Automatic deduction of the behavioral equivalence between two algebraic specifications.

Vicent-Ramon Palasí Lallana.

Barcelona, June 21, 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 a 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 types defined in it can be divided into observable and non-observable ones. Non-observable types 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 condition necessary for this to be workable is that the observable types must be already defined when we define the non-observable ones. That is, that both specifications must share the same definition of the observable types and that any operation which returns an observable result is completely defined on the observable type. We are here producing an article which removes the need for the first condition. As to the second condition, it is absolutely reasonable, for its non-fulfilment would mean that the specification had not been well defined.

The structure of this article 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, 4 and 5 we introduce the concepts and theorems which are needed to prove what is intended by this paper. Finally, in section 6, 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 skip sections 2, 3, 4, 5 and 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 $F = \{F_{w,s}\}_{(w,s) \in S^* \times S}$. The members of $F_{w,s}$ are called function symbols with domain w and sort s.

Likewise, 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 $\sigma \in F_{w_1...w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, ..., t_n \in T_{\Sigma_{w_n}}$, then $\sigma(t_1, ..., t_n) \in T_{\Sigma_s}$.

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 $\sigma_A : A_{s_1}, ..., A_{s_n} \longrightarrow A_s$.

σ_A is called interpretation of function symbol *sigma* 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 $\sigma \in F_{w_1...w_n,s}$, $t_1 \in T_{\Sigma_{w_1}}, ..., t_n \in T_{\Sigma_{w_n}}$, then $\varepsilon_A(\sigma(t_1, ..., t_n)) = \sigma_A(\varepsilon_A(t_1), ..., \varepsilon_A(t_n))$.

$\varepsilon_A(t)$ 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 Σ be a behavioral signature. Let X be a set of variables. We refer by Σ-equation to a triple (X, t_1, t_2), where $t_1, t_2 \in T_{\Sigma_s}(X)$, with $s \in S$. Whenever we write $t_1 = t_2$ (or, also, $e : t_1 = t_2$), we shall mean the equation $e = (X, t_1, t_2)$. The set of variables of an equation e will be called $vars(e)$.

Definition. Let A be a Σ-algebra. Let X be a set of variables. An application $v : X \longrightarrow A$ will be called assignment of values.

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 $e : t_1 = t_2$ if $\forall\, v : vars(e) \longrightarrow A$ it is fulfilled that: $\varepsilon_A(v^*(t_1)) = \varepsilon_A(v^*(t_2))$. If a Σ-algebra A satisfies an equation $e : t_1 = t_2$, we shall write $A \models t_1 = t_2$

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: $\varepsilon_A(w^*(t_1)) = \varepsilon_A(w^*(t_2))$.

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 $sig(SPEC) = \Sigma$ 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 $(T_{\Sigma_{SPEC}})_s$.

Definition. Given $t_1, t_2 \in T_{\Sigma_{SPEC}}$, we say that t_2 is obtained by applying an equation $e : p_1 = p_2$ to t_1 if there is an assignment of values $v : vars(e) \longrightarrow T_{\Sigma_{SPEC}}$ such that $v^*(p_1) = t_1$ and $v^*(p_2) = t_2$.

Definition. Given two terms $t, u \in T_{\Sigma_{SPEC}}$, we refer to the congruence being defined by specification SPEC as $\equiv_{SPEC}$. That is to say, we say that t $\equiv_{SPEC}$ u if there is a succession of terms belonging to $T_{\Sigma_{SPEC}} : t_1, ..., t_n$ such that t_{i+1} is obtained by applying an equation of E to t_i and, in addition, $t_1 = t$ and $t_n = u$.

Definition. Suppose a behavioral specification SPEC. We divide the set $T_{\Sigma_{SPEC}}$ into classes of equivalence defined by the relation $\equiv_{SPEC}$. We refer to the classs 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 $SPEC$, we can prove that $\forall\, t, u \in T_{\Sigma_{SPEC}}$ it is fulfilled that: $\varepsilon_{T_{SPEC}}(t) = \varepsilon_{T_{SPEC}}(u)$ if and only if $t \equiv_{SPEC} u$.

Definition. Given a behavioral specification SPEC=(Obs,S,F,E), we refer by $(T_{SPEC})_{Obs}(X)$ to the set of terms such that all their subterms are observable. That is to say, $t \in (T_{SPEC})_{Obs}(X)$ 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 $t_1, ..., tn \in (T_{SPEC})_{Obs}(X)$.

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

Definition. Given a behavioral specification SPEC=(Obs,S,F,E), we refer by E_{Obs} to the set of equations $e : t1 = t2$ such that $t1, t2 \in (T_{SPEC})_{Obs}(X)$.

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 1. 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:

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

Sublemma 2. 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,

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

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 3. Given two Σ-algebras A and B. We define $f : A_{Obs} \longrightarrow B_{Obs}$ in the following way: $\forall\, a \in A_{Obs}, \quad f(a) = \varepsilon_B(g)$ 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 $\geq$ 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 all the 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. $\square$

Lemma 4. 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 3, 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 2 guarantees that g exists. If there are several possible "g", 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 3, 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, 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. □

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

Lemma 5. 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 3, if $t \in T_\Sigma$ then $f^*(t) = t$, it is fulfilled that:

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

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

The next theorem summarizes what is proved above.

Theorem 6 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 4 and 5. $\square$

Definition 7 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.

Theorem 8 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 easily inferred from theorem 6, if we keep in mind that T_{SPEC_1} and T_{SPEC_2} are finitely generated because they are initial algebras. $\square$

4 Renamings

In this section, we introduce the concept of "renaming", which is useful to define that of reunion. We also describe some properties that are useful in next proofs.

Definition 9. 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:

- $\forall\, \sigma,\ \forall\, w1, .., wn, s \in S; \qquad \sigma \in (F_1)_{w1 \ldots wn,s}$ if and only if $\phi(\sigma) \in (F_2)_{w1 \ldots wn,s}$.
- $\forall\, \sigma \in (F_1)_{w1 \ldots wn,s},\ \forall\, w1, .., wn, s \in S; \qquad \phi(\sigma) = \sigma$ if and only if $w1, .., wn, s \in Obs$.
- For each equation $e : t_1 = t_2; \qquad t_1 = t_2 \in E_1$ if and only if $\phi^*(t_1) = \phi^*(t_2) \in E_2$, where $\phi^* : T_{\Sigma_{SPEC_1}}(X) \longrightarrow T_{\Sigma_{SPEC_2}}(X)$ is defined in the following way:
  - 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 $\phi^*(x) = \phi(\sigma)(\phi^*(t1), ..., \phi^*(tn))$

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.

Sublemma 10 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 $t \in (T_{SPEC_1})_{Obs}$ 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 9, $\phi(t) \in (F_2)_{\lambda,s}$ and, therefore, $\phi^*(t) = \phi(t) \in T_{\Sigma_{SPEC_2}}$.

  On the other hand, if $t \in (T_{SPEC_1})_{Obs}$ then $s \in Obs$ and, by the definition 9, $\phi^*(t) = \phi(t) = t$.

- **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 9, $\phi(\sigma) \in F_2$ 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 $t \in (T_{SPEC_1})_{Obs}$, then $\sigma \in (F_1)_{w1 \ldots wn,s}$, where $w1, ..., wn, s \in Obs$ and all the $t_i \in (T_{SPEC_1})_{Obs}$. Then, by definition 9, $\phi(\sigma) \in (F_2)_{w1 \ldots wn,s}$ 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 11 If $SPEC_2 \in Renam(SPEC_1)$,

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

Proof. We shall 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 in an analogous way.

By definition of $\equiv_{SPEC_1}$, there is a succession $l_1, ..., l_n$ such that each l_{i+1} is obtained by applying an equation $e_i \in E_1$ to l_i and, moreover, $l_1 = t_1$ and $l_n = t_2$.

Suppose that e_i has the form $p_1 = p_2$. Then, by definition 9, the equation $\phi^*(p_1) = \phi^*(p_2) \in E_2$. Obviously, this equation allows to obtain $\phi^*(l_{i+1})$ out of $\phi^*(l_i)$.

Consequently, we can build a succession $\phi^*(l_1), ..., \phi^*(l_n)$ where each $\phi^*(l_{i+1})$ is obtained by applying equations belonging to E_2 to $\phi^*(l_i)$ and $\phi^*(l_1) = \phi^*(t_1)$ and $\phi^*(l_n) = \phi^*(t_2)$. Therefore, $\phi^*(t_1) \equiv_{SPEC_2} \phi^*(t_2)$, which is what we wished to prove. $\Box$

Sublemma 12. 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. We shall prove this by structural induction.

- **Induction base.** If $t_2 \in T_{\Sigma_{SPEC_2}}$ and $t_2 \in (F_2)_{\lambda,s}$, then, by definition 9, there is a c such that $c \in (F_1)_{\lambda,s}$ and $\phi(c) = t_2$. Therefore, if we make t_1 be c, then it is proved.
- **Induction step.** t_2 has the form $\sigma(l_1, ..., l_n)$, where all the $l_i \in T_{\Sigma_{SPEC_2}}$. Via the hypothesis of induction, $\exists\, c_1, ..., c_n$, such that, for all i, $\phi^*(c_i) = l_i$. Since, by definition 9, there is a c 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 that t_1 be $c(c_1, ..., c_n)$, then it is proved. $\Box$

Lemma 13. 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, 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 11, $t_1 \equiv_{SPEC_1} t_2$ 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 11, 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 12:

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

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. $\square$

Sublemma 14. If $SPEC_2 \in Renam(SPEC_1)$, then $(E_1)_{Obs} = (E_2)_{Obs}$.

Proof. We must prove that $e \in (E_1)_{Obs} \Longleftrightarrow e \in (E_2)_{Obs}$. We shall prove the first implication. The other one can be analogously proved.

Since $e \in (E_1)_{Obs}$, this means that e has the form $p_1 = p_2$, where $p_1, p_2 \in (T_{SPEC_1})_{Obs}$.

By definition 9, $\phi^*(p_1) = \phi^*(p_2) \in E_2$. Now, by sublemma 10, this means that $p_1 = p_2 \in E_2$. Since p_1, p_2 are totally observable, then $e : p_1 = p_2 \in (E_2)_{Obs}$. $\square$

5 Reunions.

In this section, we introduce the concept of "reunion" and demonstrate that proving the behavioral equivalence between two specifications is equivalent to proving some theorems on their reunion.

5.1 Definition of reunion.

Definition 15. Suppose two specifications $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2)$. We say that "their observable sorts are defined when we define the non-observable ones" if the following conditions are fulfilled:

- The two specifications share the same definition of the observable sorts, that is to say, $(E_1)_{Obs} = (E_2)_{Obs}$.
- Any operator with observable result is completely defined over the observable sorts, that is to say, $\forall\, t \in T_{\Sigma_{SPEC_1}}$, there is a $t' \in (T_{SPEC1})_{Obs}$ such that, $\varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(t')$.

Definition 16. Suppose two specifications $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2)$ such that their observable sorts are defined when we define the non-observable ones. Given a specification $SPEC_3 = (Obs, S, F_3, E_3)$, such that $SPEC_3 \in Renam(SPEC_2)$. We say that $SPEC_4 = (Obs, S, F_4, E_4)$ 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:

- $F_4 = F \cup F_3$
- $E_4 = E_1 \cup E_3 \cup TRANS$ where TRANS contains all the equations which are of the following form:
  - $\sigma(t1, ..., \phi(f)(l1, .., ln), ..., tn) = \sigma(t1, ..., f(l1, .., ln), ..., tn)$.
  - $\phi(\sigma)(t1, ..., f(l1, .., ln), ..., tn) = \phi(\sigma)(t1, ..., \phi(f)(l1, .., ln), ..., tn)$

for all $\sigma \in F_{w1..wn,s}$ such that $s \in Obs$ and for all $f \in F_{y1...yn,wi}$. In these formulae, $t1, ..., tn, l1, ..., ln$ are variables of sort $w1, ..., wn, y1, ..., yn$ respectively and ϕ is the renaming bijection between $SPEC_2$ and $SPEC_3$.

Sublemma 17. 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 16, the observable sorts of $SPEC_1$ and $SPEC_2$ are defined when we defined the non-observable ones. Since, by definition 16, $SPEC_3 \in Renam(SPEC_2)$, then, by sublemma 14, $(E_2)_{Obs} = (E_3)_{Obs}$. Since the equality of sets is transitive, the sublemma is proved. □

5.2 Reason of the existence of TRANS.

In this subsection, we say that proving the equivalence of terms of $SPEC_1$ with the corresponding (by renaming) of $SPEC_3$ in the initial algebra of $SPEC_4$ is the same that proving some given theorems on $SPEC_4$ (lemma 20). This result will be used further on in the proof of the central theorem of this paper. The proof of this result is possible thanks to the equivalences that are defined by the equations of TRANS.

Sublemma 18. Let there be $SPEC_4 = SPEC_1 \uplus SPEC_2 viaSPEC_3$. $\forall\, \sigma \in F$, $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC1}}$ it is fulfilled that:

- $\phi(\sigma)(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(v_1), ..., \phi^*(v_n))$.
- $\sigma(\phi^*(v_1), ..., \phi^*(v_n)) \equiv_{SPEC_4} \sigma(v_1, ..., v_n)$

Proof. We prove this by structural induction on the terms $\phi(\sigma)(v_1, ..., v_n)$ and $\sigma(\phi^*(v_1), ..., \phi^*(v_n))$, respectively.

- **Induction base.** This term is a constant. In this case, n=0 and, therefore, the previous equations are reduced to $\phi(\sigma) \equiv_{SPEC_4} \phi(\sigma)$ and $\sigma \equiv_{SPEC_4} \sigma$ respectively, which are trivially true.

- **Induction step.** The terms are of the form $\phi(\sigma)(v_1, ..., v_n)$ and $\sigma(\phi^*(v_1), ..., \phi^*(v_n))$, respectively. We can write these terms making explicit the parameters of the v_i. That is to say, each v_i can be written as $f_i(t_{i1}, ..., t_{im_i})$, where $f_i \in F$ and $t_{i1}, ..., t_{im_i} \in T_{\Sigma_{SPEC_1}}$. By applying this plus the definition of ϕ^*, we obtain, respectively:

  $$\phi(\sigma)(f_1(t_{11}, ..., t_{1m_1}), ..., f_n(t_{n1}, ..., t_{nm_n})).$$

  $$\sigma(\phi(f_1)(\phi^*(t_{11}), ..., \phi^*(t_{1m_1})), ..., \phi(f_n)(\phi^*(t_{n1}), ..., \phi^*(t_{nm_n}))).$$

  By applying repeatedly the equations of TRANS on all the subterms of these terms, we have, respectively:

$$\phi(\sigma)(\phi(f_1)(t_{11},...,t_{1m_1}),...,\phi(f_n)(t_{n1},...,t_{nm_n})).$$

$$\sigma(f_1(\phi^*(t_{11}),...,\phi^*(t_{1m_1})),...,f_n(\phi^*(t_{n1}),...,\phi^*(t_{nm_n}))).$$

By applying the hypothesis of induction on all the subterms, we obtain:

$$\phi(\sigma)(\phi(f_1)(\phi^*(t_{11}),...,\phi^*(t_{1m_1})),...,\phi(f_n)(\phi^*(t_{n1}),...,\phi^*(t_{nm_n}))).$$

$$\sigma(f_1(t_{11},...,t_{1m_1})),...,f_n(t_{n1},...,t_{nm_n})).$$

The first equation can be written in the following way (by definition of phi^*):

$$\phi(\sigma)(\phi^*(f_1(t_{11},...,t_{1m_1})),...,\phi^*(f_n(t_{n1},...,t_{nm_n}))).$$

And, since $f_i(t_{i1},...,t_{im_i})$ is v_i, we have:

$$\phi(\sigma)(\phi^*(v_1),...,\phi^*(v_n)).$$

$$\sigma(v_1,...,v_n)$$

That is to say, by applying equations of $SPEC_4$, we have obtained the second terms of the equations we wanted to prove out of their first terms. Therefore,$\phi(\sigma)(v_1,...,v_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(v_1),...,\phi^*(v_n))$, and $\sigma(\phi^*(v_1),...,\phi^*(v_n)) \equiv_{SPEC_4} \sigma(v_1,...,v_n)$. This is what we wished to prove. $\square$

Sublemma 19. $\forall\,\sigma \in F$, $\forall\,v_1,...,v_n \in T_{\Sigma_{SPEC4}}$, $\exists\,l_1,...,l_n \in T_{\Sigma_{SPEC1}}$ such that $\sigma(v_1,...,v_n) \equiv_{SPEC_4} \sigma(l_1,...,l_n)$ i $\phi(\sigma)(v_1,...,v_n) \equiv_{SPEC_4} \phi(\sigma)(l_1,...,l_n)$.

Proof. We shall prove this by structural induction on the terms $\sigma(v_1,...,v_n)$ and $\phi(\sigma)(v_1,...,v_n)$.

- **Induction base.** If these terms are constant, then n=0 and the previous equalities are reduced to $\sigma \equiv_{SPEC_4} \sigma$ and $\phi(\sigma) \equiv_{SPEC_4} \phi(\sigma)$, which are trivially true.

- **Induction step.** If these terms are not constant, then they are of the form:

$$\sigma(v_1,...,v_n).$$

$$\phi(\sigma)(v_1,...,v_n).$$

Where each v_i can be written as $f_i(t_{i1},...,t_{im_i})$, where $f_i \in F_4$ and $t_{i1},...,t_{im_i} \in T_{\Sigma_{SPEC_4}}$. Then, we obtain:

$$\sigma(f_1(t_{11},...,t_{1m_1}),...,f_n(t_{n1},...,t_{nm_n})).$$

$$\phi(\sigma)(f_1(t_{11},...,t_{1m_1}),...,f_n(t_{n1},...,t_{nm_n})).$$

Since $f_i \in F_4$, this means that either $f_i \in F$ or $f_i \in F_3$. To make the notation clearer, we assume, without any loss of generality, that all those f_i's which belong to F are the first k ones. Those f_i's which do not belong to F , belong to F_3 then and, therefore, are of the form $\phi(s_i)$ where $s_i \in F$. So, we have:

$$\sigma(f_1(t_{11},...,t_{1m_1}),...,f_k(t_{k1},...,t_{nm_k}),\phi(s_{k+1})(t_{k+1\,1},...,t_{k+1\,m_{k+1}}),...,\phi(s_n)(t_{n1},...,t_{nm_n})).$$

$$\phi(\sigma)(f_1(t_{11},...,t_{1m_1}),...,f_k(t_{k1},...,t_{nm_k}),\phi(s_{k+1})(t_{k+1\,1},...,t_{k+1\,m_{k+1}}),...,\phi(s_n)(t_{n1},...,t_{nm_n})).$$

By applying the equations of TRANS on the last subterms:

$$\sigma(f_1(t_{11},...,t_{1m_1}),...,f_k(t_{k1},...,t_{nm_k}),s_{k+1}(t_{k+1\,1},...,t_{k+1\,m_{k+1}}),...,s_n(t_{n1},...,t_{nm_n})).$$

$$\phi(\sigma)(f_1(t_{11},...,t_{1m_1}),...,f_k(t_{k1},...,t_{nm_k}),\phi(s_{k+1})(t_{k+1\,1},...,t_{k+1\,m_{k+1}}),...,\phi(s_n)(t_{n1},...,t_{nm_n})).$$

Since $f_1,...,f_k,s_{k+1},...,sn \in T_{\Sigma_{SPEC_1}}$, via hypothesis of induction of the first equality, there is some $l_{ij} \in T_{\Sigma_{SPEC_1}}$, such that the previous expressions are equivalent to the following ones:

$$\sigma(f_1(l_{11},...,l_{1m_1}),...,f_k(l_{k1},...,l_{nm_k}),s_{k+1}(l_{k+1\,1},...,l_{k+1\,m_{k+1}}),...,s_n(l_{n1},...,l_{nm_n})).$$

$$\phi(\sigma)(f_1(l_{11},...,l_{1m_1}),...,f_k(l_{k1},...,l_{nm_k}),\phi(s_{k+1})(l_{k+1\,1},...,l_{k+1\,m_{k+1}}),...,\phi(s_n)(l_{n1},...,l_{nm_n})).$$

Let us take as $l_1,...,l_k$ the first k subterms of the last expression and as $l_{k+1},...,l_n$ the last (n-k) ones. (Notice that, in doing so, all the $l_i \in T_{\Sigma_{SPEC_1}}$).

Consequently, we get:

$$\sigma(l_1,...,l_n)$$

$$\phi(\sigma)(l_1,...,l_n)$$

This has been deduced from the initial expressions by applying the equations of $SPEC_4$. Therefore, $\sigma(v_1,...,v_n) \equiv_{SPEC_4} \sigma(l_1,...,l_n)$ and $\sigma(v_1,...,v_n) \equiv_{SPEC_4} \sigma(l_1,...,l_n)$, which is what we wished to prove. $\Box$

Lemma 20. Both statements are equivalent:

- $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_1}}, \sigma(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(v_1), ..., \phi^*(v_n))$
- $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_4}}, \sigma(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(v_1, ..., v_n)$

Proof. First, we shall prove that the first statement implies the second one. Let us suppose that the first one is fulfilled. We want to know if $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_4}}$ it is fulfilled that:

$\sigma(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(v_1, ..., v_n)$.

By sublemma 19, we have that $\exists\, l_1, ..., l_n \in T_{\Sigma_{SPEC1}}$ such that:

$$\sigma(v_1, ..., v_n) \equiv_{SPEC_4} \sigma(l_1, ..., l_n)$$

$$\phi(\sigma)(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(l_1, ..., l_n)$$

On the other hand, since $l_1, ..., l_n \in T_{\Sigma_{SPEC1}}$, by sublemma 18 we obtain:

$$\phi(\sigma)(l_1, ..., l_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(l_1), ..., \phi^*(l_n))$$

By combining the last equation with the previous one:

$$\phi(\sigma)(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(l_1), ..., \phi^*(l_n))$$

By replacing in the first equation, we have that this one is reduced to:

$$\sigma(l_1, ..., l_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(l_1), ..., \phi^*(l_n))$$

But this last equation is true, since we have assumed that the first statement is fulfilled. Therefore, if the first statement is fulfilled, then the second one is fulfilled, too.

Now, let us suppose that the second statement is fulfilled. We wish to know if $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_1}}$, it is fulfilled that:

$$\sigma(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(\phi^*(v_1), ..., \phi^*(v_n))$$

By sublemma 18, this is equivalent to:

$$\sigma(v_1, ..., v_n) \equiv_{SPEC_4} \phi(\sigma)(v_1, ..., v_n)$$

Now this last expression is fulfilled $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_4}}$, by the second statement. So, the second statement implies the first one and, therefore, the lemma is proved. □

Comment. Lemma 20 can be written in the following way:

Both statements are equivalent:

- $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_1}}, \varepsilon_{T_{SPEC_4}}(\sigma(v_1, ..., v_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(\phi^*(v_1), ..., \phi^*(v_n)))$
- $\forall\, v_1, ..., v_n \in T_{\Sigma_{SPEC_4}}, \varepsilon_{T_{SPEC_4}}(\sigma(v_1, ..., v_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(v_1, ..., v_n))$

because saying $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$ is the same as saying $t \equiv_{SPEC_4}$ u, since T_{SPEC_4} is a initial algebra.

5.3 $SPEC_4$ contains the same information as $SPEC_1$ and $SPEC_3$

In this subsection, we shall prove another intermediate result in our demonstration. This result tells us that there is the same information in both $SPEC_1$ and $SPEC_3$ as in $SPEC_4$. This was what we intended, since we defined intuitively $SPEC_4$ as the specification which contains all the information of both $SPEC_1$ and $SPEC_3$.

Lemma 21. Let there be $SPEC_4 = SPEC_1 \uplus SPEC_2$ *via* $SPEC_3$. Both statements are fulfilled:

- $\forall\, t, u \in T_{\Sigma_{SPEC1}}, \quad t \equiv_{SPEC_1} u$ if and only if $t \equiv_{SPEC_4} u$.
- $\forall\, t, u \in T_{\Sigma_{SPEC3}}, \quad t \equiv_{SPEC_3} u$ 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 $SPEC_3$. Since $(T_{SPEC_1})_{Obs} = (T_{SPEC_3})_{Obs}$, we call both T_{Obs}.

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

Now, we shall prove that $t \equiv_{SPEC_4} u$ implies $t \equiv_{SPEC_1} u$.

If $t \equiv_{SPEC_4} u$, this means that we can arrive from t at u by applying the equations of $SPEC_4$. That is to say, there is a succession of terms belonging to $T_{\Sigma_{SPEC_4}} : t_1, ..., t_n$ such that t_{i+1} is obtained by applying an equation of E_4 to t_i, and, moreover, $t_1 = t$ and $t_n = u$. Among all the possible successions we choose the shortest one and we make an induction on its length .

- **Induction base.** When the length of this succession is 1, then $t_1 = t_n = t = u$. Therefore, since $t \in T_{\Sigma_{SPEC1}}$, $t \equiv_{SPEC_1} t$ and, consequently, $t \equiv_{SPEC_1} u$.
- **Induction step.** When $n > 1$, we focus on the last application of an equation, that transforms t_{n-1} into u. That is to say, there is an equation $e : p_1 = p_2 \in E_4$ and a substitution $v : X \longrightarrow T_{\Sigma_4}$, such that $v^*(p_1) = t_{n-1}$ and $v^*(p_2) = u$.

  Notice that, $v^*(p_2) = u$ and $u \in T_{\Sigma_{SPEC_1}}$. This means that $\forall\, x, v(x) \in T_{\Sigma_{SPEC_1}}$. (This is because if a v(x) does not belong to $T_{\Sigma_{SPEC_1}}$, then there is some subterm of u which does not belong to $T_{\Sigma_{SPEC_1}}$ and, therefore, u does not belong to $T_{\Sigma_{SPEC_1}}$. This is a contradiction. Therefore, $\forall\, x, v(x) \in T_{\Sigma_{SPEC_1}}$).

  We can distinguish the following cases:

  - $e \notin TRANS$. Here we can distinguish the following subcases:

* $p_1, p_2 \in T_{\Sigma_{SPEC_1}}(X)$. Since, as we have seen before,$\forall\, x, v(x) \in T_{\Sigma_{SPEC_1}}$, it is fulfilled that $v^*(p_1), v^*(p_2) \in T_{\Sigma_{SPEC_1}}$ and, therefore, t_{n-1}, u belong to $T_{\Sigma_{SPEC_1}}$ too. That is to say, u is obtained by applying an equation of E_1 to t_{n-1}. By definition of congruence associated to $SPEC_1$, $t_{n-1} \equiv_{SPEC_1} u$. Since, via the hypothesis of induction, $t \equiv_{SPEC_1} t_{n-1}$, we get $t \equiv_{SPEC_1} u$, and this is what we wished to prove.

* $p_2 \in (T_{\Sigma_{SPEC_3}}(X) \backslash T_{Obs}(X))$. If p_2 does not belong to $T_{\Sigma_{SPEC_1}}(X)$, neither does $v^*(p_2)$. And, consequently, $u \notin T_{\Sigma_{SPEC_1}}(X)$. So, this subcase is impossible.

* $p_2 \in (T_{\Sigma_{SPEC_1}}(X) \backslash T_{Obs}(X)), p_1 \in (T_{\Sigma_{SPEC_3}}(X) \backslash T_{Obs}(X))$. Since $p_1 \in (T_{\Sigma_{SPEC_3}}(X) \backslash T_{Obs}(X))$, p_1 cannot belong to $T_{\Sigma_{SPEC_1}}(X)$, and, therefore, $e \notin E_1$. Analogously, $e \notin E_3$. Moreover, e $\notin$ TRANS. In consequence, e $\notin$ E_4 and this subcase is impossible.

* $p_2 \in T_{Obs}(X)$ and $p_1 \in (T_{\Sigma_{SPEC_3}}(X) \backslash T_{Obs}(X))$. If p_1 does not belong to $T_{\Sigma_{SPEC_1}}(X)$, then neither does $v^*(p_1)$. So, we have that $t \notin T_{\Sigma_{SPEC_1}}(X)$. But, via hypothesis of induction, $t \equiv_{SPEC_1} t_{n-1}$, this would mean that a term which does not belong to $T_{\Sigma_{SPEC_1}}(X)$ can be deduced from the equations of $SPEC_1$. This does not make sense. Therefore, this subcase is impossible.

- $e \in TRANS$. Since u belongs to $T_{\Sigma_{SPEC_1}}$, then e must be of the form:

$$\sigma(t1, ..., \phi(f)(l1, .., ln), ..., tn) = \sigma(t1, ..., f(l1, .., ln), ..., tn).$$

where we can see that there is a subterm in t_{n-1} which does not belong to $T_{\Sigma_{SPEC_1}}$. Specifically, it is the subterm resulting from the assignment of values to $\phi(f)(l1, .., ln)$.

On the other hand, this subterm must be introduced at a point in the succession , since the initial term t has no subterm which does not belong to $T_{\Sigma_{SPEC_1}}$. Now, the only way to introduce a subterm like this is by applying the equation:

$$\sigma(t1, ..., \phi(f)(l1, .., ln), ..., tn) = \sigma(t1, ..., f(l1, .., ln), ..., tn).$$

in inverse order, with the same assignment of values as we have seen above. Therefore, the application of these two equations is superfluous since the second one undoes what the first one did.

As a result, the succession of terms is not the shorter one. So this subcase is impossible, too. $\square$

Comment. Lemma 21 can be written in the following way:

- $\forall\, t, u \in T_{\Sigma_{SPEC1}}, \quad \varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(u)$ if and only if $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$
- $\forall\, t, u \in T_{\Sigma_{SPEC3}}, \quad \varepsilon_{T_{SPEC_3}}(t) = \varepsilon_{T_{SPEC_3}}(u)$ if and only if $\varepsilon_{T_{SPEC_4}}(t) = \varepsilon_{T_{SPEC_4}}(u)$

Lemma 22. 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 21:

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

That is to say,

$[t]_{\equiv_{SPEC_1}} = [u]_{\equiv_{SPEC_1}}$ 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:

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

and this is what we wished to prove.

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:

$\forall\, t \in T_{\Sigma_{SPEC_3}}$, 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 10:

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

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

$\forall\, t \in T_{\Sigma_{SPEC_1}}$, 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. $\square$

5.4 Core of the demonstration.

In this subsection, we take the last steps to prove the central theorem of this paper.

Lemma 23. If $SPEC_1$ and $SPEC_2$ are eval-equivalent, for all $t \in T_{\Sigma_{SPEC1_s}}$, 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 22, β is that of lemma 13 and φ is the bijection of eval-equivalence).

Proof. First, let us prove $\forall\, t' \in (T_{SPEC1})_{Obs}$. 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 13:

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

By applying lemma 22:

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

Now, since $t' \in (T_{SPEC1})_{Obs}$, then, by sublemma 10, $\phi^*(t') = t'$. 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 all $t \in T_{\Sigma_{SPEC1_s}}$. Since, in $SPEC_1$, the observable sorts are defined when we define the non-observable ones, there is a $t' \in (T_{SPEC1})_{Obs}$ 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 21, $\forall\, t, u \in T_{\Sigma_{SPEC_1}}, \varepsilon_{T_{SPEC_1}}(t) = \varepsilon_{T_{SPEC_1}}(u)$ 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. □

Theorem 24 Both statements are equivalent:

- $\forall\, \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall\, x_1 \in vars(w_1), ..., x_n \in vars(w_n)$ it is fulfilled that $T_{SPEC_4} \models \sigma(x_1, ..., x_n) = \phi(\sigma)(x_1, ..., x_n)$.
- $SPEC_1$ and $SPEC_2$ are eval-equivalent. (That is to say, there is a bijection φ between $(T_{SPEC_1})_{Obs}$ and $(T_{SPEC_2})_{Obs}$ such that $\forall\, t \in (T_{\Sigma_{SPEC_1}})_s$, where $s \in Obs$ it is fulfilled that $\varphi(\varepsilon_{T_{SPEC_1}}(t)) = \varepsilon_{T_{SPEC_2}}(t))$

Proof. In this proof, we will use a different version of the definition of eval-equivalence. Since $t \in (T_{\Sigma_{SPEC_1}})_s$, where $s \in Obs$, then t must have the form $\sigma(t_1, ..., t_n)$, where n ≥ 0, $\sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$. Consequently, the definition of eval-equivalence can be written in the following way:

> 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 $s \in Obs$, $\forall\, t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$, $\varphi(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, ..., t_n))) = \varepsilon_{T_{SPEC_2}}(\sigma(t_1, ..., t_n))$

First, we want to prove that the first statement implies the eval-equivalence. So we shall begin with the first statement.

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

By thedefinition 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* $s \in Obs$, $\forall\, x_1 \in vars(w_1), ..., x_n \in vars(w_n), \forall\, v : X \longrightarrow T_{SPEC_4}$ is it fulfilled that $\varepsilon_{T_{SPEC_4}}(v^*(\sigma(x_1, ..., x_n))) = \varepsilon_{T_{SPEC_4}}(v^*(\phi(\sigma)(x_1, ..., x_n)))$

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:

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

By lemma 20, the last expression 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 $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi(\sigma)(\phi^*(t_1), ..., \phi^*(t_n)))$

Which is equivalent, by definition of ϕ^*, 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 $\varepsilon_{T_{SPEC_4}}(\sigma(t_1, ..., t_n)) = \varepsilon_{T_{SPEC_4}}(\phi^*(\sigma(t_1, ..., t_n)))$

By lemma 22, we get that 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 $\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, ..., t_n))) = \beta_3(\varepsilon_{T_{SPEC_3}}(\phi^*(\sigma(t_1, ..., t_n))))$

And, since $SPEC_3 \in Renam(SPEC_2)$, by lemma 13:

$\forall \sigma \in F_{w_1 \dots w_n, s}$ *with* $s \in Obs$, $\forall t_1, \dots, t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, \dots, t_n))) = \beta_3(\beta(\varepsilon_{T_{SPEC_2}}(\sigma(t_1, \dots, t_n))))$

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

$\forall \sigma \in F_{w_1 \dots w_n, s}$ *with* $s \in Obs$, $\forall t_1, \dots, t_n \in T_{\Sigma_{SPEC_1}}$ it is fulfilled that
$\beta^{-1}(\beta_3^{-1}(\beta_1(\varepsilon_{T_{SPEC_1}}(\sigma(t_1, \dots, t_n))))) = \varepsilon_{T_{SPEC_2}}(\sigma(t_1, \dots, t_n))$

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. So we have:

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

Which is the definition of eval-equivalence.

Now, we shall prove that eval-equivalence implies the first statement. Starting from the definition of eval-equivalence:

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

By applying lemma 13, we obtain:

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

Via lemma 22:

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

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

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

By lemma 23, this is equivalent to:

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

Via the definition of ϕ^*, we obtain:

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

And, by lemma 20, this produces:

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

Via the definition of v^* and the fact that $T_{\Sigma_{SPEC_1}} \subset T_{\Sigma_{SPEC_4}}$, we have:

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

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

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

And this is the first statement, so the theorem is proved $\square$.

Comment. From now on, we shall call the first statement of this theorem "fundamental property".

Theorem 25. Both statements are equivalent:

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

Proof. It is a corollary of theorem 24 and of theorem 8. By the former the fundamental property is equivalent to eval-equivalence. By the latter, eval-equivalence is, in this case, equivalent to behavioral equivalence. $\square$

6 Conclusions

Taking up the result obtained in the previous section:

Theorem 25. Both statements are equivalent:

- (Fundamental property). $\forall\, \sigma \in F_{w_1 \ldots w_n, s}$ *with* $s \in Obs$, $\forall\, x_1 \in vars(w_1), ..., x_n \in vars(w_n)$ it is fulfilled $T_{SPEC_4} \models \sigma(x_1, ..., x_n) = \phi(\sigma)(x_1, ..., x_n)$.
- $SPEC_1$ 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 condition which says that specifications must share the definition of the observable sets. Furthermore, the intention is to prove what was stated in the previous paragragh.

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.

7 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* Master 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 Internationnal 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