Reduction of Behavioral Equivalence to Inductive Theorems

Vicent-Ramon Palasí Lallana

Report LSI–95–56–R

Reduction of behavioral equivalence to inductive theorems

Vicent-Ramon Palasí Lallana.
Departament de Llenguatges i Sistemes Informàtics.
Universitat Politècnica de Catalunya
e-mail:vicent@lsi.upc.es

Abstract

We shall demonstrate that proving the behavioral equivalence of two algebraic specifications can be reduced to proving a set of inductives theorems. So we can prove automatically this behavioral equivalence by applying automatic deduction techniques such as *proof by consistency.*

1 Introduction.

Behavioral semantics appears as a tool to grasp, in algebraic specification, the concept of module, which is very important in software engineering.

Intuitively, a module is a "blackbox", which has an interface with the outside. Only this interface can be observed: the internal functioning of the module is unknown to us. So two modules are interchangeable (or "equivalent") if their observable behavior is the same, being their non-observable functioning as it may.

This idea of module, which is so useful and intuitive, is difficult to model in the traditional approaches of algebraic specification, such as initial or final semantics.

In fact, these semantics only accept one model (up to isomorphism) of each specification and this is much too restrictive. Specifically, it may occur that two specifications that are intuitively equivalent are not so in these semantics. This produces odd effects, i.e the usual implementation of a stack is not equivalent to its usual algebraic specification.

Behavioral semantics seems to overcome these troubles. In this semantics, the types (called "sorts") are divided into observable and non-observable ones. We are only interested in the observable consequences.

This enables us to define an equivalence which grasps the intuitive notion of equivalence between modules, that is known as "behavioral equivalence", in which two specifications are equivalent if they share the observable consequences.

This paper aims at proving that this equivalence can be managed by automatic deduction techniques. Specifically, we want to prove that the problem of testing if two algebraic specifications are equivalent can be reduced to proving some given theorems in a specific initial algebra. Thus, we obtain a co-semidecision procedure for behavioral equivalence.

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.

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

Definition 2. 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 3. 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 $\sigma \in T_{\Sigma_s}(X)$.

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.*

Definition 10. 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 11. 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 12. 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 ≥ 0, then $v^*(t) = \sigma(v^*(t_1), ..., v^*(t_n))$.

Lemma 13. Let $\Sigma = (Obs, S, F)$ be a behavioral signature, X a set of variables and $v : X \longrightarrow T_\Sigma$. Suppose $s \in S$. If $t \in (T_\Sigma(X))_s$ then $v^*(t) \in (T_\Sigma)_s$.

Lemma 14. 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 $v^*(p) \notin T_{\Sigma_{SPEC}}$

Definition 15. 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 16. If A is finitely generated, the last definition is equivalent to the following one: $\forall\, v : vars(e) \longrightarrow T_\Sigma$ 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 $\varepsilon_A(v^*(t_1)) = \varepsilon_A(v^*(t_2))$.

Definition 17. 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 $(T_{\Sigma_{SPEC}})_s$. We refer by $(T_{\Sigma_{SPEC}})_{Obs}$ to $\{t \mid t \in (T_{\Sigma_{SPEC}})_s \wedge s \in Obs\}$.

Definition 18. 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 by the following four properties:

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 $v^*(p_1) \equiv_{SPEC} v^*(p_2)$

Comments.

- If we want to prove $t \equiv_{SPEC} u$ (for some given $t, u \in T_{\Sigma_{SPEC}}$), we must apply repeatedly these properties until we obtain $t \equiv_{SPEC} u$. We refer to each of these applications of properties by "step" of the demonstration. Analogously, the sequence of steps which proves $t \equiv_{SPEC} u$ is called "demonstration" of $t \equiv_{SPEC} u$ (Beware of not confusing the terms "demonstration" and "proof).

- 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 19. 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 20. 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 21. 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 22. 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 23. We refer by $(T_{SPEC})_{Obs}$ to $\{[t]_{\equiv_{SPEC}} \mid t \in (T_{\Sigma_{SPEC}})_{Obs}\}$.

Definition 24. 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 25. Given a behavioral specification SPEC, we refer by $(T_{SPEC})_{Obs}^{Tot}$ to $\{[t]_{\equiv_{SPEC}} \mid t \in Tot_{SPEC}\}$.

Definition 26. 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 27. Let Σ=(Obs,S,F) be a behavioral signature. Let A and B be two Σ-algebras. Suppose f is a function between A_{Obs} and B_{Obs}. $f^* : T_{\Sigma}(A_{Obs}) \longrightarrow T_{\Sigma}(B_{Obs})$ is defined as follows:

- 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 $f^*(t) = \sigma(f^*(t_1), .., f^*(t_n))$.

Definition 28. 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 29. 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.

Definition 30. 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.

3 Behavioral 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.

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

Sublemma 31. 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,

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

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 32. 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 ≥ 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 33. Let A and B be two Σ-algebras finitely generated. If the following statement is fulfilled

$$\forall s \in Obs \;\; \forall s_1, s_2 \in T_{\Sigma_s}, \;\; \varepsilon_A(s_1) = \varepsilon_A(s_2) \textit{ if and only if } \varepsilon_B(s_1) = \varepsilon_B(s_2) \qquad (1)$$

, then A and B 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 32, 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 31 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

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

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 32, 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 $\varepsilon_A(t)$. By applying statement (1), 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, since statement (1) is fulfilled, 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,

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

Since B is finitely generated,

$$\exists\, g \in T_\Sigma,\ \textit{such that}\ \varepsilon_B(g) = b.$$

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

$$f(\varepsilon_A(g)) = \varepsilon_B(g')\ \textit{where}\ \varepsilon_A(g') = \varepsilon_A(g)$$

Since statement (1) is fulfilled:

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

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

$$f(\varepsilon_A(g)) = b$$

which is what we wished to prove. □

Lemma 34. Let A and B two Σ-algebras. If A and B are behaviorally equivalent, then the following statement is fulfilled

$$\forall\, s \in Obs\ \ \forall\, s_1, s_2 \in T_{\Sigma_s},\ \ \varepsilon_A(s_1) = \varepsilon_A(s_2)\ \textit{if and only if}\ \varepsilon_B(s_1) = \varepsilon_B(s_2) \qquad (2)$$

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

$$\forall t \in T_{\Sigma_s},\ with\ s \in Obs,\quad f(\varepsilon_A(t)) = \varepsilon_B(t). \tag{3}$$

Now, since f is a behavioral isomorphism, then f is a bijection. Since f is a bijection, it is fulfilled that:

$$\forall s_1, s_2 \in T_{\Sigma_s},\ with\ s \in Obs,\quad \varepsilon_A(s_1) = \varepsilon_A(s_2) \text{ if and only if } f(\varepsilon_A(s_1)) = f(\varepsilon_A(s_2))$$

By statement (3), this is equivalent to:

$$\forall s_1, s_2 \in T_{\Sigma_s},\ with\ s \in Obs,\quad \varepsilon_A(s_1) = \varepsilon_A(s_2) \text{ if and only if } \varepsilon_B(s_1) = \varepsilon_B(s_2)$$

Which is statement (2). □

Theorem 35 The two following statements are equivalent:

- A and B are behaviorally equivalent.
- $\forall s \in Obs\ \ \forall s_1, s_2 \in T_{\Sigma_s},\ \ \varepsilon_A(s_1) = \varepsilon_A(s_2)$ *if and only if* $\varepsilon_B(s_1) = \varepsilon_B(s_2)$

Proof. It is a corollary of lemma 33 and of lemma 34. □

Theorem 36 Let $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2$ be two behavioral specifications with the same signature. $SPEC_1$ and $SPEC_2$ are behaviorally equivalent if and only if the following property is fulfilled:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(s_1 \equiv_{SPEC_2} s_2)$

Proof. This theorem is inferred easily from theorem 35, if we keep in mind that T_{SPEC_1} and T_{SPEC_2} are finitely generated because they are initial algebras. □

4 T-Renamings

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

4.1 Definition of T-renaming

In this subsection, the concept of T-renaming is defined. Intuitively, a specification $SPEC_1$ is a T-renaming of another specification $SPEC_2$ if we can obtain $SPEC_2$ from $SPEC_1$ by changing the names of all the sorts and all the function symbols.

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

Definition 37. We say that a behavioral specification $SPEC_2 = (Obs_2, S_2, F_2, E_2)$ is a T-renaming of another specification $SPEC_1 = (Obs_1, S_1, F_1, E_1)$ if there are two bijections $\theta : S_1 \longrightarrow S_2$ and $\phi : F_1 \longrightarrow F_2$ such that:

1. $s \in Obs_1$ if and only if $\theta(s) \in Obs_2$
2. $S_1 \cap S_2 = \varnothing$
3. $\forall \sigma,\ \forall w1, .., wn, s \in S;\qquad \sigma \in (F_1)_{w1...wn,s}$ if and only if $\phi(\sigma) \in (F_2)_{\theta(w1)...\theta(wn),\theta(s)}$.
4. $F_1 \cap F_2 = \varnothing$
5. For any equation $e : c_1 = d_1 \ \&...\&\ c_n = d_n \Rightarrow p_1 = p_2;\qquad e \in E_1$ 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 $\phi^*(x) = \phi(\sigma)(\phi^*(t1), ..., \phi^*(tn))$

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

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

Sublemma 38 Let there be $SPEC_2 \in T - Renam(SPEC_1)$. If $t \in (T_{\Sigma_{SPEC_1}})_s$, then $\phi^*(t) \in (T_{\Sigma_{SPEC_2}})_{\theta(s)}$.

Proof. If $t \in (T_{\Sigma_{SPEC_1}})_s$, then t has the form $\sigma(t1, ..., tn)$, where $\sigma \in (F_1)_{w1,...,wn,s}$; $t_1, ..., t_n \in T_{\Sigma_{SPEC_1}}$ and $n \geq 0$. Therefore, by definition 37, $\phi^*(t) = \phi^*(\sigma(t1, ..., tn)) = \phi(\sigma)(\phi^*(t1), ..., \phi^*(tn))$, where $\phi(\sigma) \in (F_2)_{\theta(w1),...,\theta(wn),\theta(s)}$; $\phi^*(t_1), ..., \phi^*(t_n) \in T_{\Sigma_{SPEC_2}}$ and $n \geq 0$. Consequently, $\phi^*(t) \in (T_{\Sigma_{SPEC_2}})_{\theta(s)}$. □

4.2 The operation of T-renaming preserves the information

In this subsection, we shall prove that, if $SPEC_2$ is a T-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 T-renaming).

In other words: what we are proving is that the initial algebra of $SPEC_2$ contains the same information as that of $SPEC_1$, that is to say, that the operation of T-renaming changes names but preserves the information.

Sublemma 39 Let there be $SPEC_2 \in T - Renam(SPEC_1)$. Let $v : vars(e) \longrightarrow T_{\Sigma_{SPEC_1}}$ be an assignment of values and ϕ a T-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 T-renaming and $w : vars(e) \longrightarrow T_{\Sigma_{SPEC_2}}$ is the assignment of values such that $w = \phi^* \circ v$, then we have:

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

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 ≥ 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 40 If $SPEC_2 \in T - Renam(SPEC_1)$,

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

Proof. We 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 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[1]:

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.

[1]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 39, $\phi^*(v^*(x)) = w^*(\phi^*(x))$, 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 T-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 39, $\phi^*(v^*(x)) = w^*(\phi^*(x))$ 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. □

5 T-Reunions.

In this section, the concept of total reunion (hence, T-reunion) is introduced and we shall prove some basic properties about it.

5.1 Definition of T-reunion.

Intuitively, a T-reunion of two specifications $SPEC_1$ and $SPEC_2$ is a specification $SPEC_4$ whose initial algebra contains all the information which the initial algebras of $SPEC_1$ and of $SPEC_2$ 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 the initial algebra of $SPEC_4$ will have more equivalences than those ones belonging to the initial algebras of $SPEC_1$ and of $SPEC_2$ individually.

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

Definition 41. Let $SPEC_1 = (Obs, S, F, E_1)$ and $SPEC_2 = (Obs, S, F, E_2)$ be two behavioral specifications. Given a specification $SPEC_3 = (Obs_3, S_3, F_3, E_3)$, such that $SPEC_3 \in T - Renam(SPEC_2)$. We say that $SPEC_4 = (Obs_4, S_4, F_4, E_4)$ is a T-reunion of $SPEC_1$ and $SPEC_2$ via $SPEC_3$ (and we write $SPEC_4 = SPEC_1 \odot SPEC_2\ via\ SPEC_3$) if:

- $Obs_4 = Obs \cup Obs_3$.
- $S_4 = S \cup S_3 \cup \gamma$ 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 $s \in S$
    $trans_s : s \times \theta(s) \longrightarrow \gamma$

  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:
  - $plus(yes, yes) = yes$
  - $\forall s \in S \quad \forall \sigma \in (F_1)_{\lambda,s}$
    $trans_s(\sigma, \phi(\sigma)) = \text{yes}.$
  - $\forall s \in S \quad \forall \sigma \in (F_1)_{w1..wn,s}$
    $trans_s(\sigma(t_1, t_2, ..., t_n), \phi(\sigma)(u_1, u_2, ..., u_n)) =$
    $plus(trans_{w_1}(t_1, u_1), plus(trans_{w_2}(t_2, u_2), ..., trans_{w_n}(t_n, u_n)...))$

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

Comments.

- In this definition, we have used the names $\gamma, yes, plus$ and $trans$ to mean the new sort and the new function symbols which are introduced in a T-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 $\gamma, yes, plus$ and $trans$.

- It is easy to see that the algorithm which creates a T-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 42 Let there be $SPEC_4 = SPEC_1 \odot SPEC_2$ *via* $SPEC_3$. It is fulfilled that:

- $\forall s \in S, \quad t \in (T_{\Sigma_{SPEC_4}})_s$ implies $t \in (T_{\Sigma_{SPEC_1}})_s$
- $\forall s \in S_3, \quad t \in (T_{\Sigma_{SPEC_4}})_s$ implies $t \in (T_{\Sigma_{SPEC_3}})_s$

Proof. We shall prove the first statement by structural induction on t. The second statement can be proved analogously, by interchanging the roles of $SPEC_1$ and $SPEC_3$.

- Induction base. In this case, $t \in (F_4)_{\lambda,s}$. By definition 41, this means that $t \in F_{\lambda,s} \cup (F_3)_{\lambda,s} \cup (F_{new})_{\lambda,s}$. We can distinguish the following cases:

  If t belongs to $(F_{new})_{\lambda,s}$, s must be γ. Now, by definition 41, $\gamma \notin S$ and, therefore, $s \notin S$. This is a contradiction. So this case is impossible.

  If t belongs to $(F_3)_{\lambda,s}$, s must belong to S_3. Now, by definition 37, since $SPEC_3 \in T - Renam(SPEC_2)$, we obtain that $S \cap S_3 = \emptyset$. Therefore, $s \notin S$. This is a contradiction. So this case is impossible.

  Consequently, the only case which may occur is $t \in (F_1)_{\lambda,s}$. This means that $t \in (T_{\Sigma_{SPEC_1}})_s$

- Induction step. In this case, t will have the form $\sigma(t_1, ..., t_n)$, where $\sigma \in (F_4)_{w1..wn,s}$ and, for any i, $t_i \in (T_{\Sigma_{SPEC_4}})_{wi}$.

  By the same reasoning that has been applied in the induction base, we have that $\sigma \in (F_1)_{w1..wn,s}$. On the one hand, this implies that $w1, ..., wn \in S$. So we can apply the hypothesis of induction on $t_1, ..., t_n$ and we obtain that, for any i, $t_i \in (T_{\Sigma_{SPEC_1}})_{wi}$

  On the other hand, since $\sigma \in (F_1)_{w1..wn,s}$ and, for any i, $t_i \in (T_{\Sigma_{SPEC_1}})_{wi}$, we obtain trivially $\sigma(t_1, ..., t_n) \in (T_{\Sigma_{SPEC_1}})_s$, which is what we wished to prove. □

Sublemma 43 The following two statements are fulfilled:

- $\forall s \in Obs, \forall x \in vars(s), \forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}, \quad v^*(x) \in (T_{\Sigma_{SPEC_1}})_s$

- $\forall s \in Obs, \forall x \in vars(\theta(s)), \forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}, \quad v^*(x) \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$

Proof. First, we shall prove the first statement. By definition of assignment of values, since $x \in vars(s)$ and $v : X \longrightarrow T_{\Sigma_{SPEC_4}}$, we have that $v^*(s) \in (T_{\Sigma_{SPEC_4}})_s$. Since $s \in Obs$, then $s \in S$ and we can apply sublemma 42. We obtain that $v^*(x) \in (T_{\Sigma_{SPEC_1}})_s$.

The proof of the second statement is analogous, but it uses the fact that $\theta(s) \in S_3$. Starting from $s \in Obs$ and $SPEC_3 \in T - Renam(SPEC_2)$, we have, by definition 37, that $\theta(s) \in Obs_3$ and, therefore, $\theta(s) \in S_3$. □

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

We have defined intuitively the T-reunion of $SPEC_1$ and $SPEC_2$ as the specification whose initial algebra contains the information present in the initial algebras 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 the initial algebra of $SPEC_4$ contains the information present in the initial algebras of $SPEC_1$ and of $SPEC_3$. Now, $SPEC_3$ is a T-renaming of $SPEC_2$, and we have already proved -in subsection 4.3- that the operation of T-renaming preserves the information. Therefore, we have that the initial algebra of $SPEC_4$ contains the information of the initial algebras present in $SPEC_1$ and of $SPEC_2$, as has just been stated.

Sublemma 44 Suppose $SPEC_4 = SPEC_1 \odot SPEC_2$ *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 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[2]:

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 $t_2 \equiv_{SPEC_4} t_1$

[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 equation applied is unconditional or, otherwise, it belongs to the induction step

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 41, 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 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 lemma 14, $t_2 \notin T_{\Sigma_{SPEC_1}}$. This is a contradiction and, consequently, this case is impossible.
   - $e \in (E_3)$. If $e \in E_3$, its right-hand side has the form $p_1 = p_2$, where $p_1, p_2 \in (T_{\Sigma_{SPEC_3}}(vars(e)))_s$, with $s \in S_3$. By definition 37, since $SPEC_3 \in T-Renam(SPEC_2)$, then $S \cap S_3 = \langle\rangle$, that is, $s \notin S$. Therefore, $p_1, p_2 \in T_{\Sigma_{SPEC_1}}(vars(e))$. By lemma 14, since $v^*(p_1) = t_1$ and $v^*(p_2) = t_2$, then $t_1, t_2 \notin T_{\Sigma_{SPEC_1}}$. This is a contradiction and, consequently, 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.

Sublemma 45 Suppose $SPEC_4 = SPEC_1 \odot SPEC_2$ *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 44, by interchanging the roles of $SPEC_1$ and $SPEC_3$. □

Lemma 46. Let there be $SPEC_4 = SPEC_1 \odot SPEC_2$ *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 45 instead of sublemma 44.

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 44. In other words, we have a demonstration of $t \equiv_{SPEC_1} u$. Consequently, the right-to-left implication is proved. □

Comment. Lemma 46 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 T-reunion includes the equations belonging to E_1 and E_3 is that, by doing so, the T-reunion contains all the information of $SPEC_1$ and of $SPEC_2$.

Now, which is the reason why a T-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 "T-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 47 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 48 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 49 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 50 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 48 and of the counter-reciprocal of sublemma 49. □

Sublemma 51 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 50, we have that "t is T-I if and only if u is T-I". □

Lemma 52. Let there be $s, t \in T_{\Sigma_{SPEC_4}}$. It is fulfilled that

$trans(s, t) \equiv_{SPEC_4} yes$ implies that $\exists\, w \in T_{\Sigma_{SPEC_1}}$ such that
$w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$.

Proof. Suppose that $trans(s,t) \equiv_{SPEC_4} yes$. Since yes is not T-I (because it does not contain any subterm which has the form $trans(s',t')$), then neither is $trans(s,t)$, by sublemma 51. Now, since this term is not T-I, by definition 47, there must be a $w \in T_{\Sigma_{SPEC_1}}$ such that $w \equiv_{SPEC_4} s$ and $\phi^*(w) \equiv_{SPEC_4} t$. □

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 53 $plus(yes, plus(yes, ..., yes)...) \equiv_{SPEC_4} yes$.

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 54. 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 53, $trans(t, \phi^*(t)) \equiv_{SPEC_4} yes$. 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 behaviorally equivalent, as we have proved in section 3). This property is stated in theorem 55.

Theorem 55 The statement

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s))$ it is fulfilled that
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \&\ trans_s(x_2, y_2) = yes \ \&\ x_1 = y_1 \Rightarrow x_2 = y_2) \wedge$
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \&\ trans_s(x_2, y_2) = yes \ \&\ x_2 = y_2 \Rightarrow x_1 = y_1)$

implies the statement

$SPEC_1$ and $SPEC_2$ are behaviorally equivalent

Proof. We shall begin with the first statement

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s))$ it is fulfilled that
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \&\ trans_s(x_2, y_2) = yes \ \&\ x_1 = x_2 \Rightarrow y_1 = y_2) \wedge$
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \&\ trans_s(x_2, y_2) = yes \ \&\ y_1 = y_2 \Rightarrow x_1 = x_2)$

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

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s)),\ \forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}$
it is fulfilled that
$(v^*(trans_s(x_1, y_1)) \equiv_{SPEC_4} v^*(yes) \ \wedge\ v^*(trans_s(x_2, y_2)) \equiv_{SPEC_4} v^*(yes)$
$\wedge\ v^*(x_1) \equiv_{SPEC_4} v^*(x_2)$ implies $v^*(y_1) \equiv_{SPEC_4} v^*(y_2)) \quad \wedge$
$(v^*(trans_s(x_1, y_1)) \equiv_{SPEC_4} v^*(yes) \ \wedge\ v^*(trans_s(x_2, y_2)) \equiv_{SPEC_4} v^*(yes)$
$\wedge\ v^*(y_1) \equiv_{SPEC_4} v^*(y_2)$ implies $v^*(x_1) \equiv_{SPEC_4} v^*(x_2))$

Now, since the only variables in the previous statement are x_1, x_2, y_1 and y_2, the definition of v^* can be applied and we obtain:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_4}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_4}})_{\theta(s)}$ it is fulfilled that
$(trans_s(s_1, t_1) \equiv_{SPEC_4} yes \ \wedge\ trans_s(s_2, t_2) \equiv_{SPEC_4} yes \ \wedge\ s_1 \equiv_{SPEC_4} s_2$ implies $t_1 \equiv_{SPEC_4} t_2)$
$(trans_s(s_1, t_1) \equiv_{SPEC_4} yes \ \wedge\ trans_s(s_2, t_2) \equiv_{SPEC_4} yes \ \wedge\ t_1 \equiv_{SPEC_4} t_2$ implies $s_1 \equiv_{SPEC_4} s_2)$

Since, for any s, $(T_{\Sigma_{SPEC_1}})_s \subset (T_{\Sigma_{SPEC_4}})_s$ i $(T_{\Sigma_{SPEC_3}})_{\theta(s)} \subset (T_{\Sigma_{SPEC_4}})_{\theta(s)}$, this implies that:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that:
$(trans_s(s_1, t_1) \equiv_{SPEC_4} yes \ \wedge\ trans_s(s_2, t_2) \equiv_{SPEC_4} yes \ \wedge\ s_1 \equiv_{SPEC_4} s_2$ implies $t_1 \equiv_{SPEC_4} t_2)$
$(trans_s(s_1, t_1) \equiv_{SPEC_4} yes \ \wedge\ trans_s(s_2, t_2) \equiv_{SPEC_4} yes \ \wedge\ t_1 \equiv_{SPEC_4} t_2$ implies $s_1 \equiv_{SPEC_4} s_2)$

Now, since $s_1, s_2 \in (T_{\Sigma_{SPEC_2}})_s$ (because $SPEC_1$ and $SPEC_2$ have the same signature), then $\phi^*(s_1), \phi^*(s_2) \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$, by sublemma 38. Therefore, since the last statement is fulfilled for all the values of t_i that belong to $(T_{\Sigma_{SPEC_4}})_{\theta(s)}$, it must be fulfilled when $t_i = \phi^*(s_i)$:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(trans_s(s_1, \phi^*(s_1)) \equiv_{SPEC_4} yes \;\wedge\; trans_s(s_2, \phi^*(s_2)) \equiv_{SPEC_4} yes \;\wedge$
$s_1 \equiv_{SPEC_4} s_2$ implies $\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2)) \;\wedge$
$(trans_s(s_1, \phi^*(s_1)) \equiv_{SPEC_4} yes \;\wedge\; trans_s(s_2, \phi^*(s_2)) \equiv_{SPEC_4} yes$
$\wedge\; \phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2)$ implies $s_1 \equiv_{SPEC_4} s_2)$

By lemma 54, we obtain that:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(true \;\wedge\; true \;\wedge\; s_1 \equiv_{SPEC_4} s_2$ implies $\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2)) \;\wedge$
$(true \;\wedge\; true \;\wedge\; \phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2)$ implies $s_1 \equiv_{SPEC_4} s_2)$

Which, by the properties of logics, is equivalent to:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2))$

By applying the first part of lemma 46 to the left-hand side of the double implication, we have that:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2))$

By definition 41, since $s_1, s_2 \in T_{\Sigma_{SPEC_1}}$, then $s_1, s_2 \in T_{\Sigma_{SPEC_2}}$, because $SPEC_1$ and $SPEC_2$ have the same signature. Now, by definition 37, since $SPEC_3 \in T\!-\!Renam(SPEC_2)$, it is fulfilled that $\phi^*(s_1), \phi^*(s_2) \in T_{\Sigma_{SPEC_3}}$. Therefore, by applying the second part of lemma 46 to the right-hand side of the double implication, we have that:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_3} \phi^*(s_2))$

Since $SPEC_3 \in T - Renam(SPEC_2)$, by applying lemma 40:

$\forall s \in Obs$, $\forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(s_1 \equiv_{SPEC_2} s_2)$

And, by theorem 36, this is equivalent to:

$SPEC_1$ and $SPEC_2$ are behaviorally equivalent

Which is what we wished to prove □

9 Proof of completeness.

In this section, we shall prove the completeness of our method. That is, we shall 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 58.

9.1 Useful properties.

First, we shall prove some properties which will be useful in order to prove theorem 58.

Lemma 56 The statement

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that
$t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge t_2 \equiv_{SPEC_4} \phi^*(s_2)$ implies
$((s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2))$

implies the statement

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that
$trans_s(s_1, t_1) \equiv_{SPEC_4} yes \wedge trans_s(s_2, t_2) \equiv_{SPEC_4} yes$ implies
$((s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2))$

Proof. Suppose the second statement is fulfilled. Suppose that it is fulfilled that $trans_s(s_1, t_1) \equiv_{SPEC_4} yes \wedge trans_s(s_2, t_2) \equiv_{SPEC_4} yes$. We want to prove that it is fulfilled that $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2)$.

Since $trans_s(s_1, t_1) \equiv_{SPEC_4} yes \wedge trans_s(s_2, t_2) \equiv_{SPEC_4} yes$ is fulfilled, by lemma *52, we have that $\exists\, w_1, w_2 \in T_{\Sigma_{SPEC_1}}$ such that $w_1 \equiv_{SPEC_4} s_1 \wedge \phi^*(w_1) \equiv_{SPEC_4} t_1$ $\wedge\ w_2 \equiv_{SPEC_4} s_2 \wedge \phi^*(w_1) \equiv_{SPEC_4} t_1$. On the one hand, since $w_1 \equiv_{SPEC_4} s_1 \wedge w_2 \equiv_{SPEC_4} s_2$, it is fulfilled that $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(w_1 \equiv_{SPEC_4} w_2)$.

On the other hand, since the first statement and $\phi^*(w_1) \equiv_{SPEC_4} t_1 \wedge \phi^*(w_1) \equiv_{SPEC_4} t_1$ are fulfilled, we can apply the first statement and we obtain $(w_1 \equiv_{SPEC_4} w_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2)$. By combining this double implication with the one in the previous paragraph, we have that: $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2)$. And this is what we wished to prove. ☐

Lemma 57. The statement

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2))$

implies the statement:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that
$t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge t_2 \equiv_{SPEC_4} \phi^*(s_2)$ implies
$((s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2))$

Proof. Suppose that the first statement is fulfilled. Suppose that it is fulfilled that $t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge t_2 \equiv_{SPEC_4} \phi^*(s_2)$. We want to prove that it is fulfilled that $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2)$.

Since it is fulfilled that $t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge t_2 \equiv_{SPEC_4} \phi^*(s_2)$, we have that $(t_1 \equiv_{SPEC_4} t_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2))$.

On the other hand, since the first statement is fulfilled, we have that $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2)$. By applying this double implication to the one in the previous paragraph, we obtain that $(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2)$, which is what we wished to prove □.

9.2 Core of the proof of completeness

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

Theorem 58 The statement

$SPEC_1$ and $SPEC_2$ are behaviorally equivalent

implies the statement

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s))$ it is fulfilled that
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes\ \&\ trans_s(x_2, y_2) = yes\ \&\ x_1 = y_1 \Rightarrow x_2 = y_2) \wedge$
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes\ \&\ trans_s(x_2, y_2) = yes\ \&\ x_2 = y_2 \Rightarrow x_1 = y_1)$

Proof. Let us suppose that $SPEC_1$ and $SPEC_2$ are behaviorally equivalent. By theorem 36, it is fulfilled that:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(s_1 \equiv_{SPEC_2} s_2)$

Since $SPEC_3 \in T - Renam(SPEC_2)$, by applying lemma 40:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_1} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_3} \phi^*(s_2))$

By applying lemma 46 to the left-hand side of the double implication:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_3} \phi^*(s_2))$

Now, since $s_1, s_2 \in (T_{\Sigma_{SPEC_2}})_s$ (because $SPEC_1$ and $SPEC_2$ have the same signature), then $\phi^*(s_1), \phi^*(s_2) \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$, by sublemma 38. Therefore, we can apply lemma 46 to the right-hand side of the double implication and we obtain:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s$ it is fulfilled that
$(s_1 \equiv_{SPEC_4} s_2)$ if and only if $(\phi^*(s_1) \equiv_{SPEC_4} \phi^*(s_2))$

By lemma 57, this implies:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that
$t_1 \equiv_{SPEC_4} \phi^*(s_1) \wedge t_2 \equiv_{SPEC_4} \phi^*(s_2)$ implies
$((s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2))$

By lemma 56, this implies:

$\forall s \in Obs,\ \forall s_1, s_2 \in (T_{\Sigma_{SPEC_1}})_s;\ t_1, t_2 \in (T_{\Sigma_{SPEC_3}})_{\theta(s)}$ it is fulfilled that
$trans_s(s_1, t_1) \equiv_{SPEC_4} yes \wedge trans_s(s_2, t_2) \equiv_{SPEC_4} yes$ implies
$((s_1 \equiv_{SPEC_4} s_2)$ if and only if $(t_1 \equiv_{SPEC_4} t_2))$

By sublemma 43, this implies:

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s)),\ \forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$trans_s(v^*(x_1), v^*(y_1)) \equiv_{SPEC_4} yes \wedge trans_s(v^*(x_2), v^*(y_2)) \equiv_{SPEC_4} yes$ implies
$((v^*(x_1) \equiv_{SPEC_4} v^*(x_2))$ if and only if $(v^*(y_1) \equiv_{SPEC_4} v^*(y_2)))$

By using the properties of logics, this is equivalent to:

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s)),\ \forall v : X \longrightarrow T_{\Sigma_{SPEC_4}}$ it is fulfilled that
$(trans_s(v^*(x_1), v^*(y_1)) \equiv_{SPEC_4} yes \wedge trans_s(v^*(x_2), v^*(y_2)) \equiv_{SPEC_4} yes$
$\wedge\ v^*(x_1) \equiv_{SPEC_4} v^*(x_2)$ implies $v^*(y_1) \equiv_{SPEC_4} v^*(y_2)) \wedge$
$(trans_s(v^*(x_1), v^*(y_1)) \equiv_{SPEC_4} yes \wedge trans_s(v^*(x_2), v^*(y_2)) \equiv_{SPEC_4} yes$
$\wedge\ v^*(y_1) \equiv_{SPEC_4} v^*(y_2)$ implies $v^*(x_1) \equiv_{SPEC_4} v^*(x_2))$

By definition of fulfilment of an equation, this is equivalent to:

$\forall s \in Obs,\ \forall x_1, x_2 \in vars(s);\ y_1, y_2 \in vars(\theta(s))$, it is fulfilled that
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \wedge trans_s(x_2, y_2) = yes \wedge x_1 = x_2 \Rightarrow y_1 = y_2) \wedge$
$(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \wedge trans_s(x_2, y_2) = yes \wedge y_1 = y_2 \Rightarrow x_1 = x_2)$

Which is what we wished to prove. □

10 End of proof.

In this section, we shall make the last step of our proof: we shall prove theorem 59.

Theorem 59. Both statements are equivalent:

- $\forall s \in Obs$, $\forall x_1, x_2 \in vars(s)$; $y_1, y_2 \in vars(\theta(s))$ it is fulfilled that
  $(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \& \ trans_s(x_2, y_2) = yes \ \& \ x_1 = y_1 \Rightarrow x_2 = y_2) \wedge$
  $(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \& \ trans_s(x_2, y_2) = yes \ \& \ x_2 = y_2 \Rightarrow x_1 = y_1)$
- $SPEC_1$ and $SPEC_2$ are behaviorally equivalent.

Proof. It is a corollary of theorem 55 and of theorem 58. □

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

11 Conclusions

Taking up the result obtained in the previous section:

Theorem 59. Both statements are equivalent:

- (Fundamental property). $\forall s \in Obs$, $\forall x_1, x_2 \in vars(s)$; $y_1, y_2 \in vars(\theta(s))$ it is fulfilled that
  $(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \& \ trans_s(x_2, y_2) = yes \ \& \ x_1 = y_1 \Rightarrow x_2 = y_2) \wedge$
  $(T_{SPEC_4} \models trans_s(x_1, y_1) = yes \ \& \ trans_s(x_2, y_2) = yes \ \& \ x_2 = y_2 \Rightarrow x_1 = y_1)$
- $SPEC_1$ 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 T-reunion of SPEC1 and SPEC2 via some arbitrary T-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.

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.

[Pal95] PALASI,V.R. *Reduction of behavioral equivalence to inductive theorems* Research Report LSI-95-56-R. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya.

[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

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1995

LSI-95-1-R "Octree simplification of polyhedral solids", Dolors Ayala and Pere Brunet.

LSI-95-2-R "A note on learning decision lists", Jorge Castro.

LSI-95-3-R "The complexity of searching implicit graphs", José L. Balcázar.

LSI-95-4-R "Design quality metrics for object-oriented software development", Alonso Peralta, Joan Serras, and Olga Slavkova.

LSI-95-5-R "Extension orderings", Albert Rubio.

LSI-95-6-R "Triangles, ruler, and compass", R. Joan-Arinyo.

LSI-95-7-R "The modifiability factor in the LESD project: definition and practical results", Nuria Castell and Olga Slavkova.

LSI-95-8-R "Learnability of Kolmogorov-easy circuit expressions via queries", José L. Balcázar, Harry Buhrman, and Montserrat Hermo.

LSI-95-9-R "A case study on prototyping with specifications and multiple implementations", Xavier Franch.

LSI-95-10-R "Evidence of a noise induced transition in fluid neural networks", Jordi Delgado and Ricard V. Solé.

LSI-95-11-R "Supporting transaction design in conceptual modelling of information systems", Joan A. Pastor-Collado and Antoni Olivé.

LSI-95-12-R "Computer (PC) assisted drawing of diagrams for forecasting soaring weather", Lluís Pérez Vidal.

LSI-95-13-R "Animats adaptation to complex environments as learning guided by evolution", Mario Martin, Màrius Garcia, and Ulises Cortés.

LSI-95-14-R "Learning to solve complex tasks by reinforcement: A new algorithm", Mario Martin and Ulises Cortés.

LSI-95-15-R "Analysis of Hoare's FIND algorithm with median-of-three partition", Peter Kirschenhofer, Conrado Martínez, and Helmut Prodinger.

LSI-95-16-R "MDCO to B-Rep conversion algorithm", Dolors Ayala, Carlos Andújar, and Pere Brunet.

LSI-95-17-R "Augmented regular expressions: A formalism to describe, recognize, and learn a class of context-sensitive languages", René Alquézar and Alberto Sanfeliu.

LSI-95-18-R "On parallel versus sequential approximation", Maria Serna and Fatos Xhafa.

LSI-95-19-R "A set of rules for a constructive geometric constraint solver", Robert Joan-Arinyo and Antoni Soto.

LSI-95-20-R "From degenerate patches to triangular and trimmed patches", Marc Vigo, Núria Pla, and Pere Brunet.

LSI-95-21-R "Learning Ordered Binary Decision Diagrams", Ricard Gavaldà and David Guijarro.

LSI-95-22-R "The complexity of learning minor closed graph classes", Carlos Domingo and John Shawe-Taylor.

LSI-95-23-R "Approximating the permanent is in RNC", J. Díaz, M. Serna, and P. Spirakis.

LSI-95-24-R "Constraint satisfaction as global optimization", Pedro Meseguer and Javier Larrosa.

LSI-95-25-R "A rule-constructive geometric constraint solver", R. Joan-Arinyo and Antoni Soto.

LSI-95-26-R "External schemas in object oriented databases" (written in Spanish), José Samos.

LSI-95-27-R "An approach to belief in strong Kleene logic", Gustavo Núñez and Matías Alvarado.

LSI-95-28-R "Dynamic belief modeling", Antonio Moreno and Ton Sales.

LSI-95-29-R "Proceedings ICLP Workshop on Deductive Databases and Abduction", to appear. H. Decker, U. Geske, A. Kakas, C. Sakama, D. Seipel, and T. Urpí (editors).

LSI-95-30-R "Difference lists and difference bags for logic programming of categorial deduction", F. Xavier Lloré and Glyn Morrill.

LSI-95-31-R "An experiment on automatic semantic tagging of dictionary senses", German Rigau.

LSI-95-32-R "Slices of meaning: a study on nouns of portions", Salvador Climent and Maria Antònia Martí.

LSI-95-33-R "An HPSG grammar of Spanish implemented in ALE: A solution for dealing with subcategorisation alternances", Salvador Climent and Xavier Farreres.

LSI-95-34-R "Optimal and nearly optimal static weighted skip lists", Conrado Martínez and Salvador Roura.

LSI-95-35-R "Evaluation of expressions in a multiparadigm framework", Xavier Burgués and Xavier Franch.

LSI-95-36-R "Equivalence between executable OOZE and algebraic specification", Vicent-Ramon Palasí Lallana.

LSI-95-37-R "Automatic deduction of the behavioral equivalence between two algebraic specifications", Vicent-Ramon Palasí Lallana.

LSI-95-38-R "A method for explaining the behaviour of conceptual models", Antoni Olivé and Maria-Ribera Sancho.

LSI-95-39-R "Simple PAC learning of simple decision lists", Jorge Castro and José L. Balcázar.

LSI-95-40-R "Parallel approximation of MAX SNP and MAX NP problems", Maria Serna and Fatos Xhafa.

LSI-95-41-R "Automatically extracting translation links using a wide coverage semantic taxonomy", German Rigau, Horacio Rodriguez, and Jordi Turmo.

LSI-95-42-R "Belief increasing in SKL model frames", Matías Alvarado and Gustavo Núñez.

LSI-95-43-R "An incremental algorithm based on edge swapping for constructing restricted Delaunay triangulations", Marc Vigo Anglada.

LSI-95-44-R "Dealing with non-functional properties in an imperative programming language", Xavier Franch.

LSI-95-45-R "Universal computation in fluid neural networks", Ricard V. Solé and Jordi Delgado.

LSI-95-46-R "Phase transitions and complex systems", Ricard V. Solé, Susanna C. Manrubia, Bartolo Luque, Jordi Delgado, and Jordi Bascompte.

LSI-95-47-R "On the average complexity of some P-complete problems", Maria Serna and Fatos Xhafa.

LSI-95-48-R "An adaptive algorithm to compute the medial axis transform of 2-D polygonal domains", R. Joan-Arinyo, L. Pérez-Vidal, and E. Gargallo-Monllau.

LSI-95-49-R "Symbolic constraints in constructive geometric constraint solving", Christoph M. Hoffmann and Robert Joan-Arinyo.

LSI-95-50-R "Lambda extensions of rewrite orderings", Jean-Pierre Jouannaud and Albert Rubio.

LSI-95-51-R "Randomization of search trees by subtree size", Conrado Martínez and Salvador Roura.

LSI-95-52-R "Succinct circuit representations and leaf languages are basically the same concept", Bernd Borchert and Antoni Lozano.

LSI-95-53-R "Generalising discontinuity", Glyn Morrill and Josep-Maria Merenciano.

LSI-95-54-R "Concurrent AVL revisited: self-balancing distributed search trees", Luc Bougé, Joaquim Gabarró, and Xavier Messeguer.

LSI-95-55-R "A co-semidecision procedure for behavioral equivalence", Vicent-Ramon Palasí Lallana.

LSI-95-56-R "Reduction of behavioral equivalence to inductive theorems", Vicent-Ramon Palasí Lallana.

LSI-95-57-R "Where should concurrent rotations take place to rebalance a distributed arbitrary search tree?", Luc Bougé, Joaquim Gabarró and Xavier Messeguer.

LSI-95-58-R "Approximating convex quadratic programming is P-complete", Maria Serna and Fatos Xhafa.

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona, Spain
secrelsi@lsi.upc.es

See also the Department WWW pages, http://www-lsi.upc.es/www/