

Invariants for Parameterised Boolean Equation Systems (extended abstract)*

S.M. Orzan and T.A.C. Willemse

Department of Mathematics and Computer Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract. The concept of invariance for Parameterised Boolean Equation Systems (PBESs), first introduced in [8], is studied in greater detail. We identify an issue with the associated theory and fix this problem by proposing a stronger notion of invariance called global invariance. A precise correspondence is proven between the solution of a PBES and the solution of its invariant-strengthened version; this enables one to exploit global invariants when solving PBESs. Furthermore, we show that global invariants are robust w.r.t. all common PBES transformations and that the existing encodings of verification problems into PBESs preserve the invariants of the processes involved. These traits provide additional support for our notion of global invariants, and, moreover, provide an easy manner for transferring (e.g. automatically discovered) process invariants to PBESs. Several examples are provided that illustrate the advantages of using global invariants in various verification problems.

1 Introduction

Parameterised Boolean Equation Systems (PBESs), introduced in [10, 9] and studied in detail in [8], provide a fundamental framework for studying and solving verification problems for complex reactive systems. Problems as diverse as model checking problems for symbolic transition systems [6, 7] and real-time systems [16]; equivalence checking problems for a variety of process equivalences [2]; and static analysis of code [4] have been encoded in the PBES framework. The solution to the encoded problem can be found by solving the PBES. Several verification tools rely on PBESs or fragments thereof, e.g. the μ CRL [7] and the mCRL2 [3] model checkers and the CADP tool-suite [5].

Solving a PBES is in general an undecidable problem, much like the problems that can be encoded in them. Nevertheless, there are pragmatic approaches to solving PBESs, such as *symbolic approximation* [8] and *instantiation* [3]. While these techniques have proved their merits in practice, the undecidability of solving PBESs in general implies that these techniques are not universally applicable.

A concept that has turned out to be very powerful, especially when combined with symbolic approximation, is the notion of an *invariant* for PBESs.

* This research has been partially funded by the Netherlands Organisation for Scientific Research (NWO) under FOCUS/BRICKS grant number 642.000.602.

For instance, invariants have been used successfully in [2] when solving PBESs encoding the branching bisimulation problem for two systems: the invariants allowed the symbolic approximation process to terminate in a few steps, whereas there was no indication that it could have terminated without the invariant. As such, the notion of an invariant is a powerful tool which adds to the efficacy of techniques and tooling such as described in [7].

An invariant for a PBES, as defined in [8] (hereafter referred to as a *local invariant*), is a relation on data variables of a PBES that provides an over-approximation of the dependencies of the solution of a particular predicate variable X on its own domain. Unfortunately, the theory of local invariants as outlined in [8], is not correct for *arbitrary* equation systems.

We show that using a local invariant can wrongfully affect the solution to a PBES. We remedy this situation by introducing the concept of a *global invariant*, and show how this notion relates to local invariants. Moreover, we demonstrate that global invariants are preserved by common solution-preserving PBES manipulation methods, viz. *unfolding*, *migration* and *substitution* [8]. An invariance theorem that allows one to calculate the solution for an equation system, using a global invariant to assist the calculation, is proved. As a result of this theorem, we are able to provide a partial answer to a generalisation of an open problem coined in [8], which concerns the solution to a particular PBES pattern. Patterns are important as they allow for a simple *look-up* and *substitute* strategy to solving a PBES. Finally, we prove that traditional *process invariants* [1] are preserved under the PBES-encoding of the first-order modal μ -calculus model checking problem [7] and the PBES-encoding of various process equivalences [2].

Related Work Invariants are indispensable in any mature verification methodology that aims at tackling complex cases, such as networks of parameterised systems [12, 13], equivalence checks between reactive system [1] and for infinite data domains in general, such as hybrid systems [14]. Much research effort is aimed at stretching the limits of verification for specific classes of systems and properties. Techniques, such as invariants, that are developed for PBESs, on the other hand, are applicable to all problems that can be encoded in them.

Several works, like [12, 14] focus on the automated and even automatic discovery of invariants for specialised classes of specifications and properties. It is likely that many of these techniques can be ported to work for specific PBESs as well. This is supported by our result that demonstrates that process invariants are preserved under the existing encodings of verification problems. An advantage of verification using PBESs is that predicates can be identified that are invariants for the PBES, but that fail to be invariants for the original process(es) involved. This is because the PBES-encoding also incorporates other information from the input to the encoded verification problem (see Section 5 for an example).

Structure In Section 2, we introduce PBESs and some basic notation. We recall the definition of local invariants and introduce global invariants in Section 3, and use these to show that the accompanying theory for local invariants has issues, which are resolved by the theory for global invariants. The influence of

solution-preserving manipulations for PBESs on global invariants is investigated in Section 4, and in Section 5, we investigate the relation between process invariants and global invariants. Two small examples illustrate several aspects of the developed theory. we present our conclusions in Section 7. **Relevant proofs are collected in the appendices; these are an excerpt of the full paper [11]. Note that the appendices will be omitted in a final version.**

2 Preliminaries

In this section, we give a brief overview of the concepts and notations that provide the basis to the theory in the remainder of this paper. We refer to [8, 11] for a more detailed account.

Predicate formulae. Predicate formulae are part of the basic building blocks for PBESs; these are basically ordinary predicates extended with predicate variables.

Definition 1. A predicate formula *is a formula defined by the below grammar:*

$$\phi ::= b \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \forall d:D. \phi \mid \exists d:D. \phi \mid X(e)$$

where b is a data term of sort **Bool**. Furthermore, $X \in \mathcal{P}$ is a (sorted) predicate variable to which we associate a data variable d_X of sort D_X ; e is a data term of the sort D_X . Data variables are taken from a set \mathcal{D} .

The set of all predicate formulae is denoted **Pred**. Predicate formulae ϕ not containing predicate variables are referred to as *simple predicates*. The set of predicate variables that occur in a formula ϕ is denoted by $\text{occ}(\phi)$. Note that negation does not occur in predicate formulae, except as an operator in data terms; $b \implies \phi$ is a shorthand for $\neg b \vee \phi$ for terms b of sort **Bool**.

Note that we use predicate variables X to which we associate a single variable d_X of sort D_X instead of vectors \mathbf{d}_X of sort \mathbf{D}_X . Note that this does not incur a loss in generality; it is merely a matter of convenience.

Predicate formulae may contain both bound and unbound data variables. We assume that the set of bound variables and the set of free variables in a predicate formula are disjoint. For a closed data term e , i.e. a data term not containing free data variables, we assume an interpretation function $[_]$ that maps the term e to the semantic data element $[e]$ it represents. For open terms, we use a *data environment* ε that maps each variable from \mathcal{D} to a data value of the intended sort. The interpretation of an open term e is denoted by $[e]\varepsilon$ and is obtained in the standard way. We write $\varepsilon[e/d]$ to stand for the environment ε for all variables different from d , and $\varepsilon[v/d](d) = v$. A similar notation applies to predicate environments. For brevity, we do not explicitly distinguish between the abstract sorts of expressions and their semantic counterparts.

Definition 2. Let θ be a predicate environment assigning a function $D_X \rightarrow \mathbb{B}$ to every predicate variable X , and let ε be a data environment assigning a value from domain D to every variable d of sort D . The interpretation $[_]\theta\varepsilon$ of a predicate formula in the context of θ and ε is either true or false, as follows:

$$\begin{aligned}
[b]\theta\varepsilon &= [b]\varepsilon & [\phi_1 \wedge \phi_2]\theta\varepsilon &= [\phi_1]\theta\varepsilon \text{ and } [\phi_2]\theta\varepsilon \\
[X(e)]\theta\varepsilon &= \text{true iff } \theta(X)([e]\varepsilon) & [\phi_1 \vee \phi_2]\theta\varepsilon &= [\phi_1]\theta\varepsilon \text{ or } [\phi_2]\theta\varepsilon \\
[\forall d:D. \phi]\theta\varepsilon &= \text{for all } v \in D, [\phi]\theta(\varepsilon[v/d]) \\
[\exists d:D. \phi]\theta\varepsilon &= \text{for some } v \in D, [\phi]\theta(\varepsilon[v/d])
\end{aligned}$$

Definition 3. Let ϕ and ψ be predicate formulae. We write $\phi \rightarrow \psi$ iff for all predicate environments θ and all data environments ε , $[\phi]\theta\varepsilon$ implies $[\psi]\theta\varepsilon$.

The symmetric closure of \rightarrow induces a *logical equivalence* on Pred , denoted \leftrightarrow . Basic properties such as commutativity, idempotence and associativity of \wedge and \vee are immediately satisfied.

Predicate Variables and Substitution. A basic operation on predicate formulae is substitution of a predicate formula for a predicate variable. To this end, we introduce *predicate functions*: predicate formulae casted to functions. As a shorthand, we write $\phi_{\langle d_X \rangle}$ to indicate that ϕ is lifted to a function ($\lambda d_X:D_X. \phi$), i.e. $\phi_{\langle d_X \rangle} \in [D_X \rightarrow \text{Pred}]$, the set of all functions from sort D_X yielding a predicate formula. The semantics of a predicate function is defined in the context of a predicate environment θ and a data environment ε :

$$[\phi_{\langle d_X \rangle}]\theta\varepsilon =_{\text{def}} \lambda v \in D_X. [\phi]\theta\varepsilon[v/d_X]$$

The substitution of $\psi_{\langle d_X \rangle}$ for a predicate variable X in a predicate formula ϕ is defined by the following set of rules:

$$\begin{aligned}
b[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} b \\
Y(e)[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} \begin{cases} \psi[e/d_X] & \text{if } Y = X \\ Y(e) & \text{otherwise} \end{cases} \\
(\phi_1 \wedge \phi_2)[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} \phi_1[\psi_{\langle d_X \rangle}/X] \wedge \phi_2[\psi_{\langle d_X \rangle}/X] \\
(\phi_1 \vee \phi_2)[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} \phi_1[\psi_{\langle d_X \rangle}/X] \vee \phi_2[\psi_{\langle d_X \rangle}/X] \\
(\forall d:D. \phi)[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} \forall d:D. \phi[\psi_{\langle d_X \rangle}/X] \\
(\exists d:D. \phi)[\psi_{\langle d_X \rangle}/X] &=_{\text{def}} \exists d:D. \phi[\psi_{\langle d_X \rangle}/X]
\end{aligned}$$

Example 1. Consider the formulae $X(f(d)) \wedge Y(g(d))$ and $\psi := Y(h(d_Y))$. Then $(X(f(d)) \wedge Y(g(d)))[\psi_{\langle d_Y \rangle}/Y]$ yields: $X(f(d)) \wedge Y(h(g(d)))$. \square

Property 1. Let ϕ, ψ be predicate formulae, and θ, ε environments. Then:

$$[\phi[\psi_{\langle d_X \rangle}/X]]\theta\varepsilon = [\phi]\theta[[\psi_{\langle d_X \rangle}]\theta\varepsilon / X]\varepsilon. \quad \square$$

For convenience, we generalise single syntactic substitutions $\phi[\psi_{\langle d_X \rangle}/X]$ to finite sequences of substitutions using the following notation:

Definition 4. Let $V = \langle X_1, \dots, X_n \rangle$ be a vector of predicate variables and let ϕ_i be an arbitrary predicate formula. The consecutive substitution $\phi \left[_{X_i \in V} \phi_i_{\langle d_{X_i} \rangle} / X_i \right]$ is defined as follows:

$$\begin{cases} \phi \left[_{X_i \in \langle \rangle} \phi_i_{\langle d_{X_i} \rangle} / X_i \right] & =_{\text{def}} \phi \\ \phi \left[_{X_i \in \langle X_1, \dots, X_n \rangle} \phi_i_{\langle d_{X_i} \rangle} / X_i \right] & =_{\text{def}} (\phi[\phi_1_{\langle d_{X_1} \rangle} / X_1]) \left[_{X_i \in \langle X_2, \dots, X_n \rangle} \phi_i_{\langle d_{X_i} \rangle} / X_i \right] \end{cases}$$

When for all ϕ_i , at most variable X_i occurs in ϕ_i and all variables in $\langle X_1, \dots, X_n \rangle$ are distinct, the consecutive substitution $\phi \left[_{X_i \in \langle X_1, \dots, X_n \rangle} \phi_i \langle d_{X_i} \rangle / X_i \right]$ yields the same for all permutations of vector $\langle X_1, \dots, X_n \rangle$, i.e. it behaves as a simultaneous substitution. In this case, we allow abuse of notation by writing $\phi \left[_{X_i \in \{X_1, \dots, X_n\}} \phi_i \langle d_{X_i} \rangle / X_i \right]$.

Parameterised Boolean Equation Systems. A Parameterised Boolean Equation System (henceforth referred to as an *equation system*) is a finite sequence of equations of the form $(\sigma X(d_X : D_X) = \phi)$; ϕ is a predicate formula in which the variable d_X is bound by the defining equation for X ; σ denotes either the least (μ) or the greatest (ν) fixed point. We denote the empty equation system by ϵ .

We say an equation system is *closed* whenever all predicate variables occurring at the right-hand side of an equation occur at the left-hand side of some equation. An equation system is *open* if it is not closed. For a given equation system \mathcal{E} , the set $\text{bnd}(\mathcal{E})$ denotes the predicate variables occurring in the left-hand side of the equations of \mathcal{E} , and the set $\text{occ}(\mathcal{E})$ denotes the set of predicate variables occurring in the predicate formulae of the equations of \mathcal{E} . The *solution* to an equation system is a predicate environment, defined as follows:

Definition 5. *Given a predicate environment θ and an equation system \mathcal{E} . The solution $[\mathcal{E}] \theta \varepsilon$ is an environment that is defined as follows:*

$$\begin{aligned} [\epsilon] \theta \varepsilon &= \theta \\ [(\sigma X(d_X : D_X) = \phi) \mathcal{E}] \theta \varepsilon &= [\mathcal{E}] (\theta \left[\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi \langle d_{X} \rangle] ([\mathcal{E}] \theta [\mathcal{X}/X]) \varepsilon / X \right]) \varepsilon \end{aligned}$$

Note that the fixed points are taken over the complete lattice of functions $([D_X \rightarrow \mathbb{B}], \sqsubseteq)$ for (possibly infinite) data sets D_X , where $f \sqsubseteq g$ is defined as the point-wise ordering: $f \sqsubseteq g$ iff for all $v \in D_X$: $f(v)$ implies $g(v)$. The predicate transformer associated to a predicate function $[\phi \langle d_{X} \rangle] \theta \varepsilon$, denoted

$$\lambda \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi \langle d_{X} \rangle] \theta [\mathcal{X}/X] \varepsilon$$

is a monotone operator [7, 8, 6]. The existence of fixed points of this operator in the lattice $([D_X \rightarrow \mathbb{B}], \sqsubseteq)$ follows immediately from Tarski's Theorem [15].

Note 1. The solution to an equation system is sensitive to the ordering of the equations: while $(\mu X = Y)(\nu Y = X)$ has \perp as solution for X and Y , the equation system $(\nu Y = X)(\mu X = Y)$ has \top as solution for X and Y . Manipulations such as unfolding, migration and substitution, however, do not affect the solution to an equation system [8]. Using the latter two, all equation systems can be solved (using a strategy called *Gauß Elimination*), *provided* that one has the techniques and tools to eliminate a predicate variable from its defining equation. One such methods is e.g. *symbolic approximation*, see [8].

3 Invariants for Equation Systems

Invariants for equation systems first appeared in [8]. We first repeat its definition:

Definition 6. Let $(\sigma X(d_X:D_X) = \phi)$ be an equation and let I be a simple predicate formula. Then I is an invariant of X iff

$$I \wedge \phi \leftrightarrow (I \wedge \phi)[(I \wedge X(d_X))_{\langle d_X \rangle} / X]$$

The above definition may appear awkward to those familiar only with invariants for transition systems. It does, however, express what is normally understood as the invariance property; the unusual appearance is a consequence of the possibility of having multiple occurrences of X in subformulae of ϕ . The invariance criterion only concerns a transfer property on equation systems: an initialisation criterion is not applicable at this level. The analogue to the initialisation criterion is, however, part of Theorem 2 (see page 8), and Theorems 40 and 42 of [8]. For completeness' sake, we repeat the latter:

Theorem 1 (See [8]). Let $(\sigma X(d_X:D_X) = \phi)$ be an equation and let I be an invariant of X . Assume that:

1. for some χ with $X \notin \text{occ}(\chi)$, we have for all equation systems \mathcal{E} and all η, ε :

$$[(\sigma X(d_X:D_X) = I \wedge \phi) \mathcal{E}] \eta \varepsilon = [(\sigma X(d_X:D_X) = \chi) \mathcal{E}] \eta \varepsilon.$$
2. for the predicate formula ψ we have $\psi \leftrightarrow \psi[(I \wedge X(d_X))_{\langle d_X \rangle} / X]$.

Then for all equation systems $\mathcal{E}_0, \mathcal{E}_1$ and all environments η, ε :

$$\begin{aligned} & [(\sigma' Y(d_Y:D_Y) = \psi) \mathcal{E}_0(\sigma X(d_X:D_X) = \phi) \mathcal{E}_1] \eta \varepsilon \\ &= [(\sigma' Y(d_Y:D_Y) = \psi[\chi_{\langle d_X \rangle} / X]) \mathcal{E}_0(\sigma X(d_X:D_X) = \phi) \mathcal{E}_1] \eta \varepsilon. \quad \square \end{aligned}$$

A common use of invariants in our setting is in the evaluation of predicate formulae ϕ using a predicate environment that is defined by an equation system \mathcal{E} , i.e. $[\phi]([\mathcal{E}] \eta \varepsilon) \varepsilon$. If one can show that $\phi \leftrightarrow \phi[(I \wedge X(d_X))_{\langle d_X \rangle} / X]$ (the analogue to the initialisation criterion for an invariant), it suffices to solve the equation system \mathcal{E} strengthened with invariant I to compute whether ϕ holds. However, this is not sound for equation systems in which *open equations* appear; these arise when encoding process equivalences [2] and model checking problems [9, 7]. A second issue is that invariants may “break” as a result of a substitution:

Example 2. Consider the following (constructed) closed equation system:

$$\begin{aligned} (\mu X(n:\mathbb{N}) = n \geq 2 \wedge Y(n)) \\ (\mu Y(n:\mathbb{N}) = Z(n) \vee Y(n+1)) \\ (\mu Z(n:\mathbb{N}) = n < 2 \vee Y(n-1)) \end{aligned} \tag{1}$$

The simple predicate formula $n \geq 2$ is an invariant for equation Y in equation system (1): $n \geq 2 \wedge (Z(n) \vee Y(n+1)) \leftrightarrow n \geq 2 \wedge (Z(n) \vee (n+1 \geq 2 \wedge Y(n+1)))$. However, substituting $n < 2 \vee Y(n-1)$ for Z in the equation system Y yields the equation system of (1):

$$\begin{aligned} (\mu X(n:\mathbb{N}) = n \geq 2 \wedge Y(n)) \\ (\mu Y(n:\mathbb{N}) = n < 2 \vee Y(n-1) \vee Y(n+1)) \\ (\mu Z(n:\mathbb{N}) = n < 2 \vee Y(n-1)) \end{aligned} \tag{2}$$

The invariant $n \geq 2$ of Y in (1) fails to be an invariant for Y in (2). Worse still, observe that for (1) we have:

- $n \geq 2 \wedge Y(n) \leftrightarrow (n \geq 2 \wedge Y(n))[n \geq 2 \wedge Y(n)]_{\langle n \rangle / Y}$, and
- for all $\mathcal{E}, \eta, \varepsilon$:

$$\begin{aligned} & \llbracket (\mu Y(n:\mathbb{N}) = n \geq 2 \wedge (Z(n) \vee Y(n+1))) \mathcal{E} \rrbracket \eta \varepsilon \\ & = \llbracket (\mu Y(n:\mathbb{N}) = (n = 1 \wedge Z(n)) \vee (n = 0 \wedge Z(n+1))) \mathcal{E} \rrbracket \eta \varepsilon. \end{aligned}$$

Note that the above alternative equation for Y was found by means of a three-step symbolic approximation. Following Theorem 1, equation system (1) should therefore be equivalent to the following equation system:

$$\begin{aligned} (\mu X(n:\mathbb{N}) &= n \geq 2 \wedge ((n = 1 \wedge Z(n) \vee (n = 0 \wedge Z(n+1)))) \\ (\mu Y(n:\mathbb{N}) &= Z(n) \vee Y(n+1)) \\ (\mu Z(n:\mathbb{N}) &= n < 2 \vee Y(n-1)) \end{aligned} \quad (3)$$

The solution to equation system (1) yields $\lambda n \in \mathbb{N}. n \geq 2$ for X . In contrast, the solution to X in equation system (3) is $\lambda n \in \mathbb{N}. \perp$. This is a direct violation of the conclusion of Theorem 1. Theorem 40 of [8] is similarly flawed. \square

Example 2 shows that identified invariants (cf. [8]) fail to remain invariants when substitution is exercised on the equation system, and, more importantly, that Theorem 1 is not correct for *every* equation system.

As we demonstrate in this paper, both issues can be remedied by using a slightly stronger invariance criterion, taking all predicate variables of an equation system into account. This naturally leads to a notion of *global invariance*; in contrast, we refer to the type of invariance defined in Def. 6 as *local invariance*.

Let $f:V \rightarrow \text{Pred}$, $V \subseteq \mathcal{P}$, be a function that maps a predicate variable to a predicate formula. We say f is *simple* iff $f(X)$ is simple for all $X \in V$. Note that the notation $f(X)$ is purely *meta-notation*; e.g. it is not affected by syntactic substitutions: $f(X)[\psi_{\langle d_X \rangle} / X]$ remains $f(X)$, since $f(X)$ is simple.

Definition 7. *The simple function $f:V \rightarrow \text{Pred}$ is said to be a global invariant for an equation system \mathcal{E} iff $V \supseteq \text{bnd}(\mathcal{E})$ and for each $(\sigma X(d_X:D_X) = \phi)$ occurring in \mathcal{E} , we have:*

$$f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i}))_{\langle d_{X_i} \rangle} / X_i \rrbracket.$$

Proposition 1. *Let $f:V \rightarrow \text{Pred}$ be a global invariant for an equation system \mathcal{E} . Let $W \subseteq V$. Then for all equations $(\sigma X(d_X:D_X) = \phi)$ in \mathcal{E} , we have:*

$$f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \llbracket_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i}))_{\langle d_{X_i} \rangle} / X_i \rrbracket. \quad \square$$

Corollary 1. *Let f be a global invariant for an equation system \mathcal{E} . Then $f(X)$, for any $X \in \text{bnd}(\mathcal{E})$ is a local invariant. \square*

Note 2. The reverse of the above corollary does not hold: if for all $X \in \text{bnd}(\mathcal{E})$, we have a predicate formula $f(X)$ that is a local invariant for X in \mathcal{E} , then f is *not* necessarily a global invariant. Consider the following equation system:

$$(\nu X(n:\mathbb{N}) = Y(n-1)) (\mu Y(n:\mathbb{N}) = X(n+1))$$

X and Y both have $n \geq 5$ as local invariants (in fact, any simple predicate is a local invariant), but $(\lambda Z \in \{X, Y\}. n \geq 5)$ fails to be a global invariant.

Let $\text{pvi}(\phi)$ yield the set of *predicate variable instantiations* in ϕ :

$$\begin{aligned} \text{pvi}(b) &= \emptyset & \text{pvi}(X(e)) &= \{X(e)\} \\ \text{pvi}(\forall d:D. \phi) &= \text{pvi}(\phi) & \text{pvi}(\phi_1 \wedge \phi_2) &= \text{pvi}(\phi_1) \cup \text{pvi}(\phi_2) \\ \text{pvi}(\exists d:D. \phi) &= \text{pvi}(\phi) & \text{pvi}(\phi_1 \vee \phi_2) &= \text{pvi}(\phi_1) \cup \text{pvi}(\phi_2) \end{aligned}$$

A sufficient condition for a function f to be a global invariant is given below:

Property 2. Let \mathcal{E} be an equation system and $f:\text{bnd}(\mathcal{E}) \rightarrow \text{Pred}$ a simple function; then f is a global invariant for \mathcal{E} if for every equation $(\sigma X(d_X:D_X) = \phi)$ in \mathcal{E} we have $f(X) \rightarrow \bigwedge_{Y(e) \in \text{pvi}(\phi)} (f(Y))[e/d_Y]$. \square

We next establish an exact correspondence between the solution of an equation system \mathcal{E} and the equation system \mathcal{E}' which is derived from \mathcal{E} by strengthening it with the global invariant. Strengthening is achieved by the operation **Apply**:

Definition 8. Let $f:V \rightarrow \text{Pred}$ be a simple function. Let \mathcal{E} be an equation system satisfying $\text{bnd}(\mathcal{E}) \subseteq V$. The equation system **Apply** (f, \mathcal{E}) is defined as follows:

$$\begin{aligned} \text{Apply}(f, \epsilon) &= \epsilon \\ \text{Apply}(f, (\sigma X(d_X:D_X) = \phi) \mathcal{E}_0) &= (\sigma X(d_X:D_X) = f(X) \wedge \phi) \text{Apply}(f, \mathcal{E}_0) \end{aligned}$$

The correctness of the above-mentioned correspondence relies, a.o., on the following lemma. The main result of this section is Theorem 2, which improves and corrects Theorem 1; it immediately follows the below lemma.

Lemma 1. Let $(\sigma X(d_X:D_X) = \phi)$ be a possibly open equation. Let $f:V \rightarrow \text{Pred}$ be a simple function such that

1. $\text{occ}(\phi) \subseteq V$.
2. $f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi)[(f(X) \wedge X(d_X))_{\langle d_X \rangle} / X]$.

Then for all environments η, ε :

$$\begin{aligned} &\lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge (\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{\langle d_X \rangle}]\eta[\mathcal{X}/X]\varepsilon)(v) \\ &= \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge (\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{\langle d_X \rangle}]\eta[\mathcal{X}/X]\varepsilon)(v). \square \end{aligned}$$

Theorem 2. Let $f:V \rightarrow \text{Pred}$ be a simple function. Let \mathcal{E} be an equation system and let η_1, η_2 be arbitrary predicate environments. If the following holds:

1. $\text{bnd}(\mathcal{E}) \cup \text{occ}(\mathcal{E}) \subseteq V$ and
2. for all $X \in V$:
 - (a) $[f(X) \wedge X(d_X)]\eta_1\varepsilon = [f(X) \wedge X(d_X)]\eta_2\varepsilon$.
 - (b) $f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi)_{[X_i \in V. (f(X_i) \wedge X_i(d_{X_i}))_{\langle d_{X_i} \rangle} / X_i]}$.

then we have for all $X \in V$:

$$[f(X) \wedge X(d_X)]([\mathcal{E}]\eta_1\varepsilon)\varepsilon = [f(X) \wedge X(d_X)]([\text{Apply}(f, \mathcal{E})]\eta_2\varepsilon)\varepsilon. \quad \square$$

Corollary 2. *Let $f:V \rightarrow \text{Pred}$ be a global invariant for an equation system \mathcal{E} . Then for all predicate formulae ϕ with $\text{occ}(\phi) \subseteq V$ and all environments η, ε :*

$$\begin{aligned} & \phi \leftrightarrow \phi \left[\prod_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i}))_{(d_{X_i})} / X_i \right] \\ & \text{implies } [\phi]([\mathcal{E}] \eta \varepsilon) \varepsilon = [\phi]([\text{Apply}(f, \mathcal{E})] \eta \varepsilon) \varepsilon \quad \square \end{aligned}$$

This means that for an equation system \mathcal{E} and a global invariant f of \mathcal{E} , it does not matter whether we use \mathcal{E} or its invariant-strengthened version $\text{Apply}(f, \mathcal{E})$ to evaluate a predicate formula ϕ for which the initialisation criterion for invariant f holds. As another consequence of Theorem 2, we have the proposition below:

Proposition 2. *Let \mathcal{E} be an equation system. Let f be a global invariant for \mathcal{E} and assume \mathcal{E} contains an equation for X of the form:*

$$(\nu X(d:D) = f(X) \wedge \bigwedge_{i \in I} Q_i e_i^1 : E_i^1 \dots Q_{m_i} e_i^{m_i} : E_i^{m_i} \cdot \psi_i \implies X(g_i(d, e_i^1, \dots, e_i^{m_i})))$$

where $Q_j \in \{\forall, \exists\}$ for any j , and for all i , ψ_i are simple predicate formulae and g_i is a data term that depends only on the values of d and $e_i^1, \dots, e_i^{m_i}$. Then X has the solution $f(X)$. \square

In the terminology of [8], the equation above is a pattern which has solution $f(X)$. This pattern is an instance of a generalisation of the unsolved pattern of [8]. This pattern appears to be extremely useful in the examples of Section 6.

4 Preservation of Global Invariants under Solution-Preserving Manipulations

One of the defects of local invariants is that this notion is not robust with respect to *substitution*. In this section, we study the robustness of (global) invariants with respect to most common solution-preserving manipulations, viz. *migration*, *unfolding* and *substitution* [8].

Theorem 3. *Let $\mathcal{E} := \mathcal{E}_0$ ($\sigma X(d_X:D_X) = \phi$) \mathcal{E}_1 \mathcal{E}_2 be an equation system with $\text{occ}(\phi) = \emptyset$. Let $\mathcal{F} := \mathcal{E}_0$ \mathcal{E}_1 ($\sigma X(d_X:D_X) = \phi$) \mathcal{E}_2 be the result of a migration. If $f:V \rightarrow \text{Pred}$ is a global invariant for \mathcal{E} then f is a global invariant for \mathcal{F} . \square*

Unfolding and substitution [8] involve replacing predicate variables with the right-hand side expressions of their corresponding equation. The difference is that unfolding operates locally and substitution is a global operation. The following lemma proves robustness of invariants under replacing variables with their corresponding right-hand side expressions.

Lemma 2. *Let \mathcal{E} be an equation system and let $f:V \rightarrow \text{Pred}$ be a global invariant for \mathcal{E} . For any predicate variable $X \in \text{bnd}(\mathcal{E})$, we denote the right-hand side of X 's defining equation in \mathcal{E} by ϕ_X . Then for all $X, Y \in \text{bnd}(\mathcal{E})$:*

$$\begin{aligned} & f(X) \wedge \phi_X[\phi_Y(d_Y)/Y] \\ \leftrightarrow & (f(X) \wedge \phi_X[\phi_Y(d_Y)/Y]) \left[\prod_{Z \in V} (f(Z) \wedge Z(d_Z))_{(d_Z)} / Z \right] \quad \square \end{aligned}$$

The above lemma immediately leads to the robustness of invariants under substitution and unfolding. This is expressed by the following theorems:

Theorem 4. *Let $\mathcal{E} := \mathcal{E}_0 (\sigma X(d_X:D_X) = \phi) \mathcal{E}_1$ be an equation system and let $f:V \rightarrow \text{Pred}$ a global invariant for \mathcal{E} . Then f is also a global invariant for the equation system $\mathcal{F} := \mathcal{E}_0 (\sigma X(d_X:D_X) = \phi[\phi_{\langle d_X \rangle}/X]) \mathcal{E}_1$. \square*

Theorem 5. *Let $\mathcal{E} := \mathcal{E}_0 (\sigma X(d_X:D_X) = \phi) \mathcal{E}_1 (\sigma' Y(d_Y:D_Y) = \psi) \mathcal{E}_2$ and $\mathcal{F} := \mathcal{E}_0 (\sigma X(d_X:D_X) = \phi[\psi_{\langle d_Y \rangle}/Y]) \mathcal{E}_1 (\sigma' Y(d_Y:D_Y) = \psi) \mathcal{E}_2$ be PBESs. If $f:V \rightarrow \text{Pred}$ is a global invariant for \mathcal{E} then f is also a global invariant for \mathcal{F} . \square*

An interesting observation is that both substitution and unfolding not only preserve existing global invariants, but also may lead to *new* global invariants. We illustrate this phenomenon with an example for unfolding.

Example 3. Let $\nu X(n:\mathbb{N}) = X(n+1)$ be an equation system. Using unfolding, we obtain the equivalent equation system $\nu X(n:\mathbb{N}) = X(n+2)$. Clearly, the function f that assigns to X the predicate formula `even`(n) is a global invariant for the latter equation. However, f is not a global invariant for the original equation. Thus, by unfolding, the set of invariants for an equation system increases. \square

5 Process Invariants and Equation Invariants

Linear process equations (LPEs) have been proposed as *symbolic* representations of general (infinite) labelled transition systems, the semantic framework for specifying and analysing complex, reactive systems. In an LPE, the state of a process is modelled by a finite vector of (possibly infinite) sorted variables, and the behaviour is described by a finite set of condition-action-effect rules. The apparent restrictiveness of this format is misleading: parallelism, (infinite) non-determinism and other operators can often be mapped losslessly onto LPEs.

Definition 9. *A linear process equation is an equation taking the form*

$$P(d:D) = \sum \left\{ \sum_{e_a:E_a} c_a(d, e_a) \implies a(f_a(d, e_a)) \cdot P(g_a(d, e_a)) \mid a \in \text{Act} \right\}$$

where $f_a:D \times E_a \rightarrow D_a$, $g_a:D \times E_a \rightarrow D$ and $c_a:D \times E_a \rightarrow \text{Bool}$ for each action label $a \in \text{Act}$. D , D_a and E_a are general data sorts. The restrictions to single sorts D and E_a are again only for brevity and do not incur a loss of generality.

In the above definition, the LPE P specifies that if in the current state d the condition $c_a(d, e_a)$ holds, for an arbitrary e_a of sort E_a , then an action a carrying data parameter $f_a(d, e_a)$ is possible and the effect of executing this action is that the state is changed to $g_a(d, e_a)$. Thus, the values of the condition, action parameter and new state may depend on the current state and the non-deterministic chosen value for variable e_a .

Definition 10. *Let P be the LPE of Def. 9. A simple predicate ι is an invariant for P iff for all actions $a \in \text{Act}$: $\iota \wedge c_a(d, e_a) \rightarrow (\iota[g_a(d, e_a)/d])$ holds.*

Model Checking. In [9, 6], the *first-order modal μ -calculus* (μ -calculus for short) is defined, a modal language for verification of data-dependent process specifications. The language is a first-order extension of the standard modal μ -calculus due to Kozen. It permits the use of data variables and parameters to capture the essential data-dependencies in the process behaviour. The grammar of the calculus is given by the following rules:

$$\begin{aligned}\phi &::= b \mid X(e) \mid \neg\phi \mid \phi \wedge \phi \mid \forall d:D. \phi \mid [\alpha]\phi \mid (\nu X(d_f:D_f := e). \phi) \\ \alpha &::= b \mid a(e) \mid \neg\alpha \mid \alpha \wedge \alpha \mid \forall d:D.\alpha\end{aligned}$$

where σ is a least or greatest fixed point sign. The meaningful formulae are those formulae for which every occurrence of a variable X is under an even number of negations. The semantics of μ -calculus formulae is defined over an LTS, induced by an LPE P , see [9, 6] for details. The global model checking problem $P \models \Phi$ and the local model checking problem $P(e) \models \Phi$, where e is an initial value for the P and Φ is a μ -calculus formula, can be translated¹ to the problem of solving the equation system $\mathbf{E}(\Phi)$ [9, 6, 7] resulting from the translation.

Theorem 6. *Let Φ be a μ -calculus formula. Let ι be an invariant for the LPE P of Def. 9. Then $(\lambda X \in \text{bnd}(\mathbf{E}(\Phi)). \iota)$ is a global invariant of $\mathbf{E}(\Phi)$. \square*

The reverse of the above theorem does not hold: if f is a global invariant for an equation system $\mathbf{E}(\Phi)$ for some formula Φ and LPE P , then f does not necessarily lead to an invariant for the process P (see the below example).

Example 4. Consider the process that models the rise and fall of a stock value of a company and may report its current value if asked.

$$\begin{aligned}M(v:\mathbb{N}) &= \sum_{m:\mathbb{N}} \mathbf{up} \cdot M(v+m) \\ &+ \mathbf{current}(v) \cdot M(v) \\ &+ \sum_{m:\mathbb{N}} m \leq v \implies \mathbf{down} \cdot M(v-m)\end{aligned}$$

Verifying that without decreases, the stock value is always above threshold T (provided it is so initially), i.e. $\nu X. [\neg \mathbf{down}]X \wedge \forall n:\mathbb{N}. [\mathbf{current}(n)](n > T)$, using an equation system boils down to solving the below equation system:

$$\nu X(v:\mathbb{N}) = (\forall m:\mathbb{N}. X(v+m)) \wedge \forall n:\mathbb{N}. v = n \implies n > T$$

Clearly, X has $v > T$ as an invariant whereas this is not an invariant for M . \square

Process Equivalences. In [2] various equivalences, such as strong and branching bisimulation, between two LPEs have been encoded as solution problems of equation systems. Branching bisimulation is the most complex of the process equivalences tackled in [2], yielding equation systems $\nu E_1 \mu E_2$, which are of alternation depth 2. Here, E_1 and E_2 are sets of equations obtained from a syntactic manipulation of the input LPEs.²

¹ the translation can be found in Appendix D for quick reference.

² the encoding can be found in Appendix E for quick reference.

Theorem 7 (See [11]). *Let P be the LPE of Def. 9. Assume ι is an invariant for LPE P . We define function f as follows:*

$$f(Z) = \begin{cases} \iota & \text{if } Z \in \text{bnd}(E_1) \\ \iota \wedge c_a(d, e) & \text{if } Z \in \text{bnd}(E_2) \end{cases}$$

Then f is a global invariant for the equation system $\nu E_1 \mu E_2$, resulting from the encoding of branching bisimulation between P and some second LPE S . \square

The remaining encodings in [2] yield similar global invariants, see [11]. The significance of the preservation of process invariants under the PBES-encoding of an equivalence lies in the fact that this helps ensuring that the solution of the equation system does not relate all unreachable states of the input processes. Relating unreachable parts of processes is often neither meaningful nor computationally tractable (in particular for infinite state systems).

6 Examples

To illustrate how invariants typically assist in solving equation systems, we provide two easily understood examples of verifications using equation systems. The first example treats the privacy problem of a rudimentary voting protocol; the second is a mutual exclusion problem for readers and writers.

6.1 Voting Protocol

The voting protocol is given by the LPE E below. The intended votes of participants are modelled by variable V , a bitlist; we write $V.i$ to indicate the vote of voter i . A high bit represents a *yes* and a low bit represents a *no* vote. Registered voters are maintained in set R and parameters y, n record the number of casted yes/no votes so far. Voting of a person is modelled by action **vote**, and it follows no particular order. The outcome of the vote is published by action **outcome**.

$$\begin{aligned} E(V:\mathcal{L}(\{0, 1\}), R:2^{\mathbb{N}}, y, n:\mathbb{N}) = \\ R = \emptyset \implies \text{outcome}(y, n) \cdot \delta \\ + \sum_{i:\mathbb{N}} i \in R \implies \text{vote}(i) \cdot E(V, R \setminus \{i\}, y + V.i, n + (1 - V.i)) \end{aligned}$$

Privacy with respect to an external observer of the voting process is warranted if this observer cannot tell whether $V.i = 0$ or $V.i = 1$ for any voter i . Formally, privacy is guaranteed if process $E(l, r, 0, 0)$ is strongly bisimilar to $E(\pi(l), r, 0, 0)$, where list $\pi(l)$ is an arbitrary permutation of list l . Strong bisimilarity is encoded using the below equation system (see [2] for the general translation).

$$\begin{aligned} (\nu X(V:\mathcal{L}(\{0, 1\}), R:2^{\mathbb{N}}, y, n:\mathbb{N}, V':\mathcal{L}(\{0, 1\}), R':2^{\mathbb{N}}, y', n':\mathbb{N}) = \\ (\forall i:\mathbb{N}. i \in R \implies (i \in R' \\ \wedge X(V, R \setminus \{i\}, y + V.i, n + (1 - V.i), V', R' \setminus \{i\}, y' + V'.i, n' + (1 - V'.i)))) \\ \wedge (\forall i:\mathbb{N}. i \in R' \implies (i \in R \\ \wedge X'(V, R \setminus \{i\}, y + V.i, n + (1 - V.i), V', R' \setminus \{i\}, y' + V'.i, n' + (1 - V'.i)))) \\ \wedge (R = \emptyset \iff R' = \emptyset) \wedge (R = \emptyset \implies (y = y' \wedge n = n'))) \\ (\nu X'(V':\mathcal{L}(\{0, 1\}), R':2^{\mathbb{N}}, y', n':\mathbb{N}, V:\mathcal{L}(\{0, 1\}), R:2^{\mathbb{N}}, y, n:\mathbb{N}) = \\ X(V, R, y, n, V', R', y', n')) \end{aligned}$$

$E(l, r, 0, 0)$ and $E(\pi(l), r, 0, 0)$ are bisimilar iff $X(l, r, 0, 0, \pi(l), r, 0, 0)$ is true. A symbolic approximation of variable X generates a non-converging series of increasingly complex equations expressing constraints on subsets of R , meaning that we cannot compute the general solution to X .

The equation system encodes the strong bisimulation relation between two processes E , i.e. both reachable and unreachable states of the two processes E will be related in the solution to X . However, we are interested only in the answer to $X(l, r, 0, 0, \pi(l), r, 0, 0)$. We state the following three predicate formulae:

- $\iota_1 := R = R'$ formalises that any difference in the set of voters is noticeable by an external observer,
- $\iota_2 := y + n = y' + n'$ formalises that the total number of expressed votes should be the same in both protocols,
- $\iota_3 := y + \sum_{i \in R} V.i = y' + \sum_{i \in R'} V'.i$ formalises, a.o., that we are dealing with permutations.

Let $\iota := \iota_1 \wedge \iota_2 \wedge \iota_3$; from Property 2, we immediately conclude that ι is an invariant for X and X' . Note that ι is a tautology when instantiated with the initial values due to the verification problem $E(l, r, 0, 0) = E(\pi(l), r, 0, 0)$. So, without affecting the answer to our verification problem, we can strengthen the equations for X and X' with ι . The variable X' , appearing in the equation for X can be removed by a substitution. We observe that for equation X :

$$(\iota \wedge (R = \emptyset \iff R' = \emptyset) \wedge (R = \emptyset \implies (y = y' \wedge n = n'))) \iff \iota$$

We find that the equation for X is of the form of Proposition 2; it therefore has solution ι . Since $X(l, r, 0, 0, \pi(l), r, 0, 0)$ holds, privacy is indeed guaranteed.

6.2 Readers-Writers Mutual Exclusion

We consider a standard mutual exclusion problem between distributed *readers* and *writers*. A total of $N > 0$ (N is some arbitrary value) readers and writers are assumed.

$$\begin{aligned} P(n_r, n_w, t:\mathbb{N}) = & t \geq 1 \implies r_s \cdot P(n_r + 1, n_w, t - 1) \\ & + n_r > 0 \implies r_e \cdot P(n_r - 1, n_w, t + 1) \\ & + t \geq N \implies w_s \cdot P(n_r, n_w + 1, t - N) \\ & + n_w > 0 \implies w_e \cdot P(n_r, n_w - 1, t + N) \end{aligned}$$

Here the actions r_s and w_s express the starting of reading and writing of a process. Likewise, the actions r_e and w_e model the ending of reading and writing. Mutual exclusion between readers and writers holds when:

1. No writer can start if readers are reading: $\nu X. [\top] X \wedge [r_s] \nu Y. ([\neg r_e] Y \wedge [w_s] \perp)$.
2. No reader can start if writers are busy: $\nu X. [\top] X \wedge [w_s] \nu Y. ([\neg w_e] Y \wedge [r_s] \perp)$.

We only treat the first property; proving the second property follows the same reasoning. The equation system that encodes the first property is given below:

$$\begin{aligned}
(\nu X(n_r, n_w, t:\mathbb{N}) = & ((t \geq 1 \implies (X(n_r + 1, n_w, t - 1) \wedge Y(n_r + 1, n_w, t - 1))) \\
& \wedge (n_r > 0 \implies X(n_r - 1, n_w, t + 1)) \wedge (t \geq N \implies X(n_r, n_w + 1, t - N)) \\
& \wedge (n_w > 0 \implies X(n_r, n_w - 1, t + N)))) \\
(\nu Y(n_r, n_w, t:\mathbb{N}) = & t < N \wedge (t \geq 1 \implies Y(n_r + 1, n_w, t - 1)) \\
& \wedge (n_w > 0 \implies Y(n_r, n_w - 1, t + N)))
\end{aligned}$$

With standard techniques, Y can only be solved using an unwieldy pattern [8], which introduces multiple quantifications and additional selector functions; symbolic approximation does not converge in a finite number of steps. The use of invariants is the most appropriate strategy here. An invariant of process P is $t = N - (n_r + n_w \cdot N)$, which, by Theorem 6 is also a global invariant for the equations X and Y . Furthermore, $n_r \geq 1$ for Y and \top for X is a global invariant. Both X and Y can be strengthened with the above invariants. The simple predicate formula $t < N$ follows from $t = N - (n_r + n_w \cdot N) \wedge n_r \geq 1$; we can therefore employ Proposition 2 and conclude that Y has solution $t = N - (n_r + n_w \cdot N) \wedge n_r \geq 1$. Substituting this solution for Y in X , using Proposition 1 to simplify the resulting equation, we find the following equivalent equation for X :

$$\begin{aligned}
(\nu X(n_r, n_w, t:\mathbb{N}) = & ((t \geq 1 \implies (X(n_r + 1, n_w, t - 1))) \\
& \wedge (n_r > 0 \implies X(n_r - 1, n_w, t + 1)) \wedge (t \geq N \implies X(n_r, n_w + 1, t - N)) \\
& \wedge (n_w > 0 \implies X(n_r, n_w - 1, t + N)) \wedge t = N - (n_r + n_w \cdot N)))
\end{aligned}$$

Another application of Proposition 2, immediately leads to the solution $t = N - (n_r + n_w \cdot N)$ for X . Thus, writers cannot start writing while readers are active if initially the values for n_r, n_w, t satisfy $t = N - (n_r + n_w \cdot N)$.

Mutual exclusion can also be expressed by a single μ -calculus formula with data variables; then invariants linking process and formula variables is required [11].

7 Closing Remarks

Techniques and concepts for solving PBESs have been studied in detail [8]. Among these is the concept of *invariance*, which has been instrumental in solving verification problems that were studied in e.g. [8, 2]. In this paper, we further studied the notion of invariance and show that the accompanying theory is flawed for PBESs in which *open* equations occur. We have proposed a stronger notion of invariance, called *global invariance*, and phrased an invariance theorem that remedies the issues of the invariance theorem of [8]. We moreover have shown that our notion of invariance is preserved by three important solution-preserving PBES manipulations. This means that, unlike the notion of invariance of [8], global invariants can be used in combination with these manipulations when solving equation systems. As a side-result, we obtain a partial answer to an open question, concerning a specific pattern for PBESs, first put forward in [8].

We continued by demonstrating that invariants for processes automatically yield global invariants in the PBESs resulting from two standard verification

encodings, viz. the encoding of the first-order modal μ -calculus model checking problem and the encoding of branching bisimulation for two (possibly infinite) transition systems. This means that in the PBES verification methodology, one can take advantage of established techniques for checking and discovering process invariants. We conjecture that many such techniques, see e.g. [12, 13], can be put to use for (automatically) discovering global invariants in PBESs. Additional research is of course needed to substantiate this conjecture.

Acknowledgements. The authors would like to thank Jan Friso Groote for valuable feedback.

References

1. M.A. Bezem and J.F. Groote. Invariants in process algebra with data. In *Proceedings Concur'94*, volume 836 of *LNCS*, pages 401–416. Springer Verlag, 1994.
2. T. Chen, B. Ploeger, J. van de Pol, and T.A.C. Willemse. Equivalence checking for infinite systems using parameterized boolean equation systems. In *Proc. 18th Int'l Conference on CONCUR*, LNCS 4703, pages 120–135. Springer, 2007.
3. A. van Dam, B. Ploeger, and T.A.C. Willemse. Instantiation for parameterised boolean equation systems. Submitted for publication, 2008.
4. M.M. Gallardo, C. Joubert, and P. Merino. Implementing influence analysis using parameterised boolean equation systems. In *Proceedings of ISOLA '06*. IEEE Computer Society Press, November 2006.
5. H. Garavel, R. Mateescu, F. Lang, and W. Serwe. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *Proceedings of CAV*, volume 4590 of *LNCS*, pages 158–163. Springer, 2007.
6. J.F. Groote and R. Mateescu. Verification of temporal properties of processes in a setting with data. In *Proc. of AMAST*, LNCS 1548, pages 74–90. Springer, 1999.
7. J.F. Groote and T.A.C. Willemse. Model-checking processes with data. *Sci. Comput. Program*, 56(3):251–273, 2005.
8. J.F. Groote and T.A.C. Willemse. Parameterised boolean equation systems. *Theor. Comput. Sci*, 343(3):332–369, 2005.
9. R. Mateescu. Local model-checking of an alternation-free value-based modal mu-calculus. In *Proc. 2nd Int'l Workshop on VMCAI*, September 1998.
10. R. Mateescu. *Vérification des propriétés temporelles des programmes parallèles*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.
11. S.M. Orzan and T.A.C. Willemse. Invariants for parameterised boolean equation systems. Technical report, Eindhoven University of Technology, 2008. To Appear; see <http://www.win.tue.nl/~timw/invariants.pdf>.
12. S. Pandav, K. Slind, and G. Gopalakrishnan. Counterexample guided invariant discovery for parameterized cache coherence verification. In *Proceedings CHARME*, volume 3725 of *LNCS*, pages 317–331. Springer, 2005.
13. A. Pnueli, S. Ruah, and L. Zuck. Automatic deductive verification with invisible invariants. In *Proceedings TACAS*, volume 2031 of *LNCS*, pages 82–97, 2001.
14. S. Sankaranarayanan, H.B. Sipma, and Z. Manna. Constructing invariants for hybrid systems. *Formal Methods in System Design*, 32(1):25–55, 2008.
15. A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Mathematics*, 5(2):285–309, June 1955.
16. D. Zhang and R. Cleaveland. Fast generic model-checking for data-based systems. In F. Wang, editor, *FORTE*, volume 3731 of *LNCS*, pages 83–97. Springer, 2005.

The appendices contain extra details, lemmata and proofs for the main theorems and propositions in the main paper. These will not be included in the final version of the paper but are included to facilitate assessing the validity of the results in the paper. An extended version of this paper, containing all proofs and additional results, is due to appear as a technical report [11].

A Substitution and Logical Equivalence

We have the following lemmata dealing with syntactic substitutions and logical equivalence. Apart from the additional insight into the subtle interactions between logical equivalence and substitutions one gains through these lemmata, they provide the necessary foundation for most of the proofs and theorems in the remaining sections. Property 1 is at the basis of the proofs of these technical lemmata; for full proofs we refer to [11].

Lemma 3. *Let ψ, ρ, χ be arbitrary predicate formulae. Then:*

$$\psi \leftrightarrow \rho \text{ implies } \chi[\psi_{\langle d_X \rangle} / X] \leftrightarrow \chi[\rho_{\langle d_X \rangle} / X].$$

Lemma 4. *Let ψ, ρ, χ be arbitrary predicate formulae. Then:*

$$\psi \leftrightarrow \rho \text{ implies } \psi[\chi_{\langle d_X \rangle} / X] \leftrightarrow \rho[\chi_{\langle d_X \rangle} / X]$$

Lemma 5. *Let ϕ, ψ and ρ be arbitrary predicate formulae. Then:*

$$(\phi[\psi_{\langle d_X \rangle} / X])[\rho_{\langle d_X \rangle} / X] \leftrightarrow \phi[\psi[\rho_{\langle d_X \rangle} / X]_{\langle d_X \rangle} / X].$$

Lemma 6. *Let ϕ, ψ, ρ be arbitrary predicate formulae. If $X \notin \text{occ}(\rho)$ and $X \neq Y$, then*

$$(\phi[\psi_{\langle d_X \rangle} / X])[\rho_{\langle d_Y \rangle} / Y] \leftrightarrow (\phi[\rho_{\langle d_Y \rangle} / Y])[\psi[\rho_{\langle d_Y \rangle} / Y]_{\langle d_X \rangle} / X].$$

B Proofs for Section 3

We only provide the proofs for Proposition 1, Lemma 1 and Theorem 2; the remaining proofs and further detail can be found in [11].

Proposition 1. *Let $f : V \rightarrow \text{Pred}$ be a global invariant for an equation system \mathcal{E} and let $W \subseteq V$. Then for every equation $(\sigma X(d_X : D_X) = \phi)$ in \mathcal{E} , we have:*

$$f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \left[_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i}))_{\langle d_{X_i} \rangle} / X_i \right] \quad (4)$$

Proof. Let $f : V \rightarrow \text{Pred}$ be a global invariant for \mathcal{E} . Let $(\sigma X(d_X : D_X) = \phi)$ be an arbitrary equation in \mathcal{E} . We prove the following property for all $W \subseteq V$:

$$f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \left[_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i}))_{\langle d_{X_i} \rangle} / X_i \right]$$

We use induction on the size of the set W .

1. Base case: $W = \emptyset$. Then $(f(X) \wedge \phi) \llbracket_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket$ is defined as $f(X) \wedge \phi$. By reflexivity of \leftrightarrow , we find that the property holds for $W = \emptyset$.
2. Induction: assume that for $W \subset V$ we have:

$$f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \llbracket_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket \quad (\text{IH})$$

Assume that $X_j \notin W$. Then:

$$\begin{aligned}
& (f(X) \wedge \phi) \llbracket_{X_i \in W \cup \{X_j\}} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket \\
\leftrightarrow & \text{\{Property of consecutive substitution\}} \\
& ((f(X) \wedge \phi) \llbracket_{X_i \in W} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket) \\
& \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket \\
\leftrightarrow & \text{\{Lemma 4 and (IH)\}} \\
& ((f(X) \wedge \phi) \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket) \\
\leftrightarrow & \text{\{Lemma 4 and } f \text{ is a global invariant\}} \\
& ((f(X) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket) \\
& \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket \\
\leftrightarrow & \text{\{Property of consecutive substitution\}} \\
& ((f(X) \wedge \phi) \llbracket_{X_i \in V \setminus \{X_j\}} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket \\
& \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket) \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket \\
\leftrightarrow & \text{\{Lemma 5\}} \\
& ((f(X) \wedge \phi) \llbracket_{X_i \in V \setminus \{X_j\}} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket \\
& \llbracket (f(X_j) \wedge f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket) \\
\leftrightarrow & \text{\{idempotence of } \wedge \text{\}} \\
& ((f(X) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket \\
& \llbracket (f(X_j) \wedge X_j(d_{X_j})) \langle d_{X_j} \rangle / X_j \rrbracket) \\
\leftrightarrow & \text{\{Property of consecutive substitution; } f \text{ is a global invariant\}} \\
& f(X) \wedge \phi
\end{aligned}$$

□

To facilitate the proof of Theorem 2, we rely on the following auxiliary lemma. The proof of this lemma is contained in [11].

Lemma 7. *Let ϕ be an arbitrary predicate formula. Let f be a simple formula satisfying: $f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket$, where $f: V \rightarrow \text{Pred}$ with $\text{occ}(\phi) \subseteq V$. Then for all environments $\eta_1, \eta_2, \varepsilon$:*

$$\begin{aligned}
& \forall Y \in V : \llbracket (f(Y) \wedge Y(d_Y)) \rrbracket \eta_1 \varepsilon = \llbracket (f(Y) \wedge Y(d_Y)) \rrbracket \eta_2 \varepsilon \\
& \text{implies} \\
& \llbracket (f(X) \wedge \phi) \rrbracket \eta_1 \varepsilon = \llbracket (f(X) \wedge \phi) \rrbracket \eta_2 \varepsilon
\end{aligned}$$

Lemma 1, which we repeat below, relates the solution to an equation that is strengthened with its local invariant (derived from a global invariant) with the

solution to the original equation. The proof is directly at the level of the semantics of an equation and employs transfinite approximation.

Lemma 1. *Let $(\sigma X(d_X:D_X) = \phi)$ be a possibly open equation. Let $f:V \rightarrow \text{Pred}$ be a simple function such that*

1. $\text{occ}(\phi) \subseteq V$
2. $f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi)[(f(X) \wedge X(d_X))_{(d_X)}/X]$

Then for all environments η, ε :

$$\begin{aligned} & \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge (\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{(d_X)}]\eta[\mathcal{X}/X]\varepsilon)(v) \\ = & \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge (\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{(d_X)}]\eta[\mathcal{X}/X]\varepsilon)(v) \end{aligned}$$

Proof. We prove this lemma by a transfinite approximation. So, we let X_α be the α -th approximation for $\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{(d_X)}]\eta[\mathcal{X}/X]\varepsilon$ and \bar{X}_α be the α -th approximation for $\sigma \mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{(d_X)}]\eta[\mathcal{X}/X]\varepsilon$, where α is an ordinal, and we show that

$$\lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge X_\alpha(v) = \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge \bar{X}_\alpha(v)$$

We find:

- For $\alpha = 0$, we must distinguish between $\sigma = \nu$ and $\sigma = \mu$. If $\sigma = \nu$, it holds that $X_0 = \bar{X}_0 = \lambda v \in D_X. \top$. For $\sigma = \mu$ we find that $X_0 = \bar{X}_0 = \lambda v \in D_X. \perp$. From both cases, it follows that $\lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge X_0(v) = \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge \bar{X}_0(v)$
- For $\alpha = \beta + 1$ a successor ordinal, we assume the following induction hypothesis:

$$\begin{aligned} & \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge X_\beta(v) \\ = & \lambda v \in D_X. [f(X)]\varepsilon[v/d_X] \wedge \bar{X}_\beta(v) \end{aligned} \tag{IH}$$

Next, we continue:

$$\begin{aligned}
& \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge X_{\beta+1}(v) \\
= & \{\text{By definition of approximation}\} \\
& \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge [\phi] \eta[X_\beta/X] \varepsilon[v/d_X] \\
= & \{\text{Semantics; } f \text{ is a simple function}\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \eta[X_\beta/X] \varepsilon[v/d_X] \\
= & \{\text{Assumption on } f\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] [(f(X) \wedge X(d_X))_{\langle d_X \rangle} / X] \eta[X_\beta/X] \varepsilon[v/d_X] \\
= & \{\text{Property 1: syntactic vs. semantic substitution}\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \\
& \quad ((\eta[X_\beta/X]) [[(f(X) \wedge X(d_X))_{\langle d_X \rangle}] \eta[X_\beta/X] \varepsilon[v/d_X] / X]) \varepsilon[v/d_X] \\
= & \{\text{Semantics; } f \text{ is a simple function; simplification of environment}\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \eta[\lambda w \in D_X. [f(X)] \eta \varepsilon[w/d_X] \wedge X_\beta(w) / X] \varepsilon[v/d_X] \\
= & \{\text{Application of (IH)}\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \eta[\lambda w \in D_X. [f(X)] \eta \varepsilon[w/d_X] \wedge \overline{X}_\beta(w) / X] \varepsilon[v/d_X] \\
= & \{\text{Semantics; } f \text{ is a simple function; rewriting environment } \eta\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \\
& \quad ((\eta[\overline{X}_\beta/X]) [[(f(X) \wedge X(d_X))_{\langle d_X \rangle}] \eta[\overline{X}_\beta/X] \varepsilon[v/d] / X]) \varepsilon[v/d_X] \\
= & \{\text{Property 1: semantic vs. syntactic substitution}\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] [(f(X) \wedge X(d_X))_{\langle d_X \rangle} / X] \eta[X_\beta/X] \varepsilon[v/d_X] \\
= & \{\text{Assumption on } f\} \\
& \lambda v \in D_X. [(f(X) \wedge \phi)] \eta[X_\beta/X] \varepsilon[v/d_X] \\
= & \{\text{By definition of approximation}\} \\
& \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge X_{\beta+1}(v)
\end{aligned}$$

– For α a limit ordinal and $\sigma = \mu$, we find:

$$\begin{aligned}
& \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge X_\alpha(v) \\
= & \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge \bigvee_{\beta < \alpha} X_\beta(v) \\
= & \lambda v \in D_X. \bigvee_{\beta < \alpha} [f(X)] \varepsilon[v/d_X] \wedge X_\beta(v) \\
\stackrel{\text{(IH)}}{=} & \lambda v \in D_X. \bigvee_{\beta < \alpha} [f(X)] \varepsilon[v/d_X] \wedge \overline{X}_\beta(v) \\
= & \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge \bigvee_{\beta < \alpha} \overline{X}_\beta(v) \\
= & \lambda v \in D_X. [f(X)] \varepsilon[v/d_X] \wedge \overline{X}_\alpha(v)
\end{aligned}$$

The case for $\sigma = \nu$ goes along the same lines. \square

The formal correspondence between the solution of an equation system \mathcal{E} and the equation system $\text{Apply}(f, \mathcal{E})$ is given by Theorem 2. We repeat this theorem below, together with its proof.

Theorem 2. *Let $f:V \rightarrow \text{Pred}$ be a simple function. Then, for all equation systems \mathcal{E} and for all environments η_1 and η_2 , if the following conditions are met:*

1. $\text{bnd}(\mathcal{E}) \cup \text{occ}(\mathcal{E}) \subseteq V$ and

2. for all $X \in V$:

- (a) $[f(X) \wedge X(d_X)]\eta_1\varepsilon = [f(X) \wedge X(d_X)]\eta_2\varepsilon$
- (b) $f(X) \wedge \phi \leftrightarrow (f(X) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket$

then we have for all $X \in V$:

$$[f(X) \wedge X(d_X)]([\mathcal{E}]\eta_1\varepsilon)\varepsilon = [f(X) \wedge X(d_X)]([\mathbf{Apply}(f, \mathcal{E})]\eta_2\varepsilon)\varepsilon \quad (5)$$

Proof. Let $f:V \rightarrow \text{Pred}$ be a simple function. We use induction on the size of \mathcal{E} .

1. Suppose $\mathcal{E} = \epsilon$. In that case the conclusion of the theorem follows immediately from assumption (2a).
2. Let \mathcal{E} be of the form $(\sigma X(d_X:D_X) = \phi) \mathcal{E}'$ for some $X \notin \text{bnd}(\mathcal{E}')$. We assume as our induction hypothesis that for all environments η'_1 and η'_2 , if the following conditions are met:
 - (a) $\text{bnd}(\mathcal{E}') \cup \text{occ}(\mathcal{E}') \subseteq V$ and
 - (b) for all $Y \in V$:
 - i. $[f(Y) \wedge Y(d_Y)]\eta'_1\varepsilon = [f(Y) \wedge Y(d_Y)]\eta'_2\varepsilon$
 - ii. $f(Y) \wedge \phi \leftrightarrow (f(Y) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket$

then for all $Y \in V$, we have

$$[f(Y) \wedge Y(d_Y)]([\mathcal{E}']\eta'_1\varepsilon)\varepsilon = [f(Y) \wedge Y(d_Y)]([\mathbf{Apply}(f, \mathcal{E}')]\eta'_2\varepsilon)\varepsilon$$

Assume that the following holds:

- (a) $\text{bnd}(\mathcal{E}) \cup \text{occ}(\mathcal{E}) \subseteq V$ and
- (b) for all $Y \in V$:
 - i. $[f(Y) \wedge Y(d_Y)]\eta_1\varepsilon = [f(Y) \wedge Y(d_Y)]\eta_2\varepsilon$
 - ii. $f(Y) \wedge \phi \leftrightarrow (f(Y) \wedge \phi) \llbracket_{X_i \in V} (f(X_i) \wedge X_i(d_{X_i})) \langle d_{X_i} \rangle / X_i \rrbracket$

We must show the below equivalence for all $Z \in V$:

$$[f(Z) \wedge Z(d_Z)]([\mathcal{E}]\eta_1\varepsilon)\varepsilon = [f(Z) \wedge Z(d_Z)]([\mathbf{Apply}(f, \mathcal{E})]\eta_2\varepsilon)\varepsilon \quad (6)$$

Let $Z \in V$ be an arbitrary predicate variable. We continue as follows:

$$\begin{aligned} & [f(Z) \wedge Z(d_Z)]([\mathcal{E}]\eta_1\varepsilon)\varepsilon \\ = & \{\text{Definition of } [\mathcal{E}]\eta_1\varepsilon\} \\ & [f(Z) \wedge Z(d_Z)]([\mathcal{E}']\eta_1[\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. \phi_{\langle d_X \rangle}][\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon)/X]\varepsilon \end{aligned}$$

Likewise, we derive:

$$\begin{aligned} & [f(Z) \wedge Z(d_Z)]([\mathbf{Apply}(f, \mathcal{E})]\eta_2\varepsilon)\varepsilon \\ = & \{\text{Definition of } [\mathbf{Apply}(f, \mathcal{E})]\eta_2\varepsilon\} \\ & [f(Z) \wedge Z(d_Z)]([\mathbf{Apply}(f, \mathcal{E}')] \\ & \eta_2[\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{\langle d_X \rangle}][\mathbf{Apply}(f, \mathcal{E}')]\eta_2[\mathcal{X}/X]\varepsilon)/X]\varepsilon \end{aligned}$$

From our assumption that $\text{bnd}(\mathcal{E}) \cup \text{occ}(\mathcal{E}) \subseteq V$, we immediately obtain $\text{bnd}(\mathcal{E}') \cup \text{occ}(\mathcal{E}') \subseteq V$, so for all $Z \neq X$, equation (6) follows from our

induction hypothesis and assuming that it holds for $Z = X$. For the latter, i.e. for $Z = X$, we must demonstrate that:

$$\begin{aligned}
& [f(X) \wedge X(d_X)]([\mathcal{E}']\eta_1[\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{\langle d_X \rangle}]([\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon)/X]\varepsilon) \\
= & [f(X) \wedge X(d_X)]([\mathbf{Apply}(f, \mathcal{E}')] \\
& \eta_2[\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{\langle d_X \rangle}][\mathbf{Apply}(f, \mathcal{E}')] \eta_2[\mathcal{X}/X]\varepsilon)/X]\varepsilon)
\end{aligned}$$

An application of the definition of semantics for predicate formulae, taking into account that f is a simple function, yields the equivalent equivalence:

$$\begin{aligned}
& [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{\langle d_X \rangle}]([\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon)([d_X]\varepsilon)) \\
= & [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D \rightarrow \mathbb{B}]. \\
& [(f(X) \wedge \phi)_{\langle d_X \rangle}][\mathbf{Apply}(f, \mathcal{E}')] \eta_2[\mathcal{X}/X]\varepsilon)([d_X]\varepsilon)
\end{aligned} \tag{7}$$

Using Lemma 1 and our assumptions, we find:

$$\begin{aligned}
& [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [\phi_{\langle d_X \rangle}]([\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon)([d_X]\varepsilon)) \\
= & [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{\langle d_X \rangle}][[\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon])([d_X]\varepsilon)
\end{aligned}$$

Using Lemma 7, our assumptions and the induction hypothesis, we find:

$$\begin{aligned}
& [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D_X \rightarrow \mathbb{B}]. [(f(X) \wedge \phi)_{\langle d_X \rangle}][[\mathcal{E}']\eta_1[\mathcal{X}/X]\varepsilon])([d_X]\varepsilon) \\
= & [f(X)]\varepsilon \wedge (\sigma\mathcal{X} \in [D \rightarrow \mathbb{B}]. \\
& [(f(X) \wedge \phi)_{\langle d_X \rangle}][\mathbf{Apply}(f, \mathcal{E}')] \eta_2[\mathcal{X}/X]\varepsilon)([d_X]\varepsilon)
\end{aligned}$$

By transitivity of equivalence, we find that equivalence (7) holds. \square

C Proofs for Section 4

We only provide the proof for Lemma 2, which we first repeat here:

Lemma 2. *Let \mathcal{E} be an equation system and let $f:V \rightarrow \text{Pred}$ be a global invariant for \mathcal{E} . For any predicate variable $X \in \text{bnd}(\mathcal{E})$, we denote the right-hand side of X 's defining equation in \mathcal{E} by ϕ_X . Then, for all predicate variables $X, Y \in \text{bnd}(\mathcal{E})$:*

$$\begin{aligned} & f(X) \wedge \phi_X[\phi_{Y\langle d_Y \rangle}/Y] \\ \leftrightarrow & (f(X) \wedge \phi_X[\phi_{Y\langle d_Y \rangle}/Y]) \ [_{Z \in V}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \end{aligned}$$

Proof. We calculate, using properties proved previously, starting from the right-hand side of the desired equality:

$$\begin{aligned} & (f(X) \wedge \phi_X[\phi_{Y\langle d_Y \rangle}/Y]) \ [_{Z \in V}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \\ \leftrightarrow & \{V = (V \setminus \{Y\}) \cup \{Y\}\} \\ & ((f(X) \wedge \phi_X[\phi_{Y\langle d_Y \rangle}/Y]) \\ & \ [_{Z \in V \setminus \{Y\}}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \) \ [(f(Y) \wedge Y(d_Y))_{\langle d_Y \rangle}/Y] \\ \leftrightarrow & \{\text{distributivity of substitution over } \wedge, f \text{ is simple}; \\ & \{\text{Lemma 6 successively applied to all } Z \in V \setminus \{X\}; \text{Lemma 3}\} \\ & (f(X) \wedge \phi_X \ [_{Z \in V \setminus \{Y\}}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \\ & \ [\phi_{Y\langle d_Y \rangle} \ [_{Z \in V \setminus \{Y\}}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] /Y]) \ [(f(Y) \wedge Y(d_Y))_{\langle d_Y \rangle}/Y] \\ \leftrightarrow & \{\text{distributivity of substitution over } \wedge, f \text{ is simple}; \\ & \{\text{Lemma 5, } (V \setminus \{Y\}) \cup \{Y\} = V\} \\ & (f(X) \wedge \phi_X \ [_{Z \in V \setminus \{Y\}}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \) \\ & \ [\phi_{Y\langle d_Y \rangle} \ [_{Z \in V}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] /Y] \\ \leftrightarrow & \{\text{Proposition 1, Lemma 3}\} \\ & (f(X) \wedge \phi_X \ [(f(Y) \wedge Y(d_Y))_{\langle d_Y \rangle}/Y] \\ & \ [\phi_{Y\langle d_Y \rangle} \ [_{Z \in V}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] /Y] \\ \leftrightarrow & \{\text{distributivity, } f \text{ is simple, Lemma 5}\} \\ & (f(X) \wedge \phi_X) \ [(f(Y) \wedge \phi_{Y\langle d_Y \rangle} \ [_{Z \in V}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z] \)_{\langle d_Y \rangle}/Y] \\ \leftrightarrow & \{\text{Proposition 1, Lemma 3}\} \\ & (f(X) \wedge \phi_X) \ [(f(Y) \wedge \phi_{Y\langle d_Y \rangle})/Y] \\ \leftrightarrow & \{\text{Lemma 5}\} \\ & (f(X) \wedge \phi_X) \ [(f(Y) \wedge Y(d_Y))_{\langle d_Y \rangle}/Y] \ [\phi_{Y\langle d_Y \rangle}/Y] \\ \leftrightarrow & \{\text{Proposition 1: } (f(X) \wedge \phi_X) \ [(f(Y) \wedge Y(d_Y))_{\langle d_Y \rangle}/Y] = f(X) \wedge \phi_X\} \\ & \{\text{distributivity, } f \text{ is simple}\} \\ & f(X) \wedge \phi_X[\phi_{Y\langle d_Y \rangle}/Y] \end{aligned}$$

□

The robustness of global invariants with respect to substitution and unfolding follows from here.

D Encoding of the μ -Calculus

Table 1. Inductive translation scheme for encoding the problem $P \models \Phi$, where $\Phi = \sigma X(d_f:D_f := e_f). \psi$ into the closed equation system $\mathbf{E}(\Phi)$. The LPE P is as given in Def. 9. We have $P(e_p) \models \Phi$ iff $X(e_p, e_f)$ holds for all e_p and e_f .

$\mathbf{E}(b)$	$= \epsilon$
$\mathbf{E}(X(e))$	$= \epsilon$
$\mathbf{E}(\phi_1 \oplus \phi_2)$	$= \mathbf{E}(\phi_1) \ \mathbf{E}(\phi_2)$
$\mathbf{E}(\mathbf{Q} \ d:D. \phi)$	$= \mathbf{E}(\phi)$
$\mathbf{E}([\alpha]\phi)$	$= \mathbf{E}(\phi)$
$\mathbf{E}(\langle \alpha \rangle \phi)$	$= \mathbf{E}(\phi)$
$\mathbf{E}(\sigma X(d_f:D_f := e). \psi)$	$= (\sigma \tilde{X}(d:D, d_f:D_f, \mathbf{Par}_{\square}(X, \Phi)) = \mathbf{RHS}_{\Phi}(\psi)) \ \mathbf{E}(\phi)$
$\mathbf{RHS}_{\Phi}(b)$	$= b$
$\mathbf{RHS}_{\Phi}(X(e))$	$= \tilde{X}(d, e, \mathbf{Par}_{\square}(X, \Phi))$
$\mathbf{RHS}_{\Phi}(\phi_1 \oplus \phi_2)$	$= \mathbf{RHS}_{\Phi}(\phi_1) \oplus \mathbf{RHS}_{\Phi}(\phi_2)$
$\mathbf{RHS}_{\Phi}(\mathbf{Q} \ d:D. \phi)$	$= \mathbf{Q} \ d:D. \ \mathbf{RHS}_{\Phi}(\phi)$
$\mathbf{RHS}_{\Phi}([\alpha]\phi)$	$= \bigwedge_{a \in \mathcal{A}ct} \forall e_a:D_a \ (c_a(d, e_a) \wedge \mathbf{match}(a(f(d, e_a)), \alpha))$ $\implies (\mathbf{RHS}_{\Phi}(\phi)[g_a(d, e_a)/d])$
$\mathbf{RHS}_{\Phi}(\langle \alpha \rangle \phi)$	$= \bigvee_{a \in \mathcal{A}ct} \exists e_a:D_a \ (c_a(d, e_a) \wedge \mathbf{match}(a(f(d, e_a)), \alpha))$ $\wedge (\mathbf{RHS}_{\Phi}(\phi)[g_a(d, e_a)/d])$
$\mathbf{RHS}_{\Phi}(\sigma X(d_f:D_f := e). \phi)$	$= \tilde{X}(d, e, \mathbf{Par}_{\square}(X, \Phi))$
$\mathbf{match}(a(v), b)$	$= b$
$\mathbf{match}(a(v), a(d))$	$= v = d$
$\mathbf{match}(a(v), a'(d))$	$= \perp$
$\mathbf{match}(a(v), \neg \alpha)$	$= \neg \mathbf{match}(a(v), \alpha)$
$\mathbf{match}(a(v), \alpha_1 \wedge \alpha_2)$	$= \mathbf{match}(a(v), \alpha_1) \wedge \mathbf{match}(a(v), \alpha_2)$
$\mathbf{match}(a(v), \forall d:D. \alpha)$	$= \forall d:D. \ \mathbf{match}(a(v), \alpha)$
$\mathbf{Par}_l(X, b)$	$= \square$
$\mathbf{Par}_l(X, X(e))$	$= \square$
$\mathbf{Par}_l(X, \phi_1 \oplus \phi_2)$	$= \mathbf{Par}_l(X, \phi_1) \ ++ \ \mathbf{Par}_l(X, \phi_2)$
$\mathbf{Par}_l(X, \mathbf{Q} \ d:D. \phi)$	$= \mathbf{Par}_{[d:D]++l}(X, \phi)$
$\mathbf{Par}_l(X, [\alpha]\phi)$	$= \mathbf{Par}_l(X, \phi)$
$\mathbf{Par}_l(X, \langle \alpha \rangle \phi)$	$= \mathbf{Par}_l(X, \phi)$
$\mathbf{Par}_l(X, \sigma Z(d_f:D_f := e). \phi)$	$= \begin{cases} l & \text{if } Z = X \\ \mathbf{Par}_{[d_f:D_f]++l}(X, \phi) & \text{otherwise} \end{cases}$

E Encoding of Branching Bisimulation

We assume a specification S and an implementation M given by the following LPEs, where $\mathcal{Act}_\tau =_{def} \mathcal{Act} \cup \{\tau\}$:

$$\begin{aligned} M(d:D) &= \sum \left\{ \sum_{e:E_a} c_a(d, e) \Longrightarrow a(f_a(d, e)) \cdot M(g_a(d, e)) \mid a \in \mathcal{Act}_\tau \right\} \\ S(d':D') &= \sum \left\{ \sum_{e:E'_a} c'_a(d', e) \Longrightarrow a(f'_a(d', e)) \cdot S(g'_a(d', e)) \mid a \in \mathcal{Act}_\tau \right\} \end{aligned}$$

The largest branching bisimulation relation that relates states of M to S and *vice versa*, can be found by solving the equation system $\nu E_1 \mu E_2$ of Table 2, and therefore also the problem whether M is branching bisimilar to S for given initial states of M and S .

Table 2. Branching bisimilarity between M and S encoded as the equation system $\nu E_1 \mu E_2$.

$$\begin{aligned} E_1 &:= \{ (X(d:D, d':D') = \bigwedge_{a \in \mathcal{Act}_\tau} ((\forall e:E. c_a(d, e) \Longrightarrow Y_a(d, d', e)) \\ &\quad \wedge (\forall e:E'. c'_a(d', e) \Longrightarrow Y'_a(d', d, e))))), \\ &\quad (X'(d':D', d:D) = X(d, d')) \} \\ E_2 &:= \{ (Y_a(d:D, d':D', e:E_a) = (\exists e':E'_\tau. c'_\tau(d', e') \wedge Y_a(d, g'_\tau(d', e'), e)) \\ &\quad \vee (X(d, d') \wedge \exists e':E'_a. c'_a(d', e') \wedge f_a(d, e) = f'_a(d', e') \\ &\quad \wedge X(g_a(d, e), g'_a(d', e')))), \\ &\quad (Y_\tau(d:D, d':D', e:E_\tau) = (\exists e':E'_\tau. c'_\tau(d', e') \wedge Y_\tau(d, g'_\tau(d', e'), e)) \\ &\quad \vee (X(d, d') \wedge (X(g_\tau(d, e), d') \\ &\quad \vee \exists e':E'_\tau. c'_\tau(d', e') \wedge X(g_\tau(d, e), g'_\tau(d', e'))))), \\ &\quad (Y'_a(d':D', d:D, e:E'_a) = (\exists e':E_\tau. c_\tau(d, e') \wedge Y'_a(d, g_\tau(d, e'), e)) \\ &\quad \vee (X'(d', d) \wedge \exists e':E_a. c_a(d, e') \wedge f_a(d, e') = f'_a(d', e) \\ &\quad \wedge X'(g'_a(d', e), g_a(d, e')))), \\ &\quad (Y'_\tau(d':D', d:D, e:E'_\tau) = (\exists e':E_\tau. c_\tau(d, e') \wedge Y'_\tau(d', g_\tau(d, e'), e)) \\ &\quad \vee (X'(d', d) \wedge (X'(g'_\tau(d', e), d) \\ &\quad \vee (\exists e':E_\tau. c_\tau(d, e') \wedge X'(g'_\tau(d', e), g_\tau(d, e'))))) \mid a \in \mathcal{Act} \} \end{aligned}$$
