Observational Refinement Process

Alexandre Madeira[1]

Department of Mathematics
University of Aveiro
Aveiro, Portugal

Abstract

In the algebraic specification of software systems, it is desirable to have freedom in the implementation process, namely for the software reuse. In this paper we will discuss two issues in order to achieve this freedom: we study the *observational stepwise refinement process* and we propose an alternative formalization of the refinement concept based on the *logical interpretation* from the *abstract algebraic logic*.
In the first topic, we go beyond the traditional assumption of maintaining the set of observable sorts during the refinement process by the possibility of changing it between the process steps, i.e., we analise the stepwise refinement with encapsulation and desencapsulation of sorts during the process. In the second topic, we suggest a formalization of the refinement concept where an equation may be mapped into a set of equations, against the refinements based on signature morphisms, where an equation is mapped into another one.

Keywords: Algebraic specification, observational equality, observational refinement, refinement via translation.

1 Introduction

The use of mathematical formalisms in the development and verification of software systems has been widely research over the times, being the algebraic specification an important topic of this study. In this context, software objects are viewed as algebras and the computations executed over them seen as terms. The algebraic specification of a software object consists of a signature together with a class of algebras that satisfy the requirements of the system. Algebras in this class are called correct realizations of the specification, and they model the possible programs that satisfy the requirements of the intended system. In the implementation process of a software component, we start with an initial specification of the system, and then we enrich it with implementation decisions in order to get a complete description of the desired program (desired algebra). This gradual process of successive refinements is known as stepwise refinement process (cf. [24,23,17,22]). Clearly, the size of the model class of the initial specification decreases as it is being enriched with new requirements since we progress from a more abstract case to a more concrete one.

[1] Email: madeira@ua.pt

In this work, we use the modeling concept defined according to the standard satisfaction relation, with the *equational logic* as the underlying logic. However, software designed according to the object orientation paradigm requires other tools, more appropriate for this process. In these software systems the data are split in the internal data (or encapsulated) and external data (or desencapsulated): the user has access to encapsulated data only via computations and has direct access to the other ones. On the user's point of view, two elements are considered indistinguishable if they produce the same output over the same computations, and two implementations may be considered as equivalents if they return the same observable result over the same computations. Therefore, this partition induces an adaptation of the modeling concept, in which, a program only needs to satisfy the specification requirements from the outside of the system's point of view, i.e., in its observable behavior. To adequate this paradigm to the algebraic approach to software development, we split the sorts of signature specification: we consider the observable sorts to represent the data which we have direct access, and the non observable sorts, to represent the encapsulated sorts. A computation of observable result is seen as an observable term. In order to achieve a precise semantics for programs with encapsulated data, this approach, named *observational approach*, suggests replacing the strict equality relation by the *observational equality* relation, in which two non observable elements are considered indistinguishable if they have the same observational behavior when executed over the same program of observable result. The study of methods for observational verification of properties can be found, for example, in works of *M. Bidoit and R. Hennicker*, of *J. Goguen, G. Malcolm* and *G. Roşu*, of *A. Bouhoula*, of *R. Diaconescu*, of *K. Futatsugi*, of *P. Padawitz* among others (cf. [1,15,13,12,21,7,9,20]).

The adjustment of the stepwise refinement process to this new perspective has been studied by several authors (cf. [17,14,3]). In all the above mentioned works, it is presumed the observational preservation of sorts between refinement steps, in the sense that, encapsulated data in one determined refinement step, are still encapsulated in the pursuing of the process. However, the change of non observable into observable sorts and vice-versa, can be useful in various situations. Specifically, on the one hand, for security and efficiency reasons in upgrades of protected software, can be necessary, sometimes during the implementation process, to encapsulate some data sorts. On the other hand, to desencapsulate sorts during the refinement process, can be advantageous in the application of proof methods (for example, when we are able to desencapsulate all sorts the observational equality relation can be interpreted by the strict equality relation). An important issue of this topic is how to control the vertical composition of observational refinements made by different observational equality relations, i.e., how to guarantee that the composition of two observational refinements made by different observational equality relations continues being an observational refinement. In the first part of this paper we study the stepwise refinement process, in which changes over the observable sorts are allowed, i.e., the observational stepwise refinement process with the variation of the set of observable sorts. Initially, is examined the data desencapsulation in the refinement process. It is characterized a class of morphisms that desencapsulate data and preserve the property of vertical composition. Let SP' be an observational refinement of SP with respect to a set of observable sorts Obs. Clearly, SP' is an observational refinement with respect to all subsets of Obs. However, the converse it is not true. We present a result that allows build from SP' a specification which is an observable refinement of SP with respect to a smaller relation, namely with respect to the set $Obs \cup \{v\}$. Part of this study is done exclusively for the equational specifications case.

Following the recent works which apply some tools and results of the *abstract algebraic logic* to the specification of software systems (cf.[19]), in the second part of the paper, we suggest an alternative formalization of the refinement concept called *refinement via translation*. This concept is based on the *logical translation* concept, a central entity of the *abstract algebraic logic* (see [5,4,6]). The definition of *translation* appears in [4] formulated for the k-logical systems. In this context, the translations are defined as $(k-l)$-mappings, which translate a k-dimensional logical into another l-dimensional one, over the same signature. A paradigmatic example of a translation of this kind is the translation of the *classical propositional calculus* into the *equational theory of boolean algebras* (cf. [4, Example 4.1.2]). An interesting aspect of the *refinements via translation*, with respect to the implementation freedom, is the fact that in this formalization, a formula may be mapped into a set of formulas, against the formalizations based on the signature morphisms, where a formula is mapped into another one.

We formalize the *refinement via translation* exclusively at the non observable case, i.e., to the case where $Obs = S$. However, the generalization of the concept to the observable case may be done in the natural way.

1.1 Preliminaries

1.1.1 Universal (sorted) Algebra

In this Section, we recall some notions of *universal sorted algebra*. A presentation of these concepts may be found in [24] (or in [8] for the one-sorted case).

1.1.2 Definitions

Let S be a non empty set whose elements are called *sorts*. An *S-sorted set* is a S-indexed family of sets $A = (A_s)_{s \in S}$. We say that a S-sorted set A is *locally finite* if, for any $s \in S$, A_s is a finite set, and we say that A is a *globally finite* if A is locally finite and $A_s = \emptyset$ except for a finite number of sorts. Observe that if S is finite, then *local* implies *global* finiteness.

Definition 1.1 [Multi-sorted binary relation] Let $A = (A_s)_{s \in S}$ be a S-sorted set. A *binary S-relation* $R \subseteq A \times A$ consists of a S-family of relations $R_s \subseteq A_s \times A_s$.

Given an element $a \in A_s$ and an equivalence relation R, we define the *equivalence class of a modulo R* as the set $a/R_s = \{b \in A_s | aR_s b\}$. The *quociente A by R* is the S-sorted set $A/R = (A/R)_{s \in S}$ such that $(A/R)_s = \{a/R_s | a \in A_s\}$.

Definition 1.2 [Signature] A *signature* Σ is a pair (S, Ω), where:

- S is a set (of sorts names);
- Ω is a $(S^* \times S)$-sorted set (of operation names);

where S^* is the set of the finite sequences of S elements.

Example 1.3 [21] Consider a cell of a computer memory where we may write and read values. This software system may be specified using the signature $\Sigma_{CELL} = (S, \Omega)$ with $S = \{\mathtt{elt}, \mathtt{cell}\}$, where `elt` represents a sort of the values to write and `cell` the sort of the cell representation, and $\Omega = \{\mathtt{put}, \mathtt{get}\}$, where `put` and `get` are used to represent write and read functions of a value in a cell:

```
[GEN]
    elt;
    cell;
[OP]
    put: elt,cell -> cell;
    get:cell -> elt;
```

Definition 1.4 [Σ-algebra] Let $\Sigma = (S, \Omega)$ be a signature. A Σ-*algebra* A consists of

- an S-sorted set $A = (A_s)_{s \in S}$, where for all $s \in S$, A_s denotes the carrier set of s.
- for any $f \in \Omega_{s_1 \ldots s_n, s}$, a function $f^A : A_{s_1} \times \cdots \times A_{s_n} \to A_s$;

Example 1.5 [21] Consider an S-sorted set B such that $B_{\mathtt{elt}} = \mathbb{N}$, $B_{\mathtt{cell}} = \mathbb{N}^*$ and the functions:

$$\mathtt{get}^B(\epsilon) = 0;$$
$$\mathtt{get}^B(n\omega) = n;$$
$$\mathtt{put}^B(m, \omega) = m\omega,$$

where ϵ represents the empty list and $\omega \in \mathbb{N}^*$ a list of natural numbers. We have that B is a Σ_{CELL}-algebra.

Definition 1.6 [Congruence relation] Let $\Sigma = (S, \Omega)$ be a signature and A be a Σ-algebra. A Σ-*congruence in* A is an S-family of symmetric, transitive and reflexive non empty relations $\approx^A = (\approx^A_s)_{s \in S}$, such that, for any function $f \in \Omega_{s_1 \ldots s_n, s}$ and for all $a_i, b_i \in A_{s_i}$,$1 \leq i \leq n$ if $a_i \approx^A_{s_i} b_i$, then $f^A(a_1, \ldots, a_n) \approx^A_s f^A(b_1, \ldots, b_n)$.

Definition 1.7 [Quociente Σ-algebra] Let A be a Σ-algebra and $\approx^A$ a Σ-congruence in A. The *quociente of* A *by* $\approx^A$ is the Σ-algebra $A/\approx^A$ defined as follows:

- $(A/\approx^A)_s = A_s/\approx^A$, for all $s \in S$;
- for any $f : s_1, \ldots, s_n \to s \in \Sigma$, and for all $a_1 \in A_{s_1}/\approx^A_{s_1}, \ldots, a_n \in A_{s_n}/\approx^A_{s_n}$, $f^{A/\approx^A}(a_1/\approx^A_{s_1}, \ldots, a_n/\approx^A_{s_n}) = f(a_1, \ldots, a_n)/\approx^A_s$.

Given a signature $\Sigma = (S, \Omega)$, we assume that there is an associate S-family $V = V_{s \in S}$ of pairwise disjoint infinite sets. An element of V_s is called *variable of sort* s, and a S-family $X \subseteq V$ is called *a set of variables for* Σ. It is required that the elements of V and the elements of Ω have different denotations.

Definition 1.8 [$T_\Sigma(X)$] Let Σ be a signature and X a set of variables for Σ. For each $s \in S$, we define $(T_\Sigma(X))_s$, the *set of* Σ-*terms of sort* s, as the smaller set $T_\Sigma(X)$ such that:

- For any $s \in S$ and $x \in X_s$ we have $x \in (T_\Sigma(X))_s$

- If there is a $f :\to s$, then $f \in (T_\Sigma(X))_s$;
- For any $f : s_1, \dots, s_n \to s \in \Sigma$, and for all term $t_i \in T_\Sigma(X)_{s_i}, 1 \leq i \leq n$, we have $f(t_1, \dots, t_n) \in T_\Sigma(X)_s$;

It is well know that $T_\Sigma(X)$ is a Σ-algebra with the operations defined in the usual way [24].

Definition 1.9 [Valuations and interpretations] Let $\Sigma = (S, \Omega)$ be a signature, X be a set of variables for Σ and A be a Σ-algebra. A *valuation* $\alpha : X \to A$ is a S-family of mappings $(\alpha_s : X_s \to A_s)_{s \in S}$. Any valuation α uniquely extends to a Σ-homomorphism $I_\alpha : T_\Sigma(X) \to A$ as follows:

(i) $I_{\alpha_s}(x) =_{def} \alpha_s(x), x \in X_s$;

(ii) for any $f : s_1, \dots, s_n \to s \in \Sigma$, for all $t_i \in T_\Sigma(X)_{s_i}$, $I_{\alpha_s}(f(t_1, \dots, t_n)) =_{def} f^A(I_{\alpha_{s_1}}(t_1), \dots, I_{\alpha_{s_n}}(t_n))$.

The mapping I_α is called the *interpretation induced by* α.

An endomorphism $\sigma : X \to T_\Sigma(X)$ is called a *substitution.* A Σ-*equation* is a triple (X, t, t') where X is a set of variables for Σ, and $t, t' \in T_\Sigma(X)_s$ for some $s \in S$. Usually, we represent a Σ-equation (X, t, t') by $(\forall X).t \approx t'$. A Σ-*conditional equation* has the form $(\forall X).t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$, where $t, t', t_i, t'_i \in T_\Sigma(X)$, $1 \leq i \leq n$. Observe that any Σ-equation may be seen as a Σ-conditional equation without premisses. We denote the set of the Σ-equations by $Eq(\Sigma)$ and the set of the Σ-conditional equations by $CEq(\Sigma)$. We define the *set of formulas over a signature* Σ, in symbols $Fm(\Sigma)$, as the set of the Σ-equations and Σ-conditional equations.

Definition 1.10 [Satisfaction relation] Let Σ be a signature, A be a Σ-algebra, $(\forall X).t \approx t'$ be a Σ-equation and $(\forall X).t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$ be a Σ-conditional equation. The Σ-algebra A *satisfies the* Σ-*equation* $(\forall X).t \approx t'$, in symbols, $A \models (\forall X).t \approx t'$ if for all valuations $\alpha : X \to A$ we have that $I_\alpha(t) = I_\alpha(t')$. The Σ-algebra A *satisfies the* Σ-*conditional equation* $(\forall X).t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$, in symbols, $A \models t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$, if for any $\alpha : X \to A$, $I_\alpha(t_i) = I_\alpha(t'_i)$, $1 \leq i \leq n$ implies $I_\alpha(t) = I_\alpha(t')$.

Given a class of Σ-algebras C, a set of Σ-equations $\{t_i \approx t'_i | i \in I\}$ and a Σ-equation $t \approx t'$, we write $C \models t \approx t'$ when for all $A \in C$, $A \models t \approx t'$ and we write $\{t_i \approx t'_i | i \in I\} \models_A t \approx t'$ when for all valuations $\alpha : X \to A$, $\{I_\alpha(t_i) = I_\alpha(t'_i) | i \in I\}$ implies $I_\alpha(t) = I_\alpha(t')$. When I is finite, we have that $\{t_i \approx t'_i | i \in I\} \models_C t \approx t'$ if and only if $\models_C t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$. The relation $\models_C$ is *finitary* if $\{t_i \approx t'_i | i \in I\} \models t \approx t'$ implies $\{t_i \approx t'_i | i \in J\} \models t \approx t'$ for some finite $J \subseteq I$.

The following proposition states some properties of the relation $\models_C$ which are well known for the one-sorted case (cf. [6]):

Proposition 1.11 *Let* Σ *be a signature and* C *be a class of* Σ*-algebras. Then,*

(i) $\models_C t \approx t$ *for any* $t \approx t \in Eq(\Sigma)$*;*

(ii) $\models_C t \approx t' \to t' \approx t$ *for any* $t \approx t' \in Eq(\Sigma)$*;*

(iii) $\models_C t \approx t' \wedge t' \approx t'' \to t \approx t''$ *for all* $t \approx t', t' \approx t'' \in Eq(\Sigma)$*;*

(iv) $\models_C t_1 \approx t'_1 \wedge \cdots \wedge t_n \approx t'_n \to f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$ *for every appropriated* $f \in \Omega$*;*

Moreover, it can be proved that, if C is the class of Σ-algebras axiomatized by the set of Σ-equations and Σ-conditional equations Φ, then the relation $\models_C$ is finitary (cf.[6] for the one-sorted case). This relation can be seen as the *consequence relation* over the set of Σ-equations (in sense of [5]) considering by the set of Σ-equations of Φ as the axioms in $\models_C$, and considering the Σ-conditional equations in Φ as the inference rules of $\models_C$.

Proposition 1.12 [6] *Let Φ be a set of Σ-equations and Σ-conditional equations, and C be the class of Σ-algebras axiomatized by Φ. We have that $\Phi \models_C t \approx t'$ if and only if, there is a finite sequence of equations $t_1 \approx t'_1, \ldots, t_n \approx t'_n$ such that $t_n \approx t'_n$ is $t \approx t'$ and for every $i = 1, \ldots, n$ one of the following conditions holds:*

(i) $t_i \approx t'_i \in \Phi$*;*

(ii) *there is a $\phi \approx \phi' \in \Phi$ and a substitution σ such that $\sigma(\phi \approx \phi')$ is $t_i \approx t'_i$;*

(iii) *there is a conditional equation $\phi_1 \approx \phi'_1 \wedge \cdots \wedge \phi_n \approx \phi'_n \to \phi \approx \phi' \in \Phi$, and a substitution σ such that $t_i \approx t'_i$ is $\sigma(\phi \approx \phi')$ and $\{\sigma(\phi_i \approx \phi'_i) | i < n\} \subseteq \{t_j \approx t'_j | j < i\}$.*

Definition 1.13 [Signature morphism] Let $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$ be signatures. A *signature morphism* $\sigma : \Sigma \to \Sigma'$, is a pair $\sigma = (\sigma_{\text{sort}}, \sigma_{\text{op}})$, where $\sigma_{\text{sorts}} : S \to S'$ and $\sigma_{\text{op}} : \Omega \to \Omega'$, is a family of functions respecting the sorts of operations names in Ω, that is, $\sigma_{op} = (\sigma_{\omega,s} : \Omega_{\omega,s} \to \Omega'_{\sigma^*_{sorts}(\omega), \sigma_{sorts}(s)})_{\omega \in S^*, s \in S}$ (where for $\omega = s_1 \ldots s_n \in S^*, \sigma^*_{sorts}(\omega) = \sigma_{sorts}(s_1) \ldots \sigma_{sorts}(s_n)$).

Definition 1.14 [Reduct Algebra] Let A' be a Σ'-algebra, and $\sigma : \Sigma \to \Sigma'$ be a signature morphism. The *σ-reduct* of A' is the Σ-algebra $A' \restriction_\sigma$ defined as follows:

- for any $s \in S$, $(A' \restriction_\sigma)_s = A'_{\sigma(s)}$, and
- for all $f : s_1, \ldots, s_n \to s \in \Sigma$,

$$f^{A' \restriction_\sigma} : A' \restriction_{\sigma s_1} \times \cdots \times A' \restriction_{\sigma s_n} \to A' \restriction_{\sigma s} = \sigma\text{op}(f)^{A'} : A'_{\sigma_{gen}(s_1)} \times \cdots \times A'_{\sigma_{gen}(s_n)} \to A'_{\sigma_{gen}(s)}.$$

Let $\sigma : \Sigma \to \Sigma'$ be a signature morphism where $\Sigma = (S, \Omega)$, $\Sigma' = (S', \Omega')$. Let V and V' be the families of sets of variables associated with Σ and Σ' respectively. It is assumed that for any $s \in S$, $V_s \subseteq V'_{\sigma(s)}$. Hence, if $X = (X_s)_{s \in S}$ is a set of variables to Σ, then X' is defined to be the following set of variables to Σ': for any $s' \in S'$, $X'_{s'} = \bigcup_{\sigma(s)=s'} X_s$ (cf. [24,17]). By this way, we define in the natural way an extension of σ from $T_\Sigma(X)$ into $T_{\Sigma'}(X')$ (see [24]). Given a equation $t \approx t'$, we write $\sigma(t \approx t')$ for $\sigma(t) \approx \sigma(t')$. For each valuation $\alpha' : X' \to A'$, the *reduct valuation* of α is the valuation $(\alpha' \restriction_\sigma) : X \to A' \restriction_\sigma$, defined by $(\alpha \restriction_\sigma)_s(x : s) = \alpha_{\sigma(s)}(x : \sigma(s))$ (see [24,16]).

Lemma 1.15 (Satisfaction Lemma [11]) *Let Σ, Σ' be signatures, A' be a Σ'-algebra and ϕ be a Σ-equation. Then,*

$$A' \models \sigma(\phi) \text{ iff } A' \restriction_\sigma \models \phi.$$

Lemma 1.16 [16] *Let Σ and Σ' be signatures, $\sigma : \Sigma \to \Sigma'$ be a signature morphism, X be a set of variables for Σ and X' a set of variables for Σ' constructed as bellow. For any valuations $\beta : X' \to A'$ e $\alpha : X \to A' \restriction_\sigma$ such that $\beta \restriction_\sigma = \alpha$, we have $\alpha(t) = \beta(\sigma(t))$.*

1.1.3 Algebraic specification

When we want to specify a software system, we should define an adequate signature, taking account the sorts and functions of the intended system, and we should express the desired functional behaviour of the signature operations, in a given logical system by axioms.

An *algebraic specification* SP is a pair $(\Sigma, Mod(SP))$ where Σ is a signature, denoted by $Sig(SP)$ and $Mod(SP)$ is a class of Σ-algebras. This class of Σ-algebras is called *model class of* SP, and a $Sig(SP)$-algebra of $Mod(SP)$ by *model of* SP. When a formula ϕ is satisfied by all the models of SP, we say that SP satisfies ϕ, and we write $SP \models \phi$. The specifications SP and SP' are semantically equivalents, if $Sig(SP) = Sig(SP')$ and $Mod(SP) = Mod(SP')$. When $Mod(SP)$ is axiomatized by a set Φ of Σ-equations and Σ-conditional equations, we represent the specification $SP = (\Sigma, Mod(SP))$ by the pair $SP = \langle \Sigma, \Phi \rangle$, where $Mod(SP) = \{A \in Alg(\Sigma) | A \models \Phi\}$. When Φ is a set of equations, the specification $SP = \langle \Sigma, \Phi \rangle$ is called an *equational specification.* Given two specifications SP and SP' we define $SP + SP'$ as the specification such that $Sig(SP{+}SP') = Sig(SP) \cup Sig(SP')$ and $Mod(SP{+}SP') = Mod(SP) \cap Mod(SP')$.

Example 1.17 [Adapted from [21]] The following expression specifies the memory cell system of Example 1.3:

```
Spec Cell =
[SORT]
    elt;
    cell;
[OP]
    put: elt,cell -> cell;
    get:cell -> elt;
[AX]
(∀e:elt)(∀c:cell)get(put(e,c)) ≈ e;
```

It is not difficult to see that the Σ_{cell}-algebra defined in Example 1.5 is a model of `CELL`.

1.2 Observational equality

The strict equality relation is often showed as too strong for algebraic specification of software with encapsulated data (Section 1). It is presented, in this section an adaptation of the usual concepts of validity satisfaction, etc., which are more appropriate to the semantic treatment of this kind of systems. As stated, the sorts of a signature are split into observable sorts and non observable sorts. This division is at the base of observational equality. The observable sorts are also known as *visible sorts* and non observable sorts as *hidden sorts.*

As suggested in Section 1, in the observational approach, two elements are considered as observational equal if they are indistinguishable when executed over the same computational experiments. In our framework, these experiments are formalized by *observable contexts*:

Definition 1.18 [Contexts and observable contexts] Let $\Sigma = (S, \Omega)$ be a signature, $Obs \subseteq S$ be a set of observable sorts, X be a set of variables for Σ and $Z = (\{z_s\})_{s \in S}$ be a S-sorted set of singular sets. An *s-context* over Σ is a term $c \in T_\Sigma(X \cup \{z_s\})_{s'}$, where the variable z_s is called *contextual variable of* c. When $s' \in Obs$, an s-context is called an *observable s-context over* Σ *(with respect to Obs)*. $\mathcal{C}_\Sigma(s)$ denotes the set of s-contexts over Σ and o $\mathcal{C}_\Sigma^{Obs}$ denotes the set of observable contexts over Σ.

Given a context c, a set of non contextual variables of c is denoted by $Var(c)$, and $c[t]$ denotes the term obtained by replacing z_s by a term $t \in T_\Sigma(X)_s$.

Definition 1.19 [Contextual equality, Observational equality and Behaviour] Let $\Sigma = (S, \Omega)$be a signature, $Obs \subseteq S$ be a set of observable sorts, $\mathcal{C}$ be an arbitrary set of contexts over Σ and A be a Σ-algebra. Two elements $a, b \in A_s$ are *contextually equal with respect to* $\mathcal{C}$, denoted by $a \approx_{\mathcal{C}}^A b$, if for any context $c \in \mathcal{C}(s)$, for all valuation $\alpha, \beta : X \cup \{Z_s\} \to A$ such that $\alpha(x) = \beta(x)$ for all $x \in X$, and $\alpha(z_s) = a$ and $\beta(z_s) = b$, we have $I_\alpha(c) = I_\beta(c)$. The contextual equality in A with respect to the set $\mathcal{C}_\Sigma^{Obs}$ is called *observational equality* and is denoted by $\approx_{Obs}^A$. The Σ-algebra $A/\approx_{Obs}^A$ is called the *behaviour of* A *with respect to* $\approx_{Obs}$.

Given a class C of Σ-algebras, $C/\approx_{Obs}$ denotes the class of Σ-algebras $\{A/\approx_{Obs}^A |A \in C\}$, and given a specification SP, $SP/\approx_{Obs}$ denotes the class $Mod(SP)/\approx_{Obs}$.

Definition 1.20 [Observational satisfaction relation] Let Σ be a signature, $Obs \subseteq S$ a set of observable sorts, $(\forall X).t \approx t'$ be a Σ-equation and $(\forall X).t_0 \approx t'_0 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$ be a Σ-conditional equation. A Σ-algebra A *observationally satisfies the equation* $(\forall X).t \approx t'$ *with respect to Obs*, in symbols, $A \models_{\approx_{Obs}} (\forall X).t \approx t'$, if for any valuation $\alpha : X \to A$, $I_\alpha(t) \approx_{Obs}^A I_\alpha(t')$. A Σ-algebra A *observationally satisfies the conditional equation* $(\forall X).t_0 \approx t'_0 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$ *with respect to Obs*, in symbols, $A \models_{\approx_{Obs}} (\forall X).t_0 \approx t'_0 \wedge \cdots \wedge t_n \approx t'_n \to t \approx t'$, if for any valuation $\alpha : X \to A$, $I_\alpha(t_i) \approx_{Obs}^A I_\alpha(t'_i)$, $1 \leq i \leq n$, implies $I_\alpha(t) \approx_{Obs}^A I_\alpha(t')$.

Remark 1.21 The previous adaptation (generalization) of the satisfaction relation, is made by replacing the strict equality by the observational equality. However, there are some other works in the literature where this generalization is made at a more abstract level, obtained by replacing the strict equality by an arbitrary parcial congruence relation, called such a context by *behavioural equality* (cf. [15,1]). Observe that this approach includes the one presented here, since the observational equality is a congruence relation (cf. [15, Fact 3.1.8]).

Example 1.22 Let `CELL1` the following specification:

```
Spec CELL1= enrich CELL  by
[AX]
```

(∀e,e':elt)(∀c:cell).put(e,put(e',c))) $\approx$ put(e,c);

We have that the Σ_{CELL}-algebra B (example 1.5) that is not a strict model of `CELL1`, since, given a $\omega \in \mathbb{N}^*$ and `e,e'` $\in \mathbb{N}$ we have that

$$\mathtt{put}^B\mathtt{(e,put}^B\mathtt{(e',}\omega\mathtt{))} \approx \mathtt{ee'}\omega$$

and $\mathtt{e}\omega \approx \mathtt{put}^B\mathtt{(e,}\omega\mathtt{)}$. However, it is not difficult to see that $B \models_{\approx_{Obs}}$ $\mathtt{put(e,put(e',x))} \approx \mathtt{put(e',x)}$, and hence, we have that B is an observable model of `CELL` not being in the strict sense.

Theorem 1.23 [1] *Let Σ be a signature and $Obs \subseteq S$ a set of observable sorts, A be a Σ-algebra and C a class of Σ-algebras. Then:*

(i) *$A \models_{\approx_{Obs}} \phi$ iff $A/\approx^A_{Obs} \models \phi$;*

(ii) *$C \models_{\approx_{Obs}} \phi$ iff $C/\approx_{Obs} \models \phi$;*

where $C\ /\approx = \{A/\approx^A_{Obs} | A \in C\}$.

1.2.1 The observational behaviour operator

Definition 1.24 [Observational behaviour class] Let $\Sigma = (S, \Omega)$ be a signature, $Obs \subseteq S$ be a set of observable sorts and C be a class of Σ-algebras. The *observational behaviour class of C with respect to Obs* is the class

$$Beh_{\approx_{Obs}}(C) =_{def} \{A \in Alg(\Sigma) | A/\approx^A_{Obs} \in C\}.$$

This definition give rise the definition of an important specifications operator: the operator **behaviour.wrt.**. Let $BehEq$ be the class of observational equalities:

- Sintax:
$$\textbf{behaviour.wrt.} : Spec, BehEq \to Spec$$
- Semantics:
$$Sig(\textbf{behaviour } SP \textbf{ wrt} \approx_{obs}) =_{def} Sig(SP)$$
$$\mathrm{Mod}(\textbf{behaviour } SP \textbf{ wrt} \approx_{Obs}) =_{def} Beh_{\approx_{Obs}}(\mathrm{Mod}(SP))$$

Hence, the model class of $\mathrm{Mod}(\textbf{behaviour } SP \textbf{ wrt} \approx)$ is the class of all the Σ-algebras which observable behaviours belongs to $\mathrm{Mod}(SP)$, i.e., the operator **behaviour** SP **wrt** $\approx$ specifies the class of Σ-algebras of the "desired observational behaviours", that is, the "observational correct realizations" of SP.

Theorem 1.25 *Let $SP = \langle \Sigma, \Phi \rangle$ be a specification and Obs a set of observable sorts to Σ. Then:*

$$\mathrm{Mod}(\textbf{\textit{behaviour}}\ SP\ \textbf{\textit{wrt}} \approx_{obs}) = \{A \in Alg(\Sigma) | A \models_{\approx_{Obs}} \Phi\}.$$

Proof. By definition of **behaviour**, we have that for any Σ-algebra A, $A \in \mathrm{Mod}(\textbf{behaviour } SP \textbf{ wrt} \approx)$ if and only if $A/\approx^A_{Obs} \in \mathrm{Mod}(SP)$. Since $SP = \langle \Sigma, \Phi \rangle$, we have that $A/\approx^A_{Obs} \in \mathrm{Mod}(SP)$ if and only if $A/\approx^A_{Obs} \models \Phi$, and therefore, by Theorem 1.23 $A \models_{\approx_{Obs}} \Phi$. □

A specification SP is *observationally closed with respect to Obs* when $Mod(SP) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt} \approx_{Obs})$. Given a signature $\Sigma = (S, \Omega)$ and a set of observable sorts $Obs \subseteq S$, the specification $SP = \langle \Sigma, \Phi \rangle$ is observationally closed with respect to Obs if for any $t_1 : s_1 \approx t'_1 : s_1 \wedge \cdots \wedge t_n : s_n \approx t'_n : s_n \to t : s \approx t' : s \in \Phi$, $s_i \in Obs$ for all $1 \leq i \leq n$. In particular, an equational specification is observationally closed with respect to any $Obs \subseteq S$ (cf. [15]).

2 The Observational Stepwise Refinement Methodology

2.1 Strict refinements

Given a specification SP of a software system, the implementation process consists in constructing a correct realization (a program) of SP, i.e., of constructing an algebra P such that $P \in Mod(SP)$, or at least a class of $Sig(SP)$-algebras SP' such that $Mod(SP') \subseteq Mod(SP)$, small enough for the desired work. Hence, in this process, we enrich SP with implementation decisions, in order to obtain a complete description of the intended program (desired algebra).

The *stepwise refinement process* (see [24,23,17]) is the systematic process by which, from an initial specification SP_0 are successively built more restrictive specifications by introducing of new requisites:

$$SP_0 \rightsquigarrow SP_1 \rightsquigarrow SP_2 \rightsquigarrow \cdots \rightsquigarrow SP_{n-1} \rightsquigarrow SP_n,$$

where for all $1 \leq i \leq n$, $SP_{i-1} \rightsquigarrow SP_i$ is a refinement.

Note that if $SP \rightsquigarrow SP'$ and $SP' \rightsquigarrow SP''$ then $SP \rightsquigarrow SP''$, since $Sig(SP) = Sig(SP') = Sig(SP'')$ and $Mod(SP'') \subseteq Mod(SP') \subseteq Mod(SP)$. This transitivity, named *vertical composition*, assure that $SP_0 \rightsquigarrow SP_n$.

Example 2.1 Consider the specifications `CELL` and `CELL1` of Examples 1.17 and 1.22. We have `CELL` $\rightsquigarrow$ `CELL1`.

As it was mentioned, during the refinement process, the specification to refine is enriched with new requirements, being natural the need to modify the signature of the initial specification, by the introduction of new sorts and functions, renaming, etc.. This can be done by a signature morphism. Based on *Satisfaction Lemma* (Lemma 1.15), we have the following generalization of refinement concept:

Definition 2.2 [σ-Refinement] Let σ be a signature morphism. The specification SP' is a σ-refinement of SP, in symbols $SP \rightsquigarrow_\sigma SP'$, if:

(i) $Sig(SP') = \sigma(Sig(SP))$ and

(ii) $Mod(SP') \upharpoonright_\sigma \subseteq Mod(SP)$,

where $Mod(SP') \upharpoonright_\sigma = \{A \upharpoonright_\sigma | A \in Mod(SP')\}$.

Note that when we consider the identity morphism id, the concept of id-refinement coincide with the refinement concept. Since the composition of two signature morphisms is a signature morphism, we have directly by the *Satisfaction Lemma* (Lemma 1.15), that the vertical composition of this kind of refinements holds. Hence, if $SP_0 \rightsquigarrow_{\sigma_1} SP_1$ and $SP_1 \rightsquigarrow_{\sigma_2} SP_2$ we have $SP_0 \rightsquigarrow_{\sigma_2 \circ \sigma_1} SP_2$, and for the case of stepwise refinement with n steps, we have $SP_0 \rightsquigarrow_{\sigma_n \circ \cdots \circ \sigma_1} SP_n$.

Example 2.3 Suppose that we need to implement a `CELL1` system to use with natural numbers. Firstly, we may translate the `CELL1` specification in this new signature (by the morphism σ):

$$\begin{array}{rlcl} \sigma : & \mathit{Sig}(\texttt{CELL1}) & \rightarrow & \mathit{Sig}(\texttt{CELLNAT}) \\ & \texttt{elt} & \rightarrow & \texttt{nat} \\ & \texttt{cell} & \rightarrow & \texttt{cell} \\ & \texttt{get} & \rightarrow & \texttt{get} \\ & \texttt{put} & \rightarrow & \texttt{put} \\ & & \rightarrow & \texttt{s} \\ & & \rightarrow & \texttt{zero} \end{array}$$

and then, introduce the axiomatic of the natural numbers set in this new specification. Now, we have that

$$\texttt{CELL1} \leadsto_\sigma \texttt{CELLNAT}$$

It follows an important characterization of the σ-refinement concept:

Theorem 2.4 *Let $SP = \langle \Sigma, \Phi \rangle$ and $SP' = \langle \Sigma', \Phi' \rangle$ specifications and $\sigma : \Sigma \rightarrow \Sigma'$ a signature morphism. Then, $SP \leadsto_\sigma SP'$ iff $SP' \models \sigma(\Phi)$.*

Proof. Suppose that $SP \leadsto_\sigma SP'$. Then, for any $A' \in \mathit{Mod(SP')}$, $A' \restriction_\sigma \in \mathit{Mod(SP)}$, i.e., $A' \restriction_\sigma \models \Phi$. Hence, by Lemma 1.15, $A' \models \sigma(\Phi)$. On the other hand, we have that $SP' \models \sigma(\Phi)$, and therefore, for any $A' \in \mathit{Mod(SP')}$, $A' \models \sigma(\Phi)$. By Lemma 1.15 $A' \restriction_\sigma \models \Phi$, and hence, $A' \restriction_\sigma \in \mathit{Mod}(SP)$. Therefore $SP \leadsto_\sigma SP'$. □

2.2 Observational refinements

The relevance of the adjustment of the concepts of refinement and σ-refinement to the observational approach, is evident, since according to this view, the preservation of requirements to refinement, is no longer strict, but just observational. Hence, a refinement is observationally correct when their observational behaviour preserves the requirements of the refined specification:

Definition 2.5 [Observational Refinement] Let SP and SP' be two specifications, Obs a set of observable sorts of $\mathit{Sig}(SP)$ and σ a signature morphism. *SP' is an observational σ-refinement with respect to Obs*, in symbols $SP \leadsto_\sigma^{\approx_{Obs}} SP'$, if

$$\textbf{behaviour } SP \textbf{ wrt } \approx_{\mathit{Obs}} \leadsto_\sigma SP',$$

that is, if:

- $\mathit{Sig}(SP') = \sigma(\mathit{Sig}(\textbf{behaviour } SP \textbf{ wrt } \approx_{\mathit{Obs}}))$ and
- $\mathit{Mod(SP')} \restriction_\sigma \subseteq \mathit{Mod(}\textbf{behaviour } SP \textbf{ wrt } \approx_{\mathit{Obs}}\mathit{)}$.

The adaptation of the stepwise refinement process to the observational case requires some attention, since different observational equalities can be considered in different refinement steps. The main question is how to control the conservation of the vertical composition during the process. Some important steps in this study are already given as, for example, the characterization of sufficient conditions for this conservation.

Given an algebra A and two congruences θ^A, θ'^A in A, we write $\theta^A \leq \theta'^A$ if for any $s \in S$, $(\theta^A)_s \subseteq (\theta'^A)_s$. The relevance of this relation between congruences when we work with observational refinements is quite intuitive. For example, given the observational equalities $\approx_{Obs}$ and $\approx_{Obs'}$, such that $\approx_{Obs} \leq \approx_{Obs'}$, if $SP \rightsquigarrow^{\approx_{Obs}} SP'$ then $SP \rightsquigarrow^{\approx_{Obs'}} SP'$, since the second relation distinguishes fewer elements than the first one. The following theorem, characterizes the sufficient conditions, to the preservation of the vertical composition in the observational refinements:

Theorem 2.6 [15] *Let SP, SP' and SP'' be specifications with $Sig(SP') = \sigma(Sig(SP))$ and $Sig(SP'') = \tau(Sig(SP'))$ and Obs, Obs' be sets of observable sorts to $Sig(SP)$ and $Sig(SP')$ such that, for any $Sig(SP')$-algebra A', $(\approx'^{A'}_{Obs'}) \restriction_\sigma \leq \approx^{(A' \restriction_\sigma)}_{Obs}$. If $SP \rightsquigarrow^{\approx_{Obs}}_{\sigma} SP'$ and $SP' \rightsquigarrow^{\approx_{Obs'}}_{\phi} SP''$, then $SP \rightsquigarrow^{\approx_{Obs}}_{\phi \circ \sigma} SP''$.*

This result is presented in [15] at the context of the *behavioural equalities* (see Remark 1.21).

We study in this paper the observational case of stepwise refinement process, with special attention to the case where it is possible to vary the set of observable sorts between process steps. The characterization of the observational stepwise refinement process present in literature, supposes the "preservation of observability" of the specifications between refinement steps, in the sense that, given two specifications SP, SP' with $Obs \subseteq S$ and $Obs' \subseteq S'$ sets of observable sorts to $Sig(SP)$ and $Sig(SP')$ such that $SP \rightsquigarrow^{\approx_{Obs}}_{\sigma} SP'$ then, for any $s \in Obs$ we have $\sigma(s) \in Obs'$ and for any $s \in S \setminus Obs$ we have $\sigma(s) \in S' \setminus Obs'$, or at least $\sigma(Obs) \subseteq Obs'$ (see [17]). However, changing observable sorts into non observable and vice-versa, can be useful in several situations. For example, according to the object oriented paradigm, only input/output data must be desencapsulated, and by security reasons (data and code) can be necessary encapsulate some types of data in a determined phase of the implementation process. On the other hand, desencapsulate data during the refinement process, can be advantageous in the verification tasks (for example, if it is possible to desencapsulate all the sorts, we may interpret the observational equality relation by the strict equality relation).

Let $\Sigma = (S, \Omega)$ a signature and $Obs \subseteq S$ and $Obs' \subseteq S$ sets of observable sorts such that $Obs \subseteq Obs'$. Observe that the relation $\approx_{Obs'}$ is more restrictive than the relation $\approx_{Obs}$ since $\mathcal{C}^{Obs}_{\Sigma} \subseteq \mathcal{C}^{Obs'}_{\Sigma}$ and hence, in the definition of the first relation we considere less contexts than the seconde one. By Definition 1.20, all the models of an equational specification SP by the relation $\models_{\approx_{Obs'}}$ also they are by relation $\models_{\approx_{Obs}}$, i.e.,

$$Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs'}) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}).$$

The progressive sort desencapsulation $Obs_1 \subseteq \cdots \subseteq Obs_n$, induce a relations chain $\approx_{Obs_1} \geq \approx_{Obs_2} \geq \cdots \geq \approx_{Obs_n}$ where for any $i, j \leq n$ such that $i \leq j$, if $a \approx_{Obs_j} b$ then $a \approx_{Obs_i} b$. It is understood of this form that the data desencapsulation "preserve" the formation of contexts, arriving thus at a first characterization of the vertical composition of the observational refinement steps with data desencapsulation: by Theorem 2.6, if we have $SP \rightsquigarrow^{\approx}_{Obs} SP'$ and $SP \rightsquigarrow^{\approx}_{Obs'} SP'$ with $Obs \subseteq Obs'$, then $SP \rightsquigarrow^{\approx}_{Obs} SP''$. Now, we will analise this preservation to the general case of the σ-refinements. The next definition characterizes a class of morphisms that assure this preservation:

Definition 2.7 [Observational morphism] Let $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$ be signatures, $Obs \subseteq S$ and $Obs' \subseteq S'$ be sets of observable sorts for Σ and Σ', and $\sigma : \Sigma \to \Sigma'$ be a signature morphism. The morphism σ is said to be an *$Obs - Obs'$−observational morphism* if for all $s \in S$, $s \in Obs$ implies $\sigma(s) \in Obs'$.

Theorem 2.8 *Let $\sigma : \Sigma \to \Sigma'$ be an $Obs - Obs'$-observational morphism and A' be a Σ'-algebra. Then,*

$$(\approx^{A'}_{Obs'}) \restriction_\sigma \leq \approx^{(A'\restriction_\sigma)}_{Obs} .$$

Proof.

Let $a, b \in A' \restriction_\sigma$ such that $a(\approx^{A'}_{Obs'})\restriction_\sigma b$. Since $a(\approx^{A'}_{Obs'})b$, we have that for any $c' \in \mathcal{C}^{Obs'}_{\Sigma'}$, for any valuations $\alpha'_1, \alpha'_2 : X' \cup \{Z_{\sigma(s)}\} \to A'$, such that $\alpha'_1(x') = \alpha'_2(x')$ for all $x' \in X'$ and $\alpha'_1(z_{\sigma(s)}) = a$ and $\alpha'_2(z_{\sigma(s)}) = b$,

(1) $$I_{\alpha'_1}(c') = I_{\alpha'_2}(c').$$

Since σ is an $Obs - Obs'$-observational morphism, we have that all the contexts of $\mathcal{C}^{Obs}_{\Sigma}$ are mapped by σ into contexts of $\mathcal{C}^{Obs'}_{\Sigma'}$, and hence, all the contexts considered in $\approx^{A'\restriction_\sigma}_{Obs}$, also they are in $\approx^{A'}_{Obs'}$ (by reduct algebra definition $c^{A'\restriction_\sigma} = \sigma(c)^{A'}$). By 1, we have in particular that for any $c \in \mathcal{C}^{Obs}_{\Sigma}$, $I_{\alpha'_1}(c) = I_{\alpha'_2}(c)$.Consider now the reduct valuations $\alpha'_1 \restriction_\sigma : X \to A' \restriction_\sigma$ and $\alpha'_2 \restriction_\sigma : X \to A' \restriction_\sigma$. By Lemma 1.16 we have that

$$\alpha'_1 \restriction_\sigma (x_s) = \alpha'_1(\sigma(x_s)) = \alpha'_1(x_{\sigma(s)}),$$
$$\alpha'_2 \restriction_\sigma (x_s) = \alpha'_2(\sigma(x_s)) = \alpha'_2(x_{\sigma(s)}),$$
$$\alpha'_1 \restriction_\sigma (z_s) = \alpha'_1(\sigma(z_s)) = \alpha'_1(z_{\sigma(s)})$$

and

$$\alpha'_2 \restriction_\sigma (z_s) = \alpha'_2(\sigma(z_s)) = \alpha'_2(z_{\sigma(s)}).$$

We have also that $\alpha'_1(z_{\sigma(s)}) = a$ and $\alpha'_2(z_{\sigma(s)}) = b$, and therefore, by unicity of I, we have that for all $c \in \mathcal{C}^{Obs}_{\Sigma}$, $I_{\alpha'_1\restriction_\sigma}(c) = I_{\alpha'_1}(c)$ and $I_{\alpha'_2\restriction_\sigma}(c) = I_{\alpha'_2}(c)$, i.e., for all valuations $\alpha'_1, \alpha'_2 : X' \to A'$, for any context $c \in \mathcal{C}^{Obs}_{\Sigma}$,

(2) $$I_{(\alpha'_1\restriction_\sigma)}(c) = I_{(\alpha'_2\restriction_\sigma)}(c).$$

On the other hand, for any valuation $\alpha : X \to A' \restriction_\sigma$, there is an valuation $\alpha' : X' \to A'$ such that $\alpha = \alpha' \restriction_\sigma$ (all valuations $\alpha_s = \alpha'_{\sigma(s)}$), and therefore, we have by 2 that for all valuations $\alpha_1, \alpha_2 : X' \to A' \restriction_\sigma$, such that $\alpha_1(x') = \alpha_2(x')$, if $x' \in X'$, $\alpha_1(z_{\sigma(s)}) = a$, $\alpha_2(z_{\sigma(s)}) = b$, then, for any $c \in \mathcal{C}^{Obs}_{\Sigma}$, we have $I_{\alpha_1}(c) = I_{\alpha_2}(c)$, i.e., $a(\approx^{A'\restriction_\sigma}_{Obs})b$. □

Thus, we arrive at the following characterization of vertical composition of observational refinements:

Corollary 2.9 *Let SP, SP' and SP'' be three specifications, σ be an $Obs - Obs'$-observable morphism and ϕ be an $Obs' - Obs''$-observable morphism. If $SP \rightsquigarrow^{\approx_{Obs}}_{\sigma} SP'$ and $SP' \rightsquigarrow^{\approx_{Obs'}}_{\phi} SP''$, then $SP \rightsquigarrow^{\approx_{Obs}}_{\phi\circ\sigma} SP''$.*

Proof. Immediate by Theorems 2.6 and 2.8. □

The introduction of the concepts presented here was made in such a way as to finding sufficiency conditions for the preservation of the vertical composition property between refinement steps. However, this composition is only assured when processed via the observational equality relation considered in the first refinement step, and that be such that $\approx_{Obs_1} \geq \cdots \geq \approx_{Obs_n}$ (i.e., such that $Obs_1 \subseteq \cdots \subseteq Obs_n$). Hence, in the refinement process

$$SP_0 \leadsto_{\sigma_1}^{\approx_{Obs_1}} SP_1 \leadsto_{\sigma_2}^{\approx_{Obs_2}} \cdots \leadsto_{\sigma_n}^{Obs_n} SP_n,$$

such that $\approx_{Obs_1} \geq \cdots \geq \approx_{Obs_n}$, we have that $SP_0 \leadsto_{\sigma_n \circ \cdots \circ \sigma_1}^{\approx_{Obs_1}} SP_n$.

The study of other characterizations of vertical composition can be interesting as, for example, according to the presented characterization, all desencapsulated sorts during the refinement steps become encapsulated at the end of the process. By the reasons mentioned above, the possibility of vertically composed observational refinements, according to the relation with more observable sorts appears often as a desirable situation. This is the characterization that we want to do next. Observe that for any equational specification $SP = \langle \Sigma, \Phi \rangle$, and for any $s \in Obs$, the relation $\approx_{Obs}$ is more restrictive than the relation $\approx_{Obs \setminus \{s\}}$, i.e., $\approx_{Obs} \leq \approx_{Obs \setminus \{s\}}$. Hence, since $A \models_{\approx_{Obs}} \Phi \Rightarrow A \models_{\approx_{Obs \setminus \{s\}}} \Phi$, we have that

$$Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \setminus \{s\}}).$$

However, it is obvious that the reciprocal does not holds. To guarantee some kind of reciprocal we have to impose some conditions about Φ. For example, if $\Phi_s = \emptyset$ for some $s \in S$ and $\Phi_{S \setminus Obs} = \emptyset$, we have $A \models_{\approx_{Obs}} \Phi \Leftrightarrow A \models_{\approx_{Obs \setminus \{s\}}} \Phi$, and hence $Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) = Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \setminus \{s\}})$. In these conditions, if we have two observational refinements $SP \leadsto_{\sigma}^{\approx_{Obs \setminus \{s\}}} SP'$ and $SP' \leadsto_{\tau}^{\approx_{Obs}} SP''$ with $SP = \langle \Sigma, \Phi \rangle$, we can compose them obtaining the observational refinement $SP \leadsto_{\tau \circ \sigma}^{\approx_{Obs}} SP''$.

Consider now the following result:

Lemma 2.10 *Let $\Sigma = (S, \Omega)$ be a signature, Obs a set of observable sorts for Σ and Φ be a set of Σ-equations. Then, for any Σ-algebra A and for any $s \in Obs$,*

$$A \models_{\approx_{Obs \cup \{v\}}} \Phi \textit{ iff } (A \models \Phi' \textit{ and } A \models_{\approx_{Obs}} \Phi)$$

where

$$\Phi' = \Phi_v \cup \{c(t) = c(t') \,|\, t \approx t' \in \Phi_h, h \in S \setminus (Obs \cup \{v\}), c \in \mathcal{C}_{\Sigma}^{\{v\}}(h)\}.$$

Proof. Suppose that $A \models_{\approx_{Obs \cup \{v\}}} \Phi$. Since $A \models_{\approx_{Obs \cup \{v\}}} \Phi_{Obs \cup \{v\}}$ implies $A \models \Phi_{Obs \cup \{v\}}$ we have that

(3) $$A \models_{\approx_{Obs}} \Phi_{Obs}$$

(4) $$A \models \Phi_v.$$

From (4) $A \models_{\approx_{Obs}} \Phi_v$. Since for any $h \in S \setminus (Obs \cup \{v\})$ $\mathcal{C}_{\Sigma}^{Obs}(h) \subseteq \mathcal{C}_{\Sigma}^{Obs \cup v}$ we have that $A \models_{\approx_{Obs}} \Phi_h$ for all $h \in S \setminus (Obs \cup \{v\})$. Therefore, $A \models_{\approx_{Obs}} \Phi$.

Let now $t \approx t' \in \Phi_h$, $h \in S \setminus (Obs \cup \{v\})$. Then, by hypothesis $A \models c(t) \approx c(t')$ for any $c \in \mathcal{C}_\Sigma^{Obs \cup v}$ (since $\mathcal{C}_\Sigma^v \subseteq \mathcal{C}_\Sigma^{Obs \cup \{v\}}$). From this together with (4) , $A \models_{\approx_{Obs}} \Phi'$.

Now, suppose that $A \models_{\approx Obs \cup \{v\}} \Phi$ and $A \models \Phi'$. Let $t \approx t \in \Phi_s$. We split the proof in three cases: (i) $s \in Obs$, (ii) $s = v$ and (iii) $s \in S \setminus Obs \cup \{v\}$. In the first case it is obvious, since $A \models_{\approx_{Obs}} t \approx t'$ implies that $A \models t \approx t'$ and hence $A \models_{\approx_{Obs \cup \{v\}}} t \approx t'$. In case (ii), $A \models_{\approx_{Obs \cup \{v\}}} t \approx t'$ since by hypothesis $A \models t \approx t'$. In the latter case we have just to see that $\mathcal{C}_\Sigma^{Obs \cup \{v\}}(s) = \mathcal{C}_\Sigma^{Obs}(s) \cup \mathcal{C}_\Sigma^{\{v\}}(s)$. In fact, from $A \models \Phi'$ we have $A \models c(t) \approx c(t')$ for any $c \in \mathcal{C}_\Sigma^{\{v\}}(s)$, and $A \models c(t) \approx c(t')$ for any $c \in \mathcal{C}_\Sigma^{Obs}(s)$ because $A \models_{\approx_{Obs}} t \approx t'$. Therefore $A \models_{\approx_{Obs \cup \{v\}}} t \approx t'$. □

Given an equational specification $SP = \langle \Sigma, \Phi \rangle$ we define the specifications SP_v as the equational specification $\langle \Sigma, \Phi' \rangle$ with Φ' defined as in Lemma 2.10. From the previous Lemma we have

$$Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \cup \{v\}}) = Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs} + SP_v)$$

By the following Theorem, from an observational refinement with respect to $\approx_{Obs}$ of an equational specification SP, we can build an observational refinement with respect to $\approx_{Obs \cup s}$ of the same SP. This result can be worth in the reuse perspective: suppose that we have an observational refinement of SP with respect to a relation $\approx_{Obs}$ and, by some reason, we need to output the data of sort v which at this moment it is not an observable sort. On the other hand, the result may be useful when, during the specification process, we have not decided yet if a sort whether or not it is an encapsulated sort. This idea is stated in the following Theorem:

Theorem 2.11 *Let Φ be a set of Σ-equations and $SP = \langle \Sigma, \Phi \rangle$ and SP' be two specifications such that $SP \rightsquigarrow^{\approx_{Obs}} SP'$. Then*

$$SP \rightsquigarrow^{\approx_{Obs \cup \{s\}}} SP' + SP_s.$$

Proof. By assumption we have that $Mod(SP) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs})$ and hence $Mod(SP) \cap Mod(SP_s) \subseteq Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) \cap Mod(SP_s)$. By Theorem 2.10 we have that $Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs}) \cap Mod(SP_s) = Mod(\textbf{behaviour } SP \textbf{ wrt } \approx_{Obs \cup \{s\}})$ and therefore $SP \rightsquigarrow^{\approx_{Obs \cup \{s\}}} SP' + SP_s$. □

Remark 2.12 Note that the definition of Φ' does not depend of the observable equations of Φ. Hence, we may extend this result to another sets of formulas. For example, this result holds for all the specifications $SP = \langle \Sigma, \Phi \rangle$ where Φ is a set of Σ-equations and observable Σ-conditional equations with observable premisses.

Example 2.13 Consider the specification `CELL1` of Example 1.22 and the following specification:

```
    Spec CELL_cell=
[GEN]
     elt;
     cell;
[OP]
```

```
    put: elt,cell -> cell;
    get:cell -> elt;
[Ax]
(∀e,e':elt)(∀c:cell).put(e,put(e',c))) ≈ put(e,c);
```

Now, for any refinement `CELL1` $\rightsquigarrow^{\approx_{\{elt\}}} SP$, we have that `CELL1` $\rightsquigarrow^{\approx_{\{elt,cell\}}} SP +$ $\texttt{CEll1}_{cell}$.

Example 2.14 [NatCell] Consider the specification `CELLNAT` from Example 2.3 with `NAT` axiomatized by `(∀x:nat).s(p(x)) ≈ x`. Consider too the following specification:

```
Spec CELLNATBOOL = enrich CELLNAT by BOOL

Spec CELLNATEQ = enrich CELLNATBOOL by
[OP]
    eq:nat,nat ->bool;
[AX]
(∀x:nat).eq(x,x) ≈ true;
(∀x,y:nat).eq(x,y) ≈ true  ⇒  eq(y,x) ≈ true;
(∀x,y,z:nat).eq(x,y) ≈ true  ∧  eq(y,z) ≈ true  ⇒  eq(x,z)  ≈  true;
(∀x:nat).eq(s(p(x)),x) ≈ true;
```

where `BOOL` represents the classical specification of the boolean algebras. Suppose that we have an observational refinement `CELLNATEQ` $\rightsquigarrow^{\approx_{\{bool\}}} SP$ and we need an observational refinement of `CELLNATEQ` with respect to $\approx_{\{bool,nat\}}$. Then, we define the specification:

`Spec` $\texttt{CELLNATEQ}_{nat}$ = `enrich CELLNAT+BOOL by`

```
[OP]
    eq:nat,nat ->bool;
[AX]
(∀x:nat).s(p(x)) ≈ x;
(∀e,e':nat)(∀c:cell).get(put(e,put(e',c))) ≈ get(put(e,c));
(∀e,e',e'':nat)(∀c:cell).get(put(e'',put(e,put(e'',c))))
                                          ≈ get(put(e'',put(e,c)));
⋮
```

and we build the desirable refinement by $SP + \texttt{CELLNATEQ}_{nat}$ (cf. Remark 2.12).

Observe that this technic sometimes has to be followed with some complementary methods, since that in the general case, the set $\mathcal{C}_{\Sigma}^{\{c\}}(h)$ is infinite and consequently SP_s is infinitary (such as in Example above). However, in most of the cases it is possible to consider only a finite set of contexts $\mathcal{C}_{\Sigma} \subseteq \mathcal{C}_{\Sigma}^{\{s\}}$ instead of the set $\mathcal{C}_{\Sigma}^{\{s\}}$, inducing this way the formation of finite specifications SP_s (cf. [1,12,19]).

3 Refinements via translation

In this Section, we look over the refinement process in a new perspective greatly influenced by *abstract algebraic logic*. We believe that the study of this formalization may be important, for example, from the point of view of software reuse. This approach is based on the notion of *logical translation*, which is a central concept considered in the abstract algebraic theory of deductive systems (see [5,4,6]). Since the presentation of this topic requires a strong notation, we formalize the *refinement via translation* concept exclusively in the non observable case, i.e., in the case where $Obs = S$. The concept generalization to the observable case can be done in the natural way.

In the sequel we formalize the notion of *logical translation* for the sorted case. Intuitively, a translation is a mapping from the set of equations into their power set: given the signatures $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$ such that $\Sigma \subseteq \Sigma'$, a $\Sigma - \Sigma'$-*translation* τ is a family indexed by S $(\tau_s(x : s, y : s))_{s \in S}$ where for each $s \in S$, $\tau_s(x : s, y : s) = (\tau_{s,s'}(x : s, y : x))_{s' \in S'}$ is a globally finite S'-sorted set of Σ'-equations $\phi(x, y) \approx \psi(x, y)$ of sort s' in two variables of sort s. Given a $\Sigma - \Sigma'$-translation τ, we define the τ-translation of a Σ-equation $t \approx t'$ of sort s, denoted by $\tau(t \approx t')$, as the S'-sorted set of Σ'-equations $(\tau_{s,s'}(t, t'))_{s' \in S'}$, and a τ-translation of a conditional equation $t_1 : s_1 \approx t'_1 : s'_1 \wedge \cdots \wedge t_n : s_n \approx t'_n : s'_n \to t : s \approx t' : s$ as the S'-sorted set of conditional Σ'-equations defined for each $s' \in S'$ and for each $\phi(t, t') \approx \psi(t, t') \in \tau_{s,s'}(t : s \approx t' : s)$ as follows:

$$\bigwedge_{i \leq n} (\bigwedge_{s \in S'} \tau_{s_i,s}(t_i : s_i \approx t'_i : s_i)) \to \phi(t, t') \approx \psi(t, t').$$

In the sequel, we identify a S-sorted set $(\tau_s(t : s, t' : s'))_{s \in S}$ with the disjoint union $\biguplus_{s \in S} \tau_s(t : s, t' : s)$.

Observe that a formula ϕ is translated by a signature morphism σ in another formula $\sigma(\phi)$; however, a logical translation maps a formula into a set of formulas. An useful tool in the sequel is given by the following Lemma:

Lemma 3.1 *Let $\sigma : Fm(\Sigma) \to Fm(\Sigma)$ be a substitution, τ be a $\Sigma - \Sigma'$-translation and ξ be a Σ-equation. We have that $\tau(\sigma(\xi)) = \sigma(\tau(\xi))$.*

Proof. Given a Σ-equation $t \approx t'$ of sort s and a $\Sigma - \Sigma'$-translation τ, we have that for any $s' \in S'$, $\tau_{s,s'}(t \approx t')$ is defined as a set of equations $\phi(t, t') \approx \psi(t, t')$ of sort s'. Hence, for any substitution $\sigma : Fm(\Sigma) \to Fm(\Sigma)$, $\sigma(\phi(t, t') \approx \psi(t, t')) = \sigma(\phi(t, t')) \approx \sigma(\psi(t, t')) = \phi(\sigma(t), \sigma(t')) \approx \psi(\sigma(t), \sigma(t'))$. On the other hand, we have that $\sigma(t \approx t') = \sigma(t) \approx \sigma(t')$, and therefore, for any $s' \in S'$, $\tau_{s,s'}(\sigma(t \approx t'))$ is defined as a set of equations $\phi(\sigma(t), \sigma(t')) \approx \psi(\sigma(t), \sigma(t'))$ of sort s'. Therefore, for any equation ξ, $\tau(\sigma(\xi)) = \sigma(\tau((\xi))$. The case of the Σ-conditional equations follows directly to the previous case. □

Definition 3.2 [Interpretation] Let SP be a specification and τ be a $Sig(SP) - \Sigma'$-translation. We say that τ *interprets* SP if there is a specification SP' with signature Σ' such that, for any $\xi \in Fm(\Sigma)$, $SP \models \xi$ if and only if $SP' \models \tau(\xi)$. In this case we say that the SP' *is a τ-interpretation of* SP.

Definition 3.3 [τ-model] Let SP be a specification, Σ' be a signature and τ be a $Sig(SP) - \Sigma'$-translation. A Σ'-algebra A' is a τ-model of SP if for any $\xi \in Fm(\Sigma)$, $SP \models \xi$ implies $A' \models \tau(\xi)$. We define the *τ-model class of SP*, denoted by $Mod_\tau(SP)$, as the class of all τ-models of SP.

Given a specification SP and a $Sig(SP) - \Sigma'$-translation τ, we define SP^τ as the specification such that $Sig(SP^\tau) = \Sigma'$ and $Mod(SP^\tau) = Mod_\tau(SP)$.

Theorem 3.4 *Let SP be a specification and τ be a $Sig(SP) - \Sigma'$-translation. If τ interprets SP, then the specification SP^τ is the τ-interpretation of SP with the largest class of models.*

Proof. Since τ interprets SP then, there is a specification SP' such that for any $\xi \in Fm(Sig(SP))$, $SP \models \xi$ if and only if $SP' \models \tau(\xi)$. on the one hand, we have that $SP \models \xi$ implies that $SP^\tau \models \tau(\xi)$, since by definition, $Mod(SP^\tau) = \{A|A$ is a τ-model of $SP\}$. On the other hand, if $SP^\tau \models \tau(\xi)$ then $SP' \models \tau(\xi)$ (since $Mod(SP') \subseteq Mod(SP'))$ and hence $SP \models \xi$. Therefore, SP^τ is a τ-interpretation of SP. Obviously, it is the largest one, since they include all the τ-models of SP.

□

Theorem 3.5 *Let $SP = \langle \Sigma, \Phi \rangle$ be a specification and τ be a $\Sigma - \Sigma'$-translation. Then, if τ interprets SP, we have that the specification SP^τ is axiomatized by the set of axioms $\tau(\Phi)$, i.e, $SP^\tau = \langle \Sigma', \tau(\Phi) \rangle$. Moreover, if Φ is finite then SP^τ is finitely axiomatized.*

Proof. On the one hand, we have that for any $A' \in Mod(SP^\tau)$ and for any Σ-formula ξ , $SP \models \xi$ implies $SP^\tau \models \tau(\xi)$. In particular, since $SP \models \Phi$, we have that $SP^\tau \models \tau(\Phi)$ and hence, $Mod(SP^\tau) \subseteq Mod(\langle \Sigma', \tau(\Phi) \rangle)$.

On the other hand, consider a Σ-algebra $A' \in \langle \Sigma', \tau(\Phi) \rangle$, i.e., such that $A' \models \tau(\Phi)$, and a equation ξ such that $SP \models \xi$, i.e., $\Phi \models_{SP} \xi$. Then, there is a finite sequence of equations $t_1 \approx t'_1, \ldots, t_n \approx t'_n$, such that $t_n \approx t'_n$ is ξ and that, for all $i = 1, \ldots, n$, $t_i \approx t'_i$ satisfies one of the three conditions of the Proposition 1.12. If $\xi \in \Phi$, then, $\tau(\xi) \in \tau(\Phi)$ and therefore $\tau(\Phi) \models_{A'} \tau(\xi)$ (by (i) of Proposition 1.12)). If ξ is $\sigma(\phi \approx \phi')$ for some substitution σ and some $\phi \approx \phi' \in \Phi$, we have that $\tau(\phi \approx \phi') \in \tau(\Phi)$, by Lemma 3.1, we have that $\tau(\xi)$ is $\sigma(\tau(\phi \approx \phi'))$ and therefore $\tau(\Phi) \models_{A'} \tau(\xi)$ (by (ii) of Proposition 1.12)). Consider now the case where there is a conditional equation $\phi_1 \approx \phi'_1 \wedge \cdots \wedge \phi_n \approx \phi'_n \to \phi \approx \phi' \in \Phi$, and a substitution σ such that $t \approx t'$ is $\sigma(\phi \approx \phi')$ and $\{\sigma(\phi_i \approx \phi'_i)|i < n\} \subseteq \{t_j \approx t'_j|j < i\}$. Hence, we have that $\{\tau(\sigma(\phi_i \approx \phi'_i))|i < n\} \subseteq \{\tau(t_j \approx t'_j)|j < i\}$, and by Lemma 3.1, $\{\sigma(\tau(\phi_i \approx \phi'_i))|i < n\} \subseteq \{\tau(t_j \approx t'_j)|j < i\}$. We have too that $\tau(\phi_1 \approx \phi'_1 \wedge \cdots \wedge \phi_n \approx \phi'_n \to \phi \approx \phi') \subseteq \tau(\Phi)$, i.e., for any $\psi(\phi : s, \phi' : s) \approx \psi'(\phi : s, \phi' : s) \in \tau_{s,s'}(\phi : s \approx \phi' : s)$,

$$(\bigwedge_{i \leq n}(\bigwedge_{s \in S'} \tau_{s_i,s}(\phi_i : s_i \approx \phi'_i : s_i)) \to \psi(\phi : s, \phi' : s) \approx \psi'(\phi : s, \phi' : s))_{s' \in S'} \subseteq \tau(\Phi).$$

Let $\bigwedge_{i \leq n}(\bigwedge_{s \in S'} \tau_{s_i,s}(\phi_i : s_i \approx \phi'_i : s_i)) \to \psi(\phi : s, \phi' : s) \approx \psi'(\phi : s, \phi' : s) \in \tau(\Phi)$ be one of these conditional equations. Since, by hypotheses $t \approx t'$ is $\sigma(\phi \approx \phi')$ we have that $\tau(t \approx t')$ is $\tau(\sigma(\phi \approx \phi'))$, and therefore, $\tau(t \approx t')$ is $\sigma(\tau(\phi \approx \phi'))$. Hence, there is an equation $\mu(t, t') \approx \mu'(t, t') \in \tau(t \approx t')$ such that $\mu(t, t') \approx \mu'(t, t')$ is

$\sigma(\psi(\phi : s, \phi' : s) \approx \psi'(\phi : s, \phi' : s))$, and hence, $A' \models \bigwedge_{i \leq n}(\bigwedge_{s \in S'} \tau_{s_i,s}(\phi_i : s_i \approx \phi'_i : s_i)) \rightarrow \psi(\phi : s, \phi' : s) \approx \psi'(\phi : s, \phi' : s)$ (by (ii) of Proposition 1.12). Therefore $\tau(\Phi) \models_{A'} \tau(t \approx t')$. Since A' is arbitrary, we have that any $A' \in \langle \Sigma', \tau(\Phi) \rangle$ is a τ-model of SP and, therefore, $Mod(\langle \Sigma', \tau(\Phi) \rangle) \subseteq SP^\tau$. Hence $\langle \Sigma', \tau(\Phi) \rangle = SP^\tau$.

Since τ maps each $\phi \in \Phi$ in a finite set $\tau(\phi)$ (by hypotheses, τ is globally finite), we have that, if Φ is finite, then $\tau(\Phi)$ is finite too, and therefore, under these conditions, SP^τ is finitely axiomatized. □

Definition 3.6 [Refinement via translation] Let SP and SP' be two specifications and τ be a translation such that τ interprets SP. We say that the specification SP' *refines via the translation* τ *the specification* SP, in symbols $SP \rightharpoondown_\tau SP'$, if $SP^\tau \rightsquigarrow SP'$.

Lemma 2.4 motivates the next characterization of the refinements via translation concept in the specifications axiomatized by a set of equations and conditional equations:

Theorem 3.7 *Let $SP = \langle \Sigma, \Phi \rangle$ and τ be a $\Sigma - \Sigma'$-translation. If τ interprets SP, then, for any specification SP' with $Sig(SP') = \Sigma'$, we have that $SP \rightharpoondown_\tau SP'$ iff $SP' \models \tau(\Phi)$.*

Proof. By Theorem 3.5 we have that $SP^\tau = \langle \Sigma', \tau(\Phi) \rangle$, and by Lemma 2.4, $SP^\tau \rightsquigarrow SP'$ if and only if $SP' \models \tau(\Phi)$. □

Corollary 3.8 *Let $SP = \langle \Sigma, \Phi \rangle$ be a specification and τ be a $\Sigma - \Sigma'$-translation. A Σ'-algebra A' is a τ-model of SP if $A' \models \tau(\Phi)$.*

Example 3.9 Consider the following specification of the natural numbers set:

```
Spec NAT =
[SORT]
    nat;
[OP]
    s:nat ->nat;
[AX]
```

$(\forall \mathtt{x,y:nat}).\mathtt{s(x)} \approx \mathtt{s(y)} \Rightarrow \ \mathtt{x} \approx \mathtt{y};$

and consider the following specification `NATEQ`:

```
Spec NATEQ =enrich BOOL by
[SORT]
    nat;
    s:nat->nat;
    eq:nat,nat->bool;
[OP]
    s:nat ->nat;
    eq:nat,nat ->bool;
[AX]
```

(i) $(\forall \mathtt{x:nat}).\mathtt{eq(x,x)} \approx \mathtt{true};$

(ii) $(\forall \mathtt{x,y:nat}).\mathtt{eq(x,y)} \approx \mathtt{true} \ \Rightarrow \ \mathtt{eq(y,x)} \approx \mathtt{true};$

(iii) $(\forall$x,y,z:nat$)$.eq(x,y) $\approx$ true $\wedge$ eq(y,z) $\approx$ true $\Rightarrow$ eq(x,z) $\approx$ true;

(iv) $(\forall$x,y:nat$)$.eq(x,y) $\approx$ true $\Rightarrow$ eq(s(x),s(y)) $\approx$ true;

(v) $(\forall$x,y:nat$)$.eq(s(x),s(y)) $\approx$ true $\Rightarrow$ eq(x,y) $\approx$ true;

Consider the translation τ such that

$$\tau_{nat,bool}(\texttt{x:nat} \approx \texttt{y:nat}) = \{\texttt{eq(x:nat,y:nat)} \approx \texttt{true}\}$$

and $\tau_{s,s'} = \emptyset$ in another cases. It is not difficult to see that NATEQ interprets NAT by τ: first note that for any equation t $\approx$ t' such that NAT $\models$ t $\approx$ t', we have that NATEQ $\models$ eq(t,t') $\approx$ true, since the translation of the proof of NAT $\models$ t $\approx$ t' (in sense of Theorem 1.12) is a proof of NATEQ $\models$ eq(t,t') $\approx$ true. The converse may be verified by induction on the length of the proof of NATEQ $\models$ eq(t,t') $\approx$ true.

For example, if eq(t,t') $\approx$ true is obtained by the conditional equation (ii) then, supposing that NATEQ $\models$ eq(t',t) $\approx$ true implies NAT $\models$ t' $\approx$ t, we have that NAT $\models$ t $\approx$ t'. Therefore, we have that NAT $\rightharpoondown_\tau$ NATEQ, since NATEQ $\models$ eq(s(x),s(y)) $\approx$ true $\Rightarrow$ eq(x,y) $\approx$ true.

Note that we can translate a specification of the natural numbers into another one, axiomatized exclusively by equations of sort bool. This fact may be important, for example, if we would like to encapsulate the sort nat.

As stated, in order to use the stepwise refinement methodology in the specification process, is needed that the vertical composition is present. Bellow, we will characterize the composition of translations:

Definition 3.10 Let τ be a $\Sigma - \Sigma'$-translation and ρ be a $\Sigma' - \Sigma''$-translation. We define $\rho.\tau$ as the follows S-family: for each $s \in S$,

$$\rho.\tau_{s,s''}(x:s, y:s) = \biguplus_{s' \in S'} \rho_{s',s''}(\tau_{s,s'}(x:s, y:s)).$$

It is not difficult to see that $\rho.\tau$ is a translation, since it is a S-family of globally finite S''-sorted sets of Σ''-equations with two variables (since by hypotheses ρ and τ are also globally finite).

Theorem 3.11 *Let τ be a $Sig(SP) - \Sigma$-translation that interprets SP and ρ be a $\Sigma - \Sigma'$-translation that interprets SP^τ. Then, the translation $\rho.\tau$ interprets SP.*

Proof. On the one hand, since τ interprets SP, there is a specification SP' such that, for any $\xi \in Fm(Sig(SP))$ $SP \models \xi$ iff $SP' \models \tau(\xi)$, and by Theorem 3.4, we have that $SP \models \xi$ iff $SP^\tau \models \tau(\xi)$. On the other hand, since ρ interprets SP^τ, then there is a specification SP'' such that for any $\psi \in Fm(\Sigma)$, $SP^\tau \models \psi$ iff $SP'' \models \rho(\psi)$. In particular, for any $\xi \in Fm(Sig(SP))$ $SP^\tau \models \tau(\xi)$ iff $SP'' \models \rho(\tau(\xi))$, i.e., $SP'' \models \rho.\tau(\xi)$. Therefore, the $Sig(SP) - \Sigma''$-translation $\rho.\tau$ interprets SP. □

4 Conclusions and future works

In this paper we presented, some formalizations of the stepwise refinement process, namely the case where it is required the preservation of the signature during the process, the case where the specifications of the refinements may differ via signature morphisms and the generalization of this process to the observational paradigm. In the latter case, we characterized the vertical composition of refinements using possibly different observational equalities, i.e., we allowed the encapsulation and desencapsulation of data sorts during the refinement process (Section 2.2). In this context may be worth to extend a refinement calculus like the system presented in [2] with rules for encapsulation and desencapsulation of sorts in the refinement steps.

In Section 3 we introduced the concept of *refinement by translation* and we used some tools from *abstract algebraic logic* to develop an introductory theory of these kind of refinements. We intend to develop these results and analise how they can be generalized to the observational case. An interesting topic in this research is the integration of this kind of refinements in stepwise refinement process present in the Sections 2.1 and 2.2. A first step in this study can be done by the generalization of the *refinement via translation* in the "mixed" case. Here, given a signature morphism $\sigma : \Sigma \to \Sigma'$ and a $\Sigma - \Sigma'$-translation τ that interprets SP, $SP \rightharpoondown_{\tau \rhd \sigma} SP'$ iff $SP^{\tau} \rightsquigarrow_{\sigma} SP'$. Another interesting topic in this study is the *equivalence between specifications* in the perspective of the logical translations. We believe that this work can be done based on the *equivalence of deductive systems*, more precisely via *interpretations* in the sense of [4]. There is another notion of logical translation, called *conservative translation* introduced by *H. Feitosa* and *I. D'Ottaviano* in [10]. It will be interesting to investigate this notion within the refinement process.

5 Acknowledgements

I would like to thank Manuel António Martins and Enrique Hernandez Manfredini for the fruitful discussions concerning this work. I am also grateful to my colleagues Jacinta Poças and Nilde Barreto for their careful reading of the paper manuscript.

References

[1] Bidoit, M., and Hennicker, R. Behavioural theories and the proof of behavioural properties. *Theor. Comput. Sci. 165*, 1 (1996), 3–55.

[2] Bidoit, M., and Hennicker, R. Proving behavioral refinements of col-specifications. In *Essays Dedicated to Joseph A. Goguen* (2006), pp. 333–354.

[3] Bidoit, M., Hennicker, R., and Kurz, A. Observational logic, constructor-based logic, and their duality. *Theor. Comput. Sci. 298*, 3 (2003), 471–510.

[4] Blok, W., and Pigozzi, D. Abstract algebraic logic and the deduction theorem. Preprint. To appear in the Bulletin of Symbolic Logic. Available at `http://www.math.iastate.edu/dpigozzi/papers/aaldedth.pdf`.

[5] Blok, W., and Pigozzi, D. Algebraizable logics. *Memoirs of the American Mathematical Society 396, Amer. Math. Soc., Providence* (1989).

[6] Blok, W. J., and Rebagliato, J. Algebraic semantics for deductive systems. *Studia Logica 74*, 1-2 (2003), 153–180.

[7] Bouhoula, A., and Rusinowitch, M. Observational proofs by rewriting. *Theor. Comput. Sci. 275*, 1-2 (2002), 675–698.

[8] Burris, S., and Sankappanavar, H. P. *A course in universal algebra.* Graduate Texts in Mathematics, Vol. 78. New York - Heidelberg Berlin: Springer-Verlag., 1981.

[9] Diaconescu, R., and Futatsugi, K. Behavioural coherence in object-oriented algebraic specification. *J. UCS 6*, 1 (2000), 74–96.

[10] Feitosa, H. A., and D'Ottaviano, I. M. L. Conservative translations. *Ann. Pure Appl. Logic 108*, 1-3 (2001), 205–227.

[11] Goguen, J., and Burstall, R. Institutions: abstract model theory for specification and programming. *J. ACM 39*, 1 (1992), 95–146.

[12] Goguen, J., and Malcolm, G. Hidden coinduction: Behavioural correctness proofs for objects. *Math. Struct. Comput. Sci. 9*, 3 (1999), 287–319.

[13] Goguen, J., and Malcolm, G. A hidden agenda. *Theor. Comput. Sci. 245*, 1 (2000), 55–101.

[14] Goguen, J., and Roşu, G. Hiding more of hidden algebra. In *Wing, Jeannette M. et al. (ed.), FM '99. Formal methods. World congress on Formal methods in the development of computing systems. Toulouse, France, September 20-24. Proceedings.* 1999.

[15] Hennicker, R. Structural specifications with behavioural operators: semantics, proof methods and applications, 1997. Habilitationsschrift.

[16] Loeckx, J., Ehrich, H.-D., and Wolf, M. Algebraic specification of abstract data types. In *Handbook of logic in computer science, Vol. 5*, Oxford Sci. Publ. Oxford Univ. Press, New York, 2000, pp. 217–316.

[17] Martins, M. A. Behavioral institutions and refinements in generalized hidden logics. *Journal of Universal Computer Science 12*, 8 (2006), 1020–1049.

[18] Martins, M. A. Closure properties for the class of behavioral models. *Theor. Comput. Sci. 379*, 1-2 (2007), 53–83.

[19] Martins, M. A., and Pigozzi, D. Behavioural reasoning for conditional equations. *Mathematical. Structures in Comp. Sci. 17*, 5 (2007), 1075–1113.

[20] Padawitz, P. Swinging types=functions+relations+transition systems. *Theor. Comput. Sci. 243*, 1-2 (2000), 93–165.

[21] Roşu, G. *Hidden Logic.* PhD thesis, University of California, San Diego, 2000.

[22] Sannella, D. Algebraic specification and program development by stepwise refinement. (Extended abstract). In *Bossi, Annalisa (ed.), Logic-based program synthesis and transformation. 9th international workshop, LOPSTR '99. Venice, Italy, September 22-24, 1999. Selected papers. Berlin: Springer. Lect. Notes Comput. Sci. 1817.* 2000, pp. 1–9.

[23] Sannella, D., and Tarlecki, A. Essential concepts of algebraic specification and program development. *Formal Asp. Comput. 9*, 3 (1997), 229–269.

[24] Sannella, D., and Tarlecki, A. *Foundations of Algebraic Specifications and Formal Program Development.* Cambridge University Press, To appear.