Extraction of Invariants in Parameterised Boolean Equation Systems

Edward Liem

Eindhoven University of Technology, The Netherlands
e.liem@student.tue.nl

Abstract. Parameterised Boolean Equation Systems (PBESs) are used to express and solve various model checking and equivalence checking problems. However, it may not always be efficient, or even possible, to find a solution to PBESs since they may encode undecidable problems. One particular technique towards finding a solution to a PBES is the concept of exploiting global PBES invariants. Although invariants have been studied extensively, there is a lack of research towards invariant discovery and exploitation in PBESs. Our paper presents PBES invariant extraction techniques inspired from various concepts found in program verification literature well as provide new conditions for invariance properties. We also present a novel graph structure, namely relevancy graphs, which characterize relevant predicate variable instances of instantiated PBES equations. Using relevancy graphs, we illustrate how invariants interact with PBESs as well as provide an alternative criteria to proving the PBES global invariant condition in simple functions.

1 Introduction

Parameterised Boolean Equation Systems (PBESs) [1] allow one to express model checking and equivalence checking problems of infinite size; by solving the PBES associated with a particular problem, one finds the solution to the original problem. Although verification problems may be undecidable, various techniques has been developed to solve their PBES encodings. These techniques include symbolic approximation [1, 2], pattern matching [1], instantiation [3], static analysis [4, 5], and invariants [6]. Our research focuses on PBES invariants. While there has been research on invariants and their application in PBESs, it is not yet clear how to automatically extract these invariants from arbitrary PBES equations. Additionally, to what extent these invariants play a role in PBES solving tools has not been thoroughly explored.

Invariants appear in a variety of areas. Their use can be seen in algebra, geometry, topology, and more [7–10]. In the realm of computer science, invariant based programming methodologies have stressed the importance of constructing invariants before program code has been established [11]. In particular, loop invariants in programs allow one to formally reason about correctness using Hoare logic [12] and have been used in various formal specification languages and tools [13–15]. Alternatively, high performance computing applications take

advantage of invariants to determine data dependencies, enabling safe and efficient execution of code in massive parallel processor architectures [16]. Furthermore, invariants have also been used to prove safety properties in various protocols [17, 18]. While invariants play a significant role in a variety of applications, formulation of these invariants may not always be trivial. Therefore, it is meaningful to find and extract these invariants automatically.

Our research utilizes techniques found in program verification literature to construct PBES invariants. Specifically, we take concepts from static analysis techniques, such as PBES guards and control flow graphs, to derive PBES invariants. We uncover the relationship between guards and invariants by showing how invariants of a given structure can be generated from guards. Additionally, we investigate invariant extraction using control flow graphs. Since work by Keiren et al. has shown how irrelevant parameters in PBESs could be eliminated by means of control flow graphs (CFGs) [5], we analyze CFG properties and show how global PBES invariants are derived from such graphs. In the process, we have uncovered flaws with the original CFG definitions and have defined additional constraints to avoid erroneous CFG constructions and properly prove our CFG derived invariant theorem. Along with the theorems in this paper, we also provide a variety of lemmas that adds onto the original analysis of global PBES invariants by Orzan et al. [6]. Moreover, we present a novel graph structure, namely *relevancy graphs*, providing an alternative characterization of global invariants and assists in visualizing the relationship between invariants and equation instantiations of PBESs. These relevancy graphs also give rise to a new approach as to how one may prove the invariance property for simple functions in PBESs. Thus, the aim of this paper is to answer the following research questions:

How can PBES invariants be derived from techniques found in various program verification literature?

Namely, our paper answers the following research questions in detail:

How are invariants derived from PBES guards?

What is the relationship between Control Flow Graphs (CFGs) and PBES global invariants?

In addition, we also address the following research question:

Are there alternative characterizations of PBES invariants and how could they be used to prove invariance properties in PBESs?

In the following section, we provide related work on the topic of invariants and PBES solving techniques. To provide theoretical background information, we discuss preliminaries in Section 3. Sections 4, 5 and 6 illustrates our work: guard derived invariants are discussed in Section 4, Section 5 shows how invariants are constructed using control flow graphs, and the concept of relevancy graphs and their relationship with PBES invariants is presented in Section 6. We conclude the paper as well as discuss future work in Section 7. For the sake of brevity and conciseness, full details are provided in Appendix A for lemmas and properties for which we have omitted the proof in the main sections of this paper.

2 Related Works

There have been numerous developments towards finding solutions in Parameterised Boolean Equation Systems. In [1, 2], symbolic approximations allow substitution of equations with their stable approximates. However, this procedure does not guarantee termination due to the infinite nature of the lattices in question. In such cases, one may instead attempt transfinite approximations. Groote et al. also formalizes the pattern matching technique [1]; PBES equations may simply be solved by looking up a solution to a pattern. Another technique is PBES instantiation; instantiation has been used to derive solutions by transforming PBESs into Boolean Equation Systems (BESs) and solving the resulting BES [3]. An alternative approach for solving PBESs is finding and solving a bisimulation quotient of a given PBES [19]. Moreover, static analysis techniques tailored for PBESs have also been explored in [4] and [5]. Specifically, Orzan et al. have developed methods for eliminating redundant constant parameters [4]. In a similar vein, Keiren et al. have defined a graph structure which characterizes the values of certain parameters of PBESs, enabling elimination of quantifiers in certain cases and potentially reducing the underlying BES instantiation [5].

Our focus is the use of invariants in the context of PBESs. Although Groote et al. [1] introduces the notion of local PBES invariants, these invariants do not hold under certain PBES transformation techniques. Later works by Orzan et al. fixes these issues by introducing the notion of global invariants [6]. In their paper, it has been observed that PBES invariants may have the potential to realize solutions that would have otherwise been difficult to extract via other PBES solving techniques.

There have been numerous studies on the application of invariants. For example, geometric invariants have been used to express unchanging structures in objects for object recognition [8]; topological invariants can be seen in spacial data modeling for geographical information systems [20]; knot invariants are used to distinguish mathematical knots in knot theory [10]; algebraic invariants express equivalence classes in equations [7]. In computer science, invariants have been used in programming to enhance the correctness of software components. Loop invariants have been used to reason on the correctness of program loops through formal reasoning, such as Hoare logic [12, 21]. Language specifications, such as JML [14], utilize loop invariants to ensure program correctness. Additionally, tools such as Dafny [15] and Spec# [13] use SMT-based verification with user defined invariants to analyze program code and generate correct programs. Furthermore, Invariant-based programming methodologies stresses the use of formalizing invariants before development of software to enforce the idea of program correctness [21, 11]. Moreover, there have been various papers on how invariants are used to verify the maintenance of safety properties in various protocols, e.g. [17, 18, 22].

In addition to the application of invariants, there has been research on automatically deriving invariants. For instance, research has been done on discovering invariants in imperative programs. In particular, the Daikon system discovers likely invariants from programs via trace analysis [23]. To generate additional

invariants, Zhang et al. has paired symbolic program execution with Daikon [24]. Symbolic approaches also appear in [25] where researchers utilize iterative techniques to strengthen and refine invariants. Others, such as [26], use a collection of automated heuristics to strengthen a given invariant; these heuristics perform operations on the reachable set of states in a finite instance of the system. Likewise, [27] has applied heuristics to strengthen invariants on failed proof attempts. Similarly, papers such as [17, 28, 18] have used a counterexample-based approach to generate invariants to strengthen a given safety property. Meanwhile, Garoche et al. has proposed automatic generation of abstract interpreters which give rise to on-the-fly invariant generation [29].

The methods for invariant generation mentioned thus far have been primarily focused on strengthening some safety property in the form of an invariant by discovering and adding so called *auxiliary* invariants. Rather than simply finding some invariant, these methods aim to automatically discover *inductive* invariants, i.e. invariants that remain true in all possible transitions in a given state space rather than a set of reachable transitions. In addition to the previously mentioned works, papers such as [30] and [31] propose methods on how to generate such auxiliary invariants using transition systems.

While some of the techniques mentioned thus far seem promising, preliminary research needs to be conducted on PBESs beforehand. For instance, it is not clear how to automatically generate statements satisfying PBES invariance conditions from a given template when working with arbitrary PBESs consisting of multiple equations with parameters of arbitrary sorts. Additionally, unlike transition systems, PBESs do not exhibit a natural transition relation, preventing us from simply applying techniques used for imperative programs.

3 Preliminaries

This section provides an introduction to the theory behind Parameterised Boolean Equation Systems (PBESs) and how invariants are defined over PBESs. We first discuss the notion of predicate formulae and the structure of PBESs. Afterwards, we introduce the notion of global invariants in the context of PBESs.

In this paper, we work with *abstract data types*; we assume non-empty data sorts typically written as letters D, E, F. We also assume the existence of a sort B representing the Booleans $\mathbb{B}$ and the sort N representing natural numbers $\mathbb{N}$. For these sorts, we assume the existence of the standard associated operations. Observe that we differentiate between syntactic objects and semantic objects by using distinguishing fonts, e.g. B for syntactic boolean sorts and $\mathbb{B}$ for semantic booleans. To represent vectors of data variables, we use bold fonts, e.g. $\mathbf{d}$. This notation also extends to vectors of data sorts $\mathbf{D}$ and vectors of data terms $\mathbf{e}$.

Let $\mathcal{P}$ be the set of predicate variables. For predicate variable $X \in \mathcal{P}$, we associate a vector of data variables $\mathbf{d}_X$ of sort $\mathbf{D}_X$.

Definition 1. Predicate formulae ϕ, ψ *are defined using the following grammar:*

$$\phi, \psi ::= b \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall d : D.\phi \mid \exists d : D.\phi \mid X(\mathbf{e})$$

where b is a data term of sort B possibly containing data variables $d \in \mathcal{D}$, $X(\mathbf{e})$ a predicate variable instance *(PVI), X a predicate variable from $\mathcal{P}$ of sort $\mathbf{D}_X \to B$, and $\mathbf{e}$ a vector of data terms of sort $\mathbf{D}_X$.*

Remark 1. We assume that data terms do not contain predicate variables. E.g. $X(X(\mathbf{e}))$ would not be a valid predicate formula.

The set of *free variables* of predicate formula ϕ, denoted $\mathsf{FV}(\phi)$, is a set of variables not bounded by a universal or existential quantifier in ϕ. This concept extends to data terms and vectors in a standard way. Additionally, we denote the set of all predicate formulas as Pred.

A predicate formula ϕ that does not contain predicate variables is a *simple predicate*. Furthermore, we say a predicate formula ϕ is *normalized* iff negations only occur before a boolean term b. Moreover, ϕ is *capture-avoiding* iff all free variables of ϕ does not have a bound occurrence in ϕ, i.e. there does not exist a subformula $(\mathsf{Q}d{:}D.\ \psi)$ of ϕ such that $d \in \mathsf{FV}(\phi)$, where $\mathsf{Q} \in \{\exists, \forall\}$. Note that a predicate formula can be transformed into an equivalent capture-avoiding formula via appropriate renamings of variables bound in quantifiers.

Throughout our paper, we may reference specific PVIs in a predicate formula. Specifically, the ith PVI in a predicate formula ϕ is denoted as $\mathsf{PVI}(\phi, i)$ for $i \in \mathbb{N}$ and $1 \leq i \leq \mathsf{npred}(\phi)$, where $\mathsf{npred}(\phi)$ is the number of PVIs in ϕ. Furthermore, we write $\mathsf{pv}(\phi, i)$ to refer to the ith predicate variable in ϕ and $\mathsf{arg}(\phi, i)$ for the arguments in the ith PVI of ϕ. In other words, $\mathsf{PVI}(\phi, i) = \mathsf{pv}(\phi, i)(\mathsf{arg}(\phi, i))$. We may also refer to the jth argument in the ith PVI of ϕ as $\mathsf{arg}_j(\phi, i)$.

We also define syntactic replacement of PVIs; given predicate formulas ϕ, ψ, the replacement of the ith PVI with ψ in ϕ is denoted as $\phi[i \mapsto \psi]$. The complete definition is provided below:

Definition 2. *Let ϕ be a predicate formula and $1 \leq i \leq \mathsf{npred}(\phi)$. The* syntactic replacement *of the ith PVI in ϕ with ψ, denoted $\phi[i \mapsto \psi]$, is defined inductively as follows:*

$$
\begin{aligned}
b[i \mapsto \psi] &= b \\
Y(e)[i \mapsto \psi] &= \begin{cases} \psi & \text{if } i = 1 \\ Y(e) & \text{otherwise} \end{cases} \\
(\forall d{:}D.\ \phi)[i \mapsto \psi] &= \forall d{:}D.\ \phi[i \mapsto \psi] \\
(\exists d{:}D.\ \phi)[i \mapsto \psi] &= \exists d{:}D.\ \phi[i \mapsto \psi] \\
(\phi_1 \wedge \phi_2)[i \mapsto \psi] &= \begin{cases} \phi_1 \wedge \phi_2[(i - \mathsf{npred}(\phi_1)) \mapsto \psi] & \text{if } i > \mathsf{npred}(\phi_1) \\ \phi_1[i \mapsto \psi] \wedge \phi_2 & \text{if } i \leq \mathsf{npred}(\phi_1) \end{cases} \\
(\phi_1 \vee \phi_2)[i \mapsto \psi] &= \begin{cases} \phi_1 \wedge \phi_2[(i - \mathsf{npred}(\phi_1)) \mapsto \psi] & \text{if } i > \mathsf{npred}(\phi_1) \\ \phi_1[i \mapsto \psi] \wedge \phi_2 & \text{if } i \leq \mathsf{npred}(\phi_1) \end{cases}
\end{aligned}
$$

At times, we perform consecutive syntactic replacement:

Definition 3. *Let ϕ be an arbitrary predicate formula and $I = \langle i_1, \ldots, i_n \rangle$ be a vector of indices such that $n \leq \mathsf{npred}(\phi)$. Let $\psi_{i_1}, \ldots, \psi_{i_n}$ be arbitrary predicate formulae.* Consecutive syntactic replacement $\phi[i \mapsto \psi_j]_{j \in \langle i_1, \ldots, i_n \rangle}$ *for predicate formulae ϕ is defined as follows:*

$$\begin{aligned} \phi[i \mapsto \psi_j]_{j \in \langle \rangle} &= \phi \\ \phi[i \mapsto \psi_j]_{j \in \langle i_1, \ldots, i_n \rangle} &= (\phi[i \mapsto \psi_{i_1}])[i \mapsto \psi_j]_{j \in \langle i_2, \ldots, i_n \rangle} \end{aligned}$$

One must exercise caution when using consecutive syntactic replacement as replacing a PVI with a predicate formula which does not contain exactly one predicate variable may lead to unintended or undefined behavior. Thus, we generally utilize consecutive replacements in a safe way, i.e. the formula replacing a PVI contains exactly one predicate variable. In the case where consecutive syntactic replacement behaves as simultaneous syntactic replacement, we may write $\phi[i \mapsto \psi_i]_{i \in I}$. In some cases, we may even write $\phi[i \mapsto \psi_i]_{i \leq \mathsf{npred}(\phi)}$ for the simultaneous syntactic replacement of all PVIs in ϕ.

Given a closed term t, i.e. $\mathsf{FV}(t) = \emptyset$, the interpretation function $[\![\_]\!]$ maps the term t to the semantic data element $[\![t]\!]$ it represents. For open terms, we utilize a *data environment* ε that maps each variable $d \in \mathcal{D}$ to a data element of the appropriate sort. The interpretation of an open term t under the context of a data environment ε is denoted as $[\![t]\!]\varepsilon$ which is evaluated in the standard way. Interpretation of vectors consisting of terms are also done in a standard way.

Predicate formulae are interpreted under the context of a data environment ε and *predicate environment* $\eta : \mathcal{P} \to (\mathbb{D} \to \mathbb{B})$, mapping predicate variables X to a function of type $\mathbb{D}_X \to \mathbb{B}$. For an arbitrary environment $\theta \in \{\varepsilon, \eta\}$, the notation $\theta[v/d]$ denotes that variable d has been assigned value v, i.e. $\theta[v/d](d') = \theta(d')$ for $d \neq d'$ and $\theta[v/d](d) = v$ otherwise. Extensions for vector substitutions are standard.

Definition 4. *Let ϕ be a predicate formula, ε be a data environment, and η be a predicate environment. The interpretation of ϕ in the context of ε and η is denoted as $[\![\phi]\!]\eta\varepsilon$ where:*

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

We write $\phi \to \psi$ iff for all predicate environments η and data environments ε, $[\![\phi]\!]\eta\varepsilon$ implies $[\![\psi]\!]\eta\varepsilon$. This gives rise to a logical equivalence between predicate formulas, where $\phi \leftrightarrow \psi$ iff $\phi \to \psi$ and $\psi \to \phi$. Additionally, we operate with the assumption of the general principle of substitutivity, i.e. if $\phi \to \psi$, then $\phi[v/d] \to \psi[v/d]$.

We now introduce *predicate functions* which casts predicate formulas to functions. To indicate that predicate formula ϕ has been lifted to a function $(\lambda \mathbf{d}_X : \mathbf{D}_X.\ \phi)$, we write $\phi_{\langle \mathbf{d}_X \rangle}$. The interpretation of $\phi_{\langle \mathbf{d}_X \rangle}$ under predicate environment η and data environment ε, written as $[\![\phi_{\langle \mathbf{d}_X \rangle}]\!]\eta\varepsilon$, is defined in the following way:

$$[\![\phi_{\langle \mathbf{d}_X \rangle}]\!]\eta\varepsilon = \lambda \mathbf{v} \in \mathbb{D}_X.\ [\![\phi]\!]\eta\varepsilon[\mathbf{v}/\mathbf{d}_X]$$

The syntactic substitution of predicate function $\psi_{\langle \mathbf{d}_X \rangle}$ for predicate variable X in predicate formula ϕ is defined as follows:

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

Rather than perform a single substitution, we may wish to perform a series of substitutions. The definition for a finite sequence of substitutions of the form $\phi[\psi_{1\langle \mathbf{d}_{X_1} \rangle}/X_1][\psi_{2\langle \mathbf{d}_{X_2} \rangle}/X_2] \ldots [\psi_{n\langle \mathbf{d}_{X_n} \rangle}/X_n]$ is as follows:

Definition 5. *Let $V = \langle X_1, \ldots, X_n \rangle$ be a vector of predicate variables and let ψ_i be arbitrary predicate formulae for all $i \in \mathbb{N}$, $1 \leq i \leq n$.* Consecutive substitution *$\phi[_{X_i \in V}\psi_{i\langle \mathbf{d}_{X_i} \rangle}/X_i]$ for predicate formula ϕ is defined as follows:*

$$\begin{aligned}
\phi\Big[_{X_i \in \langle\rangle}\psi_{i\langle \mathbf{d}_{X_i} \rangle}/X_i\Big] &= \phi \\
\phi\Big[_{X_i \in \langle X_1, \ldots, X_n \rangle}\psi_{i\langle \mathbf{d}_{X_i} \rangle}/X_i\Big] &= (\phi[\phi_{1\langle \mathbf{d}_{X_1} \rangle}/X_1])\Big[_{X_i \in \langle X_2, \ldots, X_n \rangle}\psi_{i\langle \mathbf{d}_{X_i} \rangle}/X_i\Big]
\end{aligned}$$

When variable X_i only occurs in ϕ_i for all i and all variables in the vector $\langle X_1, \ldots, X_n \rangle$ are distinct, consecutive substitution behaves as simultaneous substitution. Lemma 16 of Invariants for PBESs [6] expresses this phenomenon. Generally, we utilize consecutive substitutions in a way that behaves as simultaneous substitution and we abuse notation by writing $\phi[_{X_i \in \{X_1, \ldots, X_n\}}\phi_{i\langle \mathbf{d}_{X_i} \rangle}/X_i]$.

At times, we shift syntactic substitutions to semantic updates for environments. Since we assume that a data term b of sort B is evaluated in a standard way, we also operate under the assumption that $[\![b[e/d]]\!]\varepsilon = [\![b]\!]\varepsilon[[\![e]\!]\varepsilon/d]$.

Definition 6. *A Parameterised Boolean Equation System (*PBES *or simply* equation system*) is defined by the following grammar:*

$$\mathscr{E} ::= \epsilon \mid (\mu X(\mathbf{d}_X : \mathbf{D}_X) = \phi)\mathscr{E} \mid (\nu X(\mathbf{d}_X : \mathbf{D}_X) = \phi)\mathscr{E}$$

where ϵ is the empty PBES, μ is the least fixed point, ν is the greatest fixed point, and ϕ is a predicate formula where $\mathbf{d}_X$ of sort $\mathbf{D}_X$ is a vector of formal parameters of X and is considered bound in the equation for X.

For convenience, ϕ_X refers to the right-hand side of the defining equation for X in a PBES $\mathscr{E}$. Moreover, $\mathsf{par}(X)$ denotes the set of formal parameters of X. At times, there may be situations where equations share the same parameter name; we distinguish the differences by using superscripts. E.g. if $i \in \mathsf{par}(X)$ and $i \in \mathsf{par}(Y)$, then we write i^X to denote the parameter i in equation X. We may also write σ, which could stand for μ or ν.

An equation system is *closed* whenever all predicate variables on the right-hand side of every equation occur in the left-hand side of some equation, otherwise the equation system is considered *open*. For our paper, we only consider closed equation systems. In addition, given an equation system $\mathscr{E}$, the set of *defined variables* (notated as $\mathsf{bnd}(\mathscr{E})$) consists of all predicate variables occurring in the left-hand side of each equation in $\mathscr{E}$.

Consider all functions $f, g : \mathbb{D} \to \mathbb{B}$ for some arbitrary data set $\mathbb{D}$. The ordering $\sqsubseteq$ on the set of all functions $\mathbb{D}$ to $\mathbb{B}$, denoted as $[\mathbb{D} \to \mathbb{B}]$, is defined as $f \sqsubseteq g$ iff for all $v \in \mathbb{D}$, $f(v)$ implies $g(v)$. Observe that $([\mathbb{D} \to \mathbb{B}], \sqsubseteq)$ is a complete lattice. A *predicate transformer* associated to $[\![\phi_{\langle d_X \rangle}]\!]\eta\varepsilon$ is a function of type $[\mathbb{D}_X \to \mathbb{B}] \to [\mathbb{D}_X \to \mathbb{B}]$, defined as

$$\lambda f \in [\mathbb{D}_X \to \mathbb{B}].[\![\phi_{\langle d_X \rangle}]\!]\eta[f/X]\varepsilon$$

This predicate transformer is monotonic [1, 32]; the existence of the least and greatest fixed points of this operator in the lattice $([\mathbb{D}_X \to \mathbb{B}], \sqsubseteq)$ is guaranteed to exist by Tarski's fixed point Theorem [33]. We denote these fixed points as $\sigma f \in [\mathbb{D}_X \to \mathbb{B}].[\![\phi_{\langle d_X \rangle}]\!]\eta[f/X]\varepsilon$ where $\sigma \in \{\mu, \nu\}$.

Definition 7. *The* solution *of an equation system in the context of a predicate environment η and data environment ε is defined as follows*

$$[\![\epsilon]\!]\eta\varepsilon = \eta$$
$$[\![(\sigma X(\mathbf{d}_X : \mathbf{D}_X) = \phi)\mathscr{E}]\!]\eta\varepsilon = [\![\mathscr{E}]\!](\eta[\sigma f \in [\mathbb{D}_X \to \mathbb{B}].[\![\phi_{\langle \mathbf{d}_X \rangle}]\!](([\![\mathscr{E}]\!]\eta[f/X]\varepsilon)\varepsilon]/X)\varepsilon$$

where $\sigma \in \{\mu, \nu\}$.

Although the notion of invariants in PBESs has been first discussed in [1], Orzan et al. has shown that certain PBES manipulation techniques violate invariance requirements [6]. Therefore, we focus on a stronger notion of invariants, namely *global invariants*. Global invariants use a notion of simple functions: a function $f : V \to \mathsf{Pred}$, such that $V \subseteq \mathcal{P}$, is considered *simple* iff for all $X \in V$, the predicate $f(X)$ is simple.

Definition 8. *A simple function $f : V \to \mathsf{Pred}$ is a* global invariant *for an equation system $\mathscr{E}$ iff $V \supseteq \mathsf{bnd}(\mathscr{E})$ and for each equation $(\sigma X(\mathbf{d}_X : \mathbf{D}_X) = \phi)$ in $\mathscr{E}$, we have that:*

$$(f(X) \wedge \phi) \;\leftrightarrow\; \Big((f(X) \wedge \phi)\big[_{X_i \in V}(f(X_i) \wedge X_i(\mathbf{d}_{X_i}))_{\langle \mathbf{d}_{X_i} \rangle}/X_i\big]\Big)$$

For the remainder of the paper, whenever we refer to a simple function as an 'invariant' for a PBES, we refer to the global invariant definition (Definition 8), rather than the definition presented in [1].

Intuitively, $f(X)$ is an invariant for the equation associated with predicate variable X iff for the parameter space of the equation where the invariant holds, the solution for equation X is not changed by the addition of all invariants defined in f. Note that by adding invariants we might change the solution of the equation since the solutions with and without the invariant are only guaranteed to coincide when the invariant holds.

When working with invariants, we generally limit the scope of the free variables in an invariant to the set of parameters of the corresponding equation. For instance, if $\sigma X(\mathbf{d}_X : \mathbf{D}_X)$ is an equation in some PBES $\mathscr{E}$, we would like that $\mathsf{FV}(f(X)) \subseteq \mathbf{d}_X$ for some invariant f. This is to ensure that we do not introduce open equations to PBESs during implementation of the invariant using Theorem 35 of [6].

In our proofs, we require the observation that the evaluation of an invariant does not depend on its predicate environment. Additionally, evaluations of invariants do not depend on its data environment as long as we have guarantees on the evaluation of its free variables. The lemmas we present generalize the claims to arbitrary simple predicates. First, since we work on environments and their updates, we operate with the following remark in mind:

Remark 2. Let $\theta \in \{\eta, \varepsilon\}$ be some environment. If two updates to θ affects the same variable, the last environment update takes priority. I.e. $(\theta[v/d])[w/d](d) = w$.

We also use the next remark to prove the following lemmas.

Remark 3. For the sake of simplicity, we associate a single variable d_X of sort D_X to predicate variables X rather than use vectors $\mathbf{d}_X$ of sort $\mathbf{D}_X$. This does not result in a loss of generality as one could utilize a data sort which is able to express more complex formulae.

Lemma 1. *Let ϕ be a simple, capture-avoiding predicate such that $\mathsf{FV}(\phi) \subseteq \{d\}$ where $d : D$. Then we have that for any environments $\eta, \varepsilon, \varepsilon'$ and $v \in \mathbb{D}$,*

$$[\![\phi]\!]\eta\varepsilon[v/d] = [\![\phi]\!]\eta\varepsilon'[v/d]$$

Lemma 2. *Let ϕ be a simple predicate formula. Then for any environments η, η', ε,*

$$[\![\phi]\!]\eta\varepsilon = [\![\phi]\!]\eta'\varepsilon$$

Proof. Since ϕ is a simple predicate, there does not exist predicate variables in ϕ to evaluate. Thus, the choice of predicate environments during evaluation of ϕ is irrelevant. □

It follows from these two lemmas that we may use arbitrary data and predicate environments when evaluating simple, capture-avoiding predicates, given that we have guarantees on the evaluation of free variables:

Corollary 1. *Let ϕ be a simple, capture-avoiding predicate such that $\mathsf{FV}(\phi) \subseteq \{d\}$, where $d : D$. Then for any environments $\eta, \varepsilon, \eta', \varepsilon'$ and $v \in \mathbb{D}$,*

$$[\![\phi]\!]\eta\varepsilon[v/d] = [\![\phi]\!]\eta'\varepsilon'[v/d]$$

This corollary is useful in proofs when dealing with interpretation of invariants. Since Corollary 1 requires capture-avoiding predicates, we define capture-avoiding functions. Similar to simple functions, a function $f : V \to \mathsf{Pred}$, where $V \subseteq \mathcal{P}$, is *capture-avoiding* iff for all $X \in V$, the predicate $f(X)$ is capture-avoiding.

The following lemma is also useful. It states that we may shift syntactic substitutions of free variables to the data environment on simple, capture-avoiding predicates.

Lemma 3. *Let ϕ be a simple, capture-avoiding predicate formula and let $d \in \mathsf{FV}(\phi)$ for some $d : D$. Then, for some $e : D$ and any environments η, ε, we have*

$$[\![\phi[e/d]]\!]\eta\varepsilon = [\![\phi]\!]\eta\varepsilon[\varepsilon(e)/d]$$

4 Guards

There have been various approaches towards formula reduction techniques to solve PBESs. Examples include constant elimination [4], partial-order reductions in parity games [34], liveness analysis and control flow graphs [32, 5] to name a few. Some of these techniques analyze the syntactic structure of PBES equations to discover irrelevant subformulae. For instance, in the formula $n \neq 3 \vee X(n)$, $X(n)$ becomes irrelevant whenever $n \neq 3$. Or in other words, we need only to consider the solution of $X(3)$. One method of detecting irrelevant subformulae is to utilize *guards*: a simple predicate formula that characterizes the relevant values of a predicate variable instance. Since guards allow for various techniques to simplify PBES equations, it becomes beneficial to investigate whether there exist a relationship between guards and invariants.

Similar work has been done by Orzan et al. [4]; via static analysis of PBESs, invariants were constructed by detecting constant parameters, which includes the use of guards. However, Orzan et al. requires that equations of PBESs to be in, or transformed to, Predicate Formula Normal Form (PFNF). Transformations into PFNF may result in an exponential increase in formula size. The work presented in this section does not require formulas to adhere to a certain structure.

This section operates under Remark 3. We restate this remark for convenience:

Remark 3. For the sake of simplicity, we associate a single variable d_X of sort D_X to predicate variables X rather than use vectors $\mathbf{d}_X$ of sort $\mathbf{D}_X$. This does not result in a loss of generality as one could utilize a data sort which is able to express more complex formulae.

The need for computing guards first appear in Keiren's thesis [32], where heuristics were created for their concept of PBES control flow graphs. These control flow graphs required constraints on predicate variable instances, with stronger constraints being preferable. Unfortunately, these constraints are not always efficiently computable. Hence, efficient approximations have been developed to construct constraints, as seen in [32, 5, 34]. These heuristics analyze the syntactic conditions of PVIs within a given predicate formula to determine when a PVI is relevant. We refer to these conditions as *guards*.

Definition 9. *Given a predicate formula ϕ, a simple formula ψ is a* guard *for the ith PVI of ϕ iff $\phi \leftrightarrow \phi[i \mapsto (\psi \wedge \mathsf{PVI}(\phi, i))]$.*

For instance, the following guard function from [34] is one such heuristic for generating guards from the structure of predicate formulas.

Definition 10. *Let ϕ be a predicate formula. The function $\mathsf{guard}^i : \mathsf{Pred} \to \mathsf{Pred}$ is defined inductively as follows where $i \leq \mathsf{npred}(\phi)$:*

$$\mathsf{guard}^i(b) = \mathit{false} \qquad \mathsf{guard}^i(\forall d{:}D.\ \phi) = s(\forall d{:}D.\ \phi) \wedge \mathsf{guard}^i(\phi)$$

$$\mathsf{guard}^i(Y(e)) = \mathit{true} \qquad \mathsf{guard}^i(\exists d{:}D.\ \phi) = s(\neg\exists d{:}D.\ \phi) \wedge \mathsf{guard}^i(\phi)$$

$$\mathsf{guard}^i(\phi \wedge \psi) = \begin{cases} s(\phi) \wedge \mathsf{guard}^{i-\mathsf{npred}(\phi)}(\phi) & \text{if } i > \mathsf{npred}(\phi) \\ s(\psi) \wedge \mathsf{guard}^i(\phi) & \text{if } i \leq \mathsf{npred}(\phi) \end{cases}$$

$$\mathsf{guard}^i(\phi \vee \psi) = \begin{cases} s(\neg\phi) \vee \mathsf{guard}^{i-\mathsf{npred}(\phi)}(\phi) & \text{if } i > \mathsf{npred}(\phi) \\ s(\neg\psi) \vee \mathsf{guard}^i(\phi) & \text{if } i \leq \mathsf{npred}(\phi) \end{cases}$$

where

$$s(\phi) = \phi[i \mapsto \mathit{true}]_{i \leq \mathsf{npred}(\phi)}$$
$$s(\neg\phi) = \neg\phi[i \mapsto \mathit{false}]_{i \leq \mathsf{npred}(\phi)}$$

While Keiren et al. [5] proposed a similar guard function, we opt to use Neele's definition. As noted by Neele [34], the current definition of guards (Definition 9) is not compositional, i.e. simultaneous syntactic replacement of all guards into an equation ϕ is not maintained under $\leftrightarrow$. Neele combats this by strengthening their computation guards. A detailed proof of its compositionality is in Neele's thesis [34].

Example 1. Consider the formula $\phi = X(1) \vee X(1)$. By Definition 9, *false* is a guard for both PVIs since

$$\begin{aligned} & X(1) \vee X(1) \\ \leftrightarrow\ & \\ & (X(1) \vee X(1))[1 \mapsto \mathit{false} \wedge X(1)] \\ \leftrightarrow\ & \\ & (X(1) \vee X(1))[2 \mapsto \mathit{false} \wedge X(1)] \end{aligned}$$

However, both guards cannot be applied at the same time:

$$X(1) \vee X(1) \not\leftrightarrow \mathit{false} \leftrightarrow (X(1) \vee X(1))[1 \mapsto \mathit{false} \wedge X(1)][2 \mapsto \mathit{false} \wedge X(1)]$$

Hence, we will mainly be focused on compositional guards.

Definition 11. *Let ϕ be a predicate formula and $I \subseteq \{1, \ldots, \mathsf{npred}(\phi)\}$ be a set of PVI indices. Let ψ_i be a guard for $\mathsf{PVI}(\phi, i)$ where $i \in I$. The collection of guards ψ_i are* compositional *iff $\phi \leftrightarrow \phi[i \mapsto \psi_i \wedge \mathsf{PVI}(\phi, i)]_{i \in I}$.*

Theorem 7.38 ***(Reductions for Parity Games and Model Checking [34]).*** *For all normalized, capture-avoiding formulae ϕ, it holds that*

$$\phi \leftrightarrow \phi[i \mapsto (\mathsf{guard}^i(\phi) \wedge \mathsf{PVI}(\phi, i))]_{i \leq \mathsf{npred}(\phi)}$$

To assist in finding invariants, Orzan et al. has shown that the following property provides a sufficient condition for a simple function f to be a global invariant [6].

Property 1. Let $\mathscr{E}$ be a closed equation system. Additionally, assume that there exists a guard for all PVIs in $\mathscr{E}$. Let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple function such that for every equation $(\sigma X(d_X : D_X) = \phi_X)$ in $\mathscr{E}$ we have:

$$f(X) \to \bigwedge_{\mathsf{PVI}(\phi_X, i) = Y(e)} (f(Y))[e/d_Y]$$

Then f is a global invariant for $\mathscr{E}$.

This property states that a simple function f is a global invariant if we can prove that for all equations $\sigma X(d_X : D_X) = \phi_X$, the invariants of all PVIs in ϕ_X hold after substitution of argument values. However, this property assumes that all PVIs in a given equation are relevant regardless of their surrounding context.

Example 2. Consider the following PBES:

$$\mu X(i : N) = (\mathit{true} \vee X(1)) \wedge X(2)$$

Let $f(X) = (i = 2)$. By Property 1, to show that f is a global invariant we must show that

$$(i = 2) \to (1 = 2)$$
$$(i = 2) \to (2 = 2)$$

However, $X(1)$ is irrelevant since $(\mathit{true} \vee \psi) \leftrightarrow \mathit{true}$ for any predicate formula ψ. It is sufficient to only show $(i = 2) \to (2 = 2)$ to prove that f is a global invariant.

Therefore, we add an additional condition to this property which states that an invariant of a certain PBES equation must be considered only if its guard holds.

Property 2. Let $\mathscr{E}$ be a closed equation system. For every equation $(\sigma X(d_X : D_X) = \phi_X)$ in $\mathscr{E}$, assume the existence of compositional guards $\gamma^1_{\phi_X}, \ldots, \gamma^{\mathsf{npred}(\phi_X)}_{\phi_X}$ for ϕ_X where $\gamma^i_{\phi_X}$ is the guard for $\mathsf{PVI}(\phi_X, i)$. Let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple function such that for every equation $(\sigma X(d_X : D_X) = \phi_X)$ in $\mathscr{E}$ and all $\mathsf{PVI}(\phi_X, i) = X_i(e_i)$:

$$f(X) \wedge \gamma^i_{\phi_X} \to f(X_i)[e_i/d_{X_i}]$$

Then f is a global invariant for $\mathscr{E}$.

Proof. Consider an equation $(\sigma X(d_X : D_X) = \phi_X)$ for which $f(X) \wedge \gamma^i_{\phi_X} \to f(X_i)[e/d_{X_i}]$ holds for all $\mathsf{PVI}(\phi_X, i) = X_i(e_i)$. By definition of compositional guards, we have that

$$\phi_X \leftrightarrow \phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$$

Note that $\phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$ can be described by the following grammar since each PVI occurs within the scope of a conjunction:

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

where γ is the guard for $X(e)$.

We now perform structural induction on the structure of the subformulae ψ of $\phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$. For each subformula, it suffices to prove that

$$(f(X) \wedge \psi) \leftrightarrow (f(X) \wedge \psi)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big]$$

We first consider the base cases:

Case $\psi = b$. Immediately follows from the definition of syntactic substitution:

$$(f(X) \wedge b) \leftrightarrow (f(X) \wedge b)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big]$$

Case $\psi = \gamma^i_{\phi_X} \wedge Y(e)$. Let $Y(e) = \mathsf{PVI}(\phi_X, i)$. We then reason as follows:

$$\begin{aligned}
&(f(X) \wedge (\gamma^i_{\phi_X} \wedge Y(e)))\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big] \\
\leftrightarrow\ &\{\text{Definition of syntactic substitution; } f(X) \text{ and } \gamma^i_{\phi_X} \text{ are simple}\} \\
&f(X) \wedge (f(Y)[e/d_Y]) \wedge (\gamma^i_{\phi_X} \wedge Y(e)) \\
\leftrightarrow\ &\{\text{Assumption: } f(X) \wedge \gamma^i_{\phi_X} \to f(Y)[e/d_Y]\} \\
&f(X) \wedge (\gamma^i_{\phi_X} \wedge Y(e))
\end{aligned}$$

The remaining cases follows from the definition of substitution in a straightforward way. For brevity, we place the remainder of the proof in Appendix A.2. □

Although guards appear to resemble a structure similar to the definition of global invariants, guards alone fail to be invariants. Intuitively, guards characterize when a certain PVI is relevant in a particular PBES equation while invariants describe what must hold when exploring predicate variables with certain values. A guard shows what must hold at a particular location within a given predicate formula, and by itself, does not provide sufficient information on the values of parameters when considering the arguments of PVIs.

Example 3. Consider the following PBES.

$$\begin{aligned}
&(\mu X(i : N) = (i \leq 8) \wedge Y(i + 2)) \\
&(\mu Y(j : N) = (j = 4) \vee (j < 5 \wedge X(2j)))
\end{aligned} \tag{1}$$

The guard function from Definition 10 gives rise to the following guards:

$$\mathsf{pv}(\phi_Y, 1) = X \qquad \mathsf{pv}(\phi_X, 1) = Y$$
$$\mathsf{guard}^1(\phi_Y) = (j \neq 4) \wedge (j < 5) \qquad \mathsf{guard}^1(\phi_X) = (i \leq 8)$$

In other words, the truth of $X(2j)$ in ϕ_Y is irrelevant when $j = 4$ or if $j \geq 5$. However, observe that constructing a simple function f such that

$$f(X) = (j \neq 4) \wedge (j < 5) \qquad f(Y) = (i \leq 8)$$

will not yield proper invariants. First of all, $f(X)$ contains the variable j, which is free in the equation of X. Recall that to avoid open equations when integrating invariants into a PBES, we do not want variables that are not part of an equation's parameters to be in the invariant. The problem is symmetrical for Y. For this example, we can remedy this particular issue by swapping the variables such that f does not contain irrelevant free variables. Note that this approach may not work in general due to quantifier bound variables; we address this issue later in this section. For now, let f be the simple function such that

$$f(X) = (i \neq 4) \wedge (i < 5) \qquad f(Y) = (j \leq 8)$$

Still, the function f cannot be a global invariant.

$$\begin{aligned}
& f(Y) \wedge \phi_Y[(f(X) \wedge X(d_X))_{\langle d_X \rangle}/X] \\
\leftrightarrow\ & \{\text{Construction of } f\} \\
& (j \leq 8) \wedge \phi_Y[(f(X) \wedge X(d_X))_{\langle d_X \rangle}/X] \\
\leftrightarrow\ & \{\text{Definition of simultaneous substitution}\} \\
& (j \leq 8) \wedge (j = 4 \vee (j < 5 \wedge ((2j \neq 4) \wedge (2j < 5) \wedge X(2j)))) \\
\not\leftrightarrow\ & \{\text{Global invariant requirement}\} \\
& (j \leq 8) \wedge (j = 4 \vee (j < 5 \wedge X(2j)))
\end{aligned}$$

The global invariant requirement fails to hold if $j = 2$. The problem lies with operations within PVI arguments. In this example, the fact that we double j in $X(2j)$, the invariant of X would need to be $(i \neq 8) \wedge (i < 10)$; if $j \neq 4$ at $X(2j)$, in $X(2j)$ it must be $i \neq 8$ since $2j \neq 8$. In the end, we would like to get the following invariants from the information of guards:

$$f(X) = (i \neq 8) \wedge (i < 10) \qquad f(Y) = (j \leq 10)$$

As we have seen, one problem preventing guards from becoming invariants is substitution with arguments of PVIs. Because arguments may contain operations or functions, guards alone are not sufficient in determining the values of variables after certain operations. One option is to replace PVIs containing operators within their arguments with PVIs that do not contain operators by introducing existential quantifiers. For instance, PBES (1) can be transformed into

$$(\mu X(i : N) = (i \leq 8) \wedge \exists n{:}N.\ ((n = i + 2) \wedge Y(n)))$$
$$(\mu Y(j : N) = (j = 4) \wedge (j < 5 \wedge \exists n{:}N.\ ((n = 2j) \wedge (j < 5) \wedge X(n))))$$

While these transformations address the problem with operators in PVI arguments, such replacement of PVIs will introduce unnecessary complexity in larger equation systems when checking and proving the validity of invariants. Additionally, we wish to reason about individual predicate variable instances of equations rather than subformulae consisting of existential quantifiers. We therefore take inspiration from Hoare logic [12] and Dijkstra's predicate transformers [35]. A Hoare triple $\{P\}S\{Q\}$ describes the execution of statement S; if the precondition P holds, the postcondition Q will be maintained after the execution of S. In our case, we wish to find the strongest statement that will hold after traversing through some PVI. Thus, if $\mathsf{PVI}(\phi, i) = X(e)$, we wish to find the strongest postcondition Q such that

$$\{\gamma_\phi^i\}(d_X{:=}e)\{Q\}$$

where γ_ϕ^i is a guard for the ith PVI of predicate ϕ and $d_X := e$ represents the assignment of variable d_X with e.

Example 4. Consider PBES (1) in Example 3. We wish to find a suitable postcondition Q such that

$$\{(j \neq 4) \wedge (j < 5)\}(i := 2j)\{Q\}$$

Observe that $(i \neq 8) \wedge (i < 10)$ is a suitable option for Q.

Taking concepts from Dijkstra's predicate transformers [35], we define the following definition for the strongest postcondition function when assigning a data variable d of type D to a data term e, where ϕ is a simple predicate formula representing the precondition.

Definition 12. *Let d be a data variable of type D, e a data term of type D, and ϕ be a simple predicate formula. The predicate formula* $\mathsf{sp}(d := e, \phi)$ *represents the* strongest postcondition *when assigning a data variable d to the term e, given a precondition ϕ. It is defined as follows:*

$$\mathsf{sp}(d := e, \phi) = \exists d'{:}D.\ ((d = e[d'/d]) \wedge \phi[d'/d])$$

where d' does not occur in ϕ.

Lemma 4. *Let ϕ be some predicate formula, $d : D$ a data variable, and e a data term. We have that $\phi \rightarrow \mathsf{sp}(d := e, \phi)[e/d]$.*

Proof.

$$
\begin{aligned}
& \mathsf{sp}(x := e, \phi)[e/d] \\
\leftrightarrow\ & \{\text{Definition of } \mathsf{sp}\} \\
& (\exists d'{:}D.\ (d = e[d'/d]) \wedge (\phi[d'/d]))[e/d] \\
\leftrightarrow\ & \{\text{Substitution}\} \\
& (\exists d'{:}D.\ (d[e/d] = (e[d'/d])[e/d]) \wedge (\phi[d'/d])[e/d]) \\
\leftrightarrow\ & \{d' \text{ is fresh; } d \text{ does not exist in } (e[d'/d]) \text{ or } (\phi[d'/d])\} \\
& (\exists d'{:}D.\ (e = e[d'/d]) \wedge \phi[d'/d]) \\
\leftarrow\ & \{\text{Choose } d' = d\} \\
& \phi
\end{aligned}
$$

□

Example 5. Consider the PBES and its guards in Example 3. Using Definition 12, we have that

$$
\begin{aligned}
f(X) &= \mathsf{sp}(d_X := \mathsf{arg}(\phi_Y, 1), \mathsf{guard}^1(\phi_Y)) \\
&= \mathsf{sp}(i := 2j, (j \neq 4) \wedge (j < 5)) \\
&= \exists n{:}N.\ ((i = 2j[n/i]) \wedge ((j \neq 4) \wedge (j < 5))[n/i]) \\
&= (i = 2j) \wedge (j \neq 4) \wedge (j < 5) \\
f(Y) &= \mathsf{sp}(d_Y := \mathsf{arg}(\phi_X, 1), \mathsf{guard}^1(\phi_X)) \\
&= \mathsf{sp}(j := i + 2, (i \leq 8)) \\
&= \exists n{:}N.\ ((j = (i + 2)[n/j]) \wedge ((i \leq 8))[n/j]) \\
&= (j = i + 2) \wedge (i \leq 8)
\end{aligned}
$$

and that f is a global invariant for PBES (1).

However, unlike in Hoare logic where variables of the same names are generally of the same types, different equations in a given PBES are not restricted to using the same parameter types, leading to undesirable consequences.

Example 6. Consider the following PBES.

$$
\begin{aligned}
&(\mu X(i : N) = i \leq 5 \wedge Y(\mathsf{even}(i))) \\
&(\mu Y(i : B) = i)
\end{aligned}
$$

We have that $\mathsf{guard}^1(\phi_X) = (i \leq 5)$, and

$$
\begin{aligned}
\mathsf{sp}(i := \mathsf{even}(i), (i \leq 5)) &= \exists d'{:}B.\ ((i = (\mathsf{even}(i))[d'/i]) \wedge (i \leq 5)[d'/i]) \\
&= \exists d'{:}B.\ ((i = \mathsf{even}(d')) \wedge (d' \leq 5))
\end{aligned}
$$

However, d' is of type B. It becomes unclear on how $(d' \leq 5)$ is evaluated.

Still, with appropriate renamings of parameters, PBESs can be transformed into an equivalent equation system with unique parameter names. Thus, for the remainder of this section, we abide by the following remark.

Remark 4. Without loss of generality, we consider PBESs in which all equations contain unique parameter names. More specifically, let $\mathscr{E}$ be some PBES and let $(\sigma X(d_X : D_X) = \phi_X)$, $(\sigma Y(d_Y : D_Y) = \phi_Y)$ be any two equations in $\mathscr{E}$ where $X \neq Y$. Then, $d_X \neq d_Y$.

Observe that if unique parameter names are used, the existential quantifier in the strongest postcondition function sp becomes irrelevant when multiple equations are involved. This can be seen in Example 5. The existential quantifier is still necessary when dealing with self-recursion.

While the definition for strongest postcondition is sufficient in developing invariants as seen in Example 5, we do not wish for free variables to occur in invariants. We therefore place an additional constraint and only consider relevant variables by restricting the free variables of a potential invariant to be within the scope of a given equation's parameters.

Definition 13. *Let $\sigma X(d_X : D_X) = \phi_X$ be an equation in some PBES $\mathscr{E}$. Additionally, let γ be a guard for $\mathsf{PVI}(\phi_X, i) = Y(e)$, where $\sigma Y(d_Y : D_Y) = \phi_Y$ is also in $\mathscr{E}$. A simple formula ψ is a* relevant guard derived invariant *for $\mathsf{PVI}(\phi_X, i)$ iff $\mathsf{FV}(\psi) \subseteq \{d_Y\}$ and*

$$(\mathsf{sp}(d_Y := e, \gamma) \to \psi)$$

We can now show that guards give rise to invariants with the following theorem.

Theorem 1. *Let $\mathscr{E}$ be a closed PBES and assume for all equations $\sigma X(d_X : D_X) = \phi_X$ in $\mathscr{E}$, there exists a guard $\gamma^i_{\phi_X}$ for all PVIs $\mathsf{PVI}(\phi_X, i)$ such that $\gamma^1_{\phi_X}, \ldots, \gamma^{\mathsf{npred}(\phi_X)}_{\phi_X}$ is compositional. Let $\iota^i_{\phi_X}$ be a relevant guard derived invariant for $\mathsf{PVI}(\phi_X, i)$ constructed from $\gamma^i_{\phi_X}$. Define the simple function $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ such that for all equations $\sigma X(d_X : D_X) = \phi$, we have*

$$f(X) = \bigvee_{\mathsf{PV}(\phi_Y, i) = X} \iota^i_{\phi_Y}$$

where $\sigma Y(d_Y : D_Y) = \phi_Y$ is some equation in $\mathscr{E}$. Then f is a global invariant for the PBES $\mathscr{E}$.

Proof. Let $\sigma X(d_X : D_X) = \phi_X$ be an equation in $\mathscr{E}$ and assume the equivalence above holds. By Property 2, it suffices to prove that

$$f(X) \wedge \gamma^i_{\phi_X} \to f(X_i)[e_i / d_{X_i}]$$

for all $\mathsf{PVI}(\phi_X, i) = X_i(e_i)$, where $\gamma^i_{\phi_X}$ is a compositional guard for $\mathsf{PVI}(\phi_X, i)$. Consider the jth PVI in ϕ_X such that $1 \leq j \leq \mathsf{npred}(\phi_X)$ and $\mathsf{PVI}(\phi_X, j) = Y(e)$.

Assume that $\gamma^j_{\phi_X}$ holds. We must show that $f(Y)[e/d_Y]$ also holds. We argue as follows:

$$
\begin{aligned}
& \gamma^j_{\phi_X} \\
\rightarrow\ & \{\text{Lemma 4}\} \\
& \mathsf{sp}(d_Y := e, \gamma^j_{\phi_X})[e/d_Y] \\
\rightarrow\ & \{\text{Definition 13}\} \\
& \iota^j_{\phi_X}[e/d_Y] \\
\rightarrow\ & \{\mathsf{PV}(\phi_X, j) = Y, \text{ thus } \iota^j_{\phi_X} \rightarrow f(Y) \text{ by assumption}\} \\
& f(Y)[e/d_Y]
\end{aligned}
$$

Since $\gamma^j_{\phi_X} \rightarrow f(Y)[e/d_Y]$ for arbitrary equation $\sigma X(d_X : D_X) = \phi_X$ and any j such that $\mathsf{PVI}(\phi_X, j) = Y(e)$, via Property 2 we conclude that f is a global invariant. □

Theorem 1 expresses that the disjunct of all relevant guard derived invariants of a predicate variable is itself a global invariant. It follows from Definition 12 and 13 that stronger guards results in stronger relevant guard derived invariants, which in turn provides more meaningful global invariants. Unfortunately, due to the disjunct in our theorem, weak guards may result in weak, or completely useless, invariants. At the moment, it is not clear whether a stronger invariant construct could be extracted from relevant guard derived invariants. Additionally, from the requirements for relevant guard derived invariants, it is not clear how to exactly compute these predicates given a guard. For example, *true* will always satisfy the condition in Definition 13. Thus, when finding relevant guard derived invariants, we wish to find the strongest predicate which satisfies the requirements. Similar to guards, one option is to develop heuristics to derive such predicates.

One can recompute relevant guard derived invariants to potentially get stronger invariants. Given a PBES $\mathscr{E}$, one can add invariants derived from Theorem 1 to obtain an updated PBES $\mathscr{E}'$. We can then repeat the process for $\mathscr{E}'$: add invariants using Theorem 1 on $\mathscr{E}'$ to obtain the PBES $\mathscr{E}''$.

Example 7. Consider the following PBES taken from Example 3.

$$
\begin{aligned}
&(\mu X(i : N) = (i \leq 8) \wedge Y(i+2)) \\
&(\mu Y(j : N) = (j = 4) \vee (j < 5 \wedge X(2j)))
\end{aligned}
\tag{1}
$$

It follows from Theorem 1 that we have the following invariants:

$$
\begin{aligned}
f(X) &= (i \neq 8) \wedge (i < 10) \\
f(Y) &= (j \leq 10)
\end{aligned}
$$

Using the $\mathsf{Apply}(f, \mathcal{E})$ function from Orzan et al. [6], we get the following PBES:

$$
\begin{aligned}
&(\mu X(i : N) = ((i \neq 8) \wedge (i < 10)) \wedge (i \leq 8) \wedge Y(i+2)) \\
&(\mu Y(j : N) = (j \leq 10) \wedge (j = 4) \vee (j < 5 \wedge X(2j)))
\end{aligned}
$$

Now applying Theorem 1 again yields slightly different invariants:

$$f(X) = (i \neq 8) \wedge (i < 10)$$
$$f(Y) = (j < 10)$$

Notice that the invariant for Y has been slightly strengthened from $j \leq 10$ to $j < 10$.

Although repeated application of Theorem 1 seems promising, it is not guaranteed to terminate.

Example 8. Consider the following PBES:

$$\mu X(i : N) = (i \geq 5) \wedge X(i + 1)$$

We then have the invariant $f(X) = (i \geq 6)$ and the updated PBES:

$$\mu X(i : N) = (i \geq 6) \wedge X(i + 1)$$

Observe that repeated application of Theorem 1 will not terminate for this PBES.

Still, more research is needed to realize the potential of this method.

4.1 Application

In this section, we provide an example on how Theorem 1 may be applied. Specifically, we study the system of a one-place buffer and analyze the constant input stream property as seen in [1].

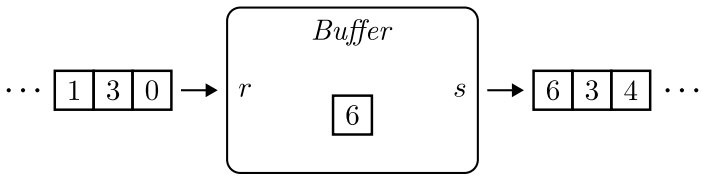


Fig. 1. One-place buffer.

Our one-place buffer reads natural numbers one-by-one from an infinite stream and outputs a stream of data. The following is a μCRL process *Buffer* which contains action r to represent reading a natural number and action s for outputting a natural number.

$$\mathbf{proc}\mathit{Buffer}(b : B, n : N) = \sum_{m:N} r(m) \cdot \mathit{Buffer}(\mathit{false}, m) \lhd b \rhd s(n) \cdot \mathit{Buffer}(\mathit{true}, n)$$

where the initial state is $\mathit{Buffer}(\mathit{true}, n)$ for any $n : N$.

We now analyze the constant input stream property on the buffer. This property states that provided an input stream k^ω where $k : N$, the output must be k^ω. I.e. the buffer does not modify the contents of a constant input. This property can be expressed and translated into the following PBES:

$$\nu X(b : B, n, k : N) = \forall l : N.\ (\forall m : N.\ (b \wedge (m = l) \implies ((l = k) \implies X(\mathit{false}, m, k)))) \wedge \\ (\neg b \wedge (l = n) \implies ((l = k) \wedge X(\mathit{true}, n, k)))$$

Using the guard function in Definition 10, we have the following guards for both PVIs in equation X:

$$\mathsf{guard}^1(\phi_X) = (b \wedge (m = l) \wedge (l = k))$$
$$\mathsf{guard}^2(\phi_X) = (\neg b \wedge (l = n) \wedge (l = k))$$

Observe that we may derive the following strongest postconditions:

$$\mathsf{sp}(\langle b := \mathit{false}, n := m, k := k\rangle, \mathsf{guard}^1(\phi_X)) = \\ \exists b' : B, n', k' : N.\ b = \mathit{false} \wedge (n = m) \wedge (k = k') \wedge (b' \wedge (m = l) \wedge (l = k'))$$

$$\mathsf{sp}(\langle b := \mathit{true}, n := n, k := k\rangle, \mathsf{guard}^2(\phi_X)) = \\ \exists b' : B, n', k' : N.\ b = \mathit{true} \wedge (n = n') \wedge (k = k') \wedge (\neg b' \wedge (l = n') \wedge (l = k'))$$

Now observe that we may derive the following guard derived invariants:

$$\gamma^1_{\phi_X} = (b = \mathit{false}) \wedge (n = k)$$
$$\gamma^2_{\phi_X} = (b = \mathit{true}) \wedge (n = k)$$

One may verify that indeed $\mathsf{FV}(\gamma^1_{\phi_X}) \subseteq \{b, n, k\}$, $\mathsf{FV}(\gamma^2_{\phi_X}) \subseteq \{b, n, k\}$, and

$$\mathsf{sp}(\langle b := \mathit{false}, n := m, k := k\rangle, \mathsf{guard}^1(\phi_X)) \to \gamma^1_{\phi_X}$$
$$\mathsf{sp}(\langle b := \mathit{true}, n := n, k := k\rangle, \mathsf{guard}^2(\phi_X)) \to \gamma^2_{\phi_X}$$

By Theorem 1, we have that a function f defined in the following way is a global invariant:

$$\begin{aligned} f(X) &= ((b = \mathit{false}) \wedge (n = k)) \vee ((b = \mathit{true}) \wedge (n = k)) \\ &= (n = k) \end{aligned}$$

We may then add our invariant into the PBES via the Apply function from [6] and use Corollary 4.12 of [1] arrive at a simplified equation system:

$$\nu X(b : B, n, k : N) = (n = k) \wedge (X(\mathit{false}, n, k) \vee X(\mathit{true}, k, k))$$

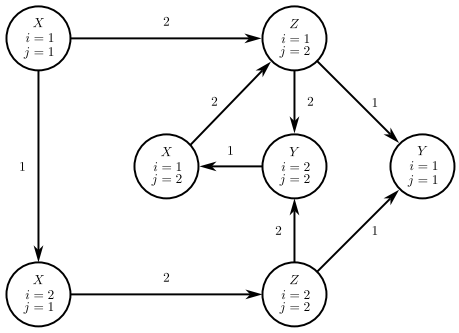
5 Control Flow Graphs

Guards have been generally used within various PBES static analysis techniques to combat the state explosion problem. Their use can be seen in the technique by Orzan et al. where methods have been developed to detect constant parameters and eliminate redundant parameters, leading to an increase of performance when applying PBES instantiation [4]. Similarly, Keiren et al. used guards to generate constraints on PBES equations, enabling the construction of control flow graphs (CFGs) which allow for detection of live parameters and elimination of irrelevant quantifiers [5]. These CFGs provide a graph structure which illustrates the possible values certain parameters in PBES equations will have when attempting to solve a certain PBES instantiation. Thus, we wish to investigate the relationship between CFGs and invariants to explore whether we are able to express the relevant values of parameters in the form of PBES invariants. Our focus of this section is to bridge the gap between CFGs and global invariants by showing how states in a CFG can be expressed in an invariant.

Example 9. As an example to express the power of CFGs, consider the following constructed example taken from [5].

$$\begin{aligned} \nu X(i,j,k,l:N) &= (i \neq 1 \vee j \neq 1 \vee X(2,j,k,l+1)) \\ &\quad \wedge (\forall m{:}N.\ Z(i,2,m+k,k)) \\ \mu Y(i,j,k,l:N) &= k = 1 \vee (i = 2 \wedge X(1,j,k,l)) \\ \nu Z(i,j,k,l:N) &= (k < 10 \vee j = 2) \wedge (j \neq 2 \vee Y(1,1,l,1)) \wedge Y(2,2,1,l) \end{aligned} \tag{2}$$

Due to the universal quantifier in ϕ_X, attempting to solve for $X(1,1,1,1)$ via instantiation into a BES does not terminate. Using techniques developed by Keiren et al., one obtains the following graph which illustrates the possible values for parameters i and j:



Since this graph shows the possible values of parameters, it is worth investigating whether we can extract invariants from these graphs to constrain the values of parameters.

Before proceeding, we first provide some preliminaries in relation to control flow graphs. For a more comprehensive and complete description on control

flow graphs and their use, we refer readers to the paper *Liveness Analysis for Parameterised Boolean Equation Systems* [5].

The construction of a control flow graph (CFG) is rather involved; CFGs rely on control flow parameters (CFPs) which are constructed from unicity constraints. Essentially, unicity constraints are a collection of guards that characterize the values of certain parameters of a PVI. Collect enough contextual information on parameters, we may realize the values of certain parameters throughout the entire equation system. In other words, if a parameter has sufficiently many constraints, we know what the value of the parameter will be at all times. Parameters of which we can determine all values given an initial instantiation are *control flow parameters (CFPs)*. Using CFPs, we can then construct a *control flow graph (CFG)* to illustrate the values of CFPs and the relationship between relevant PVIs for a predicate variable.

We now provide the definition of unicity constraints.

Definition 14. *Let* $s, t : (\mathcal{P} \times \mathbb{N} \times \mathbb{N}) \to D$ *and* $c : (\mathcal{P} \times \mathbb{N} \times \mathbb{N}) \to \mathbb{N}$ *be partial functions, where* D *is the union of terms not containing data variables. The triple* (s, t, c) *is a* unicity constraint *for some PBES* $\mathscr{E}$ *if for all* $X \in \mathsf{bnd}(\mathscr{E})$, $i, j, k \in \mathbb{N}$, *and* $e \in D$*:*

- (source) *if* $s(X, i, j) = e$ *then* $\phi_X \leftrightarrow \phi_X[i \mapsto (\mathbf{d}_j = e \wedge \mathsf{PVI}(\phi_X, i))]$
- (target) *if* $t(X, i, j) = e$ *then* $\phi_X \leftrightarrow \phi_X[i \mapsto (\mathsf{arg}_j(\phi_X, i) = e \wedge \mathsf{PVI}(\phi_X, i))]$
- (copy) *if* $c(X, i, j) = k$ *then* $\phi_X \leftrightarrow \phi_X[i \mapsto (\mathsf{arg}_k(\phi_X, i) = \mathbf{d}_j \wedge \mathsf{PVI}(\phi_X, i))]$

In other words, if $s(X, i, j)$ is defined, the formal parameter $\mathbf{d}_j$ must have the value $s(X, i, j)$ whenever $\mathsf{PVI}(\phi_X, i)$ needs to be considered in ϕ_X. Similarly, if $t(X, i, j)$ is defined, the value of the jth argument of $\mathsf{PVI}(\phi_X, i)$ has the fixed value $t(X, i, j)$. And if $c(X, i, j) = k$, the parameter $\mathbf{d}_j$ is copied to position k of $\mathsf{PVI}(\phi_X, i)$'s arguments.

We may write $s(X, i, j) = \bot$ if $s(X, i, j)$ is undefined. From hereon, assume $\mathscr{E}$ is some arbitrary PBES where $(\mathsf{source}, \mathsf{target}, \mathsf{copy})$ is its associated unicity constraint.

Remark 5. In this section, we also operate under Assumption 6.13 of [5]: if $\mathsf{source}(X, i, j) = e$ and $\mathsf{copy}(X, i, j) = k$ are defined, then $\mathsf{target}(X, i, k) = e$.

Example 10. Consider PBES (2) in Example 9. We can derive the following unicity constraints:

$$\begin{array}{rrr}
\mathsf{source}(X,1,1) = 1 & \mathsf{source}(Y,1,1) = 2 & \mathsf{source}(Z,1,2) = 2 \\
\mathsf{source}(X,1,2) = 1 & \mathsf{target}(Y,1,1) = 1 & \mathsf{target}(Z,1,1) = 1 \\
\mathsf{target}(X,1,1) = 2 & \mathsf{copy}(Y,1,2) = 2 & \mathsf{target}(Z,1,2) = 1 \\
\mathsf{target}(X,1,2) = 1 & \mathsf{copy}(Y,1,3) = 3 & \mathsf{target}(Z,1,4) = 1 \\
\mathsf{copy}(X,1,2) = 2 & \mathsf{copy}(Y,1,4) = 4 & \mathsf{copy}(Z,1,4) = 3 \\
\mathsf{copy}(X,1,3) = 3 & & \mathsf{target}(Z,2,1) = 2 \\
\mathsf{target}(X,2,2) = 2 & & \mathsf{target}(Z,2,2) = 2 \\
\mathsf{copy}(X,2,1) = 1 & & \mathsf{target}(Z,2,3) = 1 \\
\mathsf{copy}(X,2,3) = 4 & & \mathsf{copy}(Z,2,4) = 4
\end{array}$$

Before covering control flow parameters, we first define *local control flow parameters (LCFPs)* and *global control flow parameters (GCFPs)*. Intuitively, LCFPs express the set of parameters for which we know their values under self-recursion.

Definition 15. *A parameter* $\mathbf{d}_n \in \mathsf{par}(X)$ *is a* local control flow parameter (LCFP) *if for all* i *such that* $\mathsf{pv}(\phi_X, i) = X$*, either* $\mathsf{source}(X, i, n)$ *and* $\mathsf{target}(X, i, n)$ *are defined, or* $\mathsf{copy}(X, i, n) = n$*.*

Example 11. Parameters i^X, j^X, k^X are LCFPs in equation X of PBES (2). The parameter l^X in equation X is not a LCFP since there are no unicity constraints for l^X. Observe that equation X is the only equation that contains self-recursion and thus, parameters i, j, k, l of equations Y and Z are LCFPs.

GCFPs refines the set of LCFPs by removing parameters to ensure that we know the values of parameters even when they are present in any PBES equation.

Definition 16. *A parameter* $\mathbf{d}_n \in \mathsf{par}(X)$ *is a* global control flow parameter (GCFP) *if* $\mathbf{d}_n$ *is a LCFP, and for all* $Y \in \mathsf{bnd}(\mathscr{E}) \setminus \{X\}$ *and all* i *such that* $\mathsf{pv}(\phi_Y, i) = X$*, either* $\mathsf{target}(Y, i, n)$ *is defined, or* $\mathsf{copy}(\phi_Y, i, m) = n$ *for some GCFP* $\mathbf{d}_m \in \mathsf{par}(Y)$*.*

Example 12. Only i, j are GCFPs in all equations of PBES (2). Since parameter l^X is not a LCFP, the parameter l^Z of equation Z cannot be a GCFP. Therefore, the LCFP k^X is not a GCFP since l^Z is not a GCFP.

To obtain a set of CFPs, we construct an equivalence relation that also captures the set of GCFPs that are dependent on each other through copying. The relation $\sim$ on GCFPs expresses that they are *related*.

Definition 17. *Let* $\mathbf{d}_n^{\mathbf{X}}$ *and* $\mathbf{d}_m^{\mathbf{Y}}$ *be GCFPs. We say* $\mathbf{d}_n^{\mathbf{X}}$ *and* $\mathbf{d}_m^{\mathbf{Y}}$ *are* related*, denoted* $\mathbf{d}_n^{\mathbf{X}} \sim \mathbf{d}_m^{\mathbf{Y}}$*, if* $n = \mathsf{copy}(Y, i, m)$ *for some* i*.*

We now characterize the set of GCFPs which are not mutually dependent on each other. The parameters within this set are *control flow parameters (CFPs)*.

Definition 18. *Let* $\mathcal{C}$ *be a set of GCFPs, and let* $\sim^*$ *denote the reflexive, symmetric and transitive closure of* $\sim$ *on* $\mathcal{C}$*. Assume* $\approx \subseteq \mathcal{C} \times \mathcal{C}$ *is an equivalence relation that satisfies* $\sim^* \subseteq \approx$*. Then the pair* $\langle \mathcal{C}, \approx \rangle$ *is a* control structure *if for all* $X \in \mathsf{bnd}(\mathscr{E})$ *and all* $d, d' \in \mathcal{C} \cap \mathsf{par}(X)$*, if* $d \approx d'$*, then* $d = d'$*.*

Definition 19. *A formal parameter* c *is a* control flow parameter *(CFP) if there is a control structure* $\langle \mathcal{C}, \approx \rangle$ *such that* $c \in \mathcal{C}$*.*

Example 13. $\langle \{i^X, j^X, i^Y, j^Y, i^Z, j^Z\}, \approx \rangle$ is a control structure for PBES (2) from Example 9. Formal parameters i and j are CFPs for each equation.

Using a control structure $\langle \mathcal{C}, \approx \rangle$ allows all equations in a given PBES to have the same set of CFPs. Even if equations may not contain the same set of CFPs, without loss of generality, one could add CFPs that do not appear in an equation to achieve identical sets of CFPs in all equations. Additionally, we partition the parameters of equations such that the set of CFPs appear first in the parameter list of equations. We operate under the assumption as seen in [5]:

Remark 6. Assume that the set of CFPs is the same for all equations in a PBES $\mathscr{E}$, i.e. for all $X, Y \in \mathsf{bnd}(\mathscr{E})$, $d^X \in \mathsf{par}(X)$ is a CFP iff $d^Y \in \mathsf{par}(Y)$ is a CFP, and $d^X \approx d^Y$. Moreover, assume all equations in $\mathscr{E}$ are of the form $\sigma X(\mathbf{c} : \mathbf{C}, \mathbf{d_X} : \mathbf{D_X}) = \phi_X$, where $\mathbf{c}$ is a vector of CFPs and $\mathbf{d_X}$ is a vector of data parameters for the PBES equation X.

Given a set of CFPs, we now define *control flow graphs (CFGs)*. Vertices in the graph represents values of CFPs and edges represent dependencies on PVIs. Assume we wish to find the solution of $X(\mathbf{e}, \mathbf{f})$ where $\sigma X(\mathbf{c} : \mathbf{C}, \mathbf{d}_X : \mathbf{D}_X) = \phi_X$ is an equation in some PBES $\mathscr{E}$. Then the initial instantiated value of CFP $\mathbf{c}_k$ is defined as $\mathsf{init}(\mathbf{c}_k) = \mathbf{e}_k$. The set of values of CFPs are restricted by the set $\mathsf{values}(\mathbf{c}_k)$, defined as

$$\{\mathsf{init}(\mathbf{c}_k)\} \cup \bigcup_{i \in \mathbb{N}, X \in \mathsf{bnd}(\mathscr{E})} \{v \in D \mid \mathsf{source}(X, i, k) = v \vee \mathsf{target}(X, i, k) = v\}$$

For later proofs, we require that this set to be uniquely representable. For example, if $[\![5]\!] = [\![2 + 3]\!]$, we do not want both $5 \in D$ and $2 + 3 \in D$ for some uniquely representable set D. A set of data terms D not containing data variables is *uniquely representable* if for any $a, b \in D$ where $a \neq b$, $[\![a]\!] \neq [\![b]\!]$. We assume the set $\mathsf{values}(\mathbf{c})$ is uniquely representable.

Definition 20. *A* control flow graph *(CFG) of PBES $\mathscr{E}$ is a directed graph $(V, \rightarrow)$ where*

- $V \subseteq \mathsf{bnd}(\mathscr{E}) \times \mathsf{values}(\mathbf{c})$
- $\rightarrow \subseteq V \times \mathbb{N} \times V$ *is the least relation such that whenever* $(X, \mathbf{v}) \xrightarrow{i} (\mathsf{pv}(\phi_X, i), \mathbf{w})$ *then for every k, either*
  - $\mathsf{source}(X, i, k) = \mathbf{v}_k$ *and* $\mathsf{target}(X, i, k) = \mathbf{w}_k$*, or*
  - $\mathsf{source}(X, i, k) = \bot$*,* $\mathsf{copy}(X, i, k) = k$ *and* $\mathbf{v}_k = \mathbf{w}_k$*, or*
  - $\mathsf{source}(X, i, k) = \bot$*, and* $\mathsf{target}(X, i, k) = \mathbf{w}_k$

Example 14. Taking the equation from Example 9, we can construct the CFG graph in Figure 2 for when we wish to find the solution for $X(1, 1, k, l)$, where k, l can be any value of type N. Here, we see that the solution for $X(1, 1, \_, \_)$ depends on the solution of $Z(1, 2, \_, \_)$ and $X(2, 1, \_, \_)$ where $(\_)$ is some value of type N.

Control flow graphs characterize the possible values CFPs are able to hold within various equations given an initial instantiation. In other words, given an equation X in a PBES $\mathscr{E}$ and corresponding CFG $(V, \rightarrow)$, the values of X's CFPs must be $\mathbf{c}$, where $(X, \mathbf{c}) \in V$ is some vertex in the graph. In Example 14, we see that if we wish to find $X(1, 1, 1, 1)$ for equation $\nu X(i, j, k, l : N)$ in PBES (2), the values of X's CFPs (i, j) must either be $(1, 1)$, $(2, 1)$, or $(1, 2)$. Similarly, we can also characterize the potential values of formal parameters by means of invariants.

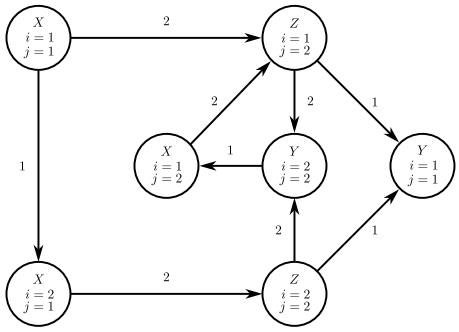


Fig. 2. Control flow graph for PBES (2).

5.1 Compositionality

Before proceeding further, we address a problem we encountered during our research. So far, we have covered the definitions of CFGs as discussed in the paper by Keiren et al. [5]. However, the definitions of CFGs are incomplete; a flaw is present with the current definitions of CFGs that allow for illogical CFGs to be constructed. As we have seen in Section 4, applying multiple syntactic replacements does not always hold under $\leftrightarrow$. The same problem comes up with CFGs since there are no compositionality requirements on unicity constraints.

Example 15. Consider the following PBES:

$$\mu X(i, j : N) = Y(0, 0) \vee Y(0, 0)$$
$$\mu Y(i, j : N) = X(0, 0) \vee X(0, 0)$$

and let $(\mathsf{source}, \mathsf{target}, \mathsf{copy})$ be a unicity constraint for this PBES defined as follows:

$$\begin{aligned} &\mathsf{target}(X, 1, 1) = 5 && \mathsf{target}(Y, 1, 1) = 2 \\ &\mathsf{target}(X, 1, 2) = 10 && \mathsf{target}(Y, 1, 2) = 4 \\ &\mathsf{target}(X, 2, 1) = 15 && \mathsf{target}(Y, 2, 1) = 6 \\ &\mathsf{target}(X, 2, 2) = 20 && \mathsf{target}(Y, 2, 2) = 8 \end{aligned}$$

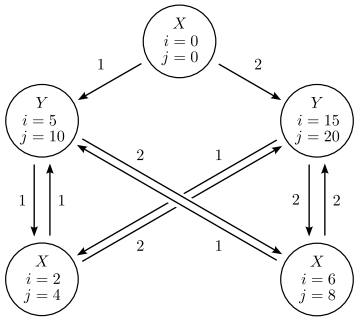
Observe that a unicity constraint defined this way is valid. For instance, consider the constraint $\mathsf{target}(X, 1, 1) = 5$. Then we have that

$$\begin{aligned} (Y(0, 0) \vee Y(0, 0)) &\leftrightarrow (Y(0, 0) \vee Y(0, 0))[1 \mapsto (0 = 5 \wedge Y(0, 0))] \\ &\leftrightarrow (0 = 5 \wedge Y(0, 0)) \vee Y(0, 0) \\ &\leftrightarrow Y(0, 0) \end{aligned}$$

In fact, we even have that

$$\phi_X \leftrightarrow \phi_X[1 \mapsto \mathit{false} \wedge Y(0, 0)]$$

If we defined our unicity constraints as above, we may show that i, j are CFPs and derive a graph such as:



Since the goal of CFGs is to show the relevant values of CFPs, we would expect our CFG to only contain the states $(X, \langle 0, 0\rangle)$ and $(Y, \langle 0, 0\rangle)$ when trying to find the solution for $X(0, 0)$. However, this is not the case. We instead want our constraint to be compositional, e.g.

$$\phi_X \leftrightarrow \phi_X[1 \mapsto (0 = 5 \wedge Y(0, 0))][2 \mapsto (0 = 10 \wedge Y(0, 0))]$$

Note that this statement is not valid since:

$$(Y(0, 0) \vee Y(0, 0)) \not\leftrightarrow ((0 = 5) \wedge Y(0, 0)) \vee ((0 = 10) \wedge Y(0, 0)) \leftrightarrow \mathit{false}$$

Therefore, we need a notion of compositionality in our constraints to ensure that we have proper CFGs.

This compositionality problem in CFGs also impacts soundness of the liveness analysis presented in [5]. In order to remedy this problem, we require that our unicity constraints be compositional. Thus, we will be utilizing the following remark:

Remark 7. We assume that the combination of unicity constraints are compositional. I.e. let (s, t, c) be unicity constraints for some PBES $\mathscr{E}$. For every $X \in \mathsf{bnd}(\mathscr{E})$, let S_X, T_X, C_X be sets of indices defined in the following way:

$$\begin{aligned} i \in S_X &\text{ if } s(X, i, i') = e_i^s \\ i \in T_X &\text{ if } t(X, i, i') = e_i^t \\ i \in C_X &\text{ if } c(X, i, i') = k_i \end{aligned}$$

We then assume that for all equations $\sigma X(\mathbf{d}_X : \mathbf{D}_X) = \phi_X$ in $\mathscr{E}$,

$$\begin{aligned} \phi_X \leftrightarrow ((\phi_X[i \mapsto (\mathbf{d}_{i'} = e_i^s \wedge \mathsf{PVI}(\phi_X, i))]_{i \in S_X}) \\ [i \mapsto (\mathsf{arg}_{i'}(\phi_X, i) = e_i^t \wedge \mathsf{PVI}(\phi_X, i))]_{i \in T_X}) \\ [i \mapsto (\mathsf{arg}_{k_i}(\phi_X, i) = \mathbf{c}_{i'} \wedge \mathsf{PVI}(\phi_X, i))]_{i \in C_X} \end{aligned}$$

To assist in our proofs, we now define a function $\mathsf{Collect} : \mathsf{Pred} \times \mathbb{N} \to \mathsf{Pred}$ which combines all unicity constraints of a particular PVI into one predicate formula.

Definition 21. *Let $\mathscr{E}$ be a PBES with a corresponding CFG $(V, \to)$. For an equation $\sigma X(\mathbf{c}, \mathbf{d}_X) = \phi_X$, and $1 \leq i \leq \mathsf{npred}(\phi_X)$, we define $\mathsf{Collect}(\phi_X, i)$ to be a predicate formula collecting all unicity constraints defined on $\mathsf{PVI}(\phi_X, i) = Y(\mathbf{e}, \mathbf{f})$.*

$$\mathsf{Collect}(\phi_X, i) = \bigwedge_{\mathbf{c}_j \in \mathbf{c}} \alpha_j^{(\phi_X, i)} \wedge \beta_j^{(\phi_X, i)} \wedge \gamma_j^{(\phi_X, i)}$$

where

$$\alpha_j^{(\phi_X, i)} = \begin{cases} (\mathbf{c}_j = \mathsf{source}(\phi_X, i, j)) & \text{if } \mathsf{source}(\phi_X, i, j) \text{ is defined} \\ \mathit{true} & \text{otherwise} \end{cases}$$

$$\beta_j^{(\phi_X, i)} = \begin{cases} (\mathbf{e}_j = \mathsf{target}(\phi_X, i, j)) & \text{if } \mathsf{target}(\phi_X, i, j) \text{ is defined} \\ \mathit{true} & \text{otherwise} \end{cases}$$

$$\gamma_j^{(\phi_X, i)} = \begin{cases} (\mathbf{e}_j = \mathbf{c}_k) & \text{if } \mathsf{copy}(\phi_X, i, j) = k \text{ is defined} \\ \mathit{true} & \text{otherwise} \end{cases}$$

The following two lemmas allow us to utilize Property 2 when proving our theorem on CFG invariants.

Lemma 5. *Let $\mathscr{E}$ be a PBES, $\sigma X(\mathbf{c}, \mathbf{d}_Y)$ an equation in $\mathscr{E}$, and $(V, \to)$ a CFG for $\mathscr{E}$. Then $\mathsf{Collect}(\phi_X, i)$ is a guard for $\mathsf{PVI}(\phi_X, i)$.*

Proof. Follows from the definition of $\mathsf{Collect}(\phi_X, i)$, unicity constraints, and Remark 7. □

Lemma 6. *Let $\mathscr{E}$ be a PBES, $\sigma X(\mathbf{c}, \mathbf{d}_Y)$ an equation in $\mathscr{E}$, and $(V, \to)$ a CFG for $\mathscr{E}$. Then the collection of guards $\mathsf{Collect}(\phi_X, 1), \ldots, \mathsf{Collect}(\phi_X, \mathsf{npred}(\phi_X, i))$ are compositional for ϕ_X.*

Proof. Since $\mathsf{Collect}$ is constructed from unicity constraints, and by Remark 7 the collection of unicity constraints are compositional, it follows that the collection $\mathsf{Collect}(\phi_X, 1), \ldots, \mathsf{Collect}(\phi_X, \mathsf{npred}(\phi_X, i))$ is compositional for ϕ_X. □

5.2 CFG Invariants

While we have fixed the problem with compositionality on unicity constraints, we must also notice that a CFG may not be minimal; there could exist nodes within the CFG which may not be reachable from an initial state. Consider the following example.

Example 16. Addition of the state $(X, \langle 2, 2 \rangle)$ with edge $(X, \langle 2, 2 \rangle) \to (Z, \langle 2, 2 \rangle)$ in the CFG of Example 14 would also be a valid CFG for PBES (2).

$$\begin{aligned} \nu X(i, j, k, l : N) &= (i \neq 1 \vee j \neq 1 \vee X(2, j, k, l + 1)) \\ &\qquad \wedge (\forall m{:}N.\ Z(i, 2, m + k, k)) \\ \mu Y(i, j, k, l : N) &= k = 1 \vee (i = 2 \wedge X(1, j, k, l)) \\ \nu Z(i, j, k, l : N) &= (k < 10 \vee j = 2) \wedge (j \neq 2 \vee Y(1, 1, l, 1)) \wedge Y(2, 2, 1, l) \end{aligned} \tag{2}$$

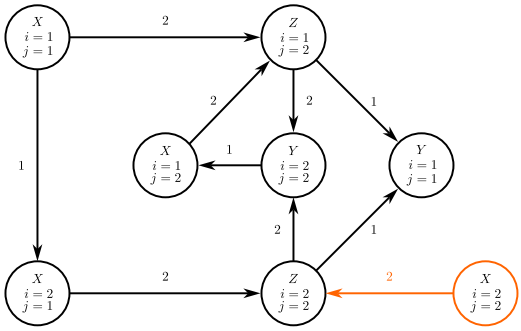


Fig. 3. A valid control flow graph for PBES (2).

When solving for $X(1,1,1,1)$, it would be unnecessary to consider the values $i = 2$ and $j = 2$ for X since we never will reach a point where we depend on $(X, \langle 2, 2\rangle)$.

Example 17. Consider the CFG in Figure 2. If one is only interested in finding the solution for $Z(1,2,k,l)$, where k, l are any values of type N, then the following CFG provides sufficient information for the relevant values the CFPs i and j will hold.

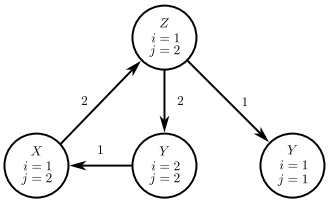


Fig. 4. Another valid control flow graph for PBES (2).

As can be seen, if one is interested in finding the solution for some predicate variable instantiation, we only need to consider states that are reachable from a certain vertex in the CFG to determine the set of values CFPs will ever hold. Therefore, to deal with irrelevant states, we provide a notion of reachability within a CFG graph.

Definition 22. *Let $\mathscr{E}$ be a PBES and $(V, \to)$ be its corresponding CFG. Let $(X, \mathbf{v}) \in V$ be some state in the CFG. Then, $R \subseteq V$ is the reachable set of states originating from (X, v). We define R as the least set satisfying:*

$$(X, \mathbf{v}) \in R, \text{ and}$$
$$(Y, \mathbf{w}) \in R \text{ if } (Y', \mathbf{w}') \to (Y, \mathbf{w}) \text{ for some } (Y', \mathbf{w}') \in R$$

In our invariant proof, we use the notion of an 'initial state' in a CFG. While Definition 20 does not have the notion of an initial state, we consider an initial

state $(X, \mathbf{v})$ to be one where we wish to find the solution for $X(\mathbf{v}, \mathbf{w})$ for some PBES $\mathscr{E}$.

Assume we are interested in finding the solution for $X(\mathbf{e}, \mathbf{f})$, where $\sigma X(\mathbf{c} : \mathbf{C}, \mathbf{d}_X : \mathbf{D}_X)$ is an equation in PBES $\mathscr{E}$. We now show that given a CFG graph $(V, \to)$ of $\mathscr{E}$, the values of the CFPs $\mathbf{c}$ of some equation $\sigma Y(\mathbf{c} : \mathbf{C}, \mathbf{d}_X : \mathbf{D}_X)$ in $\mathscr{E}$ must be equal to $\mathbf{v}$, where $(Y, \mathbf{v})$ is some state in V that is reachable from $(X, \mathbf{e})$. The following theorem formally states this idea.

Theorem 2. *Let $\mathscr{E}$ be a PBES and let the corresponding CFG graph be $(V, \to)$ with initial state $(X, \mathbf{v}) \in V$. Let $R \subseteq V$ be the set of vertices reachable from $(X, \mathbf{v})$, and let W be the set of predicate variables whose equation is reachable from $(X, \mathbf{v})$, defined as $W = \{X' \mid (X', \mathbf{v}') \in R\}$. Let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple function such that for all $Y \in W$ we have*

$$f(Y) = \bigvee_{(Y, \mathbf{v}') \in R} (\mathbf{c} = \mathbf{v}')$$

and for all $Y \notin W$, $f(Y) = \mathsf{false}$. Then f is a global invariant for PBES $\mathscr{E}$.

Proof. Let $\sigma X(\mathbf{c}, \mathbf{d}_X) = \phi_X$ be an equation in $\mathcal{E}$. From Lemma 6, we have that the collection of guards $\mathsf{Collect}(\phi_X, 1), \ldots, \mathsf{Collect}(\phi_X, \mathsf{npred}(\phi_X))$ is compositional for ϕ_X. To prove that f is a global invariant, by Property 2 it suffices to show for every $\mathsf{PVI}(\phi_Y, i) = Z(\mathbf{e}, \mathbf{f})$ that $f(Y) \wedge \mathsf{Collect}(\phi_Y, i) \to f(Z)[\mathbf{e}/\mathbf{c}][\mathbf{f}/\mathbf{d}_Z]$. Since $\mathbf{d}_Z$ does not occur in $f(Z)$, it suffices to prove

$$f(Y) \wedge \mathsf{Collect}(\phi_Y, i) \to f(Z)[\mathbf{e}/\mathbf{c}]$$

Assume $f(Y) \wedge \mathsf{Collect}(\phi_Y, i)$ holds. Then it must be the case that there exists a state $(Y, \mathbf{w}) \in R$ such that $(\mathbf{c} = \mathbf{w})$ holds. We distinguish two cases on $\mathsf{PVI}(\phi_Y, i) = Z(\mathbf{e}, \mathbf{f})$: $Y = Z$ and $Y \neq Z$.

– *Case:* $Y = Z$. Since $\mathbf{c}$ is a vector of CFPs, $\mathbf{c}$ must also be a vector of LCFPs. Because $\mathsf{pv}(\phi_Y, i) = Y$ and by definition of LCFPs, it must be the case that for all $\mathbf{c}_n \in \mathbf{c}$, either $\mathsf{source}(Y, i, n) = e_s^n$ and $\mathsf{target}(Y, i, n) = e_t^n$ for some $e_s^n, e_t^n \in \mathsf{values}(\mathbf{c}_n)$, or $\mathsf{copy}(Y, i, n) = n$. Then in $\mathsf{Collect}(\phi_Y, i)$, we have that either $(\mathbf{c}_n = e_s^n) \wedge (\mathbf{e}_n = e_t^n)$ holds or $(\mathbf{e}_n = \mathbf{c}_n)$. We investigate these two cases:
  - $(\mathbf{c}_n = e_s^n) \wedge (\mathbf{e}_n = e_t^n)$. Since $(\mathbf{c} = \mathbf{w})$, it must be the case that $\mathbf{c}_n = e_s^n = \mathbf{w}_n$. I.e. $\mathsf{source}(Y, i, n) = e_s^n = \mathbf{w}_n$.
  - $(\mathbf{e}_n = \mathbf{c}_n)$. Since $(\mathbf{c} = \mathbf{w})$ and $(\mathbf{e}_n = \mathbf{c}_n)$, we have that $\mathbf{e}_n = \mathbf{c}_n = \mathbf{w}_n$.

  Let $\mathbf{u}$ be a vector of values such that

$$\mathbf{u}_n = \begin{cases} e_t^n & \text{if } \mathsf{target}(Y, i, n) = e_t^n \text{ is defined} \\ \mathbf{w}_n & \text{if } \mathsf{copy}(Y, i, n) = n \text{ is defined} \end{cases}$$

  Observe that it must be the case that $(\mathbf{c} = \mathbf{u})[\mathbf{e}/\mathbf{c}]$ since for all n, $(\mathbf{e}_n = e_t^n)$ whenever $\mathsf{target}(Y, i, n) = e_t^n$, and $(\mathbf{e}_n = \mathbf{c}_n = \mathbf{w}_n)$ whenever $\mathsf{copy}(Y, i, n) = n$.

  Since $(Y, \mathbf{w}) \in R$, and for every n, either $\mathsf{source}(Y, i, n) = \mathbf{w}_n$ and $\mathsf{target}(Y, i, n) = e_t^n = \mathbf{u}_n$, or $\mathsf{copy}(Y, i, n) = n$ and $\mathbf{w}_n = \mathbf{u}_n$, there must be a transition in the CFG graph such that $(Y, \mathbf{w}) \to (Y, \mathbf{u})$. Then it must be the case that $(\mathbf{c} = \mathbf{u})$ is in $f(Y)$. Because $(\mathbf{c} = \mathbf{u})[\mathbf{e}/\mathbf{c}]$ holds, also $f(Y)[\mathbf{e}/\mathbf{c}]$.

– *Case:* $Y \neq Z$. Since $\mathbf{c}$ is a vector of CFPs, it must also be a vector of GCFPs. We have that $Y \neq Z$ and thus it must be the case that $\mathsf{target}(Y, i, n) = e_t^n$ for some $e_t^n \in \mathsf{values}(\mathbf{c}_n)$, or $\mathsf{copy}(Y, i, m) = n$ for all $\mathbf{c}_n \in \mathbf{c}$ and some $\mathbf{c}_m \in \mathbf{c}$. First, we investigate these two cases:
  - $\mathsf{target}(Y, i, n) = e_t^n$. Since $\mathsf{Collect}(\phi_Y, i)$ holds, it must be the case that $\mathbf{e}_n = e_t^n$.
  - $\mathsf{copy}(Y, i, m) = n$. Since $(V, \rightarrow)$ is defined using a control structure, if $\mathsf{copy}(Y, i, m) = n$ we have that $\mathbf{c}_n \sim \mathbf{c}_m$, and by Definition 18, we have that $n = m$ and $\mathsf{copy}(Y, i, n) = n$. Because $\mathsf{Collect}(\phi_Y, i)$ holds, we have $\mathbf{e}_n = \mathbf{c}_n$. We also have that $\mathbf{c} = \mathbf{w}$, and so $\mathbf{e}_n = \mathbf{w}_n$.

  Observe that if $\mathsf{source}(Y, i, n)$ is defined such that $\mathsf{source}(Y, i, n) = e_s^n$, it must be that $(\mathbf{c}_n = e_s^n)$ is in $\mathsf{Collect}(\phi_X, i)$. Since $\mathbf{c} = \mathbf{w}$, $\mathbf{w}_n = e_s^n$ whenever $\mathsf{source}(Y, i, n) = e_s^n$. Also note that by Remark 5, if $\mathsf{source}(Y, i, n) = e_s^n$ and $\mathsf{copy}(Y, i, n) = n$, it must be the case that also $\mathsf{target}(Y, i, n) = e_s^n$.

  Let $\mathbf{u}$ be a vector defined in the following way:

$$\mathbf{u}_n = \begin{cases} e_t^n & \text{if } \mathsf{target}(Y, i, n) = e_t^n \text{ is defined} \\ \mathbf{w}_n & \text{if } \mathsf{copy}(Y, i, n) = n \text{ is defined} \end{cases}$$

  Observe that $(\mathbf{c} = \mathbf{u})[\mathbf{e}/\mathbf{c}]$ since for all $\mathbf{u}_n$ either $\mathbf{e}_n = \mathbf{u}_n = \mathbf{e}_t^n$ whenever $\mathsf{target}(Y, i, n) = e_t^n$, and $\mathbf{e}_n = \mathbf{u}_n = \mathbf{w}_n$ whenever $\mathsf{copy}(Y, i, n) = n$.

  Since $(Y, \mathbf{w}) \in R$ and for every n, either

  - $\mathsf{source}(Y, i, n) = e_s^n = \mathbf{w}_n$ and $\mathsf{target}(Y, i, n) = e_t^n = \mathbf{u}_n$, or
  - $\mathsf{source}(Y, i, n) = \bot$, $\mathsf{copy}(Y, i, n) = n$ and $\mathbf{w}_n = \mathbf{u}_n$, or
  - $\mathsf{source}(Y, i, n) = \bot$ and $\mathsf{target}(Y, i, n) = e_t^n = \mathbf{u}_n$

  Then by definition of CFGs, there must be a transition such that $(Y, \mathbf{w}) \rightarrow (Z, \mathbf{u})$ and $(Z, \mathbf{u}) \in R$. Since this is the case, $(\mathbf{c} = \mathbf{u})$ is in $f(Z)$. We then have that $f(Y)[\mathbf{e}/\mathbf{c}]$ holds since $(\mathbf{c} = \mathbf{u})[\mathbf{e}/\mathbf{c}]$ holds.

As we have proven in both cases that $f(Z)[\mathbf{e}/\mathbf{c}][\mathbf{f}/\mathbf{d}_Z]$ holds assuming $f(Y) \wedge \mathsf{Collect}(\phi_Y, i)$, by Property 2, we may conclude that f is a global invariant. □

Theorem 2 shows that a PBES's CFG gives rise to an invariant. Since CFGs illustrate the possible values of CFPs, invariants from Theorem 2 characterize the set of CFP values for a given initial instance. We can then utilize Theorem 35 from [6] to solve a PBES $\mathscr{E}$ by including these invariants into $\mathscr{E}$ and solving the modified PBES.

Example 18. Consider PBES (2) in Example 9. For convenience, we write the equation system here:

$$\nu X(i, j, k, l : N) = (i \neq 1 \vee j \neq 1 \vee X(2, j, k, l + 1)) \wedge (\forall m{:}N.\ Z(i, 2, m + k, k))$$
$$\mu Y(i, j, k, l : N) = k = 1 \vee (i = 2 \wedge X(1, j, k, l))$$
$$\nu Z(i, j, k, l : N) = (k < 10 \vee j = 2) \wedge (j \neq 2 \vee Y(1, 1, l, 1)) \wedge Y(2, 2, 1, l)$$

Using Theorem 2 and the CFG in Figure 2, we obtain the following invariants:

$$f(X) = (i = 1 \wedge j = 1) \vee (i = 1 \wedge j = 2) \vee (i = 2 \wedge j = 1)$$
$$f(Y) = (i = 1 \wedge j = 1) \vee (i = 2 \wedge j = 2)$$
$$f(Z) = (i = 1 \wedge j = 2) \vee (i = 2 \wedge j = 2)$$

Adding these invariants to the original PBES yields the following:

$$\nu X(i, j, k, l : N) = f(X) \wedge (i \neq 1 \vee j \neq 1 \vee X(2, 1, k, l + 1)) \wedge (\forall m{:}N.\ Z(i, 2, m + k, k))$$
$$\mu Y(i, j, k, l : N) = f(Y) \wedge (k = 1 \vee (i = 2 \wedge X(1, 2, k, l)))$$
$$\nu Z(i, j, k, l : N) = f(Z) \wedge Y(1, 1, l, 1) \wedge Y(2, 2, 1, l)$$

Observe the simplification of equation Z and the insertion of values in PVI arguments within equations X and Y.

Invariants derived from our theorem provide an overapproximation of the values parameters may hold given an initial instantiation. Note that Theorem 2 only provides information regarding CFP values; information regarding non-CFPs are not considered. Additionally, the extent for which this theorem simplifies PBES equations have not been analyzed. For instance, we cannot exploit the full potential of CFGs by only using invariants. In particular, Keiren et al. uses CFGs to remove 'dead' parameters in a PBES by analyzing relevant values at each vertex of a corresponding CFG [5]. However, we are not able to express this technique only using invariants. Nevertheless, adding invariants may enable other PBES solving techniques. The following example shows how invariants alone are unable to remove quantifiers, unlike liveness analysis techniques, but enables redundant parameter elimination.

Example 19. Consider the following PBES:

$$\mu X(i, j : N) = \forall n{:}N.\ Y(2, n + j)$$
$$\mu Y(i, j : N) = (i = 2 \vee j < 5) \wedge X(i, 2)$$

Assume we wish to find a solution for $X(1, 1)$. Observe that for any instantiation of X, we depend on $Y(2, \_)$. Therefore, $j < 5$ is irrelevant in Y since $i = 2$ will always hold if we start with any instance of X. Furthermore, other than the subformula $j < 5$, j does not occur elsewhere in Y. In the work by Keiren et al. [5], we may replace $\forall n{:}N.\ Y(2, n + j)$ with $\forall n{:}N.\ Y(2, 1)$ and still arrive at the same solution, effectively eliminating the universal quantifier.

Unfortunately, invariants cannot express that $j \in \mathsf{par}(Y)$ is irrelevant and thus can be any value. This is due to the fact that the invariant for X cannot not assume that a parameter of Y is irrelevant. The best we can do is to propagate the information that $(i = 2 \vee j < 5)$ must always hold for Y; we are not able to express that the value of $n + j$ is not relevant in $Y(2, n + j)$.

Still, invariants may enable other techniques to simplify PBES equations. Consider the invariant $f(Y) = (i = 2)$. This invariant yields the following

simplified PBES:

$$\mu X(i, j : N) = \forall n{:}N.\ Y(2, n + j)$$
$$\mu Y(i, j : N) = (i = 2) \wedge X(i, 2)$$

Using redundant parameter elimination [4], we may remove parameter j from both equations.

6 Relevancy Graphs

As seen thus far, proving the invariance property for simple functions is mathematically involved; from Definition 8, it is difficult to intuitively understand how simple functions remain invariant throughout an equation system. To provide insight on how simple functions satisfying the global invariant property behaves in PBESs, we present a novel graph structure: relevancy graphs. Similar to guards, relevancy graphs characterize the relevant predicate variables in an equation. Unlike guards, relevancy graphs do not operate on single PVIs, rather on all predicate variable instances for which their arguments evaluate to a certain value. In this section, we present the concept of relevancy graphs and its relationship between invariants. We not only show that a global invariant gives rise to relevancy graphs with certain properties, we also provide a condition for when a simple functions satisfies the invariance property via relevancy graphs, allowing one to use relevancy graphs as an alternative proof method. In the latter half of this section, we provide examples to demonstrate how our theorems could be used. In particular, we prove the theorem on CFG derived invariants (Theorem 2) using relevancy graphs.

While various graph representations of PBESs exists, such as proof graphs [36, 37] or structure graphs [38, 32], it is difficult to express the relationship between these graphs and invariants. Proof graphs acts as a certificate of a solution for a given PBES. However, these graphs fail to illustrate the predicates that may be explored but are not necessarily needed in the final solution. Meanwhile, structure graphs provide a view on the syntactic structure of PBESs to allow for simplification of equations, but fail to provide information on when a particular PVI is irrelevant under certain parameters. Therefore, we present a new graph representation for PBES, namely *relevancy graphs*, and show the connections between invariants of a PBES and its associated relevancy graph.

Without loss of generality, this section will be operating under Remark 3. For the reader's convenience, we restate this remark:

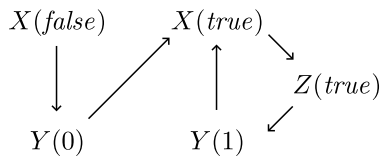
Remark 3. For the sake of simplicity, we associate a single variable d_X of sort D_X to predicate variables X rather than use vectors $\mathbf{d}_X$ of sort $\mathbf{D}_X$. This does not result in a loss of generality as one could utilize a data sort which is able to express more complex formulae.

Our relevancy graph takes inspiration from proof graphs [37]. Similar to proof graphs, we take a look at parameters with specific parameter instantiations.

Thus, for some PBES $\mathscr{E}$, the vertices of our graph are elements of the set $\mathsf{sig}(\mathscr{E}) = \{(X, v) \mid X \in \mathsf{bnd}(\mathscr{E}), v \in \mathbb{D}_X\}$, where elements of $\mathsf{sig}(\mathscr{E})$ are called *instantiations*. At times, we may write $X(v)$ for (X, v), especially in figures. Unlike proof graphs, relevancy graphs considers all possible parameter instantiations in a given equation.

Example 20. Consider the following equation system and its proof graph showing that $X(\mathit{false})$ is true. Informally, given a state $X(v)$ in a proof graph, successor states show what instantiation must be true for $X(v)$ to be true. For additional information, refer to the paper by Cranen et al. [37].

$$\begin{aligned}\mu X(b : B) &= Y(0) \vee Z(b)\\ \nu Y(n : N) &= \mathsf{odd}(n) \vee X(\mathit{true})\\ \mu Z(b : B) &= (\neg b \implies Y(0))\\ &\quad \wedge (b \implies Y(1))\end{aligned}$$



As can be seen, $X(\mathit{false})$ is true if $Y(0)$ is also true, which requires $X(\mathit{true})$ to be true and so on. However, invariants cannot determine the truth value of predicate variables in an equation and thus must consider all predicates that may affect the solution. For instance, let $f : \{X, Y, Z\} \to \mathsf{Pred}$ be a simple function such that $f(X)$ evaluates to true for all instances of X. In order for f to be an invariant in the instance $X(\mathit{false})$, we must check whether the invariant remains true for both $Y(0)$ and $Z(\mathit{false})$. Additionally, observe that the edge $Y(1) \to X(\mathit{true})$ is not necessary since $\mathsf{odd}(1)$ is true. Moreover, adding the edge $X(\mathit{false}) \to Y(5)$ would not violate proof graph definitions since the truth of $Y(5)$ does not affect the truth of $X(\mathit{false})$.

From this example, to express when an invariant may hold or not for a given PBES, we need additional information that proof graphs do not necessarily provide. In addition, relevancy graphs should omit transitions to instances of PVIs that do not affect the outcome of an equation. Rather than showing true successor instantiations for a given instance to be true, relevancy graphs capture the predicate variables that are relevant to instantiations and leaves out dependencies that are unnecessary.

Throughout this section, we abuse notation by writing $\eta[\beta/X(e)]$ for $\beta \in \mathbb{B}$ to denote the predicate environment where $\eta[\beta/X(e)](Y)(f) = \beta$ if $X = Y$ and $f = e$, and $\eta[\beta/X(e)](Y)(f) = \eta(Y)(f)$ otherwise. We now provide a formal definition for relevancy graphs.

Definition 23. *A* relevancy graph *for a closed PBES $\mathscr{E}$ is defined as a tuple $G = (S, \to)$, where*

- $S \subseteq \mathsf{sig}(\mathscr{E})$ *is a set of states,*

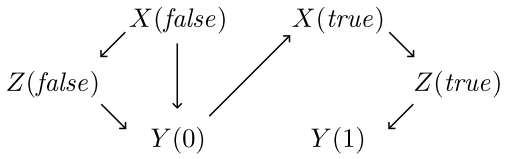
$- \rightarrow \subseteq S \times S$ *a transition relation such that for any* $(X, v) \in S$,

$$(X, v) \rightarrow (Y, w) \text{ iff}$$
$$\exists \eta, \varepsilon.\ [\![\phi_X]\!]\eta[\mathsf{true}/Y(w)]\varepsilon[v/d_X] \neq [\![\phi_X]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v/d_X]$$

We say $G = (S, \rightarrow)$ *is a relevancy graph for* (X, v) *iff* $(X, v) \in S$.

Intuitively, given a relevancy graph $G = (S, \rightarrow)$ and a specific instantiation $(X, v) \in S$, we have an edge $(X, v) \rightarrow (Y, w)$ iff changing the truth value of $Y(w)$ can affect the evaluation of ϕ_X when $d_X = v$.

Example 21. Consider the equation system found in Example 20. According to Definition 23, we have the following relevancy graph $G = (S, \rightarrow)$ with $(X, v) \in S$.



When considering $X(\mathit{false})$, we must not only consider $Y(0)$ but also $Z(\mathit{false})$. In addition, since $\mathsf{odd}(1)$ occurs in $Y(1)$, the truth value of $X(w)$, for any $w : D_X$, does not affect the truth of $Y(1)$.

Before relating invariants to relevancy graphs, we require some lemmas to assist in future proofs. In particular, we need to address minimality of relevancy graphs.

Example 22. Consider the relevancy graph of the previous example (Example 21). We may add $Y(3)$ into the graph without any additional edges and still obtain a valid relevancy graph. However, if one was interested in the solution of $X(\mathit{false})$, it would be unnecessary to include $Y(3)$.

To achieve minimal relevancy graphs, we first prove the following lemma, which shows that given some relevancy graph $G = (S, \rightarrow)$ for (X, v), a sub-graph consisting of the set of vertices reachable from some state $(X, v) \in S$ is still a relevancy graph.

Lemma 7. *Let* $\mathscr{E}$ *be a PBES and* $G = (S, \rightarrow)$ *be its corresponding relevancy graph such that* $(X, v) \in S$. *Define the reachable sub-graph originating from* (X, v) *to be* $\mathsf{Reach}_{(X,v)}(G) = (S', \rightarrow')$, *where*

$$S' = \{(X', v') \mid (X, v) \rightarrow^* (X', v')\}$$
$$\rightarrow' = \rightarrow \cap (S' \times S')$$

Then, $\mathsf{Reach}_{(X,v)}(G)$ *is also a relevancy graph for* (X, v).

The next lemma shows that relevancy graphs derived from a reachable set of vertices is minimal:

Lemma 8. *Let $\mathscr{E}$ be a PBES and $G = (S, \to)$ be its corresponding relevancy graph for some (X, v), and let $\mathsf{Reach}_{(X,v)}(G) = (S', \to')$ be the sub-graph of G consisting of vertices reachable from (X, v). Let V be a nonempty set such that $V \subset S'$, and let the graph $G' = (S'', \to'')$ be the graph $\mathsf{Reach}_{(X,v)}$ without vertices in V. Specifically,*

$$\begin{aligned} S'' &= S' \setminus V \\ \to'' &= \to' \cap (S'' \times S'') \end{aligned}$$

Then, G' cannot be a relevancy graph for (X, v).

We state that a function f is true, or holds, in a state $(X, v) \in S$ for some relevancy graph $G = (S, \to)$ if $[\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{true}$. Likewise, f is false, or fails to hold, for (X, v) if $[\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{false}$.

We first present a theorem which states that if a PBES global invariant $f(X)$ is true for a specific instance v, there exists a minimal relevancy graph containing (X, v), and for all states in the graph, the corresponding invariant also holds. In other words, the invariant holds in all states of the graph. We utilize the following helper lemma to assist in our proof.

Lemma 9. *Let $f : V \to \mathsf{Pred}$ be a simple, capture-avoiding function where $V \subseteq \mathcal{P}$. Assume that for every $X \in V$, $\mathsf{FV}(f(X)) \subseteq \{d_X\}$ with $d_X : D_X$. Let $X \in V$ and $[\![f(X)]\!]\eta'\varepsilon'[v/d_X] = \mathsf{false}$ for some $v \in \mathbb{D}_X$ and environments η', ε'. Then, for any predicate formula ϕ and arbitrary environments η, ε,*

$$\begin{aligned} &\left[\!\!\left[\phi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\ &= \\ &\left[\!\!\left[\phi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{false}/X(v)]\varepsilon \end{aligned}$$

Proof. Intuitively, if $[\![f(X)]\!]\eta'\varepsilon'[v/d_X]$ evaluates to false, the truth of $X(v)$ is irrelevant in $\phi[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i]$. In other words, due to simultaneous substitution, $X(e)$ in ϕ will be replaced with $f(X) \wedge X(e)$, and if e is eventually evaluated to v, we must also evaluate $[\![f(X)]\!]\eta'\varepsilon'[v/d_X]$, which is false. Thus, the truth of $X(e)$ in $\phi[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i]$ is irrelevant when e is evaluated to v. A complete proof, which proceeds by induction on the structure of ϕ, can be found in Appendix A.3. □

We now show that global invariants which hold in a given instance also gives rise to a relevancy graph where the invariant holds in all states.

Theorem 3. *Let $\mathscr{E}$ be a closed PBES, and let $f : V \to \mathsf{Pred}$ be a simple, capture-avoiding function where $V = \mathsf{bnd}(\mathscr{E})$. Let $\sigma X(d_X : D_X) = \phi_X$ be some equation in $\mathscr{E}$ and let the following statements hold:*

- *f is a global invariant for $\mathscr{E}$, and*
- *$[\![f(X)]\!]\eta\varepsilon[v/d_X]$ holds for some η, ε and $v \in \mathbb{D}_X$*

Then there exists a minimal relevancy graph $G = (S, \to)$ such that $(X, v) \in S$ and for all states $(Y, w) \in S$, $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds.

Proof. Let $\mathscr{E}$ be a closed PBES and $f : V \to \mathcal{P}$ be a global invariant for $\mathscr{E}$ such that $[\![f(X)]\!]\eta\varepsilon[v/d_X]$ holds for some $v \in \mathbb{D}_X$. Let $G' = (S', \to')$ be a relevancy graph with $(X, v) \in S'$. Then $G = \mathsf{Reach}_{(X,v)}(G')$ is a minimal relevancy graph containing (X, v).

We now prove that for any state $(Y, w) \in S$, $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds. We proceed with induction on the length of paths n starting from (X, v).

Base Case: $n = 0$. Holds by assumption; $[\![f(X)]\!]\eta\varepsilon[v/d_X]$ holds.

Step Case: $n \neq 0$. Assume the following induction hypothesis. For all i such that $0 \leq i < n$ and $(X, v) \to^i (Z, u)$, $[\![f(Z)]\!]\eta\varepsilon[u/d_Z]$ holds. We prove that $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds, where $(X, v) \to^n (Y, w)$. Consider some predecessor vertex $(Z, u) \in S$ such that $(X, v) \to^{n-1} (Z, u) \to (Y, w)$. By Definition 23, let η', ε' be environments such that

$$[\![\phi_Z]\!]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z] \neq [\![\phi_Z]\!]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z] \tag{3}$$

We also have that $[\![f(Z)]\!]\eta\varepsilon[u/d_Z]$ holds via the induction hypothesis. It follows from Corollary 1 that $[\![f(Z)]\!]\eta\varepsilon[u/d_Z] = [\![f(Z)]\!]\eta'\varepsilon'[u/d_Z]$. By definition of $[\![\_]\!]$ and Equation (3), the following must hold:

$$[\![f(Z) \wedge \phi_Z]\!]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z] \neq [\![f(Z) \wedge \phi_Z]\!]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z] \tag{4}$$

Since f is a global invariant, by Definition 8 we have

$$[\![f(Z) \wedge \phi_Z]\!]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z] =$$
$$\Big[\!\Big[f(Z) \wedge \phi_Z\Big[_{W_i \in V}(f(W_i) \wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\Big]\Big]\!\Big]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z]$$

and

$$[\![f(Z) \wedge \phi_Z]\!]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z] =$$
$$\Big[\!\Big[f(Z) \wedge \phi_Z\Big[_{W_i \in V}(f(W_i) \wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\Big]\Big]\!\Big]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

It follows from (4) that the following must also hold:

$$\Big[\!\Big[f(Z) \wedge \phi_Z\Big[_{W_i \in V}(f(W_i) \wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\Big]\Big]\!\Big]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z]$$
$$\neq \tag{5}$$
$$\Big[\!\Big[f(Z) \wedge \phi_Z\Big[_{W_i \in V}(f(W_i) \wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\Big]\Big]\!\Big]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

By definition of $[\![\_]\!]$ and Equation (5),

$$[\![f(Z)]\!]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z] \text{ and}$$
$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z]$$
$$\neq$$
$$[\![f(Z)]\!]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z] \text{ and}$$
$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

By Lemma 1, we have that

$$[\![f(Z)]\!]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z] = [\![f(Z)]\!]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

Therefore, it must be the case that:

$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z]$$
$$\neq \tag{6}$$
$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

For the sake of contradiction, assume that $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y] = \mathsf{false}$. By Lemma 1, we get that $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y] = [\![f(Y)]\!]\eta'(\varepsilon'[u/d_Z])[w/d_Y]$. Then, by Lemma 9, we have the following.

$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{true}/Y(w)]\varepsilon'[u/d_Z]$$
$$=$$
$$\left[\!\!\left[\phi_Z\left[_{W_i\in V}(f(W_i)\wedge W_i(d_{W_i}))_{\langle d_{W_i}\rangle}/W_i\right]\right]\!\!\right]\eta'[\mathsf{false}/Y(w)]\varepsilon'[u/d_Z]$$

However, this statement contradicts Equation (6). Since we have arrived at a contradiction, our assumption is invalid and the contrary is true: it must be the case that $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y] = \mathsf{true}$. By Corollary 1, we then have $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y] = [\![f(Y)]\!]\eta\varepsilon[w/d_Y] = \mathsf{true}$.

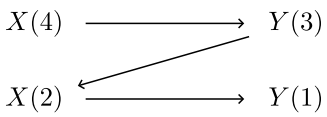
We have shown via induction that for all states $(Y, w) \in S$ satisfy $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$, given that G is a minimal dependency graph containing (X, v). Thus, we have shown that our claim holds. □

From Theorem 3, we can illustrate how invariants operate in the context of PBESs. Given an instance of some equation, say $X(v)$, and assuming the invariant $f(X)$ holds under v (i.e. $[\![f(X)]\!]\eta\varepsilon[v/d_X]$ for any η, ε), it must be the case that for any relevant predicate variable instantiation $Y(w)$ of ϕ_X must have its invariant hold under w. In other words, given an invariant f, whenever this invariant holds for the equation X the invariant must hold for all equations that are relevant in ϕ_X.

Example 23. Consider the following PBES and its corresponding global invariant f.

$$\begin{aligned} &\mu X(n:N) = (n > 0) \implies Y(n-1) && f(X) = \mathsf{even}(n) \\ &\mu Y(n:N) = (n = 0) \vee X(n-1) && f(Y) = \mathsf{odd}(n) \end{aligned}$$

Then, by Theorem 3, we can construct a relevancy graph $G = (S, \to)$ by fist starting at some instantiation for which the invariant holds, e.g. $X(4)$, and use the edge definition in Definition 23. We then obtain the following relevancy graph for this PBES.



Note that since $\mathsf{even}(4)$ holds, also $[\![f(X)]\!]\eta\varepsilon[4/d_X]$ holds. Also observe that for each instance $X(v)$ in the graph, the evaluation $[\![f(X)]\!]\eta\varepsilon[v/d_X]$ remains true, and similarly the same holds for all instances of predicate Y in the graph.

As seen in Example 23, Theorem 3 realizes a minimal relevancy graph which shows how a global invariant remains invariantly true throughout the state space of the equation system, assuming the invariant was true initially. However, while Theorem 3 shows how an invariant of a PBES remains invariant throughout the traversal of relevant predicate variable instantiations, this theorem does not give us information as to what a simple function needs to satisfy in a given relevancy graph to fulfill global invariant requirements. Intuitively, a simple function f is an invariant for some PBES $\mathscr{E}$ if there exists an associated relevancy graph $G = (S, \to)$ such that for all states $(X, v) \in S$ the invariant holds under v (i.e. for any η, ε, $[\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{true}$), and for all other states $(Y, w) \notin S$, the invariant fails to hold under w (i.e. for any η, ε, $[\![f(Y)]\!]\eta\varepsilon[w/d_Y] = \mathsf{false}$).

Example 24. Consider the PBES below.

$$\mu X(n:N) = ((0 < n < 5) \implies X(n-1)) \wedge ((n \geq 5) \implies X(n+1))$$

By Definition 23, we can construct the following minimal relevancy graph $G = (S, \to)$ such that $(X, 5) \in S$.

$$X(5) \longrightarrow X(6) \longrightarrow X(7) \longrightarrow \cdots$$

Now let $f(X) = (n \geq 3)$. Observe that for all states $(X, v) \in S$, $[\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{true}$. However, f fails to be an invariant for this PBES since $X(3)$ relies on $X(2)$,

which violates the invariant that $n \geq 3$. Therefore, for a simple function to be an invariant, it must also fail for all states not in the relevancy graph.

To assist in proving the relevancy graph invariant condition, we prove the following lemma. This lemma expresses that for some right-hand side of an equation ϕ_X of PBES $\mathscr{E}$, adding the simple function f to all PVIs does not affect the solution to the evaluation of ϕ_X assuming that there is a relevancy graph $G = (S, \to)$ for $\mathscr{E}$ such that the simple function holds under (X, v) iff the state (X, v) is in S.

Lemma 10. *Let $\mathscr{E}$ be a closed PBES and let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple, capture-avoiding function. Let $G = (S, \to)$ be a relevancy graph for $\mathscr{E}$. Assume that $(Y, w) \in S$ iff $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds for all environments η, ε. Then, for any $(X, v) \in S$ and for all environments η, ε, we have that*

$$[\![\phi_X]\!]\eta\varepsilon[v/d_X] = \left[\!\!\left[\phi_X \left[_{Y_i \in \mathsf{bnd}(\mathscr{E})} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i\right]\right]\!\!\right] \eta\varepsilon[v/d_X]$$

Proof. Here, we provide some intuition for the proof. For the complete proof, refer to Appendix A.3. Assume that we are in some state $(X, v) \in S$ of the relevancy graph. Consider some $\mathsf{PVI}(\phi_X, i)$ where $\mathsf{pv}(\phi_X, i) = Y$. If this PVI is relevant under the evaluation of ϕ_X under v, then there must be a transition in the relevancy graph such that $(X, v) \to (Y, w)$ for some $w \in \mathbb{D}_Y$. Since $(Y, w) \in S$, $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds and adding the invariant to this PVI would not affect the evaluation of ϕ_X with instance v. If this PVI is not relevant, the outcome of the evaluation of $\mathsf{PVI}(\phi_X, i)$ is irrelevant and thus knowledge on the truth of $f(Y)$ is irrelevant for PVI i of ϕ_X. □

We now prove that for some PBES $\mathscr{E}$ and an associated relevancy graph $G = (S, \to)$, a simple function $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ is a global invariant whenever f holds under state (X, v) iff $(X, v) \in S$.

Theorem 4. *Let $\mathscr{E}$ be a closed PBES and let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple, capture-avoiding function. Let $G = (S, \to)$ be a relevancy graph for $\mathscr{E}$. For all $X \in \mathsf{bnd}(\mathscr{E})$, assume that $\mathsf{FV}(f(X)) \subseteq \{d_X\}$ and let $f(X)$ be evaluated in the following way:*

$$[\![f(X)]\!]\eta\varepsilon[v/d_X] = \begin{cases} \mathsf{true} & \text{if } (X, v) \in S \\ \mathsf{false} & \text{if } (X, v) \notin S \end{cases}$$

for all $v \in \mathbb{D}_X$ and any environments η, ε. Then f is a global invariant for $\mathscr{E}$.

Proof. Let $\mathscr{E}$ be some arbitrary PBES and let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple, capture-avoiding function. Let $G = (S, \to)$ be a relevancy graph and assume f satisfies the conditions above.

By Definition 8, if for all equations $\sigma X(d_X : D_X) = \phi_X$ in $\mathscr{E}$,

$$f(X) \wedge \phi_X \leftrightarrow (f(X) \wedge \phi_X)\left[_{Y_i \in \mathsf{bnd}(\mathscr{E})} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i\right]$$

then f is a global invariant. By the definition of $\leftrightarrow$, we show that for all environments ε, η and all equations $\sigma X(d_X : D_X) = \phi_X$ the following holds.

$$[\![f(X) \wedge \phi_X]\!]\eta\varepsilon = \left[\!\!\left[(f(X) \wedge \phi_X)\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon$$

Let η, ε be any environment and $\sigma X(d_X : D_X) = \phi_X$ be any equation in $\mathscr{E}$. We distinguish two cases: $[\![f(X)]\!]\eta\varepsilon = \mathsf{true}$ and $[\![f(X)]\!]\eta\varepsilon = \mathsf{false}$.

First, consider the case where $[\![f(X)]\!]\eta\varepsilon = \mathsf{false}$. Then, we have that

$$\begin{aligned}
&[\![f(X) \wedge \phi_X]\!]\eta\varepsilon \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![f(X)]\!]\eta\varepsilon \text{ and } [\![\phi_X]\!]\eta\varepsilon \\
&= \{\text{Case } [\![f(X)]\!]\eta\varepsilon = \mathsf{false}\} \\
&\quad \mathsf{false} \\
&= \{[\![f(X)]\!]\eta\varepsilon = \mathsf{false}\} \\
&\quad [\![f(X)]\!]\eta\varepsilon \text{ and } \left[\!\!\left[\phi_X\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad \left[\!\!\left[f(X) \wedge \phi_X\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon
\end{aligned}$$

Now consider the case where $[\![f(X)]\!]\eta\varepsilon = \mathsf{true}$. Then for some $v \in \mathbb{D}_X$, $\varepsilon = \varepsilon[v/d_X]$ and $[\![f(X)]\!]\eta\varepsilon = [\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{true}$. By definition of f, since $[\![f(X)]\!]\eta\varepsilon = \mathsf{true}$, there exists a $(X, v) \in S$, otherwise $(X, v) \notin S$ and $[\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{false}$. We now argue as follows:

$$\begin{aligned}
&[\![f(X) \wedge \phi_X]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![f(X)]\!]\eta\varepsilon[v/d_X] \text{ and } [\![\phi_X]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Case } [\![f(X)]\!]\eta\varepsilon[v/d_X] = \mathsf{true}, \text{ and Lemma 10}\} \\
&\quad [\![f(X)]\!]\eta\varepsilon[v/d_X] \text{ and } \left[\!\!\left[\phi_X\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad \left[\!\!\left[f(X) \wedge \phi_X\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon[v/d_X]
\end{aligned}$$

As can be seen, we have proven that for any η, ε,

$$[\![f(X) \wedge \phi_X]\!]\eta\varepsilon = \left[\!\!\left[(f(X) \wedge \phi_X)\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta\varepsilon$$

As this argument can be applied to all $Y \in \mathsf{bnd}(\mathscr{E})$, by Definition 8, we have proven that f is a global invariant for $\mathscr{E}$. □

Theorem 4 provides an alternate characterization of global invariants in PBESs. Whenever we have some PBES $\mathscr{E}$, to verify that $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ is a global invariant, one can construct a relevancy graph such that f holds for state (X, v) iff $(X, v) \in S$. Note that this theorem is not a strict requirement for global invariants. For instance, consider Example 23 and observe that $f(X) = \mathsf{even}(n)$ is still an invariant even though $(X, 6)$ is not in the relevancy graph.

6.1 Proving with Relevancy Graphs

To show how one can utilize our theorem to show that a simple function is a global invariant, we provide a few examples on the application of Theorem 4 in this section. We later show an alternative proof for our CFG invariant described in Section 5.

While we have proven Theorem 4 under the assumption that all PBES equations have exactly one parameter, one can use pairing and projection functions to obtain more complex formulae. Thus, we show an application of Theorem 4 on a PBES with multiple parameters.

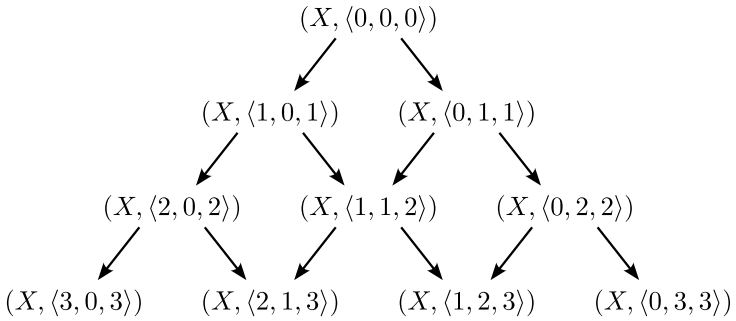
Example 25. Consider the following PBES consisting of one equation:

$$\mu X(i, j, k : N) = (k = 3) \vee X(i + 1, j, k + 1) \vee X(i, j + 1, k + 1)$$

We now prove that f defined in the following way satisfies the global invariant property:

$$f(X) = (k \leq 3) \wedge (i + j = k)$$

The following figure illustrates the minimal relevancy graph $G = (S, \to)$ such that $(X, \langle 0, 0, 0\rangle) \in S$:



Observe that for any state $(X, \mathbf{v}) \in S$, we have that $[\![f(X)]\!]\eta\varepsilon[\mathbf{v}/\mathbf{d}_X]$ holds. For any other state $(X, \mathbf{w}) \notin S$, $[\![f(X)]\!]\eta\varepsilon[\mathbf{w}/\mathbf{d}_X]$ fails to hold. Thus, by Theorem 4, f is a global invariant.

If one wants to prove a more general invariant, one must provide a more complete relevancy graph. For instance, assume that we wish f to be defined as follows:

$$f(X) = (i + j = k)$$

We can still prove that f is a global invariant using Theorem 4 by defining a relevancy graph consisting of more vertices. In particular, let $G' = (S', \rightarrow')$ be a minimal relevancy graph where $(X, \langle a, b, c\rangle) \in S'$ for all $a, b, c : \mathbb{N}$ such that $a + b = c$ holds. Then, one must argue that there cannot be edges to vertices such that the invariant is invalidated.

Control Flow Graphs
We now prove Theorem 2 using relevancy graphs. The remainder of this section operates under the remarks found in Section 5:

Remark 6. Assume that the set of CFPs is the same for all equations in a PBES $\mathscr{E}$, i.e. for all $X, Y \in \mathsf{bnd}(\mathscr{E})$, $d^X \in \mathsf{par}(X)$ is a CFP iff $d^Y \in \mathsf{par}(Y)$ is a CFP, and $d^X \approx d^Y$. Moreover, assume all equations in $\mathscr{E}$ are of the form $\sigma X(\mathbf{c} : \mathbf{C}, \mathbf{d_X} : \mathbf{D_X}) = \phi_X$, where $\mathbf{c}$ is a vector of CFPs and $\mathbf{d_X}$ is a vector of data parameters for the PBES equation X.

Before providing the formal proof, we first provide some intuition by using Example 9 as a running example. For the reader's convenience, we provide the equation system and its corresponding relevancy graph in this section:

$$\begin{aligned}
\nu X(i, j, k, l : N) &= (i \neq 1 \vee j \neq 1 \vee X(2, j, k, l + 1)) \\
&\quad \wedge (\forall m{:}N.\ Z(i, 2, m + k, k)) \\
\mu Y(i, j, k, l : N) &= k = 1 \vee (i = 2 \wedge X(1, j, k, l)) \\
\nu Z(i, j, k, l : N) &= (k < 10 \vee j = 2) \wedge (j \neq 2 \vee Y(1, 1, l, 1)) \wedge Y(2, 2, 1, l)
\end{aligned} \tag{2}$$

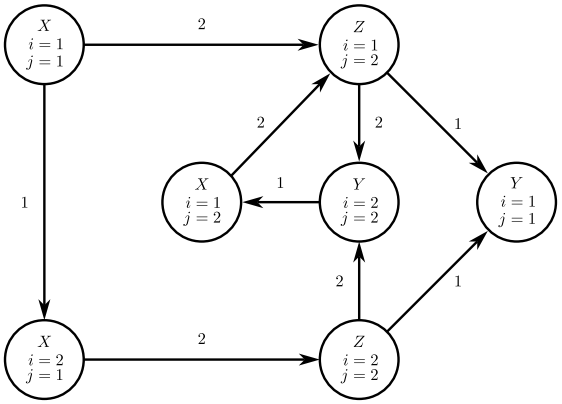


Fig. 2. CFG of PBES (2).

Recall that i, j are control flow parameters (CFPs) for this equation system, and that the CFG in Figure 2 shows that the solution for $X(1, 1, \_, \_)$ depends

on the solution of $Z(1,2,\_,\_)$ and $X(2,1,\_,\_)$. Assume we wish to find the solution for $X(1,1,1,1)$. We may construct a minimal relevancy graph $G = (S, \rightarrow)$ for PBES (2) such that $(X, \langle 1,1,1,1\rangle) \in S$:

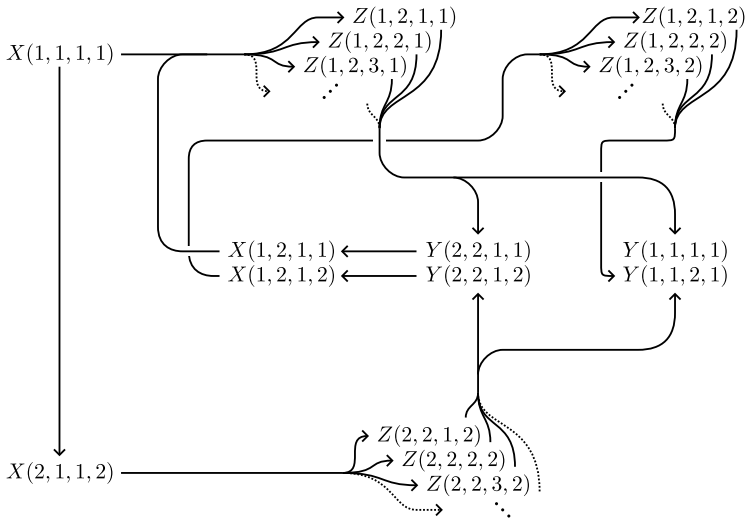


Fig. 5. Relevancy graph of PBES (2), where $(X, \langle 1,1,1,1\rangle) \in S$.

Observe the similarities to Figure 2; each state in the relevancy graph can be associated to a state in the CFG. In other words, when finding the solution for $X(1,1,1,1)$, the CFPs of each state in the relevancy graph must be related to some state in the CFG. We highlight the similarities in the following figure:

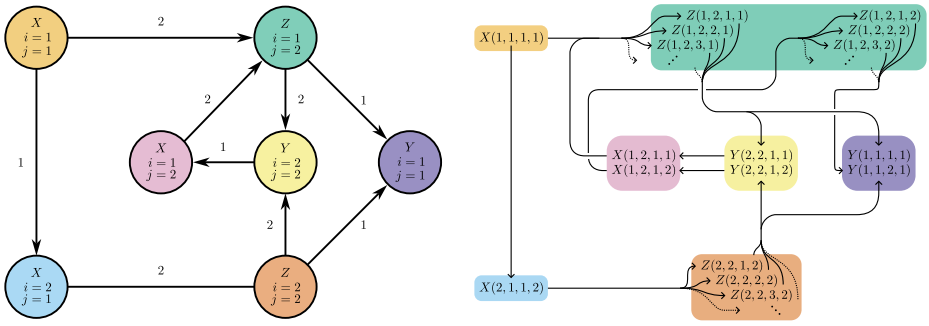


Fig. 6. Similarities between a CFG and a relevancy graph of a given PBES.

However, according to Theorem 4, we must also show that the invariant fails to hold for states not present in the graph; e.g. we must show that the invariant fails for all $(X, \langle 1, 1, a, b\rangle) \notin S$, where $a \neq 1$ and $b \neq 1$. Thus, proving the invariance property of simple functions that characterizes the CFPs in a CFG is not possible using Theorem 4 with our current relevancy graph. We can overcome this by defining a minimal relevancy graph such that all possible parameter values are included in the relevancy graph for non-CFPs. Using the running example, for all a, b, we require that $(X, \langle 1, 1, a, b\rangle)$ is in S. This applies for all corresponding states in the CFG. We then use Theorem 4 to prove that a simple function satisfying Theorem 2 is a global invariant. We now provide the formal proof:

Theorem 5. *Let $\mathscr{E}$ be a PBES and let the corresponding CFG graph be $(V, \to)$ with initial state $(X, \mathbf{v}) \in V$. Let $R \subseteq V$ be the set of vertices reachable from $(X, \mathbf{v})$, and let W be the set of predicate variables whose equation is reachable from $(X, \mathbf{v})$, defined as $W = \{X' \mid (X', \mathbf{v}') \in R\}$. Let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple function such that for all $Y \in W$ we have*

$$f(Y) = \bigvee_{(Y,\mathbf{v}')\in R} (\mathbf{c} = \mathbf{v}')$$

and for all $Y \notin W$, $f(Y) = \mathsf{false}$. Then f is a global invariant for PBES $\mathscr{E}$.

Proof. Our proof construction is similar to the one provided in Section 5. However, rather than utilizing Property 2 we use relevancy graphs. Let $\mathscr{E}$ be a PBES and let the corresponding CFG be $(V, \to_C)$ with $(X, \mathbf{v}) \in V$. Let $R \subseteq V$ be the set of reachable vertices from $(X, \mathbf{v})$. Because relevancy graphs operate under semantic data terms and $\mathbf{v}$ is a vector of syntactic data terms not containing data variables, we write $[\![\mathbf{v}]\!]$ to be the interpretation of vector $\mathbf{v}$, i.e. $\langle [\![\mathbf{v}_1]\!], \ldots, [\![\mathbf{v}_n]\!]\rangle$ where $\mathbf{v} = \langle \mathbf{v}_1, \ldots, \mathbf{v}_n\rangle$. Observe that since $\mathbf{v}$ does not contain data variables or predicate variables, we do not need environments η, ε.

Let S' be defined such that for every vertex $(Y, \mathbf{w}) \in R$, $(Y, \langle [\![\mathbf{w}]\!], \mathbf{w}'\rangle) \in S'$ for all vectors $\mathbf{w}'$ consisting of elements of type $\mathbb{D}_Y$. Let $G = (S, \to_R)$ be a minimal relevancy graph satisfying $S' \subseteq S$. We now show that $S = S'$. In other words, $S \setminus S' = \emptyset$.

For the sake of contradiction, assume that $S \setminus S' \neq \emptyset$. Then there exist a state $(Y, \mathbf{w}) \in R$ and $(Y, \langle [\![\mathbf{w}]\!], \mathbf{w}'\rangle) \in S'$ such that

$$(Y, \langle [\![\mathbf{w}]\!], \mathbf{w}'\rangle) \to_R (Z, \langle [\![\mathbf{u}]\!], \mathbf{u}'\rangle)$$

where $(Z, \mathbf{u}) \notin R$ and $(Z, \langle [\![\mathbf{u}]\!], \mathbf{u}'\rangle) \notin S'$. It follows from Definition 23 that the following must hold:

$$\begin{aligned}\exists \eta, \varepsilon.\ & [\![\phi_Y]\!]\eta[\mathsf{true}/Z([\![\mathbf{u}]\!], \mathbf{u}')]\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][\mathbf{w}'/\mathbf{d}_Y] \\ & \neq [\![\phi_Y]\!]\eta[\mathsf{false}/Z([\![\mathbf{u}]\!], \mathbf{u}')]\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][\mathbf{w}'/\mathbf{d}_Y]\end{aligned}$$

Let η, ε be the two environments from the statement above. Let I be an index set where $i \in I$ if $\mathsf{PVI}(\phi_Y, i)$ is evaluated to true in

$$[\![\phi_Y]\!]\eta[\mathsf{true}/Z([\![\mathbf{u}]\!], \mathbf{u}')]\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][\mathbf{w}'/\mathbf{d}_Y] \tag{7}$$

and false in

$$[\![\phi_Y]\!]\eta[\mathsf{false}/Z([\![\mathbf{u}]\!], \mathbf{u}')]\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][\mathbf{w}'/\mathbf{d}_Y] \tag{8}$$

Recall that $\mathsf{Collect}(\phi_Y, 1), ..., \mathsf{Collect}(\phi_Y, \mathsf{npred}(\phi_Y))$ is compositional. I.e.

$$\phi_Y \leftrightarrow \phi_Y[j \mapsto (\mathsf{Collect}(\phi_Y, j) \wedge \mathsf{PVI}(\phi_Y, j))]_{j \leq \mathsf{npred}(\phi_Y)} \tag{9}$$

Therefore, whenever $\mathsf{PVI}(\phi_Y, i)$ evaluates to true, $\mathsf{Collect}(\phi_Y, i)$ must be evaluated to true.

By Remark 6, all equations in $\mathscr{E}$ have the same set of CFPs $\mathbf{c}$. Thus, by Definitions 16 and 20, we have that the vector of CFPs $\mathbf{c}$ is also a vector of GCFPs and LCFPs. Let $\mathsf{PVI}(\phi_Y, i) = Z(\mathbf{e}, \mathbf{f})$ where $i \in I$. By Equation (9), it must be the case that whenever $\mathsf{PVI}(\phi_Y, i)$ holds, it must be the case that $\mathsf{Collect}(\phi_Y, i)$ must also hold. Additionally, due to our construction of I, we have that $\mathbf{e}$ is eventually evaluated to $[\![\mathbf{u}]\!]$ in Equations (7) and (8). Similarly, $\mathbf{c}$ will be evaluated to $[\![\mathbf{w}]\!]$ in Equations (7) and (8).

Consider a CFP $\mathbf{c}_n \in \mathbf{c}$. We investigate the cases when $\mathsf{PVI}(\phi_Y, i)$ has certain unicity constraints.

- $\mathsf{source}(\phi_Y, i, n) = e_s^n$ is defined. By Definition 21, $(\mathbf{c}_n = e_s^n)$ occurs in $\mathsf{Collect}(\phi_Y, i)$. Since $\mathsf{Collect}(\phi_Y, i)$ must hold in Equation (7), e_s^n must be equivalent to $\mathbf{w}_n$ since $\mathbf{c}$ is evaluated to $[\![\mathbf{w}]\!]$ in Equation (7). Thus, we have that $e_s^n = \mathbf{w}_n$ and may conclude that $\mathsf{source}(\phi_Y, i, n) = \mathbf{w}_n$

- $\mathsf{target}(\phi_Y, i, n) = e_t^n$ is defined. By Definition 21, $(\mathbf{e}_n = e_t^n)$ occurs in $\mathsf{Collect}(\phi_Y, i)$. Since $\mathsf{Collect}(\phi_Y, i)$ must hold in Equation (7), e_t^n must be equivalent to $\mathbf{u}_n$ since $\mathbf{e}$ is evaluated to $[\![\mathbf{u}]\!]$ in Equation (7). Therefore, $\mathsf{target}(\phi_Y, i, n) = \mathbf{u}_n$.

- $\mathsf{copy}(\phi_Y, i, m) = n$ is defined. By Definition 21, $(\mathbf{e}_m = \mathbf{c}_n)$ occurs in $\mathsf{Collect}(\phi_Y, i)$. We have seen that $\mathsf{Collect}(\phi_Y, i)$ must hold in Equation (7). Since $\mathbf{e}_m$ must be evaluated to $[\![\mathbf{u}_m]\!]$ and $\mathbf{c}_n$ is evaluated to $[\![\mathbf{w}_n]\!]$ in Equation (7), it must be that $[\![\mathbf{w}_n]\!] = [\![\mathbf{u}_m]\!]$. Since $\mathbf{w}_n$ and $\mathbf{u}_m$ come from a uniquely representable set, we have that $\mathbf{w}_n = \mathbf{u}_m$.

We now distinguish two cases on $\mathsf{PVI}(\phi_Y, i) = Z(\mathbf{e}, \mathbf{f})$: $Y = Z$ and $Y \neq Z$.

- *Case* $Y = Z$. By definition of LCFPs, for each CFP $\mathbf{c}_n$, either $\mathsf{source}(Y, i, n) = e_s^n$ and $\mathsf{target}(Y, i, n) = e_t^n$, or $\mathsf{copy}(Y, i, n) = n$ is defined for every $\mathbf{c}_n \in \mathbf{c}$. As we have investigated previously, we have that for each CFP $\mathbf{c}_n \in \mathbf{c}$, $\mathsf{source}(Y, i, n) = \mathbf{w}_n$ and $\mathsf{target}(Y, i, n) = \mathbf{u}_n$, or $\mathsf{copy}(Y, i, n) = n$ and $\mathbf{w}_n = \mathbf{u}_n$. If this were the case, there must exist a transition $(Y, \mathbf{w}) \rightarrow_C (Z, \mathbf{u})$. However, this violates the fact that $(Z, \mathbf{u}) \notin R$, and thus we have reached a contradiction in this case.

- *Case* $Y \neq Z$. By definition of GCFPs, for all CFPs $\mathbf{c}_n \in \mathbf{c}$, either $\mathsf{copy}(Y, i, m) = n$ is defined for some $\mathbf{c}_m \in \mathbf{c}$, or $\mathsf{target}(Y, i, n) = e_t^n$ is defined. Note that by Remark 6 and Definition 18, it must be the case that $n = m$ whenever

$\mathsf{copy}(Y, i, m) = n$ is defined. As seen previously, we have that for each CFG $\mathbf{c}_n \in \mathbf{c}$, either $\mathsf{copy}(Y, i, n) = n$ and $\mathbf{w}_n = \mathbf{u}_n$, or $\mathsf{target}(Y, i, n) = \mathbf{u}_n$. Additionally, if $\mathsf{source}(Y, i, n)$ is defined, we have $\mathsf{source}(Y, i, n) = \mathbf{w}_n$. Therefore, by definition of CFGs, there must exist a transition such that $(Y, \mathbf{w}) \rightarrow_C (Z, \mathbf{u})$. But this violates our definition of the relevancy graph G and the assumption that $S \setminus S' \neq \emptyset$.

Since in both cases we have arrived at a contradiction, our assumption that $S \setminus S' \neq \emptyset$ is invalid and thus the contrary is true: $S \setminus S' = \emptyset$. This implies that $(Y, \mathbf{w}) \in R$ iff $(Y, \langle [\![\mathbf{w}]\!], \mathbf{e} \rangle) \in S$ for any $\mathbf{e}$.

Now let $f : \mathsf{bnd}(\mathscr{E}) \rightarrow \mathsf{Pred}$ be a simple function such that for all $Y \in \{X' \mid (X', \mathbf{v}') \in R\}$,

$$f(Y) = \bigvee_{(Y, \mathbf{v}') \in R} (\mathbf{c} = \mathbf{v}')$$

and for all $Y \notin \{X' \mid (X', \mathbf{v}') \in R\}$, $f(Y) = \mathsf{false}$. Observe that for each state $(X, \langle [\![\mathbf{w}]\!], \mathbf{e} \rangle) \in S$, $[\![f(X)]\!]\eta\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][[\![\mathbf{e}]\!]/\mathbf{d}_X]$ holds since $(X, \langle [\![\mathbf{w}]\!], \mathbf{e} \rangle) \in S$ iff $(X, \mathbf{w}) \in R$ for any $\mathbf{e}$. It also follows that $[\![f(X)]\!]\eta\varepsilon[[\![\mathbf{w}]\!]/\mathbf{c}][[\![\mathbf{e}]\!]/\mathbf{d}_X] = \mathsf{false}$ if $(X, \mathbf{w}) \notin R$. Thus, from Theorem 4 and our construction of the relevancy graph G, we conclude that f is a global invariant. □

7 Conclusion and Future Work

Using various work done on PBESs, we relate PBES reduction techniques to PBES global invariants. Taking inspiration from static analysis techniques, we utilize the concept of guards to characterize the contextual information surrounding PVIs and derive invariants. By taking ideas from Hoare logics and Dijkstra's predicate transformers, we construct invariants by propagating guards through predicate variable instances. We have also used the concept of control flow graphs to obtain invariants. During our research, we have uncovered and fixed a slight flaw in the current definitions of CFGs, preventing arbitrary graph constructions for equations with certain structures. Along the way, we have developed new conditions for simple functions to satisfy PBES global invariant properties that acts as an extension to the work done by Orzan et al. [6]. Moreover, we presented a new graph structure to assist in intuitively grasping the concepts of invariants, but also provides an alternative characterizations of invariants.

In our paper, we mainly refer to works done in the formal methods domain even though a variety of literature outside of formal verification has been published on invariant generation. For instance, papers such as [28, 23, 18] generate invariants using syntax-guided synthesis [39], i.e. invariants automatically generated from a defined grammar. However, additional preliminary work needs to be done on PBESs to realize this approach. Nguyen et al. [40] restrict themselves to numerical data types while Schultz et al. [18] were aware of the structure and general contents of their problem. But it is unclear what the structure of generated invariants should be for arbitrary PBESs consisting of arbitrary data types. As an example, while $x < y$ would make sense for numerical data types, it may be unclear when

x and y are lists or booleans. Additionally, global PBES invariants are functions containing invariants for multiple equations of a given PBES. Generation of an invariant for an equation may require additional invariants to be placed in other equations. If one were to only generate invariant candidates for only one equation, it then becomes a question of how to convert local PBES invariants to global PBES invariants, which requires additional research. On the other hand, it is not clear how to generate invariants for multiple, or even all, equations of a given PBES simultaneously. This generative approach may even lead to a combinatorial explosion of invariant candidates.

When working with control flow graphs in Section 5, we have discovered an flaw with the original definitions of CFGs provided in [5, 32]. As Neele pointed out in his thesis [36], issues arise when performing multiple syntactic replacements on PVIs. This issue also becomes apparent in CFGs, namely in the definitions of unicity constraints; without compositionality conditions on unicity constraints, one may produce illogical CFGs with PBES of certain structures. Additional research is required to realize the implications of this flaw, especially within the mCRL2 toolset [41] where CFGs are used.

In Section 6, we introduced the notion of relevancy graphs. These graphs are still in their infancy as there are many research directions one may take. For instance, these graphs are quite similar to BES instantiations; one may derive properties which relate BES instantiations and relevancy graphs. Additionally, analysis of relevancy graphs may provide insight as to how one may find a solution to a particular instantiation. Moreover, it may be worth investigating whether invariant generation techniques found in literature apply to relevancy graphs. More work needs to be done on these graphs to realize their full potential.

Finally, while our paper was mainly focused on how PBES global invariants could be generated, we have not considered the effectiveness of these invariants via experiments. From Theorem 35 of Orzan et al. [6], we may use PBESs strengthened with its invariant to evaluate a predicate formula, but we have not analyzed equation systems with and without their invariants. For instance, one could use mCRL2 to analyze the state spaces between invariant strengthened PBESs and PBESs without invariants. Work still needs to be done to analyze the benefits of adding invariants derived from our techniques.

References

1. Groote, J.F., Willemse, T.A.C.: Parameterised boolean equation systems. Theoretical Computer Science **343**(3), 332–369 (2005). https://doi.org/https://doi.org/10.1016/j.tcs.2005.06.016, https://www.sciencedirect.com/science/article/pii/S0304397505003658, formal Methods for Components and Objects
2. Groote, J.F., Willemse, T.A.C.: Model-checking processes with data. Science of Computer Programming **56**(3), 251–273 (2005). https://doi.org/https://doi.org/10.1016/j.scico.2004.08.002, https://www.sciencedirect.com/science/article/pii/S016764230400139X
3. van Dam, A., Ploeger, B., Willemse, T.A.C.: Instantiation for parameterised boolean equation systems. In: Fitzgerald, J.S., Haxthausen, A.E., Yenigun, H. (eds.) Theoret-

ical Aspects of Computing - ICTAC 2008. pp. 440–454. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
4. Orzan, S., Wesselink, W., Willemse, T.A.C.: Static Analysis Techniques for Parameterised Boolean Equation Systems. In: Kowalewski, S., Philippou, A. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 230–245. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
5. Keiren, J.J.A., Wesselink, W., Willemse, T.A.C.: Liveness Analysis for Parameterised Boolean Equation Systems. In: Cassez, F., Raskin, J. (eds.) Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings. Lecture Notes in Computer Science, vol. 8837, pp. 219–234. Springer (2014). https://doi.org/10.1007/978-3-319-11936-6_16, https://doi.org/10.1007/978-3-319-11936-6_16
6. Orzan, S., Willemse, T.A.C.: Invariants for Parameterised Boolean Equation Systems. Theor. Comput. Sci. **411**(11-13), 1338–1371 (2010)
7. Popov, V.L., Vinberg, E.B.: Invariant Theory, pp. 123–278. Springer Berlin Heidelberg, Berlin, Heidelberg (1994). https://doi.org/10.1007/978-3-662-03073-8_2
8. Weiss, I.: Geometric invariants and object recognition. International Journal of Computer 11263on **10**(3), 207–231 (1993). https://doi.org/10.1007/BF01539536, https://doi.org/10.1007/BF01539536
9. Alexander, J.W.: Topological invariants of knots and links. Transactions of the American Mathematical Society **30**(2), 275–306 (1928)
10. Freyd, P., Yetter, D., Hoste, J., Lickorish, W.B.R., Millett, K., Ocneanu, A.: A new polynomial invariant of knots and links. Bulletin (New Series) of the American Mathematical Society **12**(2), 239 – 246 (1985)
11. Back, R.J.: Invariant based programming: basic approach and teaching experiences. Formal Aspects of Computing **21**(3), 227–244 (2009). https://doi.org/10.1007/s00165-008-0070-y, https://doi.org/10.1007/s00165-008-0070-y
12. Hoare, C.A.R.: An Axiomatic Basis for Computer Programming. Commun. ACM **12**(10), 576580 (oct 1969). https://doi.org/10.1145/363235.363259, https://doi.org/10.1145/363235.363259
13. Barnett, M., Leino, K.R.M., Schulte, W.: The Spec# Programming System: An Overview. In: Barthe, G., Burdy, L., Huisman, M., Lanet, J.L., Muntean, T. (eds.) Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. pp. 49–69. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
14. Leavens, G., Baker, A., Ruby, C.: Preliminary design of jml. ACM SIGSOFT Software Engineering Notes **31** (05 2006). https://doi.org/10.1145/1127878.1127884
15. Leino, K.R.M.: Dafny: An Automatic Program Verifier for Functional Correctness. In: Clarke, E.M., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning. pp. 348–370. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
16. Tabar, A.H., Bubel, R., Hähnle, R.: Automatic Loop Invariant Generation for Data Dependence Analysis. In: Proceedings of the IEEE/ACM 10th International Conference on Formal Methods in Software Engineering. p. 3445. FormaliSE '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3524482.3527649, https://doi.org/10.1145/3524482.3527649
17. Pandav, S., Slind, K., Gopalakrishnan, G.: Counterexample Guided Invariant Discovery for Parameterized Cache Coherence Verification. In: Borrione, D., Paul, W. (eds.) Correct Hardware Design and Verification Methods. pp. 317–331. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
18. Schultz, W., Dardik, I., Tripakis, S.: Plain and Simple Inductive Invariant Inference for Distributed Protocols in TLA+ (2022). https://doi.org/10.48550/ARXIV.2205.06360, https://arxiv.org/abs/2205.06360

19. Neele, T., Willemse, T.A.C., Groote, J.F.: Solving Parameterised Boolean Equation Systems with Infinite Data Through Quotienting. In: Bae, K., Ölveczky, P.C. (eds.) Formal Aspects of Component Software. pp. 216–236. Springer International Publishing, Cham (2018)
20. Clementini, E., Di Felice, P.: Topological invariants for lines. IEEE Transactions on Knowledge and Data Engineering **10**(1), 38–54 (1998). https://-doi.org/10.1109/69.667085
21. Dijkstra, E.W., Dijkstra, E.W., Dijkstra, E.W., Dijkstra, E.W.: A discipline of programming, vol. 613924118. prentice-hall Englewood Cliffs (1976)
22. Pnueli, A., Ruah, S., Zuck, L.: Automatic Deductive Verification with Invisible Invariants. vol. 2031 (02 2001). https://doi.org/10.1007/3-540-45319-9_7
23. Ernst, M.D., Perkins, J.H., Guo, P.J., McCamant, S., Pacheco, C., Tschantz, M.S., Xiao, C.: The Daikon system for dynamic detection of likely invariants. Science of Computer Programming **69**(1), 35–45 (2007). https://doi.org/https://-doi.org/10.1016/j.scico.2007.01.015, https://www.sciencedirect.com/science/article/pii/S016764230700161X, special issue on Experimental Software and Toolkits
24. Zhang, L., Yang, G., Rungta, N., Person, S., Khurshid, S.: Feedback-Driven Dynamic Invariant Discovery. In: Proceedings of the 2014 International Symposium on Software Testing and Analysis. p. 362372. ISSTA 2014, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2610384.2610389, https://doi.org/10.1145/2610384.2610389
25. Păsăreanu, C.S., Visser, W.: Verification of Java Programs Using Symbolic Execution and Invariant Generation. In: Graf, S., Mounier, L. (eds.) Model Checking Software. pp. 164–181. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
26. Kahsai, T., Ge, Y., Tinelli, C.: Instantiation-Based Invariant Discovery. In: Bobaru, M., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) NASA Formal Methods. pp. 192–206. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
27. Stark, J., Ireland, A.: Invariant Discovery via Failed Proof Attempts. In: Flener, P. (ed.) Logic-Based Program Synthesis and Transformation. pp. 271–288. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
28. Nguyen, T., Antopoulos, T., Ruef, A., Hicks, M.: A Counterexample-guided Approach to Finding Numerical Invariants (2019). https://-doi.org/10.48550/ARXIV.1903.12113, https://arxiv.org/abs/1903.12113
29. Garoche, P.L., Kahsai, T., Tinelli, C.: Incremental Invariant Generation Using Logic-Based Automatic Abstract Transformers. In: Brat, G., Rungta, N., Venet, A. (eds.) NASA Formal Methods. pp. 139–154. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
30. Bensalem, S., Lakhnech, Y.: Automatic Generation of Invariants. Formal Methods in System Design **15**(1), 75–92 (1999). https://doi.org/10.1023/A:1008744030390, https://doi.org/10.1023/A:1008744030390
31. Tiwari, A., Rueß, H., Saïdi, H., Shankar, N.: A Technique for Invariant Generation. In: Margaria, T., Yi, W. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 113–127. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
32. Keiren, J.J.A.: Advanced reduction techniques for model checking. Eindhoven University of Technology (2013)
33. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. (1955)
34. Neele, T.: Reductions for parity games and model checking. Ph.D. thesis, Mathematics and Computer Science (Sep 2020), proefschrift.

35. Dijkstra, E.W., Schölten, C.S.: The strongest postcondition, pp. 209–215. Springer New York, New York, NY (1990). https://doi.org/10.1007/978-1-4612-3228-5_12, https://doi.org/10.1007/978-1-4612-3228-5_12
36. Neele, T., Willemse, T.A.C., Groote, J.F.: Finding compact proofs for infinite-data parameterised Boolean equation systems. Science of Computer Programming **188**, 102389 (2020). https://doi.org/https://doi.org/10.1016/j.scico.2019.102389, https://www.sciencedirect.com/science/article/pii/S0167642319301807
37. Cranen, S., Luttik, B., Willemse, T.A.C.: Proof Graphs for Parameterised Boolean Equation Systems. In: D'Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013 – Concurrency Theory. pp. 470–484. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
38. Keiren, J., Reniers, M.A., Willemse, T.A.C.: Structural Analysis of Boolean Equation Systems. ACM Trans. Comput. Log. **13**(1), 8:1–8:35 (2012). https://-doi.org/10.1145/2071368.2071376, https://doi.org/10.1145/2071368.2071376
39. Fedyukovich, G., Prabhu, S., Madhukar, K., Gupta, A.: Quantified Invariants via Syntax-Guided Synthesis. In: Dillig, I., Tasiran, S. (eds.) Computer Aided Verification. pp. 259–277. Springer International Publishing, Cham (2019)
40. Nguyen, T., Nguyen, K., Duong, H.: SymInfer: Inferring Numerical Invariants using Symbolic States. In: 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 197–201 (2022). https://doi.org/10.1145/3510454.3516833
41. Groote, J., Mousavi, M.: Modeling and Analysis of Communicating Systems. MIT Press (2014), https://books.google.nl/books?id=fsyMEAAAQBAJ

A Complete Proofs

This section consists of the complete proofs for various lemmas and theorems found throughout this paper.

A.1 Preliminaries

Lemma 1. *Let ϕ be a simple, capture-avoiding predicate such that $\mathsf{FV}(\phi) \subseteq \{d\}$ where $d : D$. Then we have that for any environments $\eta, \varepsilon, \varepsilon'$ and $v \in \mathbb{D}$,*

$$[\![\phi]\!]\eta\varepsilon[v/d] = [\![\phi]\!]\eta\varepsilon'[v/d]$$

Proof. Let ϕ be a simple, capture-avoiding predicate formula such that $\mathsf{FV}(\phi) \subseteq \{d\}$ for some $d : D$. Then, let ε and ε' be any two data environments, and let η be any predicate environment. Observe that for some $v \in \mathbb{D}$,

$$\varepsilon[v/d](d) = \varepsilon'[v/d](d) = v$$

We proceed with induction on the structure of subformulae ψ of ϕ and show that

$$[\![\psi]\!]\eta\varepsilon[v/d] = [\![\psi]\!]\eta\varepsilon'[v/d]$$

Base Cases:

- $\psi = b$. Since $\mathsf{FV}(\phi) \subseteq \{d\}$, we have that $\mathsf{FV}(b) \setminus \{d\} \subseteq \mathsf{bnd}(\phi)$, i.e. all free variables other than d in b is bound by a universal/existential quantifier in ϕ, otherwise $\mathsf{FV}(\phi) \not\subseteq \{d\}$. Therefore, for any variable $x \in \mathsf{FV}(b) \setminus \{d\}$ of type X, by definition of $[\![\_]\!]$ we have that $\varepsilon = \varepsilon[x'/x]$ and $\varepsilon' = \varepsilon'[x'/x]$ for some $x' : \mathbb{X}$. In other words, all variables will in b will be evaluated the same in both $\varepsilon[v/d_X]$ and $\varepsilon'[v/d_X]$. We may conclude that $[\![b]\!]\eta\varepsilon[v/d_X] = [\![b]\!]\eta\varepsilon'[v/d_X]$.
- $\psi = Y(e)$. This case contradicts the fact that ϕ is a simple predicate, i.e. ϕ cannot contain any predicate variables. Therefore, this case is vacuously true.

Step Cases: Assume the following induction hypothesis: for all subforumlae ψ_i of ψ, and for any environments $\eta^*, \varepsilon^*, \varepsilon^{**}$ and $v \in \mathbb{D}$,

$$[\![\psi_i]\!]\eta^*\varepsilon^*[v/d] = [\![\psi_i]\!]\eta^*\varepsilon^{**}[v/d]$$

- $\psi = \psi_1 \wedge \psi_2$.

$$\begin{aligned}
&[\![\psi_1 \wedge \psi_2]\!]\eta\varepsilon[v/d] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![\psi_1]\!]\eta\varepsilon[v/d] \text{ and } [\![\psi_2]\!]\eta\varepsilon[v/d] \\
&= \{\text{Induction Hypothesis}\} \\
&\quad [\![\psi_1]\!]\eta\varepsilon'[v/d] \text{ and } [\![\psi_2]\!]\eta\varepsilon'[v/d] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![\psi_1 \wedge \psi_2]\!]\eta\varepsilon'[v/d]
\end{aligned}$$

- $\psi = \psi_1 \vee \psi_2$. Symmetrical to the case where $\phi = \psi_1 \wedge \psi_2$.
- $\psi = \forall e{:}E.\ \psi_1$. Note that since ϕ is capture-avoiding, $e \neq d$.

$$\begin{aligned}
&[\![\forall e{:}E.\ \psi_1]\!]\eta\varepsilon[v/d] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad \text{for all } e' \in \mathbb{E}, [\![\psi_1]\!]\eta\varepsilon[v/d][e'/e] \\
&= \{d \neq e \implies \varepsilon[v/d][e'/e] = \varepsilon[e'/e][v/d]\} \\
&\quad \text{for all } e' \in \mathbb{E}, [\![\psi_1]\!]\eta\varepsilon[e'/e][v/d] \\
&= \{\text{Induction Hypothesis}\} \\
&\quad \text{for all } e' \in \mathbb{E}, [\![\psi_1]\!]\eta\varepsilon'[e'/e][v/d] \\
&= \{\varepsilon[v/d][e'/e] = \varepsilon[e'/e][v/d]; \text{Definition of } [\![_]\!]\} \\
&\quad [\![\forall e{:}E.\ \psi_1]\!]\eta\varepsilon'[v/d]
\end{aligned}$$

- $\psi = \exists e{:}E.\ \psi_1$. Symmetrical to the case where $\psi = \forall e{:}E.\ \psi_1$.

□

Lemma 3. *Let ϕ be a simple, capture-avoiding predicate formula and let $d \in \mathsf{FV}(\phi)$ for some $d : D$. Then, for some $e : D$ and any environments η, ε, we have*

$$[\![\phi[e/d]]\!]\eta\varepsilon = [\![\phi]\!]\eta\varepsilon[\varepsilon(e)/d]$$

Proof. Let ϕ be a simple, capture-avoiding predicate formula such that $d \in \mathsf{FV}(\phi)$ for some $d : D$. We show that for any $e : D$, $[\![\phi[e/d]]\!]\eta\varepsilon = [\![\phi]\!]\eta\varepsilon[\varepsilon(e)/d]$ for any environments η, ε by induction on the structure on subformulae ψ of ϕ.

Base Cases:
- $\psi = b$. Follows by assumption that $[\![b[e/d]]\!]\varepsilon = [\![b]\!]\varepsilon[\varepsilon(e)/d]$.
- $\psi = Y(f)$. Since ψ is a simple predicate, this case is vacuously true.

Step Cases: Assume the following inductive hypothesis: for any subformulae ψ_i of ψ and for any environments η', ε', we have

$$[\![\psi_i[e/d]]\!]\eta\varepsilon = [\![\psi_i]\!]\eta\varepsilon[\varepsilon(e)/d]$$

- $\psi = \psi_1 \wedge \psi_2$.

$$\begin{aligned}
&[\![(\psi_1 \wedge \psi_2)[e/d]]\!]\eta\varepsilon \\
&= \{\text{Definition of substitution}\} \\
&\quad [\![(\psi_1[e/d]) \wedge (\psi_2[e/d])]\!]\eta\varepsilon \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![\psi_1[e/d]]\!]\eta\varepsilon \text{ and } [\![\psi_2[e/d]]\!]\eta\varepsilon \\
&= \{\text{Induction hypothesis}\} \\
&\quad [\![\psi_1]\!]\eta\varepsilon[\varepsilon(e)/d] \text{ and } [\![\psi_2]\!]\eta\varepsilon[\varepsilon(e)/d] \\
&= \{\text{Definition of } [\![_]\!] \text{ and substitution}\} \\
&\quad [\![\psi_1 \wedge \psi_2]\!]\eta\varepsilon[\varepsilon(e)/d]
\end{aligned}$$

- $\psi = \psi_1 \vee \psi_2$. Symmetrical to the case $\psi = \psi_1 \wedge \psi_2$.
- $\psi = \forall f{:}F.\ \psi_1$. Without loss of generality, assume $f \neq e$.

$$
\begin{aligned}
& [\![(\forall f{:}F.\ \psi_1)[e/d]]\!]\eta\varepsilon \\
= & \ \{\text{Definition of substitution}\} \\
& [\![\forall f{:}F.\ (\psi_1[e/d])]\!]\eta\varepsilon \\
= & \ \{\text{Definition of } [\![\_]\!]\} \\
& \text{for all } f' \in \mathbb{F}, [\![\psi_1[e/d]]\!]\eta\varepsilon[f'/f] \\
= & \ \{\text{Induction hypothesis}\} \\
& \text{for all } f' \in \mathbb{F}, [\![\psi_1]\!]\eta\varepsilon[f'/f][\varepsilon[f'/f](e)/d] \\
= & \ \{e \neq f \implies \varepsilon[f'/f](v) = \varepsilon(v)\} \\
& \text{for all } f' \in \mathbb{F}, [\![\psi_1]\!]\eta\varepsilon[f'/f][\varepsilon(e)/d] \\
= & \ \{d \in \mathsf{FV}(\phi);\ d \neq f \implies \varepsilon[f'/f][\varepsilon(e)/d] = \varepsilon[\varepsilon(e)/d][f'/f]\} \\
& \text{for all } f' \in \mathbb{F}, [\![\psi_1]\!]\eta\varepsilon[\varepsilon(e)/d][f'/f] \\
= & \ \{\text{Definition of } [\![\_]\!]\} \\
& [\![(\forall f{:}F.\ \psi_1)]\!]\eta\varepsilon[\varepsilon(e)/d]
\end{aligned}
$$

- $f(X) = \exists f{:}F.\ \psi_1$. Symmetrical to the case $\forall f{:}F.\ \psi_1$.

□

A.2 Guards

Property 2. Let $\mathscr{E}$ be a closed equation system. For every equation $(\sigma X(d_X : D_X) = \phi_X)$ in $\mathscr{E}$, assume the existence of compositional guards $\gamma^1_{\phi_X}, \ldots, \gamma^{\mathsf{npred}(\phi_X)}_{\phi_X}$ for ϕ_X where $\gamma^i_{\phi_X}$ is the guard for $\mathsf{PVI}(\phi_X, i)$. Let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple function such that for every equation $(\sigma X(d_X : D_X) = \phi_X)$ in $\mathscr{E}$ and all $\mathsf{PVI}(\phi_X, i) = X_i(e_i)$:

$$f(X) \wedge \gamma^i_{\phi_X} \to f(X_i)[e_i/d_{X_i}]$$

Then f is a global invariant for $\mathscr{E}$.

Proof. Consider an equation $(\sigma X(d_X : D_X) = \phi_X)$ for which $f(X) \wedge \gamma^i_{\phi_X} \to f(X_i)[e/d_{X_i}]$ holds for all $\mathsf{PVI}(\phi_X, i) = X_i(e_i)$. By definition of compositional guards, we have that

$$\phi_X \leftrightarrow \phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$$

Note that $\phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$ can be described by the following grammar since each PVI occurs within the scope of a conjunction:

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

where γ is the guard for $X(e)$.

We now perform structural induction on the structure of the subformulae ψ of $\phi_X[j \mapsto \gamma^j_{\phi_X} \wedge \mathsf{PVI}(\phi_X, j)]_{j \leq \mathsf{npred}(\phi_X)}$. For each subformula, it suffices to prove that

$$(f(X) \wedge \psi) \leftrightarrow (f(X) \wedge \psi)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big]$$

We first consider the base cases:

Case $\psi = b$. Immediately follows from the definition of syntactic substitution:

$$(f(X) \wedge b) \leftrightarrow (f(X) \wedge b)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big]$$

Case $\psi = \gamma^i_{\phi_X} \wedge Y(e)$. Let $Y(e) = \mathsf{PVI}(\phi_X, i)$. We then reason as follows:

$$\begin{aligned}
&(f(X) \wedge (\gamma^i_{\phi_X} \wedge Y(e)))\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big] \\
\leftrightarrow\ &\{\text{Definition of syntactic substitution; } f(X) \text{ and } \gamma^i_{\phi_X} \text{ are simple}\} \\
&f(X) \wedge (f(Y)[e/d_Y]) \wedge (\gamma^i_{\phi_X} \wedge Y(e)) \\
\leftrightarrow\ &\{\text{Assumption: } f(X) \wedge \gamma^i_{\phi_X} \rightarrow f(Y)[e/d_Y]\} \\
&f(X) \wedge (\gamma^i_{\phi_X} \wedge Y(e))
\end{aligned}$$

Assume the following induction hypothesis: for any arbitrary subformulae ψ_i of ϕ,

$$f(X) \wedge \psi_i \leftrightarrow (f(X) \wedge \psi_i)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big]$$

Case $\psi = \psi_1 \vee \psi_2$.

$$\begin{aligned}
&(f(X) \wedge \psi_1 \wedge \psi_2)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big] \\
\leftrightarrow\ &\{f(X) = f(X) \wedge f(X); \text{definition of syntactic substitution}\} \\
&(f(X) \wedge \psi_1)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big] \wedge \\
&\qquad (f(X) \wedge \psi_2)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})}(f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle}/Z\Big] \\
\leftrightarrow\ &\{\text{Induction hypothesis}\} \\
&(f(X) \wedge \psi_1) \wedge (f(X) \wedge \psi_2) \\
\leftrightarrow\ &\{\psi = \psi_1 \wedge \psi_2\} \\
&f(X) \wedge \psi
\end{aligned}$$

Case $\psi = \psi_1 \wedge \psi_2$. Analogous to the case where $\psi = \psi_1 \vee \psi_2$.

Case $\psi = \forall e{:}E.\ \psi_1$. Without loss of generality, assume f does not contain e.

$$
\begin{aligned}
& (f(x) \wedge \forall e{:}E.\ \psi_1)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})} (f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle} / Z\Big] \\
\leftrightarrow\ & \{e \text{ does not occur in } f(X)\} \\
& \forall e{:}E.\ (f(x) \wedge \psi_1)\Big[_{Z \in \mathsf{bnd}(\mathscr{E})} (f(Z) \wedge Z(d_Z))_{\langle d_Z \rangle} / Z\Big] \\
\leftrightarrow\ & \{\text{Induction Hypothesis}\} \\
& \forall e{:}E.\ (f(x) \wedge \psi_1) \\
\leftrightarrow\ & \{e \text{ does not occur in } f(X)\} \\
& f(x) \wedge \forall e{:}E.\ \psi_1
\end{aligned}
$$

Case $\psi = \exists e{:}E.\ \psi_1$. Analogous to the case where $\psi = \forall e{:}E.\ \psi_1$.

□

A.3 Relevancy Graphs

Lemma 7. *Let $\mathscr{E}$ be a PBES and $G = (S, \to)$ be its corresponding relevancy graph such that $(X, v) \in S$. Define the reachable sub-graph originating from (X, v) to be $\mathsf{Reach}_{(X,v)}(G) = (S', \to')$, where*

$$
\begin{aligned}
S' &= \{(X', v') \mid (X, v) \to^* (X', v')\} \\
\to' &= \to \cap\, (S' \times S')
\end{aligned}
$$

Then, $\mathsf{Reach}_{(X,v)}(G)$ is also a relevancy graph for (X, v).

Proof. Observe that the first condition of Definition 23 is satisfied for S'. We prove the second condition. Consider any state $(X', v') \in S'$.

($\Longrightarrow$) We prove that if $(X', v') \to' (Y, w)$, then

$$\exists \eta, \varepsilon.\ [\![\phi_X]\!]\eta[\mathsf{true}/Y(w)]\varepsilon[v'/d_{X'}] \neq [\![\phi_X]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v'/d_{X'}] \quad (10)$$

Since $(X', v') \to' (Y, w)$, it must be the case that $(X', v') \to (Y, w)$. Because $G = (S, \to)$ is a relevancy graph, (10) holds.

($\Longleftarrow$) Assume the following holds:

$$\exists \eta, \varepsilon.\ [\![\phi_X]\!]\eta[\mathsf{true}/Y(w)]\varepsilon[v/d_X] \neq [\![\phi_X]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v/d_X]$$

We have that by construction of S', $(X', v') \in S$. Since $G = (S, \to)$ is a relevancy graph, it must be the case that $(X', v') \to (Y, w)$. Because $(X, v) \to^* (X', v')$ and $(X', v') \to (Y, w)$, also $(X, v) \to^* (Y, w)$, and $(Y, w) \in S'$. It follows that also $(X', v') \to' (Y, w)$ by definition of $\to'$.

As we have proven both directions, we have that $\mathsf{Reach}_{(X,v)}(G) = (S', \to')$ is also a relevancy graph. □

Lemma 8. *Let $\mathscr{E}$ be a PBES and $G = (S, \to)$ be its corresponding relevancy graph for some (X, v), and let $\mathsf{Reach}_{(X,v)}(G) = (S', \to')$ be the sub-graph of G consisting of vertices reachable from (X, v). Let V be a nonempty set such that $V \subset S'$, and let the graph $G' = (S'', \to'')$ be the graph $\mathsf{Reach}_{(X,v)}$ without vertices in V. Specifically,*

$$S'' = S' \setminus V$$
$$\to'' = \to' \cap (S'' \times S'')$$

Then, G' cannot be a relevancy graph for (X, v).

Proof. If $(X, v) \in V$, then by definition of relevancy graphs, G' cannot be a relevancy graph for (X, v). Thus, assume $(X, v) \notin V$. For the sake of contradiction, assume that G' is a relevancy graph for (X, v). Consider a state $(X', v') \in S'$ such that $(X', v') \to' (Y, w)$ for some $(Y, w) \in V$. It must be the case that $(X', v') \in S''$, $(Y, w) \notin S''$, and $(X', v') \not\to'' (Y, w)$. Since $\mathsf{Reach}_{(X,v)}(G)$ is a relevancy graph and $(X', v') \to' (Y, w)$, it must be the case that

$$\exists \eta, \varepsilon.\ [\![\phi_X]\!]\eta[\mathsf{true}/Y(w)]\varepsilon[v'/d_{X'}] \neq [\![\phi_X]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v'/d_{X'}]$$

Since G' is also a relevancy graph and $(X', v') \in S''$, it must be the case that $(X', v') \to'' (Y, w)$. However, this contradicts our construction of G'. Therefore, our assumption is invalid; it cannot be the case that G' is a relevancy graph. □

Lemma 9. *Let $f : V \to \mathsf{Pred}$ be a simple, capture-avoiding function where $V \subseteq \mathcal{P}$. Assume that for every $X \in V$, $\mathsf{FV}(f(X)) \subseteq \{d_X\}$ with $d_X : D_X$. Let $X \in V$ and $[\![f(X)]\!]\eta'\varepsilon'[v/d_X] = \mathsf{false}$ for some $v \in \mathbb{D}_X$ and environments η', ε'. Then, for any predicate formula ϕ and arbitrary environments η, ε,*

$$\left[\!\!\left[\phi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{true}/X(v)]\varepsilon$$
$$=$$
$$\left[\!\!\left[\phi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{false}/X(v)]\varepsilon$$

Proof. Let $f : V \to \mathsf{Pred}$ be a simple function with $V \subseteq \mathcal{P}$, and for all $X \in V$, $\mathsf{FV}(f(X)) \subseteq \{d_X\}$ where $d_X : D_X$. Let $X \in V$ and $[\![f(X)]\!]\eta'\varepsilon'[v/d_X] = \mathsf{false}$ for $v : \mathbb{D}_X$ and environments η', ε'. By induction on the structure of subformulae ψ of ϕ, we show that for environments η, ε,

$$\left[\!\!\left[\psi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{true}/X(v)]\varepsilon$$
$$=$$
$$\left[\!\!\left[\psi\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\!\right]\eta[\mathsf{false}/X(v)]\varepsilon$$

Base Cases

- $\psi = b$ Since the evaluation of b does is not affected by the environment η, the claim holds in this case.

- $\psi = Z(e)$. We distinguish two subcases: $Z = X$ and $Z \neq X$.

  Subcase: $Z = X$.

  $$\begin{aligned}
  & \left[\!\!\left[X(e)\Big[_{Y_i \in V} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i \right] \right]\!\!\right] \eta[\mathsf{true}/X(v)]\varepsilon \\
  = & \{\text{Definition of substitution}\} \\
  & \left[\!\!\left[f(X)[e/d_X] \wedge X(e) \right]\!\!\right] \eta[\mathsf{true}/X(v)]\varepsilon \\
  = & \{\text{Definition of } [\![_]\!]\} \\
  & [\![f(X)[e/d_X]]\!]\eta[\mathsf{true}/X(v)]\varepsilon \text{ and } \eta[\mathsf{true}/X(v)](X)(\varepsilon(e)) \\
  = & \{\text{Lemma 3}\} \\
  & [\![f(X)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[\varepsilon(e)/d_X] \text{ and } \eta[\mathsf{true}/X(v)](X)(\varepsilon(e))
  \end{aligned}$$

  We have that either $\varepsilon(e) = v$ or $\varepsilon(e) \neq v$. First, consider the case when $\varepsilon(e) = v$.

  $$\begin{aligned}
  & [\![f(X)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[\varepsilon(e)/d_X] \text{ and } \eta[\mathsf{true}/X(v)](X)(\varepsilon(e)) \\
  = & \{\text{Case } \varepsilon(e) = v\} \\
  & [\![f(X)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[v/d_X] \text{ and } \eta[\mathsf{true}/X(v)](X)(v) \\
  = & \{[\![f(X)]\!]\eta'\varepsilon'[v/d_X] = \mathsf{false} \text{ and Corollary 1}\} \\
  & \mathsf{false} \\
  = & \{\text{Definition of environment } \eta[\mathsf{false}/X(v)]\} \\
  & \eta[\mathsf{false}/X(v)](X)(v) \\
  = & \\
  & [\![f(X)[e/d_X]]\!]\eta[\mathsf{false}/X(v)]\varepsilon \text{ and } \eta[\mathsf{false}/X(v)](X)(v) \\
  = & \{\text{Definition of substitution and } [\![_]\!]\} \\
  & \left[\!\!\left[X(e)\Big[_{Y_i \in V} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i \right] \right]\!\!\right] \eta[\mathsf{false}/X(v)]\varepsilon
  \end{aligned}$$

  Now we consider the case when $\varepsilon(e) \neq v$. Let $w = \varepsilon(e)$.

  $$\begin{aligned}
  & [\![f(X)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[\varepsilon(e)/d_X] \text{ and } \eta[\mathsf{true}/X(v)](X)(\varepsilon(e)) \\
  = & \{\text{Case } \varepsilon(e) \neq v;\ \varepsilon(e) = w\} \\
  & [\![f(X)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[w/d_X] \text{ and } \eta(X)(w) \\
  = & \{f \text{ is simple, Lemma 2; } w \neq v\} \\
  & [\![f(X)]\!]\eta[\mathsf{false}/X(v)]\varepsilon[w/d_X] \text{ and } \eta[\mathsf{false}/X(v)](X)(w) \\
  = & \{\text{Lemma 3; } w = \varepsilon(e)\} \\
  & [\![f(X)[e/d_X]]\!]\eta[\mathsf{false}/X(v)]\varepsilon \text{ and } \eta[\mathsf{false}/X(v)](X)(w) \\
  = & \{\text{Definition of } [\![_]\!] \text{ and substitution}\} \\
  & \left[\!\!\left[X(e)\Big[_{Y_i \in V} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i \right] \right]\!\!\right] \eta[\mathsf{false}/X(v)]\varepsilon
  \end{aligned}$$

Subcase: $X \neq Z$.

$$
\begin{aligned}
&\left[\!\left[Z(e)\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle}/Y_i \Big] \right]\!\right] \eta[\mathsf{true}/X(w)]\varepsilon \\
&= \{\text{Definition of substitution and } [\![\_]\!]\} \\
&\quad [\![f(Z)[e/d_Z]]\!]\eta[\mathsf{true}/X(v)]\varepsilon \text{ and } \eta[\mathsf{true}/X(v)](Z)(\varepsilon(e)) \\
&= \{\text{Lemma 3}\} \\
&\quad [\![f(Z)]\!]\eta[\mathsf{true}/X(v)]\varepsilon[\varepsilon(e)/d_Z] \text{ and } \eta[\mathsf{true}/X(v)](Z)(\varepsilon(e)) \\
&= \{f \text{ is simple; Lemma 2}\} \\
&\quad [\![f(Z)]\!]\eta[\mathsf{false}/X(v)]\varepsilon[\varepsilon(e)/d_Z] \text{ and } \eta[\mathsf{true}/X(v)](Z)(\varepsilon(e)) \\
&= \{Z \neq X; \eta[\mathsf{true}/X(v)](Z) = \eta[\mathsf{false}/X(v)](Z)\} \\
&\quad [\![f(Z)]\!]\eta[\mathsf{false}/X(v)]\varepsilon[\varepsilon(e)/d_Z] \text{ and } \eta[\mathsf{false}/X(v)](Z)(\varepsilon(e)) \\
&= \{\text{Lemma 3; Definition of } [\![\_]\!] \text{ and substitution}\} \\
&\quad \left[\!\left[Z(e)\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle}/Y_i \Big] \right]\!\right] \eta[\mathsf{false}/X(v)]\varepsilon
\end{aligned}
$$

Step Case. Assume the following induction hypothesis: for any subformulae ψ_i of ψ and any environments η^*, ε^*,

$$
\left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle}/Y_i \Big] \right]\!\right] \eta^*[\mathsf{true}/X(v)]\varepsilon^*
$$

$$
=
$$

$$
\left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle}/Y_i \Big] \right]\!\right] \eta^*[\mathsf{false}/X(v)]\varepsilon^*
$$

- $\psi = \psi_1 \wedge \psi_2$.

$$
\begin{aligned}
&\left[\!\left[(\psi_i \wedge \psi_2)\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\
&= \{\text{Definition of substitution}\} \\
&\quad \left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right.\right. \\
&\qquad \left.\left. \wedge\ \psi_2\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\
&\qquad \text{and } \left[\!\left[\psi_2\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\
&= \{\text{Induction hypothesis}\} \\
&\quad \left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{false}/X(v)]\varepsilon \\
&\qquad \text{and } \left[\!\left[\psi_2\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{false}/X(v)]\varepsilon \\
&= \{\text{Definition of } [\![\_]\!] \text{ and substitution}\} \\
&\quad \left[\!\left[(\psi_i \wedge \psi_2)\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{false}/X(v)]\varepsilon
\end{aligned}
$$

- $\psi = \psi_1 \vee \psi_2$. Symmetrical to the case where $\psi = \psi_1 \wedge \psi_2$.

- $\psi = \forall e{:}E.\ \psi_1$.

$$
\begin{aligned}
&\left[\!\left[\forall e{:}E.\ \psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \text{for all } e' \in \mathbb{E},\ \left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{true}/X(v)]\varepsilon[e'/e] \\
&= \{\text{Induction hypothesis}\} \\
&\quad \text{for all } e' \in \mathbb{E},\ \left[\!\left[\psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{false}/X(v)]\varepsilon[e'/e] \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \left[\!\left[\forall e{:}E.\ \psi_i\Big[_{Y_i \in V}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\right]\!\right]\eta[\mathsf{false}/X(v)]\varepsilon
\end{aligned}
$$

- $\psi = \exists e{:}E.\ \psi_1$. Symmetrical to the case where $\psi = \forall e{:}E.\ \psi_1$.

□

Lemma 10. *Let $\mathscr{E}$ be a closed PBES and let $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ be a simple, capture-avoiding function. Let $G = (S, \to)$ be a relevancy graph for $\mathscr{E}$. Assume*

that $(Y, w) \in S$ *iff* $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ *holds for all environments* η, ε*. Then, for any* $(X, v) \in S$ *and for all environments* η, ε*, we have that*

$$[\![\phi_X]\!]\eta\varepsilon[v/d_X] = \left[\!\!\left[\phi_X \Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i\right]\!\!\right] \eta\varepsilon[v/d_X]$$

Proof. Let $\mathscr{E}$ be a PBES, $f : \mathsf{bnd}(\mathscr{E}) \to \mathsf{Pred}$ a simple, capture-avoiding function, and $G = (S, \to)$ a relevancy graph for $\mathscr{E}$. Assume $[\![f(Y)]\!]\eta\varepsilon[w/d_Y]$ holds for any environments η, ε iff $(Y, w) \in S$. Consider some state $(X, v) \in S$ where $\sigma X(d_X : D_X) = \phi_X$ is an equation in $\mathscr{E}$, and consider environments η, ε'. We aim to prove the following:

$$[\![\phi_X]\!]\eta\varepsilon'[v/d_X] = \left[\!\!\left[\phi_X \Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i\right]\!\!\right] \eta\varepsilon'[v/d_X]$$

We proceed with induction on the structure of subformulae ψ_i of ϕ_X. In other words,

$$[\![\psi_i]\!]\eta\varepsilon[v/d_X] = \left[\!\!\left[\psi_i \Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})} (f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i} \rangle} / Y_i\right]\!\!\right] \eta\varepsilon[v/d_X]$$

for some environment ε. Without loss of generality, assume that d_X is free in ϕ_X. Observe that by Definition 4, during the evaluation of a predicate formulae, the predicate environment is not modified and thus, evaluations of subformulae rely on the same predicate environment as ϕ_X. Additionally, since d_X is free in ϕ_X, evaluations of subformulae using some environment ε must satisfy $\varepsilon = \varepsilon[v/d_X]$.

Base Cases:

- $\psi = b$. Follows from the definition of syntatic substitution.
- $\psi = Y(e)$. Let $\varepsilon[v/d_X](e) = w$.

$$\begin{aligned}
&\left[\!\!\left[Y(e) \Big[_{Z_i \in \mathsf{bnd}(\mathscr{E})} (f(Z_i) \wedge Z_i(d_{Z_i}))_{\langle d_{Z_i} \rangle} / Z_i\right]\!\!\right] \eta\varepsilon[v/d_X] \\
&= \{\text{Definition of syntatic substitution}\} \\
&\quad [\![f(Y)[e/d_Y] \wedge Y(e)]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![_]\!]\} \\
&\quad [\![f(Y)[e/d_Y]]\!]\eta\varepsilon[v/d_X] \text{ and } [\![Y(e)]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Lemma 3}\} \\
&\quad [\![f(Y)]\!]\eta\varepsilon[\varepsilon[v/d_X](e)/d_Y] \text{ and } [\![Y(e)]\!]\eta\varepsilon[v/d_X] \\
&= \{\varepsilon[v/d_X](e) = w\} \\
&\quad [\![f(Y)]\!]\eta\varepsilon[w/d_Y] \text{ and } [\![Y(e)]\!]\eta\varepsilon[v/d_X]
\end{aligned}$$

  We distinguish two cases: $(Y, w) \in S$ and $(Y, w) \notin S$.
  - $(Y, w) \in S$. By assumption, we have $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y]$ for some η', ε'. It follows from Corollary 1 that $\forall \eta^*, \varepsilon^*.\ [\![f(Y)]\!]\eta^*\varepsilon^*[w/d_Y] = \mathsf{true}$. We proceed with our proof:

$$\begin{aligned}
&[\![f(Y)]\!]\eta\varepsilon[w/d_Y] \text{ and } [\![Y(e)]\!]\eta\varepsilon[v/d_X] \\
&= \{\forall \eta^*, \varepsilon^*.\ [\![f(Y)]\!]\eta^*\varepsilon^*[w/d_Y] = \mathsf{true}\} \\
&\quad [\![Y(e)]\!]\eta\varepsilon[v/d_X]
\end{aligned}$$

* $Y(w) \notin S$. It must be the case that $(X, v) \not\rightarrow (Y, w)$. Since G is a relevancy graph for $\mathscr{E}$, by Definition 23,

$$\neg\exists\eta^*, \varepsilon^*.\ [\![\phi_X]\!]\eta^*[\mathsf{true}/Y(w)]\varepsilon^*[v/d_X] \neq [\![\phi_X]\!]\eta^*[\mathsf{false}/Y(w)]\varepsilon^*[v/d_X]$$

Rewriting the quantifier yields the following:

$$\forall\eta^*, \varepsilon^*.\ [\![\phi_X]\!]\eta^*[\mathsf{true}/Y(w)]\varepsilon^*[v/d_X] = [\![\phi_X]\!]\eta^*[\mathsf{false}/Y(w)]\varepsilon^*[v/d_X] \tag{11}$$

In other words, the outcome of the evaluation of $[\![Y(e)]\!]\eta\varepsilon[v/d_X] = \eta(Y)(\varepsilon[v/d_X](e)) = \eta(Y)(w)$ does not affect the solution to $[\![\phi_X]\!]\eta\varepsilon'[v/d_X]$. Still, we may show that we arrive at $[\![Y(e)]\!]\eta\varepsilon[v/d_X]$. By definition of $[\![\_]\!]$ and Equation (11), we have that

$$\begin{aligned} & [\![Y(e)]\!]\eta\varepsilon[v/d_X] \\ = \\ & [\![Y(e)]\!]\eta[\mathsf{true}/Y(w)]\varepsilon[v/d_X] \\ = \\ & [\![Y(e)]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v/d_X] \end{aligned} \tag{12}$$

By assumption, $[\![f(Y)]\!]\eta'\varepsilon'[w/d_Y] = \mathsf{false}$. It follows from Corollary 1 that

$$\forall\eta^*, \varepsilon^*.\ [\![f(Y)]\!]\eta^*\varepsilon^*[w/d_Y] = \mathsf{false}$$

We now argue as follows:

$$\begin{aligned} & [\![f(Y)]\!]\eta\varepsilon[w/d_Y] \text{ and } [\![Y(e)]\!]\eta\varepsilon[v/d_X] \\ = & \{\forall\eta^*, \varepsilon^*.\ [\![f(Y)]\!]\eta^*\varepsilon^*[w/d_Y] = \mathsf{false}\} \\ & \mathsf{false} \\ = & \{\varepsilon[v/d_X](e) = w\} \\ & [\![Y(e)]\!]\eta[\mathsf{false}/Y(w)]\varepsilon[v/d_X] \\ = & \{\text{Equation (12)}\} \\ & [\![Y(e)]\!]\eta\varepsilon[v/d_X] \end{aligned}$$

Step Cases: Assume the following induction hypothesis: for all subformulae ψ_i of ϕ_X and any environments η^*, ε^*, the following holds:

$$[\![\psi_i]\!]\eta^*\varepsilon^*[v/d_X] = \left[\!\!\left[\psi_i\left[_{Y_j \in \mathsf{bnd}(\mathscr{E})}(f(Y_j) \wedge Y_j(d_{Y_j}))_{\langle d_{Y_j}\rangle}/Y_j\right]\right]\!\!\right]\eta^*\varepsilon^*[v/d_X]$$

- $\phi_X = \psi_1 \wedge \psi_2$

$$
\begin{aligned}
&[\![\psi_1 \wedge \psi_2]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad [\![\psi_1]\!]\eta\varepsilon[v/d_X] \text{ and } [\![\psi_2]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Induction hypothesis}\} \\
&\quad \Big[\!\Big[\psi_1\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[v/d_X] \text{ and} \\
&\qquad \Big[\!\Big[\psi_2\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \Big[\!\Big[\psi_1\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big] \wedge \\
&\qquad \psi_2\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of substitution}\} \\
&\quad \Big[\!\Big[(\psi_1 \wedge \psi_2)\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[v/d_X]
\end{aligned}
$$

- $\phi_X = \psi_1 \vee \psi_2$. Symmetrical to the case $\phi_X = \psi_1 \wedge \psi_2$.

- $\forall e{:}E.\ \psi_1$.

$$
\begin{aligned}
&[\![\forall e{:}E.\ \psi_1]\!]\eta\varepsilon[v/d_X] \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \text{for all } e' \in \mathbb{E}.\ [\![\psi_1]\!]\eta\varepsilon[v/d_X][e'/e] \\
&= \{d_X \text{ is free in } \phi_X;\ d_X \neq e\} \\
&\quad \text{for all } e' \in \mathbb{E}.\ [\![\psi_1]\!]\eta\varepsilon[e'/e][v/d_X] \\
&= \{\text{Induction hypothesis}\} \\
&\quad \text{for all } e' \in \mathbb{E}.\ \Big[\!\Big[\psi_1\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[e'/e][v/d_X] \\
&= \{\text{Definition of } [\![\_]\!]\} \\
&\quad \Big[\!\Big[\forall e{:}E.\ \psi_1\Big[_{Y_i \in \mathsf{bnd}(\mathscr{E})}(f(Y_i) \wedge Y_i(d_{Y_i}))_{\langle d_{Y_i}\rangle}/Y_i\Big]\Big]\!\Big]\eta\varepsilon[v/d_X]
\end{aligned}
$$

- $\exists e{:}E.\ \psi_1$. Symmetrical to the case $\forall e{:}E.\ \psi_1$.

□