 206-227.

[EG 94] Ehrig, H., Große-Rhode, M., *Functorial theory of parameterized specifications in a general specification framework*, TCS **135** (1994), pp. 221-226.

[EK 83] Ehrig, H., Kreowski, H.-J., *Compatibility of parameter Passing and Implementation of Parameterized Data Types*, TCS Vol 27 N° 3, (1983), pp. 255-286.

[EKO 95] Ehrig, H., Kreowski, H.-J., Orejas, Fernando., *Correctness of Actualization for Parameterized Implementation Concepts based on Constructors and Abstractors*, Bull. EATCS, 56, 1995, pp. 79-85.

[EM 85] Ehrig, H., Mahr, B., *Fundamentals of Algebraic Specification 1. Equations and Initial Semantics*, EATCS Monographs on Theoretical Computer Science, Vol. 6, Springer (1985).

[GB 84] Goguen, J. A., Burstall, R. M., *Introducing Institutions*, Proc. Logics of Progr. Workshop, Springer LNCS n°. 164 (1984), pp. 221-256.

[Hoare 72] Hoare, C.A.R., *Proofs of correctness of data representations*, Acta Informatica 1, (1972), pp. 271-281.

[ONS 93] Orejas, F., Navarro, M., Sánchez, A., *Implementations and behavioural equivalence: a survey*, Springer LNCS **655** (1993), pp. 93-125.

[ONS 96] Orejas F., Navarro, M., Sánchez, A., *Algebraic Implementation of Abstract Data Types: A Survey of Concepts and New Compositionality Results*, Mathematical Structures in Computer Science **6** (1996), 33-67.

[ST 2] Sannella, D. T., Tarlecki, A., *Implementations of Patameterized Specifications*, Springer LNCS n^0. 140 (1982), pp. 473-488.

[ST 88] Sannella, D. T., Tarlecki, A., *Toward formal development of programs from algebraic specifications: Implementations revisited*, Acta Informatica **25** (1988), pp. 233-281.

Technical University of Berlin,
Franklinstraße 28/29,
10587 Berlin
e-mail: ehrig@cs.tu-berlin.de

Universität Bremen,
Postfach 33 04 40,
28334 Bremen
e-mail: kreo@informatik.uni-bremen.de

Universitat Politecnica de Catalunya,
E-08028 Barcelona (Spain)
e-mail: orejas@lsi.upc.es

Recibido: 7 de Mayo de 1996
Revisado: 20 de Enero de 1997