A co-semidecision procedure for behavioral equivalence

Vicent Palasí

Report LSI-95-55-R

A co-semidecision procedure for behavioral equivalence

Vicent-Ramon Palasí Lallana.

Barcelona, November 23, 1995.

Abstract

We shall demonstrate that proving the behavioral equivalence of two algebraic specifications is equivalent to proving a set of theorems in a given initial algebra. Thus, it is possible to prove automatically this behavioral equivalence by use of automatic deduction techniques.

1 Introduction.

Both in theory and in practice, the concept of equivalence between two modules is a matter of great importance. Intuitively, this notion could be defined by saying that two modules are equivalent if "they perform the same task" and, therefore, can be interchanged in any software system. The importance attached to this notion is further increased by the growing success of object-oriented programming. As a matter of fact, all the software in a firm which follows the object-oriented philosophy is nothing more than a huge collection of classes. It is, then, a matter of vital importance to know which of these classes are equivalent in order not to increase the size of the collection with redundant information.

In the domain of algebraic specification, this general concept becomes definite in the equivalence of algebraic specifications. Each algebraic specification could be considered as the definition of a certain set of modules. At this point the problem posed is that of the equivalence between algebraic specifications, that is to say, the fact of knowing when the sets of modules which have been defined for various specifications are equivalent. Unfortunately, in the field of algebraic specification the word "equivalence" has several different meanings.

The most widely-used meaning is that of equivalence based on initial semantics. Two specifications would be considered as equivalent if their initial algebras were the same. If we examine this approach at all, we shall clearly realize its shortcomings, particularly for practical purposes. By way of example, the classic algebraic specification of a stack would not be equivalent to another specification in which the stack is defined from an array and a pointer. This fact appears as curious, for the usual implementation of a stack is, as a general rule, precisely that of array + pointer.

The problem in question is that initial semantics are excessively restrictive. Several alternative semantics to this one have been suggested. One of them, which is called "final semantics", is not good enough to solve the problem, either. In particular, the two specifications of battery previously mentioned would not be considered as equivalent by this semantics, either.

Fortunately, there is actually a solution to this problem. The way is to define the equivalence between two specifications out of behavioral semantics ([SaW83]), [SaT85], [Rei81], [Rei84], though here we shall be using the approach of [Niv87]). According to this semantics, that which defines a specification is regarded as a "blackbox" and, consequently, the sorts (that is to say, the types) defined in it can be divided into observable and non-observable ones. Non-observable sorts are not visible in themselves from outside but only through operations which return observable results.

As regards this approach, two algebraic specifications would be equivalent if, for all the computations possible, the "observable consequences" of the computations on the two specifications were the same. Logically, we shall not be interested in the non-observable consequences, since they are not visible.

The purpose of this paper is to prove that, in minimum reasonable conditions, this concept of equivalence can be automatically proved. In other words, that proving that two specifications are behaviorally equivalent is the same as proving a few theorems by using automatic deduction techniques.

Let us be more specific. If we want to reduce the behavioral equivalence between two specifications to automatic theorem proof, we are faced with the problem of there being very little research available on automatic deduction with several different specifications. In order to sort this problem out, we have decided on reducing the problem to one single specification. That is to say, out of two specifications SPEC1 and SPEC2 (of which we are interested in examining their behavioral equivalence) we shall build a new specification SPEC4, which we shall call reunion of SPEC1 and SPEC2, and which contains all the information which was included in those two. Therefore, proving the behavioral equivalence between SPEC1 and SPEC2 is reduced to proving a few given theorems on SPEC4.

The conditions required for this to be workable is that the observable sorts must be already defined when we define the non-observable ones and that there is no equation which has only variables on its right-hand side. We are here producing an article which removes the need for these conditions.

The structure of this paper is as follows: in section 2 we define the basic concepts concerning algebraic specification and behavioral semantics which are necessary in order to follow the reasoning. In sections 3 to 10, we introduce the concepts and theorems which are needed to prove what is intended in this paper. Finally, in section 11, we make a summary of the applications of what has been proved and we expound our views on the future lines of research.

Experts in algebraic specification may skip over section 2, though it would be advisable for them to look at it so as to become acquainted with the notation. As far as the demonstrations contained in this article are concerned, they have been classified, according to the degree of importance, into sublemmas, lemmas and theorems. Thus, the reader is able to choose the degree of detail he prefers in order to read the article. Those readers who should wish to go through all the reasoning will examine all the demonstrations, and half-interested readers will be able to focus on the lemmas and theorems, whereas those interested in great results may only read the latter. Finally, those who are not interested in formal demonstrations but in the conclusions that can be drawn from them are advised to directly read the last section.

2 Basic notions.

In this section, we state the notions that are necessary to understand the rest of the paper. We describe some basic definitions on algebraic specification and behavioral semantics. For the sake of uniformity in the notation, we have chosen to express all these definitions in behavioral theory terms, though most of them are not exclusive to this theory but are rather general results over algebraic specification.

Definition. A S-set C is a family of sets indexed by S, $C = \{C_s\}_{s \in S}$.

Definition. A behavioral signature Σ is a triple $\Sigma = (Obs, S, F)$ where S is a set whose members are called sorts, Obs is a subset of S whose members are called observable sorts and F is a $S^* \times S$-set $F = \{F_{w,s}\}_{(w,s) \in S^* \times S}$.

If $\sigma \in F_{w,s}$, where $w = w1 \times ... \times wn$ with $w1, ..., wn, s \in S$, we say that σ is a function symbol with domains w1,...,wn and sort s. We refer to this by either $\sigma \in F_{w1..wn,s}$ or $\sigma : w1 \times ... \times wn \longrightarrow s$.

S is called $sorts(\Sigma)$ and F is called $opns(\Sigma)$. Variables of sort s are called vars(s).

Definition. Let Σ=(Obs,S,F) be a behavioral signature and X a set of variables. The sets $T_{\Sigma_s}(X)$ are defined in the following way:

- If $x \in X$ and $x \in vars(s)$, then $x \in T_{\Sigma_s}(X)$.

- If $\sigma \in F_{\lambda,s}$, then $\sigma \in T_{\Sigma_s}(X)$.
- If $\sigma \in F_{w_1 \ldots w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, \ldots, t_n \in T_{\Sigma_{w_n}}$, then $\sigma(t_1, \ldots, t_n) \in T_{\Sigma_s}$.

The members of $T_{\Sigma_s}(X)$ are called *terms of sort s*.

Definition. Let Σ be a behavioral signature. We define $T_\Sigma(X) = \{T_{\Sigma_s}\}_{s \in S}$ and $T_\Sigma = T_\Sigma(\emptyset)$. The members of $T_\Sigma(X)$ are called *terms* and those of T_Σ *ground terms*.

Definition. Let Σ=(Obs,S,F) be a behavioral signature. A Σ-algebra A is a tuple (A_S, A_F) where $A_S = \{A_s\}_{s \in S}$ and $A_F = \{\sigma_A\}_{\sigma \in F}$ so that:

- if $\sigma \in F_{\lambda,s}$, then $\sigma_A \in A_s$.
- if $\sigma \in F_{w_1..w_n,s}$, then $\sigma_A : A_{s_1}, \ldots, A_{s_n} \longrightarrow A_s$.

σ_A is called interpretation of function symbol σ in A. We call $A_{Obs} = \{A_s\}_{s \in Obs}$.

Definition. Let A be a Σ-algebra. A computation over A is a term of $T_\Sigma(A_{Obs})$. An observable computation is a term t of $T_\Sigma(A_{Obs})_s$, with $s \in Obs$.

Definition. Let t be a computation over a Σ-algebra A. We define $\varepsilon_A(t)$ in the following way:

- If $t \in A_{Obs}$, then $\varepsilon_A(t) = t$.
- If $\sigma \in F_{\lambda,s}$ where $s \in S$, then $\varepsilon_A(\sigma) = \sigma_A$.
- If $\sigma \in F_{w_1 \ldots w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, \ldots, t_n \in T_{\Sigma_{w_n}}$, then $\varepsilon_A(\sigma(t_1, \ldots, t_n)) = \sigma_A(\varepsilon_A(t_1), \ldots, \varepsilon_A(t_n))$.

$\varepsilon_A(t)$ is called evaluation of t in A.

Definition. We say that a Σ-algebra is finitely generated if the evaluation $\varepsilon_A : T_\Sigma \longrightarrow A$ is exhaustive.

Comment. Notice that we only consider the evaluation in A over ground terms in this definition. That is to say, A is finitely generated if $\forall\, a \in A, \quad \exists\, t \in T_\Sigma$ such that $\varepsilon_A(t) = a$.

Definition. Let $\Sigma = (Obs, S, F, E)$ be a behavioral signature. Let X be a set of variables. We refer by Σ-equation with arity n (or by Σ-equation with n conditions) to a (2*n+3)-tuple $(X, c_1, d_1, \ldots, c_n, d_n, t_1, t_2)$, where $t_1, t_2 \in T_{\Sigma_s}(X)$; $c_1, d_1 \in T_{\Sigma_{s_1}}(X)$; ...; $c_n, d_n \in T_{\Sigma_{s_n}}(X)$, with $s, s_1, s_n \in S$.

Whenever we write $c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ (or, also, $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$), we shall mean the equation $e = (X, c_1, d_1, \ldots, c_n, d_n, t_1, t_2)$. The set of variables of an equation e will be called $vars(e)$.

A Σ-equation with arity 0 is called *simple or unconditional equation.* A Σ-equation e, in which it is fulfilled that $p_1, p_2 \in vars(e)$, is called *equation which has only variables on its right-hand side.*

Lemma. If A is finitely generated, for each assignment v, there is an application $w : X \longrightarrow T_\Sigma$ such that $v = \varepsilon_A \circ v$. In this paper, when we deal with finitely generated algebras, we use indifferently the name "assignment of values" to refer either to v or to w.

Definition. Given an assignment of values v, (where v may be of the two kinds which we have earlier said), and given a term $t \in T_\Sigma(X)$, we define $v^*(t)$ as follows:

- If $t \in X$, then $v^*(t) = v(t)$.
- If t has the form $\sigma(t_1, ..., t_n)$, with n *geq* 0, then $v^*(t) = \sigma(v^*(t_1), ..., v^*(t_n))$.

Definition. We say that a Σ-algebra A satisfies an equation $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ if $\forall v : vars(e) \longrightarrow A$ it is fulfilled that: $\varepsilon_A(v^*(c_1)) = \varepsilon_A(v^*(d_1)) \wedge ... \wedge \varepsilon_A(v^*(c_n)) = \varepsilon_A(v^*(d_n))$ implies $\varepsilon_A(v^*(t_1)) = \varepsilon_A(v^*(t_2))$. If a Σ-algebra satisfies an equation e, we shall write $A \models e$

Lemma. If A is finitely generated, the last definition is equivalent to the following one: $\forall w : vars(e) \longrightarrow T_\Sigma$ it is fulfilled that: $\varepsilon_A(w^*(c_1)) = \varepsilon_A(w^*(d_1)) \wedge ... \wedge \varepsilon_A(w^*(c_n)) = \varepsilon_A(w^*(d_n))$ implies $\varepsilon_A(w^*(t_1)) = \varepsilon_A(w^*(t_2))$.

Definition. A behavioral specification is a 4-tuple SPEC=(Obs,S,F,E), where Σ=(Obs,S,F) is a behavioral signature and E is a set of Σ-equations. We define *sig*(*SPEC*) = Σ and *eqns*(*SPEC*)=E. We refer by $T_{\Sigma_{SPEC}}$ to T_Σ if $\Sigma = sig(SPEC)$. Likewise, we define $T_{\Sigma_{SPEC}}(X)$ and $(T_{\Sigma_{SPEC}})_s$. We refer by $(T_{\Sigma_{SPEC}})_{Obs}$ to $\{t \mid t \in (T_{\Sigma_{SPEC}})_s \wedge s \in Obs\}$.

Definition. Given a specification SPEC=(Obs,S,F,E), we refer by $\equiv_{SPEC}$ to the congruence being defined by SPEC. That is to say, $\equiv_{SPEC}$ is defined as follows:

1. $\forall t \in T_{\Sigma_{SPEC}}$ it is fulfilled $t \equiv_{SPEC} t$.
2. $\forall t, u \in T_{\Sigma_{SPEC}}$ $t \equiv_{SPEC} u$ implies $u \equiv_{SPEC} t$.
3. $\forall t, u, v \in T_{\Sigma_{SPEC}}$ $t \equiv_{SPEC} u \wedge u \equiv_{SPEC} v$ implies $t \equiv_{SPEC} v$.
4. $\forall e : (c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2) \in E$. $\forall v : vars(e) \longrightarrow T_\Sigma$ it is fulfilled that $v^*(c_1) \equiv_{SPEC} v^*(d_1) \wedge ... \wedge v^*(c_n) \equiv_{SPEC} v^*(d_n)$ implies $v^*(p_1) \equiv_{SPEC} v^*(p_2)$

Comment. Since $\equiv_{SPEC}$ is a congruence, the following property is fulfilled:

$\forall\, \sigma \in F_{w1...wn,s}$, with $w1, ..., wn, s \in S$, $\forall\, s_1, t_1 \in (T_{\Sigma_{SPEC}})_{w1}, ..., s_n, t_n \in (T_{\Sigma_{SPEC}})_{wn}$
If $s_1 \equiv_{SPEC} t_1, ..., s_n \equiv_{SPEC} t_n$ then $\sigma(s_1, ..., s_n) \equiv_{SPEC} \sigma(t_1, ..., t_n)$

This property will be widely used in next proofs and we shall call it "property of congruence".

Definition. Suppose a behavioral specification SPEC. We divide the set $T_{\Sigma_{SPEC}}$ into classes of equivalence defined by relationship $\equiv_{SPEC}$. We refer to the class of equivalence which contains the ground term t as $[t]_{\equiv_{SPEC}}$.

Definition. Given a behavioral specification SPEC with sig(SPEC)=Σ, we refer by quotient term algebra of this specification to the Σ-algebra which is defined as follows:

- $A_S = \{A_s\}_{s \in S}$, where $A_s = \{[t]_{\equiv_{SPEC}}\}$, with $t \in (T_{\Sigma_{SPEC}})_s$.
- $A_F = \{\sigma_A\}$ where $\sigma_A([t_1]_{\equiv_{SPEC}}, ..., [t_n]_{\equiv_{SPEC}}) = [s]_{\equiv_{SPEC}}$ if and only if $\sigma(t_1, ..., t_n) \equiv_{SPEC} s$.

Definition. Let SPEC be a behavioral specification and A an algebra. We say that A is initial w.r.t. SPEC if it is isomorphic to the quotient term algebra of SPEC. In this paper, we shall use the symbol T_{SPEC} to refer indifferently either to the quotient term algebra or to any initial algebra.

Lemma. Given a behavioral specification *SPEC*, it can be proved that $\forall\, t, u \in T_{\Sigma_{SPEC}}$ it is fulfilled that $\varepsilon_{T_{SPEC}}(t) = \varepsilon_{T_{SPEC}}(u)$ if and only if $t \equiv_{SPEC} u$ if and only if $[t]_{\equiv_{SPEC}} = [u]_{\equiv_{SPEC}}$.

Definition. We refer by $(T_{SPEC})_{Obs}$ to $\{[t]_{\equiv_{SPEC}} \mid t \in (T_{\Sigma_{SPEC}})_{Obs}\}$.

Definition. Given a behavioral specification SPEC=(Obs,S,F,E), we refer by $Tot_{SPEC}(X)$ to the set of terms such that all their subterms are observable. That is to say, $t \in Tot_{SPEC}(X)$ if:

- t is a variable of sort s, where s $\in$ Obs.
- t has the form $\sigma(t_1, ..., t_n)$, where $\sigma \in F_{w_1...wn,s}$; $w_1, ..., w_n, s \in Obs$ and $t_1, ..., tn \in Tot_{SPEC}(X)$.

Obviously, $Tot_{SPEC}(X)$ is a subset of $T_\Sigma(X)$. We call the members of $Tot_{SPEC}(X)$ totally observable terms.

Definition. Given a behavioral specification SPEC, we refer by $(T_{SPEC})^{Tot}_{Obs}$ to $\{[t]_{\equiv_{SPEC}} \mid t \in Tot_{SPEC}\}$.

Definition. Given a behavioral specification SPEC=(Obs,S,F,E), we refer by E_{Obs} to the set of equations $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ such that $c_1, d_1, ..., c_n, d_n, t_1, t_2 \in Tot_{SPEC}(vars(e))$.

Definition. Let Σ=(Obs,S,F) be a behavioral signature. Let also be two Σ-algebras A and B. A behavioral morphism is a function f between A_{Obs} and B_{Obs} such that for each observable computation t over A it is fulfilled that: $f(\varepsilon_A(t)) = \varepsilon_B(f^*(t))$.
If this function is bijective, we call it "behavioral isomorphism".

Definition. Let Σ be a behavioral signature. Let also be two Σ-algebras A and B. We say that A and B are behaviorally equivalent if there is a behavioral isomorphism between A and B.

3 Eval-equivalence.

In this section, we shall prove that the notion of behavioral equivalence between two algebras, which has been defined out of behavioral isomorphisms, may be defined out of the interpretations of the ground terms in the algebras, if these ones are finitely generated.

This introduces a new notion, the "eval-equivalence", which will be used in our demonstrations and which will be the same thing as behavioral equivalence, in the case of working with finitely generated algebras.

In this section, we assume that Σ is a behavioral signature of the form Σ=(Obs,S,F).

Definition *.Prop1. Given two Σ-algebras A and B, we say that they are eval-equivalent if there is a bijection φ between A_{Obs} and B_{Obs} such that:

$$\forall t \in T_{\Sigma_s},\ with\ s \in Obs, \quad \varphi(\varepsilon_A(t)) = \varepsilon_B(t)$$

Sublemma *.Prop2. If a Σ-algebra is finitely generated, then for each observable computation t, there is a ground term g with the same interpretation in A. That is to say,

$$\forall t \in T_\Sigma(A_{Obs})_s,\ with\ s \in Obs, \quad \exists g \in T_\Sigma\ such\ that\ \varepsilon_A(t) = \varepsilon_A(g)$$

Proof. We shall prove this by structural induction.

- Induction base. Suppose that t is a constant. We have two possible subcases.
  - $t \in T_\Sigma$. In this case, g is t and the lemma is fulfilled.

  - $t \in A_{Obs}$. In this case, since A is finitely generated, there is a $c \in T_\Sigma$ such that $\varepsilon_A(c) = t$. On the other hand, by definition of ε_A, $\varepsilon_A(t) = t$. Therefore, $\varepsilon_A(c) = \varepsilon_A(g)$. Then, if we make g be c, we shall have that $\varepsilon_A(t) = \varepsilon_A(g)$. This is what we wished to prove.

- Induction step. Suppose that t has the form $f(t_1, ..., t_n)$, where $f \in opns(\Sigma)$ and $t_1, ..., t_n \in T_\Sigma(A_{Obs})_s$, with $s \in Obs$. Then, by definition of ε_A, we have:

  $$\varepsilon_A(f(t_1, ..., t_n)) = f_A(\varepsilon_A(t_1), ..., \varepsilon_A(t_n))$$

  Via the hypothesis of induction, there are $g_1, ..., g_n \in T_\Sigma$ such that $\varepsilon_A(g_1) = \varepsilon_A(t_1), ..., \varepsilon_A(g_n) = \varepsilon_A(t_n)$. Therefore,

  $$f_A(\varepsilon_A(t_1), ..., \varepsilon_A(t_n)) = f_A(\varepsilon_A(g_1), ..., \varepsilon_A(g_n))$$

  And, by definition of ε_A,

  $$f_A(\varepsilon_A(g_1), ..., \varepsilon_A(g_n)) = \varepsilon_A(f(g_1, ..., g_n))$$

  By making all the previous expressions equal, we have

  $$\varepsilon_A(t) = \varepsilon_A(f(g_1, ..., g_n))$$

  Where $f(g_1, ..., g_n) \in T_\Sigma$. So, if we define g as $f(g_1, ..., g_n)$, we have what we wished to prove. □

Sublemma *.Prop3. Given two Σ-algebras A and B. We define $f : A_{Obs} \longrightarrow B_{Obs}$ in the following way: $\forall\, a \in A_{Obs}$, $f(a) = \varepsilon_B(g)$ where $g \in T_\Sigma$ such that $\varepsilon_A(g) = a$. Out of f, we define $f^* : T_\Sigma(A_{Obs}) \longrightarrow T_\Sigma(B_{Obs})$ as:

- If $t \in A_{Obs}$, then $f^*(t) = f(t)$.
- If t has the form $\sigma(t_1, ..., t_n)$, where n ≥ 0, then $f^*(t) = \sigma(f^*(t_1), .., f^*(t_n))$.

Then, it is fulfilled that:

- If $t \in T_\Sigma$, then $f^*(t) = t$.
- If A and B are finitely generated, then $\forall\, t \in T_\Sigma(A_{Obs})$, $\exists\, g \in T_\Sigma$ such that $\varepsilon_A(g) = \varepsilon_A(t)$ and $\varepsilon_B(f^*(t)) = \varepsilon_B(g)$.

Proof. We shall prove the first property by structural induction.

- Induction base. If $t \in F_{\lambda,s}$, where $s \in S$, then $f^*(t) = t$, by definition.

- Induction step. If t has the form $\sigma(t_1, ..., t_n)$, with n > 0, then $f^*(t) = \sigma(f^*(t_1), .., f^*(t_n))$. But since, via the hypothesis of induction, all $f^*(t_i) = t_i$, then $f^*(t) = \sigma(t_1, .., t_n) = t$.

Now, we prove the second property by structural induction:

- Induction base. If $t \in A_{Obs}$, then $f^*(t) = f(t)$. Since $f(t) \in B_{Obs}$, we have that $\varepsilon_B(f^*(t)) = \varepsilon_B(f(t)) = f(t)$. On the other hand, via the definition of f, $f(t) = \varepsilon_B(g')$ where $\varepsilon_A(g') = t$. Consequently, $\varepsilon_B(f^*(t)) = \varepsilon_B(g')$. Since $\varepsilon_A(t) = t$, then $\varepsilon_A(t) = \varepsilon_A(g')$. Therefore, if we make g be g', the property is proved.

- Induction step Examine the case in which t has the form $\sigma(t_1, ..., t_n)$, with n $\geq$ 0. Via hypothesis of induction we have that for any t_i, there is a g_i such that $\varepsilon_A(g_i) = \varepsilon_A(t_i)$ and $\varepsilon_B(f^*(t_i)) = \varepsilon_B(g_i)$.

  By applying this hypothesis of induction and the definition of ε_A, we write:
  $\varepsilon_A(\sigma(t_1, ..., t_n)) = \sigma_A(\varepsilon_A(t_1), ..., \varepsilon_A(t_n)) = \sigma_A(\varepsilon_A(g_1), ..., \varepsilon_A(g_n)) =$
  $\varepsilon_A(\sigma(t_1, ..., t_n))$.

  On the other hand, we have, via the hypothesis of induction and the definitions of ε_B and f^*:
  $\varepsilon_B(f^*(\sigma(t_1, ..., t_n))) = \varepsilon_B(\sigma_B(f^*(t_1), ..., f^*(t_n))) = \varepsilon_B(\sigma_B(\varepsilon_B(g_1), ..., \varepsilon_B(g_n))) =$
  $\varepsilon_B(\sigma(g_1, ..., g_n))$.

  Therefore, if we take $\sigma(g_1, ..., g_n)$ as g, we shall prove the property.

  So, we have proved the sublemma. □

Lemma *.Prop4. Let A and B be two Σ-algebras finitely generated. If A and B are eval-equivalent, then they are behaviorally equivalent.

Proof. In order to prove that they are behaviorally equivalent, we must prove that there is a behavioral isomorphism f between A and B.

We define f: $A_{Obs} \longrightarrow B_{Obs}$ in the following way:
For each a $\in A_{Obs}$, we have f(a)=$\varepsilon_B(g)$ where g $\in T_\Sigma$ such that $\varepsilon_A(g) = a$.

(The exhaustivity of ε_A guarantees that g exists, because A is finitely generated. To avoid the problem that there may be several possible "g"'s, we define an arbitrary order between ground terms and we choose the first of them w.r.t this order.)

If f has been defined in the previous way, f^* has the form $f^* : T_\Sigma(A_{Obs}) \longrightarrow T_\Sigma(B_{Obs})$ such that, by sublemma *.Prop3, it is fulfilled that:
$\forall\, t \in T_\Sigma(A_{Obs}), \exists\, g \in T_\Sigma$ such that $\varepsilon_A(g) = \varepsilon_A(t)$ and $\varepsilon_B(f^*(t)) = \varepsilon_B(g)$.

(Sublemma *.Prop2 guarantees that g exists. If there are several possible "g"'s, we apply the same solution as above.)

We want to check if f is a behavioral isomorphism. We must prove two things: f is a behavioral morphism and f is bijective.

Now, we shall prove that f is a behavioral morphism. We want to see that

$$\forall\, t \in T_\Sigma(A_{Obs})_s,\ \textit{with } s \in Obs, \quad f(\varepsilon_A(t)) = \varepsilon_B(f^*(t)).$$

On the one hand, we have, by definition of f, since $\varepsilon_A(t) \in A_{Obs}$:

$$f(\varepsilon_A(t)) = \varepsilon_B(g),\ \textit{where } g \in T_\Sigma \textit{ such that } \varepsilon_A(g) = \varepsilon_A(t).$$

On the other hand, by sublemma *.Prop3, since $t \in T_\Sigma(A_{Obs})$:

$$\varepsilon_B(f^*(t)) = \varepsilon_B(g'),\ \textit{where } g' \in T_\Sigma \textit{ such that } \varepsilon_A(g') = \varepsilon_A(t).$$

Therefore, proving the statement of equality $f(\varepsilon_A(t)) = \varepsilon_B(f^*(t))$ has been reduced to proving:

$$\varepsilon_B(g) = \varepsilon_B(g').$$

On the other hand, we have that $\varepsilon_A(g) = \varepsilon_A(g')$, because the two terms of this equation are equal to $\varepsilon_A(t)$. In consequence, be φ as it may, it is fulfilled that $\varphi(\varepsilon_A(g)) = \varphi(\varepsilon_A(g'))$. Since, via the definition of eval-equivalence, we have that:

$$\varphi(\varepsilon_A(g)) = \varepsilon_B(g) \qquad \varphi(\varepsilon_A(g')) = \varepsilon_B(g')$$

We obtain $\varepsilon_B(g) = \varepsilon_B(g')$, which is what we wished to prove.

Now, we shall prove that f is bijective. First, we shall check that f is injective, that is to say,

$$\forall\, a, b \in A_{Obs},\ f(a) = f(b) \Longrightarrow a = b$$

Via the definition of f, $f(a) = \varepsilon_B(g)$ and $f(b) = \varepsilon_B(g')$, where $\varepsilon_A(g) = a$ and $\varepsilon_A(g') = b$. Therefore, the equation f(a)=f(b) becomes:

$$\varepsilon_B(g) = \varepsilon_B(g')$$

Now, we know by eval-equivalence that there is a bijection φ such that $\varphi(\varepsilon_A(g)) = \varepsilon_B(g)$ and $\varphi(\varepsilon_A(g')) = \varepsilon_B(g')$. Thus, the previous equality is transformed into:

$$\varphi(\varepsilon_A(g)) = \varphi(\varepsilon_A(g'))$$

Now, since φ is bijective, this produces:

$$\varepsilon_A(g) = \varepsilon_A(g')$$

That, as we have seen before, is equivalent to:

$a = b$

And this is what we intended to prove.

Now, let us check that f is exhaustive, that is to say,

$\forall\, b \in B_{Obs}\, \exists\, a \in A_{Obs}$ *such that* $f(a) = b$

Since B is finitely generated,

$\exists\, g \in T_\Sigma$, *tal que* $\varepsilon_B(g) = b$.

Then, a is $\varepsilon_A(g)$, because, as we shall prove next, $f(\varepsilon_A(g)) = b$.

$f(\varepsilon_A(g)) = \varepsilon_B(g')$ *where* $\varepsilon_A(g') = \varepsilon_A(g)$

Trivially [1],

$\varphi(\varepsilon_A(g')) = \varphi(\varepsilon_A(g))$

And, by eval-equivalence,

$\varepsilon_B(g') = \varepsilon_B(g)$

And, since $f(a) = \varepsilon_B(g')$ and $b = \varepsilon_B(g)$, we have:

$f(a) = b$

which is what we wished to prove. □

Lemma *.Prop5. Let A and B be two Σ-algebras. If A and B are behaviorally equivalent, they are eval-equivalent.

Proof. Since A and B are behaviorally equivalent:

$\forall\, t \in T_\Sigma(A_{Obs})_s$, *with* $s \in Obs$, $\quad f(\varepsilon_A(t)) = \varepsilon_B(f^*(t))$.

Since all the observable ground terms are observable computations, we have:

$\forall\, t \in T_{\Sigma_s}$, *with* $s \in Obs$, $\quad f(\varepsilon_A(t)) = \varepsilon_B(f^*(t))$.

And, by sublemma *.Prop3, if $t \in T_\Sigma$ then $f^*(t) = t$, it is fulfilled that:

$\forall\, t \in T_{\Sigma_s}$, *with* $s \in Obs$, $\quad f(\varepsilon_A(t)) = \varepsilon_B(t)$.

[1] Notice that g doesn't have to be g', though $\varepsilon_A(g)$ and $\varepsilon_A(g')$ can be equal

Which, if we make f be φ, is the definition of eval-equivalence. □

The next theorem summarizes what is proved above.

Theorem *.Prop6 Let A and B be two finitely generated Σ-algebras. A and B are behaviorally equivalent if and only if they are eval-equivalent.

Proof. It is a direct consequence of lemmas *.Prop4 and *.Prop5. □

Definition *.Prop7 Let $SPEC_1$ and $SPEC_2$ be two behavioral specifications over the same signature Σ. We say that $SPEC_1$ and $SPEC_2$ are eval-equivalent if their respective initial algebras are eval-equivalent.

Lemma *.PropEV Let $SPEC_1$ and $SPEC_2$ be two behavioral specifications which have the same signature Σ. The definition of eval-equivalence between $SPEC_1$ and $SPEC_2$ can be written as follows:

There is a bijection φ between $(T_{SPEC_1})_{Obs}$ and $(T_{SPEC_2})_{Obs}$ such that
$\forall \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$,
$\forall t_1, \ldots, t_n \in T_{\Sigma_{SPEC_1}}, \varphi(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, \ldots, t_n))) = \varepsilon_{T_{SPEC_2}}(\sigma(t_1, \ldots, t_n))$

Proof. "$SPEC_1$ and $SPEC_2$ are equivalent" means "the initial algebras of $SPEC_1$ and $SPEC_2$ are equivalent". Therefore, if we apply the definition of eval-equivalence to algebras $T_{\Sigma_{SPEC_1}}$ and $T_{\Sigma_{SPEC_2}}$, we have the following statement:

There is a bijection φ between $(T_{SPEC_1})_{Obs}$ and $(T_{SPEC_2})_{Obs}$ such that
$\forall t \in (T_{\Sigma_{SPEC_1}})_s$, *with* $s \in Obs$, $\varphi(\varepsilon_{T_{SPEC_1}}(t)) = \varepsilon_{T_{SPEC_2}}(t)$

Since $t \in (T_{\Sigma_{SPEC_1}})_s$, where $s \in Obs$, then t must have the form $\sigma(t_1, \ldots, t_n)$, where n ≥ 0, $\sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, and $t_1, \ldots, t_n \in T_{\Sigma_{SPEC_1}}$. By replacing these equivalences in the last statement, we obtain what we wished to prove. □

Theorem *.Prop8 Let $SPEC_1$ and $SPEC_2$ be two behavioral specifications over the same signature Σ. $SPEC_1$ and $SPEC_2$ are behaviorally equivalent if and only they are eval-equivalent.

Proof. This theorem is inferred easily from theorem *.Prop6, if we keep in mind that T_{SPEC_1} and T_{SPEC_2} are finitely generated because they are initial algebras. □

4 Renamings

In this section, we introduce the concept of "renaming", which will be useful to define that of reunion in section 5. We also describe some properties of renamings that will be useful for next proofs.

4.1 Definition of renaming

In this subsection, the concept of renaming is defined. Intuitively, a specification $SPEC_1$ is a renaming of another specification $SPEC_2$ if we can obtain $SPEC_2$ from $SPEC_1$ by changing the names of the function symbols. To be more exact, only the names of the function symbols which are related to the non-observable sorts are changed. The names of the function symbols with observable parameters and result remain the same.

In more formal terms, the concept of renaming is defined as follows:

Definition *.Prop9. We say that a behavioral specification $SPEC_2 = (Obs, S, F_2, E_2)$ is a renaming of another specification $SPEC_1 = (Obs, S, F_1, E_1)$ if there is a bijection $\phi : F_1 \longrightarrow F_2$ such that:

1. $\forall \sigma,\ \forall w1,..,wn,s \in S;\quad \sigma \in (F_1)_{w1...wn,s}$ if and only if $\phi(\sigma) \in (F_2)_{w1...wn,s}$.

2. $\forall \sigma \in F_1 \cup F_2$ it is fulfilled that

   $$\sigma \in ((F_1)_{w1...wn,s} \cap (F_2)_{w1...wn,s}) \text{ if and only if } w1,..,wn,s \in Obs$$

3. $\forall \sigma \in (F_1)_{w1...wn,s},\ \forall w1,..,wn,s \in S;\quad \phi(\sigma) = \sigma$ if and only if $w1,..,wn,s \in Obs$.

4. For each equation $e : c_1 = d_1 \,\&...\&\, c_n = d_n \Rightarrow p_1 = p_2;\quad e \in E_1$ if and only if $\phi^*(c_1) = \phi^*(d_1) \,\&...\&\, \phi^*(c_n) = \phi^*(d_n) \Rightarrow \phi^*(p_1) = \phi^*(p_2) \in E_2$, where $\phi^* : T_{\Sigma_{SPEC_1}}(X) \longrightarrow T_{\Sigma_{SPEC_2}}(X)$ is defined as follows:

   - If x is a variable, $\phi^*(x) = x$.
   - If x has the form $\sigma(t1, ..., tn)$, where $\sigma \in (F_1)_{w1,...,wn,s}$, then $\phi^*(x) = \phi(\sigma)(\phi^*(t1), ..., \phi^*(tn))$

Comments.

- Condition 2 could have been written as follows:

  $$\sigma \in ((F_1)_{w1...wn,s} \cap (F_2)_{w1...wn,s}) \Longrightarrow w1,..,wn,s \in Obs$$

  because the reciprocal implication can be easily deduced from conditions 1 and 3. However, we have preferred to write the double implication for the sake of clearness.

- It is easy to see that the algorithm which creates a renaming of a behavioral specification has a linear complexity w.r.t the input.

Notation. If $SPEC_2$ is a renaming of $SPEC_1$, we write: $SPEC_2 \in Renam(SPEC_1)$. The bijection ϕ is called renaming bijection.

4.2 Totally observable equations and terms of a renaming

In this subsection, we shall prove that if $SPEC_2$ is a renaming of $SPEC_1$, then the totally observable equations and terms of $SPEC_1$ and $SPEC_2$ are the same. This result (and some properties which we shall use for its demonstration) will be useful for next sections.

Sublemma *.Prop10 If $SPEC_2 \in Renam(SPEC_1)$, it is fulfilled that $\forall t \in T_{\Sigma_{SPEC_1}}$:

- $\phi^*(t) \in T_{\Sigma_{SPEC_2}}$
- if $t \in Tot_{SPEC_1}$ it is fulfilled that $\phi^*(t) = t$.

Proof. We prove this by structural induction:

- Induction base. Let $t \in (F_1)_{\lambda,s}$, where $s \in S$. By definition *.Prop9, $\phi(t) \in (F_2)_{\lambda,s}$ and, therefore, $\phi^*(t) = \phi(t) \in T_{\Sigma_{SPEC_2}}$.

  On the other hand, if $t \in Tot_{SPEC_1}$ then $s \in Obs$ and, by definition *.Prop9, $\phi^*(t) = \phi(t) = t$.

- Induction step. t has the form $\sigma(t_1, ..., t_n)$, where all the $t_i \in T_{\Sigma_{SPEC_1}}$ and $\sigma \in F_1$. By definition *.Prop9, $\phi(\sigma) \in F_2$ and, via the hypothesis of induction, all the $\phi^*(t_i) \in T_{\Sigma_{SPEC_2}}$. Consequently, $\phi^*(t) = \phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n)) \in T_{\Sigma_{SPEC_2}}$, which is what we wished to prove.

  On the other hand, if $t \in Tot_{SPEC_1}$, then $\sigma \in (F_1)_{w1...wn,s}$, where $w1, ..., wn, s \in Obs$ and all the $t_i \in Tot_{SPEC_1}$. Then, by definition *.Prop9, $\phi(\sigma) \in (F_2)_{w1...wn,s}$ and, via the hypothesis of induction, all the $\phi^*(t_i) = t_i$. Replace in $\phi^*(t) = \phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n))$ and we obtain that $\phi^*(t) = t$, which is what we wished to prove. □

Sublemma *.Prop81 Let $SPEC_1$ and $SPEC_2$ two specifications such that $SPEC_2 \in Renam(SPEC_1)$. Let X be any set of variables. Suppose $t_1 \in (T_{\Sigma_{SPEC_1}})(X)$ and $t_2 \in Tot_{SPEC_2}(X)$ such that $\phi^*(t_1) = t_2$. Then $t_1 = t_2$.

Proof. We shall prove this by structural induction on the term t_1.

- Induction base. We can distinguish two subcases.
  - Suppose $t_1 \in (F_1)_{\lambda,s}$, where $s \in S$. By definition *.Prop9, $\phi^*(t_1) = \phi(t_1)$ and $\phi(t_1) \in (F_2)_{\lambda,s}$. That is to say, $t_2 \in (F_2)_{\lambda,s}$. Now, since $t_2 \in Tot_{SPEC_2}$, this means that $s \in Obs$. Therefore, by condition 3 of definition *.Prop9, $\phi(t_1) = t_1$. That is to say, $t_2 = t_1$.
  - Suppose $t_1 \in X$. By definition *.Prop9, $\phi^*(t_1) = t_1$ and, consequently, $t_1 = t_2$.

- Induction step. Suppose t_1 has the form $\sigma(l_1, ..., l_n)$, where, for each i, $l_i \in T_{\Sigma_{SPEC_1}}$ and $\sigma \in (F_1)_{w1..wn,s}$. Consequently, by definition *.Prop9, t_2 is $\phi(\sigma)(m_1, ..., m_n)$ where $\phi(\sigma) \in (F_2)_{w1..wn,s}$, $m_1 = \phi^*(l_1), ..., m_n = \phi^*(l_n)$. Since $t_2 \in Tot_{SPEC_2}$, then $w1, ..., wn \in Obs$ and, on the other hand, $m_1, ..., m_n \in Tot_{SPEC_2}$. Now, since $w1, ..., wn \in Obs$, by condition 3 of definition *.Prop9, $\phi(\sigma) = \sigma$. On the other hand, since $m_1, ..., m_n \in Tot_{SPEC_2}$, by the hypothesis of induction, $l_1 = m_1, ..., l_n = m_n$. That is to say, t_2 is $\sigma(l_1, ..., l_n)$. Therefore, $t_1 = t_2$, which is what we wished to prove. □

Sublemma *.Prop80 Let $SPEC_1$ and $SPEC_2$ be two specifications such that $SPEC_2 \in Renam(SPEC_1)$. Let X be any set of variables. Suppose $t \in T_{\Sigma_{SPEC_1}}(X)$ and $x \in X$ such that $\phi^*(t) = x$. Then $t \in X$.

Proof. We shall prove this by contradiction. Suppose $t \notin X$. Then, t has the form $\sigma(t_1 ... t_n)$. Therefore, x (which is equal to $\phi^*(t)$) is $\phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n))$. Consequently, x is not a variable, which is a contradiction. □

Sublemma *.Prop65 Let X be a set of variables. Suppose $SPEC_1$ and $SPEC_2$ such that $SPEC_2 \in Renam(SPEC_1)$. It is fulfilled that $Tot_{SPEC_1}(X) = Tot_{SPEC_2}(X)$.

Proof. We must prove that $\forall t \in Tot_{SPEC_1}(X)$ it is fulfilled that $t \in Tot_{SPEC_2}(X)$. We shall prove this property by structural induction on t.

First, we prove that, if $w1, ..., wn, s \in Obs$, $t \in (F_1)_{w1..wn,s}$ if and only if $t \in (F_2)_{w1..wn,s}$. Suppose that $t \in (F_1)_{w1..wn,s}$ with w1,...,wn,s ∈ Obs. Then, since $SPEC_2$ is a renaming of $SPEC_1$, by condition 2 of definition *.Prop9, $t \in ((F_1)_{w1..wn,s} \cap (F_2)_{w1..wn,s})$. And, consequently, $t \in (F_2)_{w1..wn,s}$. Thus, we have proved that $t \in (F_1)_{w1..wn,s}$ implies $t \in (F_2)_{w1..wn,s}$. The reciprocal implication can be proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_2$.

- Induction step. We can distinguish two cases:
  - t is a variable. Then, since $t \in Tot_{SPEC_1}(X)$, the sort of t is $s \in Obs$. Now, since $SPEC_1$ and $SPEC_2$ have the same observable sorts, then $t \in Tot_{SPEC_2}(X)$. Thus, we have proved that $t \in Tot_{SPEC_1}(X)$ implies $t \in Tot_{SPEC_2}(X)$. The reciprocal implication can be proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_2$.

  - $t : \lambda \longrightarrow s$. Therefore, if $t \in Tot_{SPEC_1}(X)$, then $t \in (F_1)_{\lambda,s}$ with s ∈ Obs. By applying the property stated at the beginning of this proof, we have that $t \in (F_2)_{Obs}$ and, consequently, $t \in (T_{\Sigma_{SPEC_2}})_{\lambda,s}$. Thus, we have proved that $t \in Tot_{SPEC_1}(X)$ implies $t \in Tot_{SPEC_2}(X)$. The reciprocal implication can be proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_2$.

- Induction step. t has the form $\sigma(t_1,...,t_n)$. If $t \in Tot_{SPEC_1}$, all the subterms of t must be observable, and, consequently, $t_1,...,t_n \in Tot_{SPEC_1}(X)$. On the one hand, $t_1,...,t_n \in Tot_{SPEC_2}(X)$, via the hypothesis of induction. On the other hand, since $\sigma \in (F_1)_{w1...wn,s}$, we have that $\sigma \in (F_2)_{w1...wn,s}$, by applying the property given at the beginning on this proof. Consequently, $\sigma(t_1,...,t_n)$ belongs to $Tot_{SPEC_2}(X)$. Thus, we have proved that $t \in Tot_{SPEC_1}(X)$ implies $t \in Tot_{SPEC_2}(X)$. The reciprocal implication can be proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_2$. □

Corollary *.Prop66. Suppose $SPEC_1$ and $SPEC_2$ such that $SPEC_2 \in Renam(SPEC_1)$. It is fulfilled that $Tot_{SPEC_1} = Tot_{SPEC_2}$.

Proof. It is sublemma *.Prop65, when X is equal to $\emptyset$. □.

Sublemma *.Prop14. If $SPEC_2 \in Renam(SPEC_1)$, then $(E_1)_{Obs} = (E_2)_{Obs}$.

Proof. We must prove that $e \in (E_1)_{Obs}$ if and only if $e \in (E_2)_{Obs}$.

First, we shall prove the implication which goes from left to right. Since $e \in (E_1)_{Obs}$, then e has the form $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ where $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_1}(X)$.

By definition *.Prop9, $\phi^*(c_1) = \phi^*(d_1)$ &...& $\phi^*(c_n) = \phi^*(d_n) \Rightarrow \phi^*(t_1) = \phi^*(t_2) \in E_2$. Now, by sublemma *.Prop10, this means that $c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2 \in E_2$. That is to say, $e \in E_2$. Since $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_1}(X)$, by sublemma *.Prop65, $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_2}(X)$. Moreover, $e \in E_2$. Therefore, $e \in (E_2)_{Obs}$.

Now, we shall prove the implication which goes from right to left. Since $e \in (E_2)_{Obs}$, then e has the form $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ where $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_2}(X)$.

By sublemma *.Prop12, there are $c'_1,...,c'_n,d'_1,...,d'_n,t'_1,t'_2 \in Tot_{SPEC_1}(X)$ such that $\phi^*(c'_1) = c_1,...,\phi^*(c'_n) = c_n, \phi^*(d'_1) = d_1,...,\phi^*(d'_n) = d_n, \phi^*(t'_1) = t_1, \phi^*(t'_2) = t_2$. Therefore, $e : \phi^*(c'_1) = \phi^*(d'_1)$ &...& $\phi^*(c'_n) = \phi^*(d'_n) \Rightarrow \phi^*(t'_1) = \phi^*(t'_2) \in E_2$. By definition *.Prop9, we have that $c'_1 = d'_1$ &...& $c'_n = d'_n \Rightarrow t'_1 = t'_2 \in E_1$.

Now, since $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_2}(X)$, by applying repeatedly sublemma *.Prop81 on expressions $\phi^*(c'_1) = c_1,...,\phi^*(c'_n) = c_n, \phi^*(d'_1) = d_1,...,\phi^*(d'_n) = d_n, \phi^*(t'_1) = t_1, \phi^*(t'_2) = t_2$, we have that $c'_1 = c_1,...,c'_n = c_n, d'_1 = d_1,...,d'_n = d_n, t'_1 = t_1, t'_2 = t_2$. That is to say, e is the same equation as $c'_1 = d'_1$ &...& $c'_n = d'_n \Rightarrow t'_1 = t'_2$. Consequently, $e \in E_1$.

Now, since $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_2}(X)$, by sublemma *.Prop65, $c_1,...,c_n,d_1,...,d_n,t_1,t_2 \in Tot_{SPEC_1}(X)$. That is to say, $e \in (E_1)_{Obs}$, which is what we wished to prove. □

4.3 Congruence and classes of equivalence of a renaming

In this subsection, we shall prove that, if *SPEC*$_2$ is a renaming of *SPEC*$_1$, the congruences of *SPEC*$_1$ and *SPEC*$_2$ are the same (if we do not care about the changes of names entailed by a renaming). This enables to define a bijection between the classes of equivalence of *SPEC*$_1$ and *SPEC*$_2$. Both facts will be useful for next proofs..

In other words: what we prove is that *SPEC*$_2$ contains the same information as *SPEC*$_1$, that is to say, that the operation of renaming changes names but preserves the information.

Sublemma *.Prop26 Let $v : vars(e) \longrightarrow T_{\Sigma_{SPEC_1}}$ be an assignment of values and ϕ a renaming bijection between *SPEC*$_1$ and *SPEC*$_2$. If $\phi^* : T_{\Sigma_{SPEC_1}} \longrightarrow T_{\Sigma_{SPEC_2}}$ is defined out of ϕ as in the definition of renaming and $w : vars(e) \longrightarrow T_{\Sigma_{SPEC_2}}$ is the assignment of values such that$w = \phi^* \circ v$, then we have:

$\forall\, t \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\phi^*(v^*(t)) = w^*(\phi^*(t))$

Proof. We shall prove this by structural induction on t.

- Induction base. Suppose that t is a variable. We wish to prove that:

  $$\phi^*(v^*(t)) = w^*(\phi^*(t))$$

  Since t is a variable, by the definitions of ϕ^* and v^*, we have that $v^*(t) = v(t)$ and $\phi^*(t) = t$. So the last equality becomes:

  $$\phi^*(v(t)) = w^*(t)$$

  Analogously, since t is a variable, we have that $w^*(t) = w(t)$ and, consequently:

  $$\phi^*(v(t)) = w(t)$$

  But, since w is defined as $w = \phi^* \circ v$, this becomes:

  $$\phi^*(v(t)) = \phi^*(v(t))$$

  which is a trivial equality and, therefore, the induction base is proved.

- Induction step. Suppose that t has the form $\sigma(t_1, ..., t_n)$ where n $\geq$ 0. Then $\phi^*(v^*(t))$ can be written as:

  $$\phi^*(v^*(\sigma(t_1, ..., t_n)))$$

By definition of v^*, this is equivalent to:

$$\phi^*(\sigma(v^*(t_1), ..., v^*(t_n)))$$

By definition of ϕ^*, this is equivalent to:

$$\phi(\sigma)(\phi^*(v^*(t_1)), ..., \phi^*(v^*(t_n)))$$

By the hypothesis of induction, this expression becomes:

$$\phi(\sigma)(w^*(\phi^*(t_1)), ..., w^*(\phi^*(t_n)))$$

Which, by definition of w^*, is equivalent to:

$$w^*(\phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n)))$$

By definition of ϕ^*, this is equivalent to:

$$w^*(\phi^*(\sigma(t_1, ..., (t_n)))$$

And, since t is $\sigma(t_1, ..., t_n)$, then this is equivalent to:

$$w^*(\phi^*(t))$$

which is what we wished to prove. □

Sublemma *.Prop11 If $SPEC_2 \in Renam(SPEC_1)$,

$$\forall t_1, t_2 \in T_{\Sigma_{SPEC_1}}, \quad t_1 \equiv_{SPEC_1} t_2 \text{ if and only if } \phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$$

Proof. We prove that $t_1 \equiv_{SPEC_1} t_2$ implies $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$. The reciprocal implication can be proved analogously.

In order to demonstrate $t_1 \equiv_{SPEC_1} t_2$, we must apply repeatedly the definition of $\equiv_{SPEC_1}$. We refer by "step" to each of these applications. We make an induction on the number of steps of the demonstration of $t_1 \equiv_{SPEC_1} t_2$.

If $t_1 \equiv_{SPEC_1} t_2$, by definition of $\equiv_{SPEC}$, one of the following four cases may occur[2]:

1. In this case, $t_1 = t_2$. Trivially, $\phi^*(t_1) = \phi^*(t_2)$ and, therefore, $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$, since $\equiv_{SPEC_2}$ is reflexive.

[2] Case 1 belongs to the induction base, cases 2 and 3 belong to the induction step. Case 4 belongs to the induction base when the applied equation is unconditional or, otherwise, it belongs to the induction step

2. In this case, $t_1 \equiv_{SPEC_1} t_2$ because $t_2 \equiv_{SPEC_1} t_1$. Since the demonstration of the latter relationship of congruence has a step less than the former, we can apply the hypothesis of induction. We have that $\phi^*(t_2) \equiv_{SPEC_2} \phi^*(t_1)$ and, therefore, $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$, since$\equiv_{SPEC_2}$ is symmetrical.

3. In this case, $t_1 \equiv_{SPEC_1} t_2$ because $t_1 \equiv_{SPEC_1} t_3$ and $t_3 \equiv_{SPEC_1} t_2$. Since the demonstrations of $t_1 \equiv_{SPEC_1} t_3$ and $t_3 \equiv_{SPEC_1} t_2$ are shorter (in number of steps) than the former, we can apply the hypothesis of induction. So we have $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_3)$ and $\phi^*(t_3) \equiv_{SPEC_2} \phi^*(t_2)$. Since $\equiv_{SPEC_2}$ is transitive, we have that $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$.

4. In this case, $t_1 \equiv_{SPEC_1} t_2$ because there is an equation $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow p_1 = p_2$, where $e \in E_1$ and an assignment of values v:X $\longrightarrow T_{\Sigma_{SPEC_1}}$ such that

   (a) $v^*(p_1) = t_1$ and $v^*(p_2) = t_2$, and, moreover,

   (b) $v^*(c_1) \equiv_{SPEC_1} v^*(d_1)$, ..., $v^*(c_n) \equiv_{SPEC_1} v^*(d_n)$

   On the one hand, since demonstrations $v^*(c_1) \equiv_{SPEC_1} v^*(d_1)$, ..., $v^*(c_n) \equiv_{SPEC_1} v^*(d_n)$ are shorter (in number of steps) than the former, by the hypothesis of induction, we have: $\phi^*(v^*(c_1)) \equiv_{SPEC_2} \phi^*(v^*(d_1)), \ldots, \phi^*(v^*(c_n)) \equiv_{SPEC_2} \phi^*(v^*(d_n))$. Since, by sublemma *.Prop26, $\phi^*(v^*(x)) = w^*(\phi^*(x))$, we have the following result: $w^*(\phi^*(c_1)) \equiv_{SPEC_2} w^*(\phi^*(d_1)), \ldots, w^*(\phi^*(c_n)) \equiv_{SPEC_2} w^*(\phi^*(d_n))$.

   On the other hand, since $SPEC_2$ is a renaming of $SPEC_1$ and $e \in E_1$, we have that ($\phi^*(c_1) = \phi^*(d_1)$ &...& $\phi^*(c_n) = \phi^*(d_n) \Rightarrow \phi^*(p_1) = \phi^*(p_2)) \in E_2$. At this point, we can apply the fourth case of the definition of $\equiv_{SPEC_2}$ on this equation and on the result of the previous paragraph. We have that $w^*(\phi^*(p_1)) \equiv_{SPEC_2} w^*(\phi^*(p_2))$.

   By sublemma *.Prop26, $\phi^*(v^*(x)) = w^*(\phi^*(x))$ and, therefore, $\phi^*(v^*(p_1)) \equiv_{SPEC_2} \phi^*(v^*(p_2))$. Now, since $v^*(p_1) = t_1$ i $v^*(p_2) = t_2$, we have that $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$, which is what we wished to prove. □

Sublemma *.Prop60. Suppose $SPEC_2 \in Renam(SPEC_1)$. Let X be a set of variables. It is fulfilled that $\forall\, t_2 \in T_{\Sigma_{SPEC_2}}(X) \quad \exists\, t_1 \in T_{\Sigma_{SPEC_1}}(X)$ such that $\phi^*(t_1) = t_2$.

Proof. We shall prove this by structural induction.

- Induction base. We can distinguish two cases:

  - Since ϕ is a bijection, if $t_2 \in (F_2)_{\lambda,s}$, then there is a c such that $c \in (F_1)_{\lambda,s}$ and $\phi(c) = t_2$. By definition *.Prop9, $\phi^*(c) = \phi(c)$. Therefore, if we make c be t_1, we obtain what we wished.

- If $t_2 \in X$, then $t_2 \in T_{\Sigma_{SPEC_1}}(X)$. Since t_2 is a variable, it is fulfilled that $\phi^*(t_2) = t_2$, by definition *.Prop9. Therefore, if we make t_1 be t_2, we obtain what we wished.

- Induction step. t_2 has the form $\sigma(l_1, ..., l_n)$, where, for any i, $l_i \in T_{\Sigma_{SPEC_2}}$. By the hypothesis of induction, $\exists\, c_1, ..., c_n$, such that, for any i, $\phi^*(c_i) = l_i$. Since, by definition *.Prop9, there is a $c \in F_1$ such that $\phi(c) = \sigma$, we have that $\phi^*(c(c_1, ..., c_n))= \phi(c)(\phi^*(c_1), ..., \phi^*(c_n))=\sigma(l_1, ..., l_n)$. Therefore, if we make t_1 be $c(c_1, ..., c_n)$, we obtain what we wished.

Sublemma *.Prop12. If $SPEC_2 \in Renam(SPEC_1)$, $\forall\, t_2 \in T_{\Sigma_{SPEC_2}}$ $\exists\, t_1 \in T_{\Sigma_{SPEC_1}}$ such that $\phi^*(t_1) = t_2$.

Proof. It is a corollary of sublemma *.Prop60 when X is equal to $\emptyset$. □

Lemma *.Prop13. If $SPEC_2 \in Renam(SPEC_1)$, then there is a bijection $\beta : (T_{SPEC_1})_{Obs} \longrightarrow (T_{SPEC_2})_{Obs}$, such that:

$$\forall\, t \in (T_{\Sigma_{SPEC_1}})_s, \textit{ with } s \in Obs, \quad \beta(\varepsilon_{T_{SPEC_1}}(t)) = \varepsilon_{T_{SPEC_2}}(\phi^*(t))$$

Proof. Given a class of equivalence $[t]_{\equiv_1}$, we define $\beta([t]_{\equiv_1}) = ([\phi^*(t)]_{\equiv_2})$.

Since we know, by definition of quotient term algebra, that for any specification SPEC:

$$\varepsilon_{T_{SPEC}}(t) = [t]_{\equiv_{SPEC}}$$

Replacing in the definition of β:

$$\beta(\varepsilon_{T_{SPEC_1}}(t)) = \varepsilon_{T_{SPEC_2}}(\phi^*(t))$$

which is what we intended to prove. Now, we shall see that β is a bijection. By sublemma *.Prop11, $t_1 \equiv_{SPEC_1} t_2$ if and only if $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$, that is to say, $[t_1]_{\equiv_{SPEC_1}} = [t_2]_{\equiv_{SPEC_1}}$ if and only if $[\phi^*(t_1)]_{\equiv_{SPEC_2}} = [\phi^*(t_2)]_{\equiv_{SPEC_2}}$. Therefore, β is well defined because it does not depend on the particular representative of the class that we use.

β is injective, that is to say:

$$\forall [t_1]_{\equiv_{SPEC_1}}, [t_2]_{\equiv_{SPEC_1}}; \quad \beta([t_1]_{\equiv_{SPEC_1}}) = \beta([t_2]_{\equiv_{SPEC_1}}) \text{ implies } [t_1]_{\equiv_{SPEC_1}} = [t_2]_{\equiv_{SPEC_1}}$$

We start from:

$$\beta([t_1]_{\equiv_{SPEC_1}}) = \beta([t_2]_{\equiv_{SPEC_1}})$$

By definition of β:

$$[\phi^*(t_1)]_{\equiv_{SPEC_2}} = [\phi^*(t_2)]_{\equiv_{SPEC_2}}$$

This is the same as:

$$\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$$

Now, by sublemma *.Prop11, this is equivalent to:

$$t_1 \equiv_{SPEC_1} t_2$$

That is to say,

$$[t_1]_{\equiv_{SPEC_1}} = [t_2]_{\equiv_{SPEC_1}}$$

and this is what we wished to prove.

Now, we shall see that β is exhaustive:

$$\forall [t_2]_{\equiv_{SPEC_2}}, \exists [t_1]_{\equiv_{SPEC_1}}; \quad \beta([t_1]_{\equiv_{SPEC_1}}) = [t_2]_{\equiv_{SPEC_2}}$$

We start from sublemma *.Prop12:

$$\forall\, t_2 \in T_{\Sigma_{SPEC_2}} \exists\, t_3 \in T_{\Sigma_{SPEC_1}} \text{ such that } \phi^*(t_3) = t_2$$

This implies

$$\forall\, t_2 \in T_{\Sigma_{SPEC_2}} \exists\, t_3 \in T_{\Sigma_{SPEC_1}} \text{ such that } \phi^*(t_3) \equiv_{SPEC_2} t_2$$

That is to say,

$$\forall [t_2]_{\equiv_{SPEC_2}} \exists [t_3]_{\equiv_{SPEC_1}} \text{ such that } [\phi^*(t_3)]_{\equiv_{SPEC_2}} = [t_2]_{\equiv_{SPEC_2}}$$

By definition of β:

$$\forall [t_2]_{\equiv_{SPEC_2}} \exists [t_3]_{\equiv_{SPEC_1}} \text{ such that } \beta([t_3]_{\equiv_{SPEC_1}}) = [t_2]_{\equiv_{SPEC_2}}$$

That is to say, if we make $[t_1]_{\equiv_{SPEC_1}}$ be $[t_3]_{\equiv_{SPEC_1}}$, then the exhaustivity is proved. □

5 Reunions.

In this section, the concept of reunion is introduced and we prove some basic properties about it.

5.1 Definition of reunion.

Intuitively, a reunion of two specifications $SPEC_1$ and $SPEC_2$ is a specification $SPEC_4$ which contains all the information which $SPEC_1$ and $SPEC_2$ have individually. A naive idea to do this could be to build a specification which has all the equations belonging to $SPEC_1$ and $SPEC_2$. But, if we do that, since $SPEC_1$ and $SPEC_2$ have the same signature, there will be a naming conflict and $SPEC_4$ will have more equivalences than those ones belonging to $SPEC_1$ and $SPEC_2$ individually.

The solution is to avoid the naming conflict, by using a renaming of $SPEC_2$, instead of $SPEC_2$. As we have seen, the operation of renaming preserves the information but changes the names.

The conditions required for this to be workable appear in definitions *.Prop27 and *.Prop15. (We are working on a paper which removes the need for these conditions).

Definition *.Prop27 Let $SPEC = (Obs, S, F, E)$ be a behavioral specification. Let $\equiv_{Obs}$ be the congruence defined only by the equations of E_{Obs}. We say that, in $SPEC$, *the definition of observable sorts does not modify the definition of the non-observable ones* if it is fulfilled that:

1. Any operation which has an observable result is totally defined over the observable sort, that is to say, $\forall\, t \in (T_{\Sigma_{SPEC}})_s$, with $s \in Obs$, there is a $t' \in Tot_{SPEC}$ such that, $\varepsilon_{T_{SPEC}}(t) = \varepsilon_{T_{SPEC}}(t')$.
2. $\forall\, t_1, t_2 \in Tot_{SPEC}$ it is fulfilled that

$$t_1 \equiv_{Obs} t_2 \text{ if and only if } t_1 \equiv_{SPEC} t_2$$

Comment. In this second condition, the left-to-right implication is superfluous, since it is fulfilled in any behavioral specification because Obs is a subset of E. However, we have preferred to write the double implication for the sake of clearness.

Definition *.Prop15. Let $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2)$ be two behavioral specifications. We say that $SPEC_1$ and $SPEC_2$ *share the same definition of observable sorts* if the following conditions are fulfilled:

- $(E_1)_{Obs} = (E_2)_{Obs}$.
- All the equations belonging to $SPEC_1$ which have only variables on their right-hand sides belong to $(E_1)_{Obs}$. It happens the same with $SPEC_2$.
- In $SPEC_1$ and in $SPEC_2$, the definition of non-observable sorts does not modify that of the observable ones.

Definition *.Prop16. Let $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2)$ be two behavioral specifications which share the same definition of the observable sorts. Given a specification $SPEC_3 = (Obs, S, F_3, E_3)$, such that $SPEC_3 \in Renam(SPEC_2)$. We say that $SPEC_4 = (Obs, S_4, F_4, E_4)$ is a reunion of $SPEC_1$ and $SPEC_2$ via $SPEC_3$ (and we write $SPEC_4 = SPEC_1 \uplus SPEC_2 \; via \; SPEC_3$) if:

- $S_4 = S \cup \gamma$ where $\gamma \notin S$.
- $F_4 = F \cup F_3 \cup F_{new}$ where F_{new} contains the following function symbols:
  - $yes : \longrightarrow \gamma$
  - $plus : \gamma \times \gamma \longrightarrow \gamma$
  - For any $s \in S$

    $trans_s : s \times s \longrightarrow \gamma$

  where $yes, \_plus\_, trans \notin (F \cup F_3)$
- $E_4 = E_1 \cup E_3 \cup E_{new}$ where E_{new} contains the following equations:
  - plus(yes,yes) = yes
  - $\forall s \in S \quad \forall \sigma \in (F_1)_{\lambda,s}$

    $trans_s(\sigma, \phi(\sigma)) =$ yes.
  - $\forall s \in S \quad \forall \sigma \in (F_1)_{w1..wn,s}$

    $trans_s(\sigma(t_1, t_2, ..., t_n), \phi(\sigma)(u_1, u_2, ..., u_n)) =$
    $plus(trans_{w_1}(t_1, u_1), plus(trans_{w_2}(t_2, u_2), ..., trans_{w_n}(t_n, u_n)...))$

  where $E_{new} \cap (E_1 \cup E_3) = \emptyset$

Comments.

- In this definition, we have used the names γ, *yes*, *plus* and *trans* to mean the new sort and the new function symbols which are introduced in a reunion. There may be some trouble if, in $SPEC_1$ and $SPEC_2$, any of these names have already been used (because, as we have seen, $\gamma \notin S$ and $yes, plus, trans \notin (F \cup F_3)$) . This naming conflict is avoided easily by using names other than γ, *yes*, *plus* and *trans*.
- It is easy to see that the algorithm which creates a reunion out of two behavioral specifications has a linear complexity w.r.t the input.

5.2 Basic properties.

In this subsection, we shall prove some properties which will be useful for next proofs.

Sublemma *.Prop41 Suppose *SPEC*$_4$=*SPEC*$_1$ $\uplus$ *SPEC*$_2$ via *SPEC*$_3$. Then, $\forall \sigma \in F \cup F_3$ it is fulfilled that

$$\sigma \in (F_{w1...wn,s} \cap (F_3)_{w1...wn,s}) \text{ if and only if } w1, .., wn, s \in Obs$$

Proof. This sublemma is obtained by applying condition 2 of definition *.Prop9 to *SPEC*$_2$ and *SPEC*$_3$. This application is possible because *SPEC*$_3$ is a renaming of *SPEC*$_2$, by definition *.Prop16. □

Sublemma *.PropCI It is fulfilled that:

- $Tot_{SPEC_4} \subseteq (T_{\Sigma_{SPEC_1}})_{Obs}$
- $Tot_{SPEC_4} \subseteq (T_{\Sigma_{SPEC_3}})_{Obs}$

Proof. We shall prove the first statement. The second statement can be proved analogously, by interchanging the roles of *SPEC*$_1$ and *SPEC*$_3$.

To be more specific, we shall prove that:

$$t \in Tot_{SPEC_4} \text{ implies } t \in (T_{\Sigma_{SPEC_1}})_{Obs}$$

We shall prove this by structural induction on t.

- Induction base. In this case, $t \in (F_4)_{\lambda,s}$ with $s \in Obs$. By definition *.Prop16, $F_4 = F \cup F_3 \cup F_{new}$. Now, $t \notin F_{new}$, since there is no function symbol in F_{new} which has an observable sort. Therefore, $t \in F \cup F_3$. Now, since $s \in Obs$, by sublemma *.Prop41, we have that $t \in F_{\lambda,s} \cap (F_3)_{\lambda,s}$ and, trivially, $t \in F_{\lambda,s}$, with $s \in Obs$. Consequently, $t \in (T_{\Sigma_{SPEC_1}})_{Obs}$, by the definition of this one.
- Induction step. In this case, t has the form $\sigma(t_1, ..., t_n)$, where $\sigma \in (F_4)_{w1..wn,s}$, $w1, ..., wn, s \in Obs$ and $t_1, ..., t_n \in Tot_{SPEC_4}$. Then, we have that $\sigma \in F_{w1..wn,s}$ by the same reasoning that has been applied to t in the induction base. We obtain that $t_1, ..., t_n \in (T_{\Sigma_{SPEC_1}})_{Obs}$, by the hypothesis of induction. And, consequently, $\sigma(t_1, ..., t_n) \in (T_{\Sigma_{SPEC_1}})_{Obs}$, which is what we wished to prove. □

Sublemma *.PropLI $\forall t \in (T_{\Sigma_{SPEC_1}})_{Obs}, \phi^*(t) \in (T_{\Sigma_{SPEC_3}})_{Obs}$.

Proof. t must have the form $\sigma(t_1, ..., t_n)$, where $n \geq 0$, $\sigma \in F_{w1..wn,s}$, $s \in Obs$ i $t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$. Since *SPEC*$_3$ $\in$ *Renam*(*SPEC*$_2$), then, by definition *.Prop16,

$\phi(\sigma) \in (F_3)_{w1..wn,s}$, where $s \in Obs$.

On the other hand, since $SPEC_1$ and $SPEC_2$ have the same signature, we have $t_1, ..., t_n \in T_{\Sigma_{SPEC_2}}$ and, by sublemma *.Prop10, $\phi^*(t_1), ..., \phi^*(t_n) \in T_{\Sigma_{SPEC_3}}$ Therefore, $\phi^*(\sigma(t_1, ..., t_n)) = \phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n)) \in (T_{\Sigma_{SPEC_3}})_{Obs}$, by the definition of this one. □

Sublemma *.Prop17. If $SPEC_4 = SPEC_1 \uplus SPEC_2$ *via* $SPEC_3$, then $(E_1)_{Obs} = (E_3)_{Obs}$.

Proof. $(E_1)_{Obs} = (E_2)_{Obs}$ is fulfilled, because, by definition *.Prop16, $SPEC_1$ and $SPEC_2$ share the same definition of the observable sorts. Since, by definition *.Prop16, $SPEC_3 \in Renam(SPEC_2)$, then, by sublemma *.Prop14, $(E_2)_{Obs} = (E_3)_{Obs}$. Since the equality of sets is transitive, the sublemma is proved. □

Sublemma *.Prop50 In $SPEC_3$ the definition of the non-observable sorts does not modify that of the observable ones.

Proof. First, we prove that $SPEC_3$ fulfills the first condition of definition *.Prop27. We want to prove that $\forall t \in (T_{\Sigma_{SPEC_3}})_s$, with $s \in Obs$, there is a $t' \in Tot_{SPEC_3}$ such that, $t \equiv_{SPEC_3} t'$.

Suppose that $t \in (T_{\Sigma_{SPEC_3}})_s$. Since $SPEC_3 \in Renam(SPEC_2)$, by sublemma *.Prop12, there is $l \in (T_{\Sigma_{SPEC_2}})_s$ such that $\phi^*(l) = t$.

By definition *.Prop16, in $SPEC_2$, the definition of the observable sorts does not modify the definition of the non-observable ones. Therefore, we can apply the first condition of definition *.Prop27. Consequently, there is a $l' \in Tot_{SPEC_2}$ such that, $l \equiv_{SPEC_2} l'$.

Now, by sublemma *.Prop11, $\phi^*(l) \equiv_{SPEC_3} \phi^*(l')$, that is to say, $t \equiv_{SPEC_3} \phi^*(l')$. Moreover, by sublemma *.Prop10, $\phi^*(l') = l'$, that is to say, $t \equiv_{SPEC_3} l'$. And, since $l' \in Tot_{SPEC_2}$, by sublemma *.Prop66, $l' \in Tot_{SPEC_3}$. Therefore, if we make t' be l', we have what we wished.

Now, we shall prove that $SPEC_3$ fulfills the second condition of definition *.Prop27. We want to prove that $\forall t_1, t_2 \in Tot_{SPEC_3}$ it is fulfilled that $t_1 \equiv_{SPEC_3} t_2$ if and only if $t_1 \equiv_{Obs_3} t_2$ where $\equiv_{Obs_3}$ is the congruence defined only by the equations belonging to $(E_3)_{Obs}$.

Since $t_1, t_2 \in Tot_{SPEC_3}$, by sublemma *.Prop66, $t_1, t_2 \in Tot_{SPEC_2}$. Therefore, by sublemma *.Prop10, $\phi^*(t_1) = t_1$ i $\phi^*(t_2) = t_2$.

On the other hand, by sublemma *.Prop11, we have that $t_1 \equiv_{SPEC_2} t_2$ if and only if $\phi^*(t_1) \equiv_{SPEC_3} \phi^*(t_2)$. By the result of the last paragraph, we can reduce this to $t_1 \equiv_{SPEC_2} t_2$

if and only if $t_1 \equiv_{SPEC_3} t_2$.

Now, since in $SPEC_2$, the definition of the non-observable sorts does not modify that of the observable ones, we have $t_1 \equiv_{SPEC_2} t_2$ if and only if $t_1 \equiv_{Obs_2} t_2$. By the result of the last paragraph, we have that $t_1 \equiv_{SPEC_3} t_2$ if and only if $t_1 \equiv_{Obs_2} t_2$, where $\equiv_{Obs_2}$ is the congruence defined by the equations belonging to $(E_2)_{Obs}$.

Now, by sublemma *.Prop14, $(E_2)_{Obs} = (E_3)_{Obs}$ and, therefore, $\equiv_{Obs_2}$ is the same congruence as $\equiv_{Obs_3}$. Consequently, we can write $t_1 \equiv_{SPEC_3} t_2$ if and only if $t_1 \equiv_{Obs_3} t_2$, which is what we wished to prove. □

Sublemma *.Prop51 *SPEC*$_1$ and *SPEC*$_3$ share the same definition of the observable sorts.

Proof. We want to prove that both *SPEC*$_1$ and *SPEC*$_3$ fulfill the conditions which definition *.Prop15 has. The first condition is fulfilled by sublemma *.Prop17. *SPEC*$_1$ fulfills the second and the third ones by definition *.Prop16. *SPEC*$_3$ fulfills the third one by sublemma *.Prop50. Therefore, the only part left to be proved is that *SPEC*$_3$ fulfills the second condition of definition *.Prop15.

So, we want to prove that, in *SPEC*$_3$, all the equations which have only variables on their right-hand side belong to $(E_3)_{Obs}$. Let $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow t_1 = t_2$ be an equation which has only variables on its right-hand side (that is to say, $t_1, t_2 \in vars(e)$).

Since all the terms in this equation belong to $T_{\Sigma_{SPEC_3}}$, by sublemma *.Prop12, there are $c'_1, ..., c'_n, d'_1, ..., d'_n, t'_1, t'_2$ such that $c_1 = \phi^*(c'_1), ..., c_n = \phi^*(c'_n), d_1 = \phi^*(d'_1), ..., d_n = \phi^*(d'_n), t_1 = \phi^*(t'_1), t_2 = \phi^*(t'_2)$. That is to say, e can be written as follows: $e : \phi^*(c'_1) = \phi^*(d'_1)$ &...& $\phi^*(c'_n) = \phi^*(d'_n) \Rightarrow \phi^*(t'_1) = \phi^*(t'_2)$.

Since $e \in E_3$, by definition *.Prop9, equation $e' : c'_1 = d'_1$ &...& $c'_n = d'_n \Rightarrow t'_1 = t'_2 \in E_2$. On the other hand, since $t_1 = \phi^*(t'_1)$, $t_2 = \phi^*(t'_2)$ and t_1, t_2 are variables, by sublemma *.Prop80, we have that t'_1, t'_2 are variables, too.

So, we have that e' is a equation which has only variables on its right-hand side. Now, *SPEC*$_1$ and *SPEC*$_2$ share the definition of the observable sorts. Therefore, by definition *.Prop15, $e' \in (E_2)_{Obs}$. Now, by definition of $(E_2)_{Obs}$, this means that $c'_1, ..., c'_n, d'_1, ..., d'_n, t'_1, t'_2 \in Tot_{SPEC_2}$. By applying sublemma *.Prop10, we have that $c'_1 = \phi^*(c'_1), ..., c'_n = \phi^*(c'_n), d'_1 = \phi^*(d'_1), ..., d'_n = \phi^*(d'_n), t'_1 = \phi^*(t'_1), t'_2 = \phi^*(t'_2)$, that is to say, $c_1 = c'_1, ..., c_n = c'_n, d_1 = d'_1, ..., d_n = d'_n, t_1 = t'_1, t_2 = t'_2$. That is to say, e' is the same equation as e.

This means that $e \in (E_2)_{Obs}$ and, by sublemma *.Prop14, $e \in (E_3)_{Obs}$, which is what we wished to prove. □

6 $SPEC_4$ contains all the information which there is in $SPEC_1$ and in $SPEC_3$

We have defined intuitively the reunion of $SPEC_1$ and $SPEC_2$ as the specification which contained the information of $SPEC_1$ and of $SPEC_2$. In this section, we shall prove formally that this statement is true.

Actually, what we shall prove is that $SPEC_4$ contains the information of $SPEC_1$ and of $SPEC_3$. Now, $SPEC_3$ is a renaming of $SPEC_2$, and we have already proved -subsection 4.3- that the operation of renaming preserves the information. Therefore, we have that $SPEC_4$ contains the information of $SPEC_1$ and $SPEC_2$, as has just been stated.

Sublemma *.Prop40. Let $SPEC = (Obs, S, F, E)$ be a behavioral specification. Let $\Sigma' = (Obs', S', F')$ be any signature. Let X be any set of variables. Let $v : X \longrightarrow T_{\Sigma'}$ be an assignment of values . Suppose $p \in T_{\Sigma'}(X)$. Then, it is fulfilled that $v^*(p) \in T_{\Sigma_{SPEC}}$ implies $p \in T_{\Sigma_{SPEC}}(X)$.

Proof. We shall prove this by structural induction on p.

- Induction base.
  - $p \in X$. Then $p \in T_{\Sigma_{SPEC}}(X)$. Therefore, this sublemma is reduced to $v^*(p) \in T_{\Sigma_{SPEC}}$ implies true. Now, this is equivalent to true.
  - $p \in F'$. In this case, by definition of v^*, $v^*(p) = p$. Therefore, this sublemma is reduced to $p \in T_{\Sigma_{SPEC}}$ implies $p \in T_{\Sigma_{SPEC}}(X)$, which is trivially true.
- Induction step. p has the form $\sigma(p_1, ..., p_n)$. Then, $v^*(p)$ is $v^*(\sigma(p_1, ..., p_n)) = \sigma(v^*(p_1), ..., v^*(p_n))$. Since $v^*(p) \in T_{\Sigma_{SPEC}}$, then $\sigma \in F$ and $\forall i \ v^*(p_i) \in T_{\Sigma_{SPEC}}$. By the hypothesis of induction, this means that $p_i \in T_{\Sigma_{SPEC}}(X)$. And, since $\sigma \in F$, then $\sigma(p_1, ..., p_n) \in T_{\Sigma_{SPEC}}(X)$. This is what we wished to prove. □

Comment. This sublemma can be written as follows:

Let $SPEC = (Obs, S, F, E)$ be a behavioral specification. Let $\Sigma' = (Obs', S', F')$ be any signature. Let X be any set of variables. Let $v : X \longrightarrow T_{\Sigma'}$ be an assignment of values . Suppose $p \in T_{\Sigma'}(X)$. Then, it is fulfilled that $p \notin T_{\Sigma_{SPEC}}(X)$ implies $v^*(p) \notin T_{\Sigma_{SPEC}}$

because this formulation is counter-reciprocal w.r.t the previous one.

Sublemma *.Prop55 Suppose $SPEC_4 = SPEC_1 \uplus SPEC_2$ *via* $SPEC_3$. Suppose $t_1, t_2 \in T_{\Sigma_{SPEC_1}}$. If we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$, then we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

Proof. In $SPEC_1$, the definition of non-observable sorts does not modify that of the observable ones. Therefore, if $t_1, t_2 \in Tot_{SPEC_1}$ then, by definition *.Prop27, we have that $t_1 \equiv_{Obs} t_2$, where $\equiv_{Obs}$ is the congruence defined out of the equations of $(E_1)_{Obs}$. Since $(E_1)_{Obs} \subseteq E_1$, then we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

Now, we shall prove the sublemma when $t_1, t_2 \notin Tot_{SPEC_1}$. In order to demonstrate $t_1 \equiv_{SPEC_4} t_2$, we must apply repeatedly the definition of $\equiv_{SPEC_4}$. We refer by "step" to each of these applications. We make an induction on the number of steps which the demonstration of $t_1 \equiv_{SPEC_4} t_2$ has.

If $t_1 \equiv_{SPEC_4} t_2$, by definition of $\equiv_{SPEC_4}$, one of the following four cases may occur[3]:

1. In this case, $t_1 = t_2$. Trivially, since $\equiv_{SPEC_4}$ is reflexive, there is a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

2. In this case, $t_1 \equiv_{SPEC_4} t_2$ because $t_2 \equiv_{SPEC_4} t_1$. Since the demonstration of the latter relationship of congruence has one step less than the former, we can apply the hypothesis of induction on it. We obtain that the demonstration of $t_2 \equiv_{SPEC_4} t_1$ does not use equations that do not belong to E_1. Consequently, by applying the symmetrical property of $\equiv_{SPEC_4}$, we can obtain a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

3. In this case, $t_1 \equiv_{SPEC_4} t_2$ because$t_1 \equiv_{SPEC_4} t_3$ and $t_3 \equiv_{SPEC_4} t_2$. Since the subdemonstrations of $t_1 \equiv_{SPEC_4} t_3$ and $t_3 \equiv_{SPEC_4} t_2$ are shorter (in number of steps) than the demonstration of $t_1 \equiv_{SPEC_4} t_2$, we can apply the hypothesis of induction. So we obtain that ($t_1 \equiv_{SPEC_4} t_3$) and ($t_3 \equiv_{SPEC_4} t_2$) have demonstrations which do not use equations that do not belong to E_1. By applying the transitive property of $\equiv_{SPEC_4}$, we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

4. In this case, $t_1 \equiv_{SPEC_4} t_2$ because there is a equation $e : c_1 = d_1$ &...& $c_n = d_n \Rightarrow p_1 = p_2$, where $e \in E_4$ and an assignment of values $v : X \longrightarrow T_{\Sigma_{SPEC_4}}$ such that

   - $v^*(p_1) = t_1$ and $v^*(p_2) = t_2$, and, moreover
   - $v^*(c_1) \equiv_{SPEC_4} v^*(d_1)$, ..., $v^*(c_n) \equiv_{SPEC_4} v^*(d_n)$

   We can distinguish the following cases:

   - If $e \in E_{new}$. As we have seen in definition *.Prop16, all the equations belonging to E_{new} have the form $l_1 = l_2$, where the sort of l_2 is γ. If e belongs to E_{new}, then

[3] Case 1 belongs to the induction base, cases 2 and 3 belong to the induction step. Case 4 belongs to the induction base when the equation applied is unconditional or, otherwise, it belongs to the induction step

p_2 must be of the sort γ. Now, $\gamma \notin S$ and, therefore, $p_2 \notin T_{\Sigma_{SPEC_1}}(vars(e))$. Since $v^*(p_2) = t_2$, by sublemma *.Prop40, $t_2 \notin T_{\Sigma_{SPEC_1}}$. This is a contradiction and, consequently, this case is impossible.

- $e \in (E_3 \backslash (E_3)_{Obs})$. We refer by s to the sort of p_1 and p_2. We can distinguish two cases:
  - $s \in Obs$. Be v as it may, we have that $v^*(p_1), v^*(p_2)$ must be of sort s, too. And, therefore, t_1, t_2 must be of sort s. Since $t_1, t_2 \in T_{\Sigma_{SPEC_1}}$ then, $t_1, t_2 \in (T_{\Sigma_{SPEC_1}})_s$ with $s \in Obs$.

    By definition *.Prop15, in $SPEC_1$ the definition of non-observable sorts does not modify that of the observable ones. Consequently, by second condition of definition *.Prop27, there are $l_1, l_2 \in (Tot_{SPEC_1})$ such that $l_1 \equiv_{SPEC_1} t_1$ and $l_2 \equiv_{SPEC_1} t_2$. Now, since $E_1 \subset E_4$ and $t_1 \equiv_{SPEC_4} t_2$, we can obtain $l_1 \equiv_{SPEC_4} l_2$, because $\equiv_{SPEC_4}$ is transitive.

    Now, since $l_1, l_2 \in Tot_{SPEC_1}$, by second condition of definition *.Prop27, $l_1 \equiv_{Obs} l_2$, where $\equiv_{Obs}$ is the congruence defined by the equations belonging to $(E_1)_{Obs}$. Now, $(E_1)_{Obs} \subseteq E_1$ and, therefore, $l_1 \equiv_{SPEC_1} l_2$. Since $l_1 \equiv_{SPEC_1} t_1$ and $l_2 \equiv_{SPEC_1} t_2$, $t_1 \equiv_{SPEC_1} t_2$, because $\equiv_{SPEC_1}$ is transitive. That is to say, there is a demonstration of $t_1 \equiv_{SPEC_4} t_2$, which does not use equations that do not belong to E_1.

  - $s \notin Obs$. By sublemma *.Prop51, $SPEC_1$ and $SPEC_3$ share the same definition of observable sorts. By definition *.Prop15, this means that all the equations which have only variables on their right-hand side must belong to $(E_3)_{Obs}$. Now, if $p_1, p_2 \in vars(e)$, there must be an equation which has only variables on its right-hand side and does not belong to $(E_3)_{Obs}$. This is a contradiction. Therefore, either $p_1 \notin vars(e)$ or $p_2 \notin vars(e)$. Suppose, without loss of generality, that $p_1 \notin vars(e)$.

    Since p_1 is not a variable, p_1 has the form $\sigma(l_1, ..., l_n)$ where $\sigma \in (F_3)_{w1..wn,s}$. Since $s \notin Obs$, by sublemma *.Prop41, $\sigma \notin (F_{w1..wn,s} \cap (F_3)_{w1..wn,s})$. Since, $\sigma \in (F_3)_{w1..wn,s}$, then $\sigma \notin F_{w1..wn,s}$ and, therefore, $p_1 \notin T_{\Sigma_{SPEC_1}}(vars(e))$. By sublemma *.Prop40, be v as it may, $v^*(p_1) \notin T_{\Sigma_{SPEC_1}}$. And, consequently, $t_1 \notin T_{\Sigma_{SPEC_1}}$, which is a contradiction. Therefore, this case is impossible.
- $e \in E_1$ By the hypothesis of induction, we have subdemonstrations of $v^*(c_1) \equiv_{SPEC_4} v^*(d_1), ..., v^*(c_n) \equiv_{SPEC_4} v^*(d_n)$ which does not use equations that do not belong to E_1. By applying e to these subdemonstrations, we obtain a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_1.

We prove next that we have seen all the possible cases. Since $e \in E_4$ and, by definition *.Prop16, $E_4 = E_1 \cup E_3 \cup E_{new}$. By properties of sets, we have $E_4 = (E_1 \backslash (E_1)_{Obs}) \cup (E_1)_{Obs} \cup (E_3 \backslash (E_3)_{Obs}) \cup (E_3)_{Obs} \cup E_{new}$. Since, by sublemma *.Prop17, $(E_1)_{Obs} = (E_3)_{Obs}$, then $E_4 = (E_1 \backslash (E_1)_{Obs}) \cup (E_1)_{Obs} \cup (E_3 \backslash (E_3)_{Obs}) \cup E_{new}$. And, by properties of sets, this is equivalent to $E_4 = E_1 \cup (E_3 \backslash (E_3)_{Obs}) \cup E_{new}$. Consequently, we have seen all the possible cases. □

Sublemma *.Prop56 Suppose $SPEC_4 = SPEC_1 \uplus SPEC_2\, via\, SPEC_3$. Suppose $t_1, t_2 \in T_{\Sigma_{SPEC_3}}$. If we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$, we have a demonstration of $t_1 \equiv_{SPEC_4} t_2$ which does not use equations that do not belong to E_3.

Proof. It is obtained out of the proof of sublemma *.Prop55, by interchanging the roles of $SPEC_1$ and $SPEC_3$. □

Lemma *.Prop57. Let there be $SPEC_4 = SPEC_1 \uplus SPEC_2\; via\; SPEC_3$. The following two statements are fulfilled:

- $\forall\, t, u \in T_{\Sigma_{SPEC_1}}, \quad t \equiv_{SPEC_1} u$ if and only if $t \equiv_{SPEC_4} u$.
- $\forall\, t, u \in T_{\Sigma_{SPEC_3}}, \quad t \equiv_{SPEC_3} u$ if and only if $t \equiv_{SPEC_4} u$.

Proof. We shall prove the first statement. The second one is proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_3$ and by using sublemma *.Prop56 instead of sublemma *.Prop55.

The fact that $t \equiv_{SPEC_1} u$ implies $t \equiv_{SPEC_4} u$ is obvious, because all the equations which appear in $SPEC_1$, also appear in$SPEC_4$. Consequently, the congruence defined by $SPEC_4$ includes that of $SPEC_4$.

Since $t, u \in T_{\Sigma_{SPEC_1}}$, if $t \equiv_{SPEC_4} u$, we have a demonstration of $t \equiv_{SPEC_4} u$ which does not use equations that do not belong to E_1 , by sublemma *.Prop55. In other words, we have a demonstration of $t \equiv_{SPEC_1} u$. Consequently, the right-to-left implication is proved. □

Comment. Lemma *.Prop57 can be written in the following way:

- $\forall\, t, u \in T_{\Sigma_{SPEC_1}}, \quad \varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(u)$ if and only if $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$
- $\forall\, t, u \in T_{\Sigma_{SPEC_3}}, \quad \varepsilon_{T_{SPEC_3}}(t) = \varepsilon_{T_{SPEC_3}}(u)$ if and only if $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$

7 Reason for the existence of E_{new}

We have seen in the previous section that the reason why a reunion includes the equations belonging to E_1 and E_3 is that, by doing so, the reunion contains all the information of $SPEC_1$ and of $SPEC_2$.

Now, which is the reason why a reunion includes the equations belonging to E_{new}? The answer to this question is that the equations of E_{new} enable us to express the fact that a term is the "renaming" of another one, that is, that a given t_2 is equal to $\phi^*(t_1)$. Moreover, E_{new} enables us to express this in an inductive theorem.

Specifically, we want to prove that it is fulfilled that $trans(t_1, t_2) \equiv_{SPEC_4} yes$ if and only if $t_2 \equiv_{SPEC_4} \phi^*(t_1)$. This statement has left-to-right and right-to-left implications, which will be proved in separate subsections.

7.1 Left-to-right implication.

In order to prove the left-to-right implication, the concept of trans-irreducibility will be of much help to us.

Definition *.PropA1 Let there be $l \in T_{\Sigma_{SPEC_4}}$. We say that l is trans-irreducible (T-I, hence) if it contains a subterm $trans_m(s,t)$ (with $m \in$ S) such that $\forall\, w \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that, either *not* $w \equiv_{SPEC_4} s$ or *not* $\phi^*(w) \equiv_{SPEC_4} t$

In other words, a term l is not T-I, if for any of its subterms which have the form $trans_m(s,t)$ (with $m \in$ S), $\exists\, w \in T_{\Sigma_{SPEC_1}}$ such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$

Comment. For the sake of clearness, we refer by $trans(s,t)$ to $trans_m(s,t)$, because the subindex of *trans* can be deduced easily (since m is the sort of s and t).

Sublemma *.PropAM Let $l \in T_{\Sigma_{SPEC_4}}$ be a T-I term and let u be the result that we obtain by applying an equation $e \in E_4$ to l. Then u is T-I.

Proof. Since l is T-I, there must be a subterm $trans(s,t)$ such that $\forall\, w \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that, either *not* $w \equiv_{SPEC_4} s$ or *not* $\phi^*(w) \equiv_{SPEC_4} t$.

We can distinguish the following cases:

- If e is applied on any subterm different to $trans(s,t)$, u will preserve the same subterm $trans(s,t)$ and, therefore, u will be T-I, too.
- If e is applied on $trans(s,t)$, the following three cases may occur:
  - It is applied on t. Then u will contain a subterm of the form $trans(s,t')$ where $t' \equiv_{SPEC_4} t$. Since l is T-I, for each w which belongs to $T_{\Sigma_{SPEC_1}}$, one of the following conditions must occur:
    - *not* $w \equiv_{SPEC_4} s$.

  * *not* $\phi^*(w) \equiv_{SPEC_4} t$. Since $t' \equiv_{SPEC_4} t$ and $\equiv_{SPEC_4}$ is transitive, if $\phi^*(w) \equiv_{SPEC_4} t'$ was fulfilled, then it would be fulfilled that $\phi^*(w) \equiv_{SPEC_4} t$ too. Now, this is a contradiction. Therefore, *not* $\phi^*(w) \equiv_{SPEC_4} t'$ is proved.

  That is to say, u is T-I, since it contains a subterm of the form $trans(s, t')$ such that $\forall\, w \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that, either *not* $w \equiv_{SPEC_4} s$ or *not* $\phi^*(w) \equiv_{SPEC_4} t'$.

- It is applied on s. Then u will contain a subterm of the form $trans(s', t)$ where $s' \equiv_{SPEC_4} s$. Since l is T-I, for each w which belongs to $T_{\Sigma_{SPEC_1}}$, one of the following conditions must occur:

  * *not* $w \equiv_{SPEC_4} s$. Since $s' \equiv_{SPEC_4} s$ and $\equiv_{SPEC_4}$ is transitive, if $w \equiv_{SPEC_4} s'$, then it would fulfilled that $w \equiv_{SPEC_4} s$. Now, this is a contradiction. Therefore, *not* $w \equiv_{SPEC_4} s'$ is proved.
  * *not* $\phi^*(w) \equiv_{SPEC_4} t$.

  That is to say, u is T-I, since it contains a subterm of the form $trans(s', t)$ such that $\forall\, w \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that, either *not* $w \equiv_{SPEC_4} s'$ or *not* $\phi^*(w) \equiv_{SPEC_4} t$.

- It is applied on the whole subterm $trans(s, t)$. Since this subterm begins by *trans*, only two equations can be applied.

  * $trans(\sigma, \phi(\sigma)) = yes$ This equation is impossible to apply, since its application entails that $\exists\, w \in T_{\Sigma_{SPEC_1}}$ such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$ (in this case, $w = s$). Now, we have chosen $trans(s, t)$ as the subterm of l such that *not* $\exists\, w \in T_{\Sigma_{SPEC_1}}$ which fulfills that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$. So we have a contradiction here and this case is impossible.

  * $trans(\sigma(s_1, ..., s_n), \phi(\sigma)(t_1, ..., t_n)) = plus(trans(s_1, t_1), plus(trans(s_2, t_2)... trans(s_n, t_n)...))$. Then the subterm of l onto which the equation is applied must have the form $trans(\sigma(v^*(s_1), ..., v^*(s_n)), \phi(\sigma)(v^*(t_1), ..., v^*(t_n))$ and the resulting subterm of u has the form $plus(trans(v^*(s_1), v^*(t_1)), plus(trans(v^*(s_2), v^*(t_2))...trans(v^*(s_n), v^*(t_n))...))$

    Suppose that, for any i, $\exists\, w_i$ such that $w_i \equiv_{SPEC_4} v^*(s_i)$ and $\phi^*(w_i) \equiv_{SPEC_4} v^*(t_i)$. Then, it is fulfilled that $\sigma(w_1, ..., w_n) \equiv_{SPEC_4} \sigma(v^*(s_1), ..., v^*(s_n))$ and
    $\phi(\sigma(w_1, ..., w_n)) = \phi(\sigma)(\phi^*(w_1), ..., \phi^*(w_n)) \equiv_{SPEC_4} \phi(\sigma)(v^*(t_1), ..., v^*(t_n))$. That is to say, if we make w be $\sigma(w_1, ..., w_n)$, then $\exists\, w$ such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$. Now, this is a contradiction, since we had chosen $trans(s, t)$ as the subterm which fulfilled that there is no w such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$.

Therefore, we can deduce that there is a i such that, either *not* $w_i \equiv_{SPEC_4} v^*(s_i)$ or *not* $\phi^*(w_i) \equiv_{SPEC_4} v^*(t_i)$. Now, since $trans(v^*(s_i), v^*(t_i))$ is a subterm of u, it is fulfilled that u is T-I, which is what we wished to prove.□

Sublemma *.PropA2 Let $l \in T_{\Sigma_{SPEC_4}}$ be a term and let u be the result that we obtain by applying an equation $e \in E_4$ to t. Then, if l is not T-I, neither is u.

Proof. We shall prove this by contradiction. Suppose that l is not T-I and u is T-I. Therefore, when we apply the equation e, we must introduce a subterm $trans(s,t)$ such that $\forall\, w \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that, either *not* $w \equiv_{SPEC_4} s$ or *not* $\phi^*(w) \equiv_{SPEC_4} t$. Now, there are only two equations which can introduce a subterm *trans*:

1. $trans(\sigma, \phi(\sigma)) = yes$, in inverse order. If we apply this equation, by making w be σ, it is fulfilled that $\exists\, w \in T_{\Sigma_{SPEC_1}}$ such that $w \equiv_{SPEC_4} s$ and, moreover, $\phi^*(w) \equiv_{SPEC_4} t$. That is to say, the subterm introduced does not fulfill the conditions which must be fulfilled and, in consequence, u is not T-I.

2. $trans(\sigma(s_1, ..., s_n), \phi(\sigma)(t_1, ..., t_n)) = plus(trans(s_1, t_1), plus(trans(s_2, t_2)... \; trans(s_n, t_n)$ ...)), in inverse order. Then, the subterm which is introduced by the equation has the form $trans(\sigma(v^*(s_1), ..., v^*(s_n)), \phi(\sigma)(v^*(t_1), ..., v^*(t_n))$ and the corresponding subterm of l has the form
$plus(trans(v^*(s_1), v^*(t_1)), plus(trans(v^*(s_2), v^*(t_2))...trans(v^*(s_n), v^*(t_n))...))$

   Now, l is not T-I. That is, for any i, $\exists\, w_i \in T_{\Sigma_{SPEC_1}}$ such that $w_i \equiv_{SPEC_4} v^*(s_i)$ and $\phi^*(w_i) \equiv_{SPEC_4} v^*(t_i)$. Then, we have that, if we make w be $\sigma(w_1, ..., w_n)$, it is fulfilled that $\exists\, w \in T_{\Sigma_{SPEC_1}}$ such that $w \equiv_{SPEC_4} \sigma(v^*(s_1), ..., v^*(s_n))$ and $\phi^*(w) \equiv_{SPEC_4} \phi(\sigma)(v^*(t_1), ..., v^*(t_n))$. Therefore, u is not T-I. □

Corollary *.PropA3 Suppose $l \in T_{\Sigma_{SPEC_4}}$ and let u be the result which we obtain by applying an equation $e \in E_4$ to t. Then, l is T-I if and only if u is T-I.

Proof. It is the immediate consequence of sublemma *.PropAM and of the counter-reciprocal of sublemma *.PropA2. □

Sublemma *.PropA4 Let there be $t, u \in T_{\Sigma_{SPEC_4}}$ such that $t \equiv_{SPEC_4} u$. Then, it is fulfilled that t is T-I if and only if u is T-I.

Proof. If $t \equiv_{SPEC_4} u$, by definition of $\equiv_{SPEC_4}$, one of the following four cases may occur:

1. In this case, $t = u$. The sublemma can be reduced to "t is T-I if and only if t is T-I", which is trivial.

2. In this case, $t \equiv_{SPEC_4} u$ because $u \equiv_{SPEC_4} t$. Since the demonstration of the last relationship of congruence is shorter (in number of steps) than that of the first one, we can apply the hypothesis of induction on it. We obtain that "u is T-I if and only if t is T-I". Since the double implication is symmetrical, we have what we wished.

3. In this case, $t \equiv_{SPEC_4} u$ because $t \equiv_{SPEC_4} v$ and $v \equiv_{SPEC_4} u$. Since the subdemonstrations of $t \equiv_{SPEC_4} v$ and $v \equiv_{SPEC_4} u$ are shorter (in number of steps) than that of $t \equiv_{SPEC_4} u$, we can apply the hypothesis of induction on them. So we have "t is T-I if and only if v is T-I" and "v is T-I if and only if u is T-I". Since the double implication is transitive, we have what we wished.

4. In this case, $t \equiv_{SPEC_4} u$ because u is the term which we obtain when we apply an equation e to t. Now, by corollary *.PropA3, we have that "t is T-I if and only if u is T-I". □

Lemma *.PropAC. Let there be $s_1, ..., s_n, t_1, ..., t_n \in T_{\Sigma_{SPEC_4}}$. It is fulfilled that:

$trans(\sigma(s_1, ..., s_n), \phi(\sigma)(t_1, ...t_n)) \equiv_{SPEC_4} yes$ implies that, for any i, $\exists\, w_i \in T_{\Sigma_{SPEC_1}}$ such that $w_i \equiv_{SPEC_4} s_i$ and $\phi^*(w_i) \equiv_{SPEC_4} t_i$

Proof. By applying an equation belonging to E_{new}, we have that $trans(\sigma(s_1, ..., s_n), \phi(\sigma)(t_1, ...t_n)) \equiv_{SPEC_4} plus(trans(s_1, t_1), plus(..., trans(s_n, t_n)))$. But, since $\equiv_{SPEC_4}$ is transitive, we obtain that $plus(trans(s_1, t_1), plus(..., trans(s_n, t_n))) \equiv_{SPEC_4} yes$. Now, since yes is not T-I (because it does not contain any subterm which has the form $trans(s, t)$), then $plus(trans(s_1, t_1), plus(..., trans(s_n, t_n)))$ is not T-I, by sublemma *.PropA4.

Then, since the last term is not T-I, any subterm of this term which has the form $trans(s, t)$ must fulfill that $\exists\, w$ such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$. This means that, for any i, $\exists\, w_i \in T_{\Sigma_{SPEC_1}}$ such that $w_i \equiv_{SPEC_4} s_i$ and $\phi^*(w_i) \equiv_{SPEC_4} t_i$, which is what we wished to prove. □

7.2 Right-to-left implication.

Now, we shall prove the right-to-left implication of the statement stated at the beginning of this section.

Sublemma *.PropA6 $plus(yes, plus(yes, ..., yes)...) \equiv_{SPEC_4} yes$.
Proof. We shall prove this by induction on the structure of the term.

- Induction base. In this case, we must prove that $plus(yes, yes) \equiv_{SPEC_4} yes$. Now, this is trivial, since there is an equation $plus(yes, yes) = yes$.

- Induction step. In this case, the term is $plus(yes, plus(yes, ..., yes)...)$. By applying the hypothesis of induction on $plus(yes, ..., yes)$, we have that $plus(yes, ..., yes) \equiv_{SPEC_4}$ yes. Therefore, by applying the property of congruence on the term, we have $plus(yes, plus(yes, ..., yes)...) \equiv_{SPEC_4} plus(yes, yes)$. Now, as we have seen, $plus(yes, yes) \equiv_{SPEC_4} yes$. Consequently, since $\equiv_{SPEC_4}$ is transitive, we obtain what we wished. □

Lemma *.PropA5. Let there be $t, u \in T_{\Sigma_{SPEC_1}}$. It is fulfilled that

$u \equiv_{SPEC_4} \phi^*(t)$ implies $trans(t, u) \equiv_{SPEC_4} yes$

Proof. We shall prove this by structural induction on t.

Induction base. If $t \in (F_4)_{\lambda,s}$, then $\phi^*(t) = \phi(t)$ and, therefore, $u \equiv_{SPEC_4} \phi(t)$. Now, there is the equation $trans(t, \phi(t)) = yes$, then $trans(t, \phi(t)) \equiv_{SPEC_4} yes$. Since $u \equiv_{SPEC_4} \phi(t)$ and $\equiv_{SPEC_4}$ is a congruence, then we have $trans(t, u) \equiv_{SPEC_4} yes$.

Induction step. If t has the form $\sigma(t_1, ..., t_n)$, then $\phi^*(t)$ has the form $\phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n))$. Therefore, $trans(t, \phi^*(t))$ is, by applying the equation $trans(\sigma(x_1, ..., x_n), \phi(\sigma)(y_1, ..., y_n)) =$ $plus(trans(x_1, y_1), plus(trans(x_2, y_2)...trans(x_n, y_n)...))$, the term $plus(trans(t_1, \phi^*(t_1)),$ $plus(trans(t_2, \phi^*(t_2))...trans(t_n, \phi^*(t_n))...))$. By the hypothesis of induction, for any i it is fulfilled that $trans(t_i, \phi^*(t_i)) \equiv_{SPEC_4} yes$. Then $trans(t, \phi^*(t))$ $\equiv_{SPEC_4} plus(yes, plus(yes, ..., yes)...)$. By sublemma *.PropA6, $trans(t, \phi^*(t)) \equiv_{SPEC_4} yes$. And, since $u \equiv \phi^*(t)$, then $trans(t, u) \equiv_{SPEC_4} yes$. □

8 Proof of soundness.

In this section, we shall prove the soundness of our method. That is, we shall prove that, if some inductive theorems are fulfilled in the initial algebra of $SPEC_4$, then $SPEC_1$ and $SPEC_2$ are eval-equivalent (and, therefore, behaviorally equivalent, as we have proved in section 3). This property is stated in theorem *.PropV1.

8.1 Useful properties.

Now, let us prove some properties which will help us to prove theorem *.PropV1.

Lemma *.Prop22. Let there be $SPEC_4 = SPEC_1 \uplus SPEC_2$ *via* $SPEC_3$. There are two bijections β_1 and β_3, such that :

- $\forall t \in T_{\Sigma_{SPEC_1}}$, it is fulfilled that $\varepsilon_{T_{SPEC_4}}(t) = \beta_1(\varepsilon_{T_{SPEC_1}}(t))$.

- $\forall\, t \in T_{\Sigma_{SPEC_1}}$, it is fulfilled that $\varepsilon_{T_{SPEC_4}}(\phi^*(t)) = \beta_3(\varepsilon_{T_{SPEC_3}}(\phi^*(t)))$.

Proof. Let us prove the first statement.

Since T_{SPEC_1} and T_{SPEC_4} are initial algebras, their members are classes of equivalence of the congruence that has been defined among their ground terms. So, we can define β_1 as:

$$\forall\, t \in T_{\Sigma_{SPEC_1}}, \quad \beta_1([t]_{\equiv_{SPEC_1}}) = [t]_{\equiv_{SPEC_4}}.$$

Notice that β_1 is well defined, since it does not depend on the representative of the class that we are using, because, by lemma *.Prop21:

$$t \equiv_{SPEC_1} u \text{ if and only if } t \equiv_{SPEC_4} u$$

That is to say,

$$[t]_{\equiv_{SPEC_1}} = [u]_{\equiv_{SPEC_1}} \text{ if and only if } [t]_{\equiv_{SPEC_4}} = [u]_{\equiv_{SPEC_4}}$$

Starting from the definition of β_1, and since T_{SPEC_1} and T_{SPEC_4} are initial algebras, we have:

$$\forall\, t \in T_{\Sigma_{SPEC_1}}, \text{ it is fulfilled that } \varepsilon_{T_{SPEC_4}}(t) = \beta_1(\varepsilon_{T_{SPEC_1}}(t)).$$

and this is what we wished to prove.

Now, we want to prove that β_1 is a bijection. First, let us see that β_1 is injective, that is:

$$\forall [t]_{\equiv_{SPEC_1}}, [u]_{\equiv_{SPEC_1}} \in (T_{SPEC_1})_{Obs} \text{ it is fulfilled that}$$
$$\beta_1([t]_{\equiv_{SPEC_1}}) = \beta_1([u]_{\equiv_{SPEC_1}}) \text{ implies } [t]_{\equiv_{SPEC_1}} = [u]_{\equiv_{SPEC_1}}$$

By definition of β_1, this is equivalent to:

$$\forall [t]_{\equiv_{SPEC_1}}, [u]_{\equiv_{SPEC_1}} \in (T_{SPEC_1})_{Obs} \text{ it is fulfilled that}$$
$$[t]_{\equiv_{SPEC_4}} = [u]_{\equiv_{SPEC_4}} \text{ implies } [t]_{\equiv_{SPEC_1}} = [u]_{\equiv_{SPEC_1}}$$

By definition of quotient term algebra, this is equivalent to:

$$\forall\, t, u \in (T_{\Sigma_{SPEC_1}})_{Obs} \text{ it is fulfilled that } t \equiv_{SPEC_4} = u \text{ implies } t \equiv_{SPEC_1} = u$$

Which is true, by sublemma *.Prop57.

Now, let us prove that β_1 is exhaustive, that is:

$$\forall [t]_{\equiv_{SPEC_4}} \in (T_{SPEC_4})_{Obs}^{Tot}, \exists [u]_{\equiv_{SPEC_1}} \in (T_{SPEC_1})_{Obs} \text{ such that } \beta_1([u]_{\equiv_{SPEC_1}}) = [t]_{\equiv_{SPEC_4}}$$

By definition of β_1, this is equivalent to:

$$\forall [t]_{\equiv_{SPEC_4}} \in (T_{SPEC_4})^{Tot}_{Obs}, \exists [u]_{\equiv_{SPEC_1}} \in (T_{SPEC_1})_{Obs} \text{ such that } [u]_{\equiv_{SPEC_4}} = [t]_{\equiv_{SPEC_4}}$$

By definition of quotient term algebra:

$$\forall\, t \in Tot_{SPEC_4}, \exists\, u \in (T_{\Sigma_{SPEC_1}})_{Obs} \text{ such that } u \equiv_{SPEC_4} t$$

By sublemma *.PropCI, $Tot_{SPEC_4} \subseteq (T_{\Sigma_{SPEC_1}})_{Obs}$. Consequently, if we make u be t, then it is proved.

Now, let us prove the second statement. By repeating the same procedure as we have done with the first one, we can prove the following:

$$\forall\, t \in T_{\Sigma_{SPEC_3}}, \text{ it is fulfilled that } \varepsilon_{T_{SPEC_4}}(t) = \beta_3(\varepsilon_{T_{SPEC_3}}(t))$$

Now, since $T_{\Sigma_{SPEC_1}} = T_{\Sigma_{SPEC_2}}$, because the two specifications have the same signature and, on the other hand, $SPEC_3 \in Renam(SPEC_2)$, we obtain by sublemma *.Prop10:

$$\forall\, t \in T_{\Sigma_{SPEC_1}}, \phi^*(t) \in T_{\Sigma_{SPEC_3}}$$

Therefore, by replacing t by ϕ^*, we have:

$$\forall\, t \in T_{\Sigma_{SPEC_1}}, \text{ it is fulfilled that } \varepsilon_{T_{SPEC_4}}(\phi^*(t)) = \beta_3(\varepsilon_{T_{SPEC_3}}(\phi^*(t)))$$

and this is what we wished to prove. □

Lemma *.Prop23. If $SPEC_1$ and $SPEC_2$ are eval-equivalent, for any $t \in T_{\Sigma_{SPEC1_s}}$, with $s \in Obs$ it is fulfilled that $\beta_3(\beta(\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t))))) = \varepsilon_{T_{SPEC_4}}(t)$,
(where β_1, β_3 are the bijections of lemma *.Prop22, β is the bijection of lemma *.Prop13 and φ is the bijection of eval-equivalence).

Proof. First, let us prove $\forall\, t' \in Tot_{SPEC1}$. Since $SPEC_1$ and $SPEC_2$ are eval-equivalent, we obtain:

$$\varphi(\varepsilon_{T_{SPEC_1}}(t')) = \varepsilon_{T_{SPEC_2}}(t')$$

By applying lemma *.Prop13:

$$\varphi(\varepsilon_{T_{SPEC_1}}(t')) = \beta^{-1}(\varepsilon_{T_{SPEC_3}}(\phi^*(t')))$$

By applying lemma *.Prop22:

$$\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t'))) = \beta^{-1}(\beta_3^{-1}(\varepsilon_{T_{SPEC_4}}(\phi^*(t'))))$$

Now, since $t' \in Tot_{SPEC_1}$, then, by sublemma *.Prop10, $\phi^*(t') = t'$. So we have:

$$\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t'))) = \beta^{-1}(\beta_3^{-1}(\varepsilon_{T_{SPEC_4}}(t')))$$

If we pass all bijections to the left-hand side of the equation:

$$\beta_3(\beta(\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t'))))) = \varepsilon_{T_{SPEC_4}}(t')$$

This is what we wished to prove. Now, we shall prove this for any $t \in T_{\Sigma_{SPEC1_s}}$. Since, in $SPEC_1$, the observable sorts are defined when we define the non-observable ones, there is a $t' \in Tot_{SPEC1}$ such that:

$$\varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(t')$$

Now, as we have proved above,

$$\beta_3(\beta(\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t'))))) = \varepsilon_{T_{SPEC_4}}(t')$$

And, since, by lemma *.Prop21, $\forall\, t, u \in T_{\Sigma_{SPEC_1}}, \varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(u)$ if and only if $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$. By replacing in the previous expression, we obtain:

$$\beta_3(\beta(\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(t))))) = \varepsilon_{T_{SPEC_4}}(t)$$

And this is what we wished to prove. □

8.2 Core of the proof of soundness.

Now, let us prove the theorem which states the soundness of our method.

Theorem *.PropV1 The statement

$\forall\, \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall\, x_1, y_1 \in vars(w_1)$; ...; $x_n, y_n \in vars(w_n)$ it is fulfilled that $T_{SPEC_4} \models trans(\sigma(x_1, ..., x_n), \phi(\sigma)(y_1, ..., y_n)) = yes \Rightarrow \sigma(x_1, ..., x_n) = \phi(\sigma)(y_1, ..., y_n)$

implies the statement

$SPEC_1$ and $SPEC_2$ are eval-equivalent

Proof. We shall begin with the first statement.

$\forall\, \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall\, x_1, y_1 \in vars(w_1)$; ...; $x_n, y_n \in vars(w_n)$ it is fulfilled that $T_{SPEC_4} \models trans(\sigma(x_1, ..., x_n), \phi(\sigma)(y_1, ..., y_n)) = yes \Rightarrow \sigma(x_1, ..., x_n) = \phi(\sigma)(y_1, ..., y_n)$

By definition of fulfilment of an equation in a given algebra, the last expression is equivalent to:

$\forall\, \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall\, x_1, y_1 \in vars(w_1)$; ...; $x_n, y_n \in vars(w_n)$,
$\forall\, v : X \longrightarrow T_{SPEC_4}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(v^*(trans(\sigma(x_1, ..., x_n), \phi(\sigma)(y_1, ..., y_n)))) = \varepsilon_{T_{SPEC_4}}(v^*(yes))$
implies $\varepsilon_{T_{SPEC_4}}(v^*(\sigma(x_1, ..., x_n))) = \varepsilon_{T_{SPEC_4}}(v^*(\phi(\sigma)(y_1, ..., y_n)))$

Now, since the only variables of the expressions $\sigma(x_1, ..., x_n)$ and $\phi(\sigma)(x_1, ..., x_n)$ are the x_i's, we can apply the definition of v^* and we have:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, t_1, ..., s_n, t_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(trans(\sigma(s_1, ..., s_n), \phi(\sigma)(t_1, ..., t_n)))) = \varepsilon_{T_{SPEC_4}}(yes)$
implies $\varepsilon_{T_{SPEC_4}}(\sigma(s_1, ..., s_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(t_1, ..., t_n))$

Since this property is fulfilled for all the t_i's, it must be fulfilled specifically when $t_i = \phi^*(s_i)$.

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(trans(\sigma(s_1, ..., s_n), \phi(\sigma)(\phi^*(s_1), ..., \phi^*(s_n))) = \varepsilon_{T_{SPEC_4}}(yes)$
implies $\varepsilon_{T_{SPEC_4}}(\sigma(s_1, ..., s_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(\phi^*(s_1), ..., \phi^*(s_n)))$

By definition of ϕ^*, we have:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(trans(\sigma(s_1, ..., s_n), \phi^*(\sigma(s_1, ..., s_n))) = \varepsilon_{T_{SPEC_4}}(yes)$
implies $\varepsilon_{T_{SPEC_4}}(\sigma(s_1, ..., s_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(s_1, ..., s_n))$

By lemma *.PropA5, we have that the left-hand side of the implication is fulfilled and, therefore, the last equality can be reduced to:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
true implies $\varepsilon_{T_{SPEC_4}}(\sigma(s_1, ..., s_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(s_1, ..., s_n))$

That is,

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(\sigma(s_1, ..., s_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(s_1, ..., s_n))$

By lemma*.Prop22, we obtain that this is equivalent to:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(s_1, ..., s_n))) = \beta_3(\varepsilon_{T_{SPEC_3}}(\phi^*(\sigma(s_1, ..., s_n))))$

And, since $SPEC_3 \in Renam(SPEC_2)$, by lemma *.Prop13:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(s_1, ..., s_n))) = \beta_3(\beta(\varepsilon_{T_{SPEC_2}}(\sigma(s_1, ..., s_n))))$

Now, β, β_1, β_3 are bijections. The bijections are inverses and their inverses are bijections too. So we have:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\beta^{-1}(\beta_3^{-1}(\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(s_1, ..., s_n))))) = \varepsilon_{T_{SPEC_2}}(\sigma(s_1, ..., s_n))$

Since $\beta^{-1}, \beta_1, \beta_3^{-1}$ are bijections, their composition is a bijection too. So, $\varphi = \beta^{-1} \circ \beta_3^{-1} \circ \beta_1$ is a bijection. Moreover, since $\beta : (T_{SPEC_2})_{Obs} \longrightarrow (T_{SPEC_3})_{Obs}$, $\beta_1 : (T_{SPEC_1})_{Obs} \longrightarrow (T_{SPEC_4})^{Tot}_{Obs}$ and $\beta_3 : (T_{SPEC_3})_{Obs} \longrightarrow (T_{SPEC_4})^{Tot}_{Obs}$, then $\varphi : (T_{SPEC_1})_{Obs} \longrightarrow (T_{SPEC_2})_{Obs}$. Therefore, we obtain:

There is a bijection φ between $(T_{SPEC_1})_{Obs}$ and $(T_{SPEC_2})_{Obs}$ such that
$\forall \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall s_1, \ldots, s_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\varphi(\varepsilon_{T_{SPEC_1}}(\sigma(s_1, \ldots, s_n))) = \varepsilon_{T_{SPEC_2}}(\sigma(s_1, \ldots, s_n))$

Which is the definition of eval-equivalence between $SPEC_1$ and $SPEC_2$, according to lemma *.PropEV. □

9 Proof of completeness.

In this section, we shall prove the completeness of our method. That is, we prove that, if $SPEC_1$ and $SPEC_2$ are eval-equivalent (and, therefore, behaviorally equivalent, as we have proved in section 3), then some inductive theorems are fulfilled in the initial algebra of $SPEC_4$. This property is stated in theorem *.PropV2.

9.1 Useful properties.

First, we shall prove some properties which will be useful in order to prove theorem *.PropV2.

Lemma *.PropB1 The statement

$\forall s_1, \ldots, s_n, t_1, \ldots, t_n \in T_{\Sigma_{SPEC_4}}$
$t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge \ldots \wedge t_n \equiv_{SPEC_4} \phi^*(s_n)$ implies $\sigma(s_1, \ldots, s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$

implies the statement

$\forall s_1, \ldots, s_n, t_1, \ldots, t_n \in T_{\Sigma_{SPEC_4}}$
$trans(\sigma(s_1, \ldots, s_n), \phi(\sigma)(t_1, \ldots, t_n)) \equiv_{SPEC_4} yes$ implies $\sigma(s_1, \ldots, s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$

Proof. Suppose that the first statement is fulfilled. Suppose that it is fulfilled that $trans(\sigma(s_1, \ldots, s_n), \phi(\sigma)(t_1, \ldots, t_n)) \equiv_{SPEC_4} yes$. We want to prove that it is fulfilled that $\sigma(s_1, \ldots, s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$.

Since it is fulfilled that $\sigma(s_1, \ldots, s_n), \phi(\sigma)(t_1, \ldots, t_n)) \equiv_{SPEC_4} yes$, by sublemma *.PropAC, we have that, for any i, $\exists\, w_i \in T_{\Sigma_{SPEC_1}}$ such that $w_i \equiv_{SPEC_4} s_i$ and $\phi^*(w_i) \equiv_{SPEC_4} t_i$. On the one hand, this implies that $\sigma(w_1, \ldots, w_n) \equiv_{SPEC_4} \sigma(s_1, \ldots, s_n)$, by property of congruence.

On the other hand, since, for any i, it is fulfilled that $\phi^*(w_i) \equiv_{SPEC_4} t_i$, we can apply the first statement. We obtain that $\sigma(w_1, ..., w_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, ..., t_n)$. Then, since $\equiv_{SPEC_4}$ is transitive, we have that $\sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, ..., t_n)$, which is what we wished to prove. □

Lemma *.PropB2 The statement

$$\forall s_1, ..., s_n \in T_{\Sigma_{SPEC_4}}$$
$$\sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(s_1), ..., \phi^*(s_n))$$

implies the statement

$$\forall s_1, ..., s_n, t_1, ..., t_n \in T_{\Sigma_{SPEC_4}}$$
$$t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge ... \wedge t_n \equiv_{SPEC_4} \phi^*(s_n) \text{ implies } \sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$$

Proof. Suppose that the first statement is fulfilled. Suppose that $t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge ... \wedge t_n \equiv_{SPEC_4} \phi^*(s_n)$. We want to prove that $\sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$.

Since the first statement is fulfilled, we have that $\sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(s_1), ..., \phi^*(s_n))$. Now, since $t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge ... \wedge t_n \equiv_{SPEC_4} \phi^*(s_n)$, we obtain that $\phi(\sigma)(t_1, ..., t_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(s_1), ..., \phi^*(s_n))$, by property of congruence. Since $\equiv_{SPEC_4}$ is transitive, we have that $\sigma(s_1, ..., s_n) \equiv_{SPEC_4} \phi(\sigma)(t_1, .., t_n)$, which is what we wanted to prove. □

Lemma *.PropB3 The following statements are equivalent:

- $\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$
- $\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

Proof. We shall prove that the first statement implies the second one. The reciprocal implication is trivial, since $T_{\Sigma_{SPEC_1}} \subset T_{\Sigma_{SPEC_4}}$.

Now, let us focus on the second statement. In order that expression $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$ makes sense, $\sigma(t_1, ..., t_n)$ must belong to $T_{\Sigma_{SPEC_1}}$, because if it does not, ϕ^* cannot be applied. (Remember that, by definition *.Prop9, the domain of ϕ^* is $T_{\Sigma_{SPEC_1}}$).

Since $\sigma(t_1, ..., t_n) \in T_{\Sigma_{SPEC_1}}$, then $t_1, ..., t_n$ must belong to $T_{\Sigma_{SPEC_1}}$. Consequently, the second statement can be written as follows:

$\forall \sigma \in F_{w_1...w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

Now, this is the first statement. Therefore, the lemma is proved. □

9.2 Core of the proof of completeness

Now, we shall prove the theorem which states the completeness of our method.

Theorem *.PropV2 The statement

$SPEC_1$ and $SPEC_2$ are eval-equivalent

implies the statement

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall x_1, y_1 \in vars(w_1)$; ...; $x_n, y_n \in vars(w_n)$ it is fulfilled that $T_{SPEC_4} \models trans(\sigma(x_1, ..., x_n), \phi(\sigma)(y_1, ..., y_n)) = yes \Rightarrow \sigma(x_1, ..., x_n) = \phi(\sigma)(y_1, ..., y_n)$

Proof. Let us start from the definition of eval-equivalence between $SPEC_1$ and $SPEC_2$. By lemma *.PropEV, there is a bijection φ between $(T_{SPEC_1})_{Obs}$ and $(T_{SPEC_2})_{Obs}$ such that:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\varphi(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, ..., t_n))) = \varepsilon_{T_{SPEC_2}}(\sigma(t_1, ..., t_n))$

By applying lemma *.Prop13, we obtain:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\varphi(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, ..., t_n))) = \beta^{-1}(\varepsilon_{T_{SPEC_3}}(\phi^*(\sigma(t_1, ..., t_n))))$

Via lemma *.Prop22:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)))) = \beta^{-1}(\beta_3^{-1}(\varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))))$

If we pass all bijections to the left-hand side, we have

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\beta_3(\beta(\varphi(\beta_1^{-1}(\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)))))) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

By lemma *.Prop23, this is equivalent to:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

Now, by lemma *.PropB3, this is equivalent to:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, ..., t_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

Via the definition of ϕ^*, we obtain:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, \ldots, t_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{T_{SPEC_4}}(\sigma(t_1, \ldots, t_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(\phi^*(t_1), \ldots, \phi^*(t_n))))$

By lemma *.PropB2, the last statement implies the following one:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, \ldots, t_n, u_1, \ldots, u_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{SPEC_4}(u_1) = \varepsilon_{SPEC_4}(\phi^*(t_1)) \wedge \ldots \wedge \varepsilon_{SPEC_4}(u_n) = \varepsilon_{SPEC_4}(\phi^*(t_n))$ implies
$\varepsilon_{SPEC_4}(\sigma(t_1, \ldots, t_n)) = \varepsilon_{SPEC_4}(\phi(\sigma)(u_1, .., u_n))$

By lemma *.PropB1, this statement implies the following one:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall t_1, \ldots, t_n, u_1, \ldots, u_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{SPEC_4}(trans(\sigma(t_1, \ldots, t_n), \phi(\sigma)(u_1, \ldots, u_n))) = \varepsilon_{SPEC_4}(yes)$ implies
$\varepsilon_{SPEC_4}(\sigma(t_1, \ldots, t_n)) = \varepsilon_{SPEC_4}(\phi(\sigma)(u_1, .., u_n))$

Via the definition of v^*, this is equivalent to:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall x_1, y_1 \in vars(w_1); \ldots; x_n, y_n \in vars(w_n)$
$\forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$\varepsilon_{SPEC_4}(v^*(trans(\sigma(x_1, \ldots, x_n), \phi(\sigma)(y_1, \ldots, y_n)))) = \varepsilon_{SPEC_4}(v^*(yes))$ implies
$\varepsilon_{SPEC_4}(v^*(\sigma(x_1, \ldots, x_n))) = \varepsilon_{SPEC_4}(v^*(\phi(\sigma)(y_1, .., y_n)))$

Which, by definition of fulfilment of an equation, is equivalent to:

$\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall x_1, y_1 \in vars(w_1); \ldots; x_n, y_n \in vars(w_n)$ it is fulfilled that
$T_{SPEC_4} \models trans(\sigma(x_1, \ldots, x_n), \phi(\sigma)(y_1, \ldots, y_n)) = yes \Rightarrow \sigma(x_1, \ldots, x_n) = \phi(\sigma)(y_1, .., y_n)$

And this is the second statement, so the theorem is proved. □.

10 End of proof.

In this section, we shall make the last steps of our proof. Actually, the hardest part of our work is already done and now we must only draw some trivial conclusions.

Theorem *.Prop24. Both statements are equivalent:

- $\forall \sigma \in F_{w_1 \ldots w_n,s}$ *with* $s \in Obs$, $\forall x_1 \in vars(w_1), \ldots, x_n \in vars(w_n)$ it is fulfilled $T_{SPEC_4} \models \sigma(x_1, \ldots, x_n) = \phi(\sigma)(x_1, \ldots, x_n)$.
- $SPEC_1$ and $SPEC_2$ are eval-equivalent.

Proof. It is a corollary of theorem *.PropV1 and of theorem *.PropV2. □

Comment. Hence, we shall call the first statement of this theorem "fundamental property".

Theorem *.Prop25. Both statements are equivalent:

- (Fundamental property). $\forall \sigma \in F_{w_1 \dots w_n, s}$ *with* $s \in Obs$, $\forall x_1 \in vars(w_1), \dots, x_n \in vars(w_n)$ it is fulfilled $T_{SPEC_4} \models \sigma(x_1, \dots, x_n) = \phi(\sigma)(x_1, \dots, x_n)$.
- $SPEC_1$ and $SPEC_2$ are behaviorally equivalent.

Proof. It is a corollary of theorem *.Prop24 and of theorem *.Prop8. By the former the fundamental property is equivalent to eval-equivalence. By the latter, eval-equivalence is, in this case, equivalent to behavioral equivalence. □

11 Conclusions

Taking up the result obtained in the previous section:

Theorem *.Prop25. Both statements are equivalent:

- (Fundamental property). $\forall \sigma \in F_{w_1 \dots w_n, s}$ *with* $s \in Obs$, $\forall x_1 \in vars(w_1), \dots, x_n \in vars(w_n)$ it is fulfilled $T_{SPEC_4} \models \sigma(x_1, \dots, x_n) = \phi(\sigma)(x_1, \dots, x_n)$.
- $SPEC_1$ and $SPEC_2$ are behaviorally equivalents.

This means that proving the behavioral equivalence between SPEC1 and SPEC2 is equivalent to proving the fundamental property in initial algebra of SPEC4 (where SPEC4 is the reunion of SPEC1 and SPEC2 via some arbitrary renaming SPEC3).

Now, the fundamental property is only a set of theorems and there are techniques for proving the fulfillment or non-fulfillment of theorems in initial algebras. These are the systems for theorem proof via inductionless induction, which are based on rewriting techniques. By submitting the theorems of the fundamental property to these systems we can know whether two algebraic specifications are behaviorally equivalent or not.

We can prove that the algorithm which builds SPEC4 is of a quadratic complexity in relation to the number of equations. We can also prove (as will be done in a subsequent article) that, if we direct the equations of TRANS in a given way, if the rewriting systems derived from SPEC1 and SPEC2 are canonical, so is the system of SPEC4.

The lines of research in the near future are several. The work is on removing the necessary conditions which say that specifications must share the definition of the observable sets and that there are no equations which have only variables on their right-hand side. Furthermore, the intention is to prove what was stated in the previous paragraph.

The possible applications of this theoretical work would be two. On the one hand, it would be useful in order to build tools for verifying the equivalence between programming modules and, more precisely, between the classes of object-oriented programming. Thus, if we have these classes formally specified, we shall be able to tell when a new class is useful or, otherwise, when it only adds redundant information to our collection.

On the other hand, this paper is part of the thesis entitled "Automatic Verification of Object-Oriented Programming". The underlying idea is as follows: for a given program and its algebraic specification, we shall build a specification which is equivalent to the program. Then, we shall see via automatic deduction whether the two specifications are behaviorally equivalent and, if it is so, the program will then be correct. As one may see, this proof of the behavioral equivalence would be along the lines proposed in this paper.

12 References

[EhM85] EHRIG,H. MAHR,B. *Fundamentals of Algebraic Specification.* ETACS monographs on theoretical computer science, Springer-Verlag, Berlin,1985.

[Mey88] MEYER,B. *Object-Oriented Software Construction.* C.A.R. Hoare Series Editor, Prentice-Hall, University Press, Cambridge, 1988.

[Niv87] NIVELA,P. *Semántica de Comportamiento en Lenguajes de Especificación* PhD thesis, directed by Fernando Orejas Valdés, Barcelona, 1987. Universitat Politècnica de Catalunya. Facultat d'Informàtica.

[Rei81] REICHEL,H. *Behavioural equivalence -a unifying concept for initial and final specification methods* Proceedings third Hungarian Computer Science Conference. Budapest (1981) 27-39.

[Rei84] REICHEL,H. *Behavioral validity of equations in abstract data types* Contributions to General Algebra 3, Proceedings of the Vienna Conference, Verlag B. G.Teubner, Sttugart (1985) 301-324.

[SaT85] SANNELLA,D. TARLECKY, A. *On observational equivalence and algebraic specification* Journal of Computer and System Science, 34 (1987) 150-178.

[SaW83] SANNELLA,D WIRSING,M. *A kernel language for algebraic specification and implementation* Proceedings International Conference on Foundations of Computation Theory. Sweden. Springer LNCS 158 (1983) 413-427.

[Wir90] WIRSING, M. *Algebraic Specification* In Handbook of Theoretical Computer Science, volume B. Elsevier, Amsterdam. 677-788